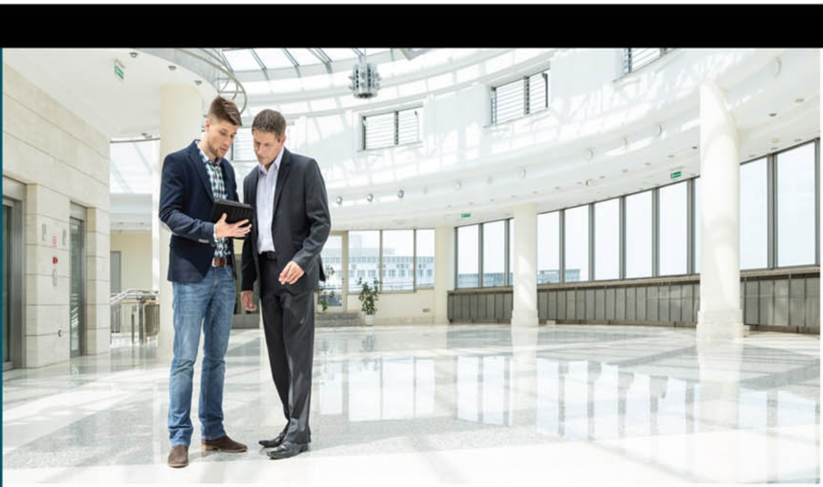


Cisco Intelligent WAN (IWAN)

Brad Edgeworth, CCIE No. 31574
Jean Marc Barozet
David Prall, CCIE No. 6508
Anthony Lockhart
Nir Ben-Dvora

ciscopress.com

Cisco Intelligent WAN (IWAN)

Brad Edgeworth, CCIE No. 31574

Jean-Marc Barozet

David Prall, CCIE No. 6508

Anthony Lockhart

Nir Ben-Dvora

Cisco Press

800 East 96th Street

Indianapolis, Indiana 46240 USA

Cisco Intelligent WAN (IWAN)

Brad Edgeworth, Jean-Marc Barozet, David Prall, Anthony Lockhart, Nir Ben-Dvora

Published by:
Cisco Press
800 East 96th Street
Indianapolis, IN 46240 USA

Printed in the United States of America

First Printing November 2016

Library of Congress Control Number: 2016954607

ISBN-13: 978-1-58714-463-9

ISBN-10: 1-58714-463-8

Warning and Disclaimer

This book is designed to provide information about the Cisco Intelligent WAN (IWAN) and Software Defined Wide Area Networking (SD-WAN). Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an “as is” basis. The authors, Cisco Press, and Cisco Systems, Inc. shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the discs or programs that may accompany it.

The opinions expressed in this book belong to the author and are not necessarily those of Cisco Systems, Inc.

Trademark Acknowledgments

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Cisco Press or Cisco Systems, Inc., cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Special Sales

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact intlcs@pearson.com.

Feedback Information

At Cisco Press, our goal is to create in-depth technical books of the highest quality and value. Each book is crafted with care and precision, undergoing rigorous development that involves the unique expertise of members from the professional technical community.

Readers' feedback is a natural continuation of this process. If you have any comments regarding how we could improve the quality of this book, or otherwise alter it to better suit your needs, you can contact us through email at feedback@ciscopress.com. Please make sure to include the book title and ISBN in your message.

We greatly appreciate your assistance.

Editor-in-Chief: Mark Taub

Production Line Manager: Brett Bartow

Business Operation Manager, Cisco Press: Ronald Fligge

Acquisitions Editor: Michelle Newcomb

Managing Editor: Sandra Schroeder

Development Editor: Ellie Bru

Senior Project Editor: Tracey Croom

Copy Editor: Barbara Wood

Technical Editor(s): Denise Fishburne, Tom Kunath

Editorial Assistant: Vanessa Evans

Cover Designer: Chuti Prasertsith

Composition: codeMantra

Indexer: Lisa Stumpf

Proofreader: H.S. Rupa

CISCO

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV
Amsterdam, The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

About the Authors

Brad Edgeworth, CCIE No. 31574 (R&S & SP), is a Systems Engineer at Cisco Systems. Brad is a distinguished speaker at Cisco Live where he has presented on a variety of topics. Before joining Cisco, Brad worked as a network architect and consultant for various Fortune 500 companies. Brad's expertise is based on enterprise and service provider environments with an emphasis on architectural and operational simplicity. Brad holds a Bachelor of Arts degree in computer systems management from St. Edward's University in Austin, TX. Brad can be found on Twitter @BradEdgeworth.

Jean-Marc Barozet is a Principal Engineer with the Intelligent WAN (IWAN) Product Management team, helping to architect and lead the Cisco SD-WAN solution. Jean-Marc has more than 25 years of experience in the networking industry and has been with Cisco for more than 19 years. He previously was Consulting Systems Engineer in the Cisco sales organization in France. Before joining Cisco, Jean-Marc worked as a consulting engineer for Alcatel Business Systems. He works closely with the largest Cisco customers to design complex large-scale networks in both the enterprise and service provider verticals. He holds an engineering degree from the National Institute of Applied Sciences of Lyon (INSA). Jean-Marc can be found on Twitter @jbarozet.

David Prall, CCIE No. 6508 (R/S, SP, & Security), is a Communications Architect on the Enterprise Networking Technical Strategy team. David has been with Cisco more than 16 years. He previously held Consulting Systems Engineer and Systems Engineer positions within the US federal area supporting civilian agencies. He is currently focused on the Cisco Intelligent WAN (IWAN), primarily complex routing and switching designs to include design, deployment, and troubleshooting of large networks using IOS, IOS XE, and IOS XR. Areas of expertise include IPv6, multicast, MPLS, fast convergence, and quality of service. He holds a Bachelor of Science degree in computer science from George Washington University. David works out of the Herndon, VA, office.

Anthony Lockhart is a Technical Marketing Engineer for the Enterprise Networking Group at Cisco Systems focusing on WAN optimization. Before joining Cisco, Anthony worked as a network engineer and consultant for various companies over his 19-year IT career. He lives in Kentucky with his wife, Colleen, and they enjoy traveling and sightseeing. Anthony received a bachelor's degree in history and sociology from Houston Baptist University in Houston, TX.

Nir Ben-Dvora is a Technical Leader, Engineering for the Core Software Group at Cisco Systems. Nir has more than 25 years of experience in software development, management, and architecture. Nir has been with Cisco for more than 17 years working as a software architect for various network services products in the areas of security, media, application recognition, visibility, and control. In recent years, Nir has been working as a Software Architect for the Application Visibility and Control (AVC) solution and as part of the Intelligent WAN (IWAN) architecture team. Nir owns multiple patents on network services and is a coauthor of IETF RFC 6759. Nir holds a Master of Science degree in electrical engineering and business management from Tel Aviv University and always likes to learn about and experiment with new technologies. Nir lives in Herzliya, Israel, with his wife, Naama, and their three boys.

About the Technical Reviewers

Tom Kunath, CCIE No. 1679 (R&S), is a Solutions Architect in Cisco Advanced Services, working with Cisco customers to plan, design, and implement large-scale, complex networks. Tom has more than 20 years of experience in the networking industry and in recent years has been exclusively focused on SD-WAN and the Cisco IWAN solution, helping to shape the architecture and lead successful deployments on several early adopter networks. Before joining Cisco, Tom worked at Juniper Networks' Professional Services Group as a Resident Engineer supporting several service provider IP and MPLS backbones, and prior to that as a Principal Consultant at International Network Services (INS). Tom is a frequent speaker on the Cisco Live circuit and is the coauthor of the 2011 Cisco Press Publication *Enterprise Network Testing*.

Denise "Fish" Fishburne, CCDE No. 20090014, CCIE No. 2639 (R&S, SNA), is an Engineer and Team Lead with the Customer Proof of Concept Lab (CPOC) in North Carolina. Fish is a geek who absolutely adores learning and passing it on. She works on many technologies in the CPOC, but her primary technical strength is troubleshooting. Fish has been with Cisco since 1996 and CPOC since 2001 and has been a regular speaker at Networkers/Cisco Live since 2006. Cisco Live is a huge passion for Fish! As such, in 2009, she got even more deeply involved with it by becoming a Cisco Live Session Group Manager. Look for Fish swimming in the bits and bytes all around you, or just go to www.NetworkingWithFish.com.

Dedications

Brad Edgeworth:

This book is dedicated to my family. To my wife, Tanya: Thank you for your support and patience. To my daughter, Teagan: Go after your dreams; anything is possible. And to my dog, Louie: I hope you are chasing tennis balls in heaven. I miss you.

Jean-Marc Barozet:

This book is dedicated to my supportive wife, Laurence—thank you for always being there for me—and to my beloved children, Amélie, Marie, Julien, and Pierre.

David Prall:

I would like to thank my wife, Crystal, and six children, Robbie, Cullen, Abby, Braden, Mackenzie, and Hudson, who allowed me to sit at their sporting and scouting events while working on my laptop.

Anthony Lockhart:

This book is dedicated to my family. To my wife, Colleen: Thank you for always being there for me and putting up with my insane work hours. To my lovely children, Naaman, Tailor, Keegan, and Collin: I love you! And a special shout-out to my stepdaughter, Sienna: What's up?

Nir Ben-Dvora:

Dedicated with love to my wife, Naama, and our three boys, Assaf, Aylon, and Elad, who inspire me in whatever I do.

Acknowledgments

Thank you to our technical editors, Denise and Tom, for making this a better product by pointing out our mistakes and showing us a different perspective for presenting various concepts.

Special thanks go to the Cisco Press team for their assistance and insight throughout this project.

Many people within Cisco have provided feedback, suggestions, and support to make this a great book. Thanks to all who have helped in the process, especially Steven Allspach, Sarel Altshuler, Shashikant Bhadoria, David Block, Bryan Byrne, Patrick Charretour, Gary Davis, Murali Erraguntala, Kelly Fleshner, Mani Ganesan, Jason Gooley, Amos Graitzer, Lior Katzri, Guy Keinan, Arshad Khan, Mike Korenbaum, Manish Kumar, Steve Moore, Noah Ofek, Craig Smith, Sharon Sturdy, Mike Sullenberger, Scott Van de Houten, David Yaron, Tina Yu, and the authors' management teams.

We are extremely grateful to be around all of these people who want to see this knowledge shared with others.

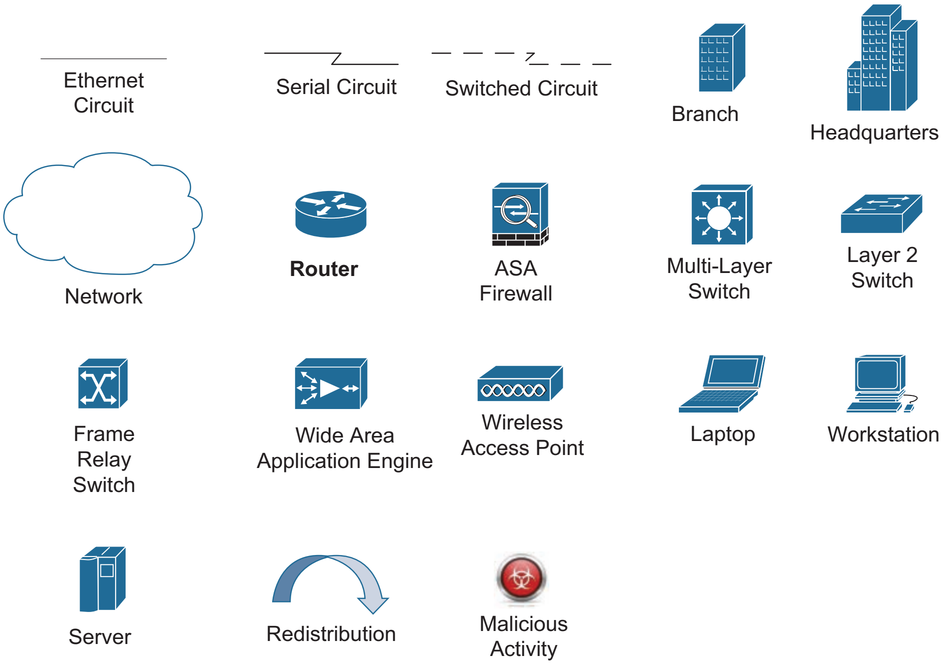
Contents at a Glance

Contents

Part V QoS

Part VI Direct Internet Access

Icons Used in This Book



Command Syntax Conventions

The conventions used to present command syntax in this book are the same conventions used in the IOS Command Reference. The Command Reference describes these conventions as follows:

- **Boldface** indicates commands and keywords that are entered literally as shown. In actual configuration examples and output (not general command syntax), boldface indicates commands that are manually input by the user (such as a **show** command).
- *Italic* indicates arguments for which you supply actual values.
- Vertical bars (|) separate alternative, mutually exclusive elements.
- Square brackets ([ ]) indicate an optional element.
- Braces ({ }) indicate a required choice.
- Braces within brackets ([{ }]) indicate a required choice within an optional element.

Foreword

The world is changing fast. And demands on the network are growing exponentially. More than ever before, businesses need technology to provide speed, flexibility, and information in a cost-effective manner across their systems and processes. The Cisco Intelligent WAN (IWAN) helps companies in any market segment connect the lifeblood of their organization—their branch locations—to business value located anywhere on the network. Whether your branch is a retail store, a healthcare clinic, or a remote office, branches are a critical component of business. These are the places where organizations interface with customers and other citizens, where most business intelligence is acquired, and where the bulk of employees work. It's crucial that the branch play a large role in any organization's plans for digitization and value.

As the leader of the Cisco Systems Engineering team, I have the privilege of working with the best networking professionals in the industry. Working on the front lines of the customer relationship, our SE teams are uniquely positioned to provide feedback from our vast customer base back to the Cisco innovation engine, our development team. Cisco has thousands of systems engineers globally working with our customers every day, and they gain great insights into the top issues facing IT and our customers' businesses in general. The feedback collected from our customers and Systems Engineering team led to the development of IWAN.

In many traditional WAN implementations, customers, vendors, suppliers, and employees who are located in the branch often cannot receive optimal service, and their capabilities are limited. The Cisco IWAN allows IT to remove those limitations by enabling intelligence on the WAN. With IWAN's ability to simplify VPNs and allow more control, applications such as guest Internet traffic, public cloud services, and partner cloud applications can be offloaded immediately with the appropriate quality of service levels. And with visibility to the application level, applications that are dependent upon data center connectivity can perform better. Last, given the need for all these use cases to be secure, you will see the value of IWAN in providing secure connectivity for your applications while providing better service and improved performance.

This book was written by an all-star team, including Brad Edgeworth, one of the key leaders in our Systems Engineering organization. Holding multiple CCIE certifications, this team of contributing authors present at both internal and external events, which means they can explain the technology and how it helps businesses. Their depth of experience and knowledge is demonstrated in this book as they address IWAN, its features, benefits, and implementation, and provide readers insight into the top issues facing IT: security, flexibility, application visibility, and ease of use. These are the most important issues facing the WAN and IT in general.

The Cisco IWAN solution helps businesses achieve their goals, and this book will help IT departments get the most out of these solutions. The book describes IWAN and its implementation in an easy-to-understand format that will allow network professionals to take full advantage of this solution in their environments. In doing so, it will allow those IT professionals to deliver tremendous business value to their organizations. At Cisco, we believe that technology can truly help businesses define their strategy and value in the market. And we believe that IT can help deliver that value through speed, agility, and responsiveness to their customers and their businesses.

Michael Koons

VP Systems Engineering and Technology,

Cisco Systems

Introduction

The Cisco Intelligent WAN (IWAN) enables organization to deliver an uncompromised experience over any WAN transport. With the Cisco IWAN architecture, organizations can provide more bandwidth to their branch office connections using cost-effective WAN transports without affecting performance, security, or reliability.

The authors' goal was to provide a multifunction self-study book that explains the technologies used in the IWAN architecture that would allow the reader to successfully deploy the technology. Concepts are explained in a modular structure so that the reader can learn the logic and configuration associated with a specific feature. The authors provide real-world use cases that will influence the design of your IWAN network.

Knowledge learned from this book can be used for deploying IWAN via CLI or other Cisco management tools such as Cisco Prime Infrastructure or Application Policy Infrastructure Controller Enterprise Module (APIC-EM).

Who Should Read This Book?

This book is for network engineers, architects, and consultants who want to learn more about WAN networks and the Cisco IWAN architecture and the technical components that increase the effectiveness of the WAN. Readers should have a fundamental understanding of IP routing.

How This Book Is Organized

Although this book can be read cover to cover, it is designed to be flexible and allow you to easily move between chapters and sections of chapters so that you can focus on just the material that you need.

Part I of the book provides an overview of the evolution of the WAN.

- **Chapter 1, "Evolution of the WAN":** This chapter explains the reasons for increased demand on the WAN and why the WAN has become more critical to businesses in any market vertical. The chapter provides an introduction to Cisco Intelligent WAN (IWAN) and how it enhances user experiences while lowering operational costs.

Part II of the book explains transport independence through the deployment of Dynamic Multipoint VPN (DMVPN).

- **Chapter 2, "Transport Independence":** This chapter explains the history of WAN technologies and the current technologies available to network architects. Dynamic Multipoint VPN (DMVPN) is explained along with the benefits that it provides over other VPN technologies.

- **Chapter 3, "Dynamic Multipoint VPN":** This chapter explains the basic concepts of DMVPN and walks the user from a simple topology to a dual-hub, dual-cloud topology. The chapter explains the interaction that NHRP has with DMVPN because that is a vital component of the routing architecture.

- **Chapter 4, "Intelligent WAN (IWAN) Routing":** This chapter explains why EIGRP and BGP are selected for the IWAN routing protocols and how to configure them. In addition to explaining the logic for the routing protocol configuration, multicast routing is explained.
- **Chapter 5, "Securing DMVPN Tunnels and Routers":** This chapter examines the vulnerabilities of a network and the steps that can be taken to secure the WAN. It explains IPsec DMVPN tunnel protection using pre-shared keys and PKI infrastructure. In addition, the hardening of the router is performed through the deployment of Zone-Based Firewall (ZBFW) and Control Plane Policing (CoPP).

Part III of the book explains how to deploy intelligent routing in the WAN.

- **Chapter 6, "Application Recognition":** This chapter examines how an application can be identified through the use of traditional ports and through deep packet inspection. Application classification is essential for proper QoS policies and intelligent routing policies.
- **Chapter 7, "Introduction to Performance Routing (PfR)":** This chapter discusses the need for intelligent routing and a brief evolution of Cisco Performance Routing (PfR). The chapter also explains vital concepts involving master controllers (MCs) and border routers (BRs) and how they operate in PfR version 3.
- **Chapter 8, "PfR Provisioning":** This chapter explains how PfRv3 can be configured and deployed in a topology.
- **Chapter 9, "PfR Monitoring":** This chapter explains how PfR can be examined to verify that it is operating optimally.
- **Chapter 10, "Application Visibility":** This chapter discusses how PfR can view and collect application performance on the WAN.

Part IV of the book discusses and explains how application optimization integrates into the IWAN architecture.

- **Chapter 11, "Introduction to Application Optimization":** This chapter covers the fundamentals of application optimization and how it can accelerate application responsiveness while reducing demand on the current WAN.
- **Chapter 12, "Cisco Wide Area Application Services (WAAS)":** This chapter explains the Cisco WAAS architecture and methods that it can be inserted into a network. In addition, it explains how the environment can be sized appropriately for current and future capacity.
- **Chapter 13, "Deploying Application Optimizations":** This chapter explains how the various components of WAAS can be configured for the IWAN architecture.

Part V of the book explains the specific aspects of QoS for the WAN.

- **Chapter 14, "Intelligent WAN Quality of Service (QoS)":** This chapter explains NBAR-based QoS policies, Per-Tunnel QoS policy, and other changes that should be made to accommodate the IWAN architecture.

Part VI of the book discusses direct Internet access and how it can reduce operational costs while maintaining a consistent security policy.

- **Chapter 15, "Direct Internet Access (DIA)":** This chapter explains how direct Internet access can save operational costs while providing additional services at branch sites. The chapter explains how ZBFW or Cisco Cloud Web Security can be deployed to provide a consistent security policy to branch network users.

Part VII of the book explains how IWAN can be deployed.

- **Chapter 16, "Deploying Cisco Intelligent WAN":** This chapter provides an overview of the steps needed to successfully migrate an existing WAN to Cisco Intelligent WAN.

The book ends with a closing perspective on the future of the Cisco software-defined WAN (SD-WAN) and the management tools that are being released by Cisco.

Learning in a Lab Environment

This book contains new features and concepts that should be tested in a lab environment first. Cisco VIRL (Virtual Internet Routing Lab) provides a scalable, extensible network design and simulation environment that includes several Cisco Network Operating System virtual machines (IOSv, IOS-XRv, CSR 1000V, NX-OSv, IOSvL2, and ASAv) and has the ability to integrate with third-party vendor virtual machines or external network devices.

The authors will be releasing a VIRL topology file so that readers can learn the technologies as they are explained in the book. More information about VIRL can be found at http://virl.cisco.com.

Additional Reading

The authors tried to keep the size of the book manageable while providing only necessary information about the topics involved. Readers who require additional reference material may find the following books to be a great supplementary resource for the topics in this book:

- Bollapragada, Vijay, Mohamed Khalid, and Scott Wainner. *IPSec VPN Design*. Indianapolis: Cisco Press, 2005. Print.
- Edgeworth, Brad, Aaron Foss, and Ramiro Garza Rios. *IP Routing on Cisco IOS, IOS XE, and IOS XR*. Indianapolis: Cisco Press, 2014. Print.
- Karamanian, Andre, Srinivas Tenneti, and Francois Dessart. *PKI Uncovered: Certificate-Based Security Solutions for Next-Generation Networks*. Indianapolis: Cisco Press, 2011. Print.
- Seils, Zach, Joel Christner, and Nancy Jin. *Deploying Cisco Wide Area Application Services*. Indianapolis: Cisco Press, 2008. Print.
- Szigeti, Tim, Robert Barton, Christina Hattingh, and Kenneth Briley Jr. *End-to-End QoS Network Design: Quality of Service for Rich-Media & Cloud Networks, Second Edition*. Indianapolis: Cisco Press, 2013. Print.

This page intentionally left blank

Chapter 1

Evolution of the WAN

This chapter covers the following topics:

- WAN connectivity
- Increasing demands on enterprise WANs
- Quality of service for the WAN
- Branch Internet connectivity and security
- Cisco Intelligent WAN

A router's primary job is to provide connectivity between networks. Designing and maintaining a LAN is straightforward because equipment selection, network design, and the ability to install or modify cabling are directly under the control of the network engineers.

WANs provide connectivity between multiple LANs that are spread across a broad area. Designing and supporting a WAN add complexity because of the variety of network transports, associated limitations, design choices, and costs of each WAN technology.

WAN Connectivity

WAN connectivity uses a variety of technologies, but the predominant methods come from *service providers (SPs)* with three primary solutions: leased circuits, Internet, and Multiprotocol Label Switching (MPLS) VPNs.

Leased Circuits

The cost to secure land rights and to purchase and install cables between two locations can present a financial barrier to most companies. Service providers can deliver dedicated circuits between two locations at a specific cost. Leased circuits can provide high-bandwidth and secure connectivity. Regardless of link utilization, leased lines provide guaranteed bandwidth between two locations because the circuits are dedicated to a specific customer.

Internet

The Internet was originally created based on the needs of the U.S. Department of Defense to allow communication even if a network segment is destroyed. The Internet's architecture has evolved so that it now supports the IP protocol (IPv4 and IPv6) and consists of a global public network connecting multiple SPs. A key benefit of using the Internet as a WAN transport is that both locations do not have to use the same SP. A company can easily establish connectivity between sites using different SPs.

When a company purchases Internet connectivity, bandwidth is guaranteed only to networks under the control of the same SP. If the path between networks crosses multiple SPs, bandwidth is not guaranteed because the peering link can be oversubscribed depending upon the peering agreement between SPs. Bandwidth for peering links is typically smaller than the bandwidth of the native SP network. At times congestion may occur on the peering link, adding delay or packet loss as packets traverse the peering link.

Figure 1-1 illustrates a sample topology in which bandwidth contention can occur on peering links. AS100 guarantees 1 Gbps of connectivity to R1 and 10 Gbps of connectivity to R3. AS200 guarantees 10 Gbps of connectivity to R4, and AS300 guarantees 1 Gbps of connectivity to R2. AS100 and AS200 peer with a 10 Gbps circuit, and AS200 peers with AS300 with two 10 Gbps circuits. With normal traffic flows R1 can communicate at 1 Gbps rates with R2. However, if R3 is transmitting 10 Gbps of data to R4, 11 Gbps of traffic must travel across the 10 Gbps circuit into AS200. Because the peering links are not dedicated to a specific customer, some traffic is delayed or dropped because of oversubscription of the 10 Gbps link. Bandwidth or latency cannot be guaranteed when packets travel across peering links.

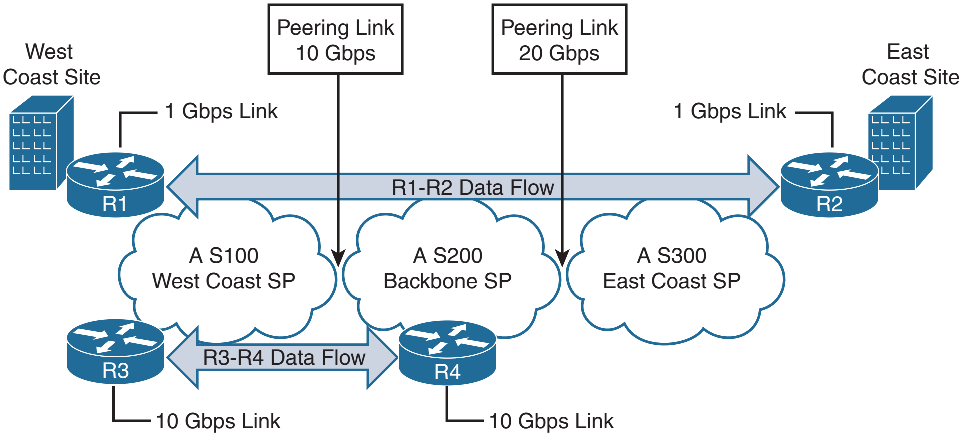


Figure 1-1 *Bandwidth Is Not Guaranteed on the Internet*

Quality of service (QoS) is based on granting preference to one type of network traffic over another. QoS design is based on trust boundaries, classification, and prioritization. Because the Internet is composed of multiple SPs, the trust boundary continually changes. Internet SPs trust and prioritize only network traffic that originates from their devices. QoS is considered a best effort when using the Internet as a transport. Some organizations may deem the Internet unacceptable because this requirement cannot be met.

Multiprotocol Label Switching VPNs (MPLS VPNs)

Service providers use MPLS to provide a scalable peer-to-peer architecture that provides a dynamic method of tunneling for packets to transit from SP router to SP router without looking into the packet's contents. Such networks forward traffic based upon the outermost label of the packet and do not require examination of the packet's header or payload. As packets cross the core of the network, the source and destination IP addresses are not checked as long as a destination label exists in the packet. Only the SP *provider edge (PE)* routers need to know how to forward unlabeled packets toward the customer router.

The MPLS VPNs are able to forward customer networks via two options depending upon the customer's requirements:

- **Layer 2 VPN (L2VPN):** The SP provides connectivity to customer routers by creating a virtual circuit between the nodes. The SP emulates a cable or network switch and does not exchange any network information with the customer routers.
- **Layer 3 VPN (L3VPN):** The SP routers create a virtual context, known as a *Virtual Route Forwarding (VRF)* instance, for each customer. Every VRF provides a method for routers to maintain a separate routing and forwarding table for each VPN network on a router. The SP communicates and exchanges routes with the *customer edge (CE)* routers. L3VPN exchanges IPv4 and IPv6 packets between PE routers.

The SPs own all the network components in an MPLS VPN network and can guarantee specific QoS levels to the customer. They price their services based on *service-level agreements (SLAs)* that specify bandwidth, QoS, end-to-end latency, uptime, and additional guarantees. The price of the connectivity typically correlates to higher demands in the SLAs to offset additional capacity and redundancy in their infrastructure.

Increasing Demands on Enterprise WANs

WAN traffic patterns have evolved since the 1990s. At first, a majority of network traffic remained on the LAN because people with similar job functions were grouped together in a building. Sharing files and interacting via email was localized, so WAN links typically transferred data between email servers, or for users accessing the corporate intranet. Over time, WAN circuits have seen an increase of network traffic as explained in the following sections.

Server Virtualization and Consolidation

Server CPUs have become faster and faster, allowing servers to do more processing. IT departments realized that consolidating file or email servers consumed fewer resources (power, network, servers, and staff) and lowered operational costs. Server consolidation reached a new height with the introduction of x86 server virtualization. Companies virtualized physical servers into *virtual machines*. An unintended consequence of server consolidation was that WAN utilization increased because servers were located at data centers (DCs), not in branch offices.

Cloud-Based Services

An organization's IT department is responsible for maintaining business applications such as word processing, email, and e-commerce. Application sponsors must work with IT to accommodate costs for staffing, infrastructure (network, workstations, and servers) for day-to-day operations, architecture, and disaster recovery.

Cloud-based providers have emerged from companies like SalesForce.com, Amazon, Microsoft, and Google. Cloud SPs assume responsibility for the cost of disaster recovery, licensing, staff, and hardware while providing flexibility and lower costs to their customers. The cost of a cloud-based solution can be spread across the length of the contract. Changing vendors in a cloud-based model does not have the same financial impact as implementing an application with in-house resources.

Connectivity to cloud providers is established with dedicated circuits or through Internet portals. Some companies prefer a dedicated circuit because they manage the security aspect of the application at the point of attachment. However, providing connectivity through the Internet gives employees the same experience whether they are in the office or working remotely.

Collaboration Services

Enterprise organizations historically maintained a network for voice and a network for computer data. Phone calls between cities were classified as long distance, allowing telephone companies to charge the party initiating the call on a per-minute basis.

By consolidating phone calls onto the data network using *voice over IP (VoIP)*, organizations were able to reduce their operating costs. Companies did not have to maintain both voice and data circuits between sites. Legacy private branch exchanges (PBXs) no longer needed to be maintained at all the sites, and calls between users in different sites used the WAN circuit instead of incurring per-minute long-distance charges.

Expanding upon the concepts of VoIP, collaboration tools such as Cisco WebEx now provide virtual meeting capability by combining voice, computer screen sharing, and interactive webcam video. These tools allow employees to meet with other employees, meet with customers, or provide training seminars without requiring attendees to be in the same geographic location. WebEx provides a significant reduction in operating costs because travel is no longer required. Management has realized the benefits of WebEx but has found video conferencing or Cisco TelePresence even more effective. These tools provide immersive face-to-face interaction, involving all participants in the meeting, thereby increasing the attention of all attendees. Decisions are made faster because of the reduced delay, and people are more likely to interact and share information with others over video.

Voice and video network traffic requires prioritization on a network. Voice traffic is sensitive to latency between endpoints, which should be less than 150 ms one way. Video traffic is more tolerant of latency than voice. Latency by itself causes a delay before the voice is heard, turning a phone call (two-way audio) into a CB radio (one-way). While this is annoying, people can still communicate. Jitter is the varying delay between packets as they arrive in a network and can cause gaps in the playback of voice or video streams. If packet loss, jitter, or latency is too high, users can become frustrated with choppy/distorted audio, video tiling, or one-way phone calls that drastically reduce the effectiveness of these technologies.

Bring Your Own Device (BYOD)

In 2010, employees began to use their personal computers, smartphones, and tablets for work. This trend is known as *bring your own device (BYOD)*. Companies allowed their employees to BYOD because they anticipated an increase in productivity, cost savings, and employee satisfaction as a result.

However, because these devices are not centrally managed, corporations must take steps to ensure that their intellectual property is not compromised. Properly designed networks ensure that BYOD devices are separated from corporate-managed devices.

Smartphones and tablets for BYOD contain a variety of applications. Some may be used for work, but others are not. Application updates are an average size of 2 MB to 25 MB; some operating system updates are 150 MB to 750 MB in size. When users update multiple applications or the operating system (OS) on their device, it consumes network bandwidth from business-related applications.

Note Some users connect their smartphones and tablets to corporate networks purely to avoid data usage fees associated with their wireless carrier contracts.

Guest Internet Access

Many organizations offer guest networks for multiple reasons, including convenience and security:

- **Convenience:** Enterprises commonly provide their vendors, partners, and visitors with Internet access as a convenience. Providing connectivity allows access to the company’s network for email, VPN access for files, or to a lab environment, making meetings and projects productive.

- **Security:** Separating the secured corporate resources (workstations, servers, and so on) from unmanaged devices creates a security boundary. If an unmanaged device becomes compromised because of malware or a virus, it cannot communicate with corporate devices.

Quality of Service for the WAN

Network users expect timely responsiveness from their network applications. Most LAN environments provide gigabit connectivity to desktops, with adequate links between network devices to prevent link saturation. Network engineers deploy QoS policies to grant preference of one type of network traffic over a different type. Although QoS policies should be deployed everywhere in a network, they are a vital component of any WAN edge design, where bandwidth is often limited because of cost and/or availability.

Media applications (voice and/or video) are sensitive to delay and packet loss and are often granted the highest priority in QoS policies. Typically, non-business-related traffic (Internet) is assigned the lowest QoS priority (best effort). All other business-related traffic is categorized and assigned an appropriate QoS priority and bandwidth based upon the business justification.

A vital component of QoS is the classification of network traffic according to the packet's header information. Typically traffic is classified by class maps, which use a combination of protocol (TCP/UDP) and communication ports. Application developers have encountered issues with traffic passing through corporate firewalls on nonstandard ports or protocols. They have found methods to tunnel their application traffic over port 80, allowing instant messaging (IM), web conferencing, voice, and a variety of other applications to be embedded in HTTP. In essence, HTTP has become the new TCP.

HTTP is not sensitive to latency or loss of packets and uses TCP to detect packet loss and retransmission. Network engineers might assume that all web-browsing traffic can be marked as best effort because it uses HTTP, but other applications that are nested in HTTP can be marked incorrectly as well.

Deep packet inspection is the process of looking at the packet header and payload to determine the actual application for that packet. Packets that use HTTP or HTTPS header information should use deep packet inspection to accurately classify the application for proper QoS marking. Providing proper network traffic classification ensures that the network engineers can deploy QoS properly for every application.

Branch Internet Connectivity and Security

The Internet provides a wealth of knowledge and new methods of exchanging information with others. Businesses host web servers known as *e-commerce* servers to provide company information or allow customers to shop online. Just as with any aspect of society, criminals try to obtain data illegally for personal gain or blackmail. Security is deployed in a layered approach to provide effective solutions to this problem.

Firewalls restrict network traffic to e-commerce servers by specifying explicit destination IP addresses, protocols, and ports. Email servers scan email messages for viruses and phishing attempts. Hackers have become successful at inserting viruses and malware into well-known and respected websites. Content-filtering servers can restrict access to websites based on the domain-based classification and can dynamically scan websites for malicious content.

Internet access is provided to the branch with either a centralized or a distributed model. Both models are explained in the following sections.

Centralized Internet Access

In the centralized Internet access model, one centralized or regional site provides Internet connectivity. This model simplifies the management of Internet security policy and device configuration because network traffic flows through a minimal number of access points. This reduces the size of the security infrastructure and its associated maintenance costs.

The downside of the centralized model is that all network traffic from remote locations to the Internet is also backhauled across the WAN circuit. This can cause congestion on the enterprise WAN and centralized Internet access circuits during peak usage periods unless the Internet circuit contains sufficient bandwidth for all sites and the WAN circuits are sized to accommodate internal network traffic as well as the backhauled Internet traffic. Although Internet circuits have a low cost, the backhauled network traffic travels on more expensive WAN circuits. In addition, backhauling Internet traffic may add latency between the clients and servers on the Internet. The latency occurs for recreational web browsing as well as access to corporate cloud-based applications.

Figure 1-2 illustrates the centralized Internet model. All Internet traffic from R2 or R3 must cross the WAN circuit where it is forwarded out through the headquarters Internet connection.

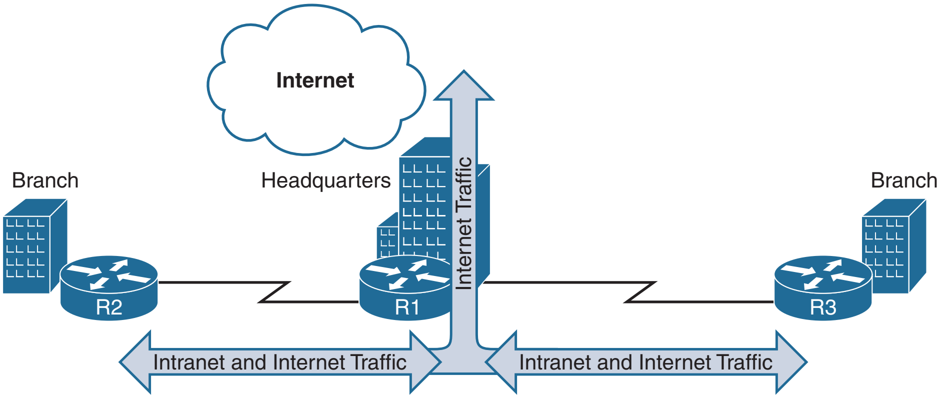


Figure 1-2 *Centralized Internet Connectivity Model*

Distributed Internet Access

In the distributed Internet access model, Internet access is available at all sites. Access to the Internet is more responsive for users in the branch, and WAN circuits carry only internal network traffic. Figure 1-3 illustrates the distributed Internet model. R2 and R3 are branch routers that can provide access to the Internet without having to traverse the WAN links. R2 and R3 route packets to the Internet out of their Internet circuits, reducing the need to backhaul Internet traffic across costly WAN circuits.

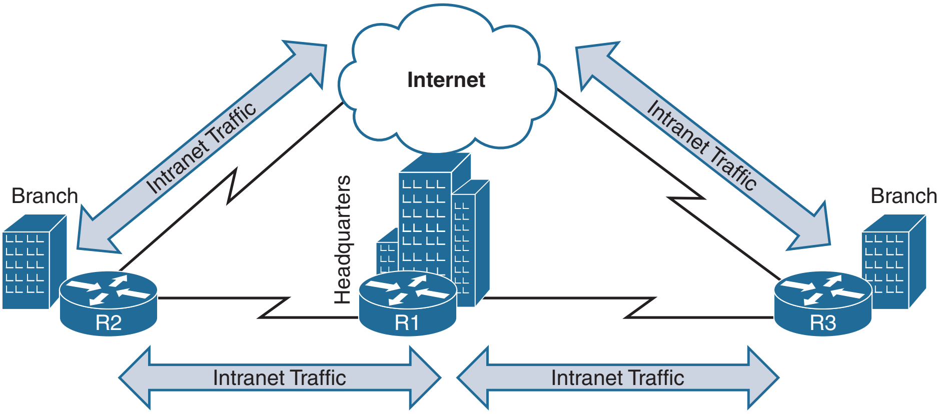


Figure 1-3 *Distributed Internet Connectivity Model*

This model requires that the security policy be consistent at all sites, and that appropriate devices be located at each site to enforce those policies. These requirements can be a burden to some companies' network and/or security teams.

Cisco Intelligent WAN

Cisco Intelligent WAN (IWAN) architecture provides organizations with the capability to supply more usable WAN bandwidth at a lower cost without sacrificing performance, security, or reliability. Cisco IWAN is based upon four pillars: transport independence, intelligent path control, application optimization, and secure connectivity.

Transport Independence

Cisco IWAN uses *Dynamic Multipoint VPN (DMVPN)* to provide transport independence via overlay routing. Overlay routing provides a level of abstraction that simplifies the control plane for any WAN transport, allowing organizations to deploy a consistent routing design across any transport and facilitating better traffic control and load sharing, and supports routing protocols, removing any barriers to *equal-cost multipathing (ECMP)*. Overlay routing provides transport independence so that a customer can select any WAN technology: MPLS VPN (L2 or L3), metro Ethernet, direct Internet, broadband, cellular 3G/4G/LTE, or high-speed radios. Transport independence makes it easy to mix and match transport options or change SPs to meet business requirements.

For example, a new branch office requires network connectivity. Installing a physical circuit can take an SP six to 12 weeks to provision after the order is placed. If the order is not placed soon enough or complications are encountered, WAN connectivity for the branch is delayed. Cisco IWAN's transport independence allows the temporary use of a cellular modem until the physical circuit is installed without requiring changes to the router's routing protocol configuration, because DMVPN resides over the top of the cellular transport. Changing transports does not impact the overlay routing design.

Intelligent Path Control

Routers forward packets based upon destination address, and the methodology for path calculation varies from routing protocol to routing protocol. Routing protocols do not take into consideration packet loss, delay, jitter, or link utilization during path calculation, which can lead to using an unsuitable path for an application. Technologies such as IP SLAs can measure the path's end-to-end characteristics but do not modify the path selected by the routing protocol.

Performance Routing (PfR) provides intelligent path control on an application basis. It monitors application performance on a traffic class basis and can forward packets on the best path for that application. In the event that a path becomes unacceptable, PfR can switch the path for that application until the original path is within application specifications again. In essence, PfR ensures that the path taken meets the requirements set for that application.

PfR has been enhanced multiple times for Cisco intelligent path control, integrating with DMVPN and making it a vital component of the IWAN architecture. It provides improved application monitoring, faster convergence, simple centralized configuration, service orchestration capability, automatic discovery, and single-touch provisioning.

Providing a highly available network requires elimination of *single points of failure (SPoFs)* to accommodate hardware failure and other failures in the SP infrastructure. In addition to redundancy, the second circuit can provide additional bandwidth with the use of transport independence and PfR. This can reduce WAN operating expenses in any of the IWAN deployment models.

Figure 1-4 depicts a topology that provides R1 connectivity to R5 across two different paths. R1 and R5 have identified DMVPN tunnel 100 as the best path with the routing protocol used and continue to send VoIP traffic up to that tunnel's capacity. R1 uses the same tunnel for sending and transferring files. The total amount of network traffic exceeds tunnel 100's bandwidth capacity. The QoS policies on the tunnel ensure that the VoIP traffic is not impacted, but file transfer traffic is impacted. The DMVPN tunnel 200 could be used to transfer files with intelligent path control.

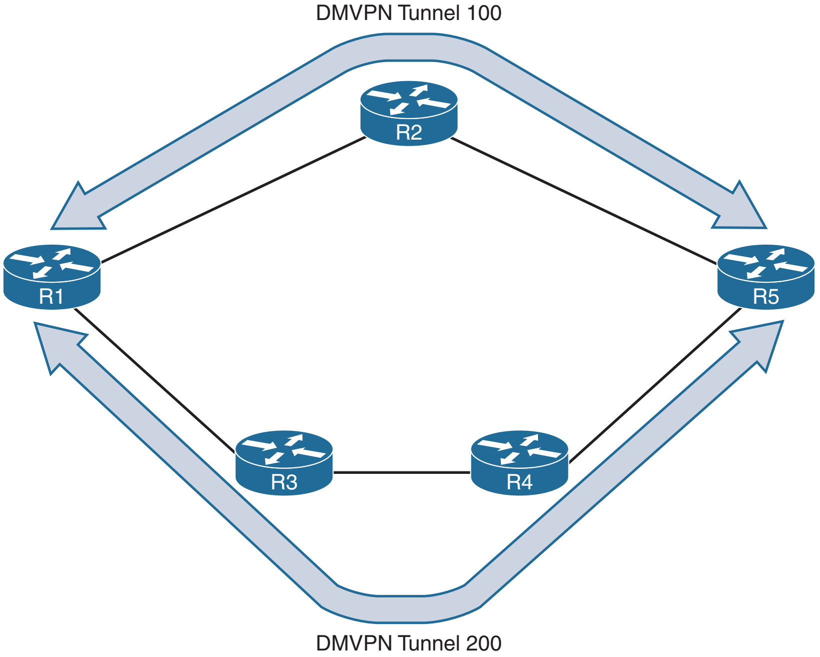


Figure 1-4 *Path Optimizations with Intelligent Path Control*

PfR overcomes scenarios like the one described previously. With PfR, R1 can send VoIP traffic across DMVPN tunnel 100 and send file transfer traffic toward DMVPN tunnel 200. PfR allows both DMVPN tunnels to be used while still supporting application requirements and not dropping packets.

Note Some network engineers might correlate PfR with MPLS traffic engineering (TE). MPLS TE supports the capability to send specific marked QoS traffic down different TE tunnels but lacks the granularity that PfR provides for identifying an application.

Application Optimization

Most users assume that application responsiveness across a WAN is directly related to available bandwidth on the network link. This is an incorrect assumption because application responsiveness directly correlates to the following variables: bandwidth, path latency, congestion, and application behavior.

Most applications do not take network characteristics into account and rely on underlying protocols like TCP for communicating between computers. Applications are typically designed for LAN environments that provide high-speed links that do not have congestion and are “chatty.” Chatty applications transmit multiple packets in a back-and-forth manner, requiring an acknowledgment in each direction.

Cisco Wide Area Application Service (WAAS) and Akamai Connect technologies provide a complete solution for overcoming the variables that impact application performance across a WAN circuit. They are transparent to the endpoints (clients and servers) as well as devices between the WAAS/Akamai Connect devices.

Cisco WAAS incorporates *data redundancy elimination (DRE)* technology to identify data patterns in network traffic and reduce the patterns with a signature as the packets traverse the network. Cisco WAAS examines packets, looking for patterns in 256-byte, 1 KB, 4 KB, and 16 KB increments, and creates a signature for each of those patterns. If a pattern is sent a second time, the first WAAS device replaces the data with a signature. The signature is sent across the WAN link, and the second WAAS device replaces the signature with the data. This drastically reduces the size of the packet as it crosses the WAN but keeps the original payload between the communicating devices.

Cisco WAAS and Akamai Connect also provide a method of caching objects locally. Caching repeat content locally shrinks the path between two devices and can reduce latency on chatty applications. For example, the latency between a branch PC and the local object cache (WAAS/Akamai Connect) is 5 ms, which is a shorter delay than waiting for the file to be retrieved from the server, which takes approximately 100 ms. Only the initial file transfer takes the 100 ms delay, and subsequent requests for the same file are provided locally from the cache with a 5 ms response time.

Secure Connectivity

A router's primary goal is to forward packets to a destination. However, corporate routers with direct Internet access need to be configured appropriately to protect them from malicious outsiders so that users may access external content (Internet) but external devices can access only appropriate corporate resources. The following sections describe the components of IWAN's secure connectivity pillar.

Zone-Based Firewall

Access control lists (ACLs) provide the first capability for filtering network traffic on a router. They control access based on protocol, source IP address, destination IP address, and ports used. Unfortunately they are stateless and do not inspect packets to detect if hackers are using a port that they have found open.

Stateful firewalls are capable of looking into Layers 4 through 7 of a network packet and verifying the state of the transmission. Stateful firewalls can detect if a port is being piggybacked and can mitigate *distributed denial of service (DDoS)* intrusions.

Cisco Zone-Based Firewall (ZBFW) is the latest integrated stateful firewall in Cisco routers that reduces the need for a second security device. Cisco ZBFW uses a zone-based configuration. Router interfaces are assigned to specific security zones, and then interzone traffic is explicitly permitted or denied based on the security policy. This model provides flexibility and overcomes the administration burden on routers with multiple interfaces in the same security zone.

Cloud Web Security

Ensuring a consistent security policy for distributed Internet access can cause headaches for the network and *information security (InfoSec)* engineers who must support it. Most businesses deploy a content security device at each location, resulting in additional hardware cost and increasing the management of security devices.

Cisco IWAN routers can use distributed Internet access while also providing a consistent, centrally managed security policy by connecting to Cisco Cloud Web Security (CWS). Cisco CWS provides content security with unmatched zero-day threat and malware protection for any organization without the burden of purchasing and managing a dedicated security appliance for each branch location.

In essence, all HTTP/HTTPS traffic exiting a branch over the WAN is redirected (proxied) to one of the closest CWS global data centers. At that time, the access policy is checked based on the requesting user, location, or device to verify proper access to the Internet. All traffic is scanned for potential security threats. This allows organizations to use a distributed Internet access architecture while maintaining the security required by InfoSec engineers.

Software-Defined Networking (SDN) and Software-Defined WAN (SD-WAN)

Managing all the components of a network can be a daunting task. *Software-defined networking (SDN)* enables organizations to accelerate application deployment and delivery, dramatically reducing IT costs through policy-enabled workflow automation. The SDN technology enables cloud architectures by delivering automated, on-demand application delivery and mobility at scale. It enhances the benefits of DC virtualization, increasing resource flexibility and utilization and reducing infrastructure costs and overhead.

SDN accomplishes these business objectives by consolidating the management of network and application services into centralized, extensible orchestration platforms that can automate the provisioning and configuration of the entire infrastructure. Common centralized IT policies bring together disparate IT groups and workflows. The result is a modern infrastructure that can deliver new applications and services in minutes, rather than days or weeks as was required in the past.

SDN has been primarily focused on the LAN. *Software-defined WAN (SD-WAN)* is SDN directed strictly toward the WAN. It provides a complete end-to-end solution for the WAN and should remove the complexity of deciding on transports while meeting the needs of the applications that ride on top of it. As applications are deployed, the centralized policy can be updated to accommodate the new application. Combining the Cisco prescriptive IWAN architecture with the *Application Policy Infrastructure Controller—Enterprise Module (APIC-EM)* simplifies WAN deployments by providing a highly intuitive, policy-based interface that helps IT abstract network complexity and design for business intent. The business policy is automatically translated into network policies that are propagated across the network. This solution enables IT to quickly realize the benefits of an SD-WAN by lowering costs, simplifying IT, increasing security, and optimizing application performance.

Summary

This chapter provided an overview of the technologies and challenges faced by network architects tasked with deploying a WAN for organizations of any size. The Cisco IWAN architecture was developed to deliver an uncompromised user experience over any WAN technology. Network engineers are able to address the evolution and increase of WAN traffic by providing more bandwidth to their remote sites by

- Removing complications with a specific WAN circuit type by providing transport independence with DMVPN tunnels
- Increasing application performance and link utilization for all circuits while using intelligent path control (PfRv3)
- Implementing optimization technologies that reduce bandwidth consumption across a WAN circuit and enabling a local cache to reduce latency
- Migrating from a centralized Internet connectivity model to a distributed Internet connectivity model while maintaining a consistent security policy with Cisco CWS

The Cisco Intelligent WAN solution provides a scalable architecture with an ROI that can be measured in months and not years. In addition to cost savings, improved application responsiveness directly correlates to increased business productivity. The Cisco policy and orchestration tools (APIC-EM) simplify the deployment of services and deliver a complete SD-WAN solution.

This page intentionally left blank

Chapter 2

Transport Independence

This chapter covers the following topics:

- WAN transport technologies
- Peer-to-peer networks
- Virtual private networks
- Benefits of transport independence

The primary focus of WANs is to provide the ability to exchange network traffic across geographic distances. In addition, scalability and cost are important factors for the design of a WAN. The provisioning and operation of a WAN can be a complex task for any network engineer because of the variety and behaviors of technologies available.

WAN Transport Technologies

Every technology has different costs and features (technical and ability to scale) that impact the overall design strategy for a WAN. The following sections provide an overview of WAN technologies as they were developed.

Dial-Up

Computers and routers connect with each other with analog modems that use the telephone companies' *public switched telephone network (PSTN)*. Using the existing PSTN infrastructure provides flexibility because connectivity can be established dynamically to any phone number. Telephone lines are required to support only voice, which needs 9600 bps of bandwidth. The speed that the first modem supported, 300 bps, has increased over time to up to 56,000 bps (56 kbps) where audio quality is ideal. Occasionally line quality can cause the modem session to disconnect, resulting in network routing convergence or packet loss while the call is reestablished. Dial-up access is a low-cost, flexible solution when temporary low-bandwidth connectivity is required.

Note Running routing protocols over a dial-up line can consume additional bandwidth and can cause additional convergence in the routing domain when the dial-up session disconnects. Typically static routes are used with dial-up connectivity.

Leased Circuits

Overcoming the speed barrier of dial-up connectivity requires the use of dedicated network links between the two sites. The cost to secure land rights, purchase cables and devices, and install the cables and acquire the hardware presents a financial barrier to most companies.

Telephone companies use the existing infrastructure and sell access between devices as dedicated leased lines. Leased lines provide high-bandwidth secure connections but lack the flexibility of dial-up. Regardless of link utilization, leased lines provide guaranteed bandwidth because the circuits are dedicated between sites in the SP environment.

Most SPs now install fiber-optic cables between locations; each cable contains multiple strands that support multiple connections. Only a portion of the cable strands is consumed, which means that additional capacity can be consumed internally or leased to other companies. Unused fiber-optic cables are called *dark fiber* circuits. Some companies specialize in locating and reselling dark fiber circuits. Customers are provided with a transport medium but are responsible for installing devices at each end for any routing or switching capabilities. Dark fiber is not really a service but more of an asset.

Organizations can take advantage of the full capability of fiber-optic cable (commonly 10 Gbps) by using *dense wavelength-division multiplexing (DWDM)* devices. These allow a 10 Gb interface to be assigned a specific optical wavelength, and because some fiber-optic cable supports multiple optical wavelengths, a single strand of cable supports multiple 10 Gb links. Bandwidth is guaranteed because only two devices can communicate on the same wavelength, and no wavelength conflicts with other wavelengths.

Virtual Circuits

Early connectivity for PSTN and leased lines allowed only point-to-point communication between two devices on the circuit. This complicated capacity planning from a telephone provider's perspective and kept operating costs high. As the demand increased for circuits to carry data traffic in lieu of voice traffic, the need for a more efficient infrastructure increased.

Technologies like X25, Frame Relay, and ATM introduced the concept of sharing a line through the use of virtual circuits. Virtual circuits provide a tunnel for traffic to follow a specific path in an SP's network. They allow multiple conversations to exist on physical links, reducing the SP's operational costs and allowing communication with multiple devices using the same physical interface.

Virtual circuits provide the following advantages:

- Communication is allowed with more than two devices at the same time using the same interface.
- Interfaces can operate with different bandwidth settings between devices.

Note Virtual circuits introduced the capability for point-to-multipoint, full-mesh, and partial-mesh Layer 2 networks that use the same Layer 3 network address. Point-to-multipoint and partial-mesh topologies deviate from standard broadcast and point-to-point transports and may require special configuration of routing protocols to work properly.

Peer-to-Peer Networks

With improvements in the Ethernet protocol, Ethernet links can support speeds of from 1 Mbps to 100 Gbps, exceeding other serial technologies such as T1 and SONET. For example, OC-768 serial circuits can transport only up to 38.5 Gbps. In addition, Ethernet is more cost-effective than serial technologies when the cost is compared on a per-bit basis.

The amount of bandwidth required at a remote site can vary depending upon the number of users and data services required at each site. Some remote sites may be able to support Ethernet connectivity, but other sites can connect only via serial links because of circuit length requirements. If two sites connect via two different transport media (such as Ethernet and serial) and must be able to communicate with each other, the SP typically provides the connectivity indirectly by routing (Layer 3) between the two sites.

Figure 2-1 illustrates an SP providing connectivity between Site 1 and Site 2 using different network media. There are CE routers located at the edge of the customer network and on the customer's site. The CE routers connect to PE routers that are located at the SP's site. The PE routers connect to *provider (P)* routers or other PE routers.

CE1 connects to PE1 via an Ethernet link (172.16.11.0/24), and CE3 connects to PE3 via a serial link (172.16.33.0/24). CE1 exchanges routes with PE1 via a dynamic routing protocol which is then advertised to P2 and then on to PE3. CE2's routes are exchanged in an identical fashion toward CE1. In this model, the SP network provides network connectivity to both locations for both customers.

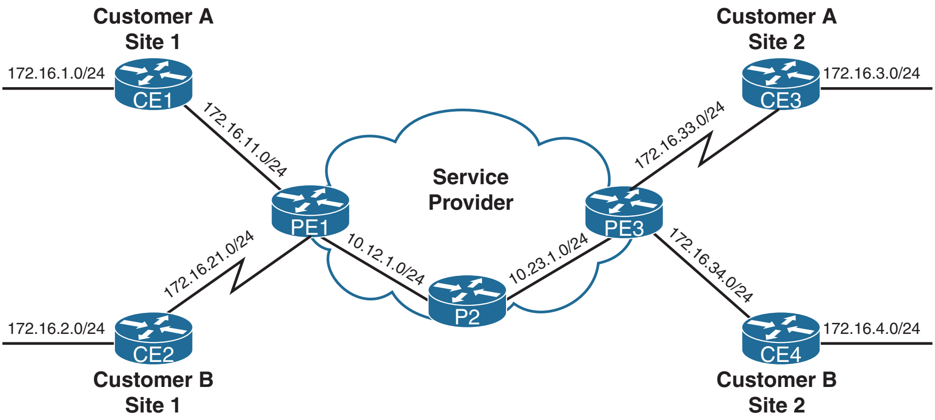


Figure 2-1 *Peer-to-Peer Connectivity*

A peer-to-peer network distributes all aspects of the network among all its peers. A problem with peer-to-peer networks is that the SP participates in the network. Any existing networks in the SP's network cannot be used in the customer's network, and all IP address spaces are shared with everyone attached to it, including other customers.

In Figure 2-1, the peer-to-peer network is shared between two customers (Customer A and Customer B). Customer A cannot use the 172.16.2.0/24 network or the 172.16.4.0/24 network because Customer B is using them. This limitation can cause issues when two different companies use private IP addressing ranges (RFC 1918) and want to use the same network segment.

Note The Internet is the world's largest peer-to-peer network.

Broadband Networks

Home consumers have requested more and more bandwidth at an economical scale. Because of the large geographic area that required service, telephone, and cable companies needed a technology that could reuse portions of their existing infrastructure while providing more bandwidth to consumers. Multiplexing (combining) multiple frequencies on the same link enabled them to provide a higher amount of bandwidth to consumers. These types of networks are referred to as *broadband networks* because they combine multiple frequencies (bands) to provide bandwidth.

Broadband networks exist in two formats:

- **Cable modem:** Connectivity is provided through the existing coaxial infrastructure used by cable television providers. Initial cable modem speeds were 1 to 2 Mbps but currently provide 100+ Mbps. Because the coaxial infrastructure is shared among several locations, the bandwidth is shared and can fluctuate based on a neighbor's bandwidth usage. Although bandwidth may peak to what is stated in the SLA, setting the bandwidth to the average is recommended.

- **DSL:** There are different variants of DSL, but in essence it uses the existing copper wires in the PSTN network that provide telephone service to a house. DSL uses higher frequencies that humans cannot hear to transmit data. It does not use a shared infrastructure but is generally limited to a small distance (about 2 to 5 km) from the telephone company's head-end device. The farther away from the head-end device, the lower the available bandwidth. DSL can reach speeds in excess of 40 Mbps.

Broadband has since expanded in scale and is no longer considered a home consumer technology. Many telephone and cable companies also provide broadband services to businesses.

Cellular Wireless Networks

With the invention of the smartphone, cellular wireless providers allowed phones to send and transmit data in addition to placing voice calls. Cellular wireless networks provide the following benefits:

- **Reachability:** Not all businesses operate in places where an SP has circuits. In some remote locations (such as mountains or islands) cables for connectivity may not exist. The coverage of cellular wireless networks enables businesses to take the network into geographic locations that were never available via wired connections.
- **Backup connectivity:** The speed of cellular networks offers businesses the ability to run their operations over what was typically used as an out-of-band management link. This means that cellular wireless networks can provide a second connection where multiple SPs are not available. This includes voice and video support over cellular in the event of a primary network outage.
- **Timeliness:** Terrestrial circuits can typically take up to 90 days to deploy at new locations. The quick activation of cellular allows businesses to bring locations and services online in days or even hours instead of weeks or months.
- **Ability to host networks that change in location:** Some mobile vendors (such as food trucks) need network connectivity to verify credit card transactions. Another growing sector is the use of cellular wireless networks to track a patient's vital signs while the patient is at home, reducing the patient's need to visit a healthcare facility, or making care available to patients who are unable to travel.
- **Ability to host networks in motion:** Some businesses require the ability to provide network connectivity to devices in motion. For example, passenger trains may provide WiFi service to their passengers, track critical on-board sensors, or combine GPS sensors to track the vehicle location.

Virtual Private Networks (VPNs)

Ensuring that an SP can provide connectivity to two different locations can be a problem. An SP may have only regional connectivity and not global capabilities. Providing connectivity to customers with a global footprint is not an easy task for many SPs, particularly those with only a regional presence. In the event that a global SP cannot be found, regional SPs must be interconnected. This scenario increases operational complexity when connectivity must be provided between sites in different countries or continents. Scenarios like this may prevent a peer-to-peer solution; however, Internet connectivity is common throughout the world and easy to obtain almost anywhere.

A VPN provides connectivity to private networks over a public network, such as the Internet. A VPN operates by tunneling, encrypting the payload, or both. With VPN tunneling, packets destined to travel between private networks are encapsulated and assigned new packet headers that allow the packets to traverse the public network. A VPN tunnel is classified as an *overlay network* because the VPN network is built on top of an existing transport network, also known as an *underlay network*. After authenticating with the remote VPN endpoint, the VPN tunnel is established for packets to traverse to the private network. A VPN can leverage public network transport such as the Internet to provide global connectivity using only one transport.

Note VPNs are not restricted to use on the Internet. Many organizations use VPNs across corporate networks to ensure that sensitive data is not compromised in transit between geographic locations.

Within VPN tunnels, the new packet headers provide a method of forwarding a packet across the public network without exposing the private network's original packet headers. This allows the packet to be forwarded between the two endpoints without requiring any routers to extract information from the payload (original packet headers and data). After the packet reaches the remote endpoint, the VPN tunnel headers are decapsulated (removed). The endpoint checks the original headers and then forwards the packet through the appropriate interface to the private network.

There are multiple VPN protocols with various benefits and design considerations for encapsulating, encrypting, and transferring data across public networks. *Internet Protocol Security (IPsec)*, *Secure Sockets Layer (SSL)*, *Datagram Transport Layer Security (DTLS)*, and *Point-to-Point Tunneling Protocol (PPTP)* are some of the most common protocols used today for VPNs. The following sections explain the most common VPN types.

Remote Access VPN

A company's employees or partners can install VPN software on a client device (workstation, tablet, or phone) that can establish a VPN tunnel to the VPN server. The client must know the VPN server's public IP address, but the VPN server does not need to know the client's public IP address in advance. The VPN server assigns a private IP address to the client so that devices in the remote private network know where to send return network traffic.

Typically the VPN server defines any access or security policies on the client. The VPN server can require that all traffic from the client be sent across the VPN tunnel, known as *full tunnel*, or specify that traffic to only specific destination networks route across the VPN tunnel, known as *split tunneling*.

Site-to-Site VPN Tunnels

Remote access VPN solutions do not scale well for locations with multiple VPN clients like branch offices. A better scalable solution uses a dedicated VPN client (router, firewall, or security appliance) to provide connectivity to the remote VPN server via a site-to-site VPN tunnel.

Each VPN endpoint must know the public IP address of the other. In addition, the endpoints must identify the *interesting traffic* (source and destination networks) that will use the VPN tunnel. If there is a mismatch of interesting traffic, either the packets are not encapsulated properly or the packets are not decapsulated and are dropped.

When the first VPN endpoint receives interesting traffic, the VPN tunnel is established. Private network packets are encapsulated, routed across the public network toward the remote endpoint, and decapsulated. Depending on the endpoint configuration, the VPN tunnel remains up indefinitely, or an idle timer can be configured. The idle timer is reset if packets come across the tunnel. If the timer reaches zero, the VPN tunnel is torn down.

Note Typical IPsec tunnels (those that are not generic routing encapsulated [GRE]) do not assign a network (subnet) to the tunnel, which prevents Interior Gateway Protocol (IGP) routing protocols from operating on the tunnel. Advertising the remote routes to downstream devices requires the use of *reverse-route injection* or other techniques.

Hub-and-Spoke Topology

Organizations with more than two locations typically establish site-to-site VPNs between the headquarters and the branch sites using a *hub-and-spoke* network topology. Network traffic from one branch to a different branch must travel from the branch (spoke) to the headquarters (hub) location and then back to the remote branch.

Figure 2-2 illustrates a typical hub-and-spoke topology between R1, R2, and R3.

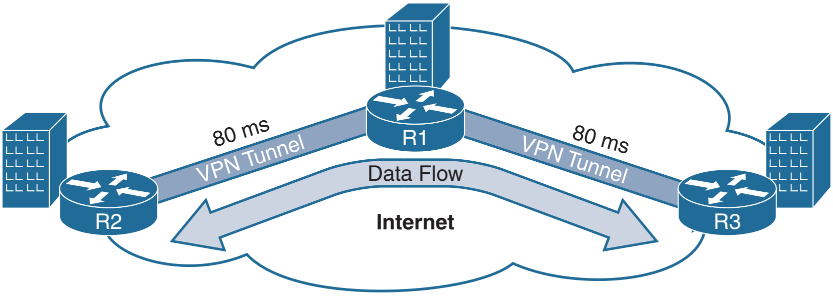


Figure 2-2 *Hub-and-Spoke Topology with Site-to-Site VPNs*

R1 is the hub, and R2 and R3 are spokes. Network latency from either spoke to the hub is 110 ms, which is generally an acceptable latency for most applications. The end-to-end latency between spokes (R2 to R3) is 220 ms, which may be acceptable for most applications but can cause problems for others. For example, VoIP recommends a maximum delay of 150 ms. Exceeding the recommended delay can affect call audio quality for users at R2 talking with users at R3.

Allocating proper bandwidth for the hub site is critical. The hub site is the typical destination for most network traffic, but traffic between other spokes must transit the hub circuit twice, one time inbound and another leaving the hub to the other branch site. Although oversubscription of spoke bandwidth is common, bandwidth oversubscription at the hub site should not be common.

Note Adding a new site to a hub-and-spoke topology requires additional configuration (endpoint IP addressing, defining interesting traffic, cryptography, route table manipulation) on the new spoke router and on the hub router. The formula *n*∗2 provides the number of VPN interfaces (tunnel endpoints) that must be configured for the routing domain, where *n* reflects the number of sites. In Figure 2-2 there are two sites, which require four VPN tunnel interfaces (two on R1, one on R2, and one on R3).

Full-Mesh Topology

To address some of the deficiencies of the hub-and-spoke model, VPN tunnels can be established between R2 and R3, thereby forming a full-mesh topology. Doing so can reduce the end-to-end latency between all sites, assuming that the latency in the transport network between R2 and R3 is less. In this scenario the latency is 110 ms, which provides sufficient voice call quality and improves the responsiveness of other applications. In addition, network traffic between the sites does not consume bandwidth from R1 unnecessarily.

Figure 2-3 shows all three sites with direct site-to-site connectivity. This illustrates a full-mesh topology because all sites are connected to all other sites regardless of size.

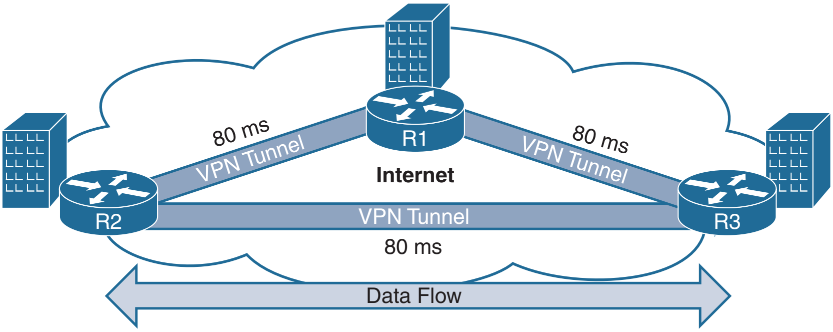


Figure 2-3 *Full-Mesh Topology with Site-to-Site VPNs*

Full-mesh topologies using site-to-site VPN tunnels can introduce a variety of issues as the number of sites increases. Those issues are

- **Manageability of VPN endpoints:** When a new site needs connectivity, the VPN tunnel interface needs to be configured on the new router and the remote router. The formula $n(n-1)$ provides the number of VPN tunnel interfaces that must be configured, where n reflects the number of sites. In Figure 2-3 there are three sites, but the topology requires the creation of six VPN tunnel endpoints. The problem is further exacerbated when 10 sites are required because 90 VPN tunnel interfaces must be managed.
- **Consumption of router CPU and memory**: Maintaining a high number of active VPN tunnels consumes more CPU and memory resources. All routers need to be sized accordingly to accommodate any load generated for maintaining the VPN tunnels.

Most VPN designs for large organizations typically place a hub in a major geographic region. The hub is connected via full-mesh, and then the interregion traffic is contained via a hub-and-spoke model.

Multiprotocol Label Switching (MPLS) VPNs

Service providers use *Multiprotocol Label Switching (MPLS)* to provide a scalable peer-to-peer architecture that allows packets to transit the SP network from PE router to PE router without the need to look into the packets' contents. The MPLS VPNs use at least two MPLS labels, one for the forwarding of packets between PE routers and the other to identify which customer's network the packet belongs to. Because a customer network is specified in the second MPLS label, IP addressing is not shared between customers. This allows the SP to offer VPN connectivity for multiple customers that often use the same private IP address space (RFC 1918).

An MPLS network forwards traffic based upon the outermost MPLS label of a packet. The MPLS labels are placed before the IP headers (source IP and destination IP), so none of the transit routers require examination of the packet's IP header or payload. As packets cross the core of the network, the source and destination IP addresses are never checked as long as the forwarding label exists in the packet. Only the PE routers need to know how to send the packets toward the CE router. An MPLS VPN is considered an overlay network because network traffic is forwarded on the SP's underlay network using MPLS labels.

All MPLS VPNs are categorized by two technologies on the PE router: Layer 2 and Layer 3 VPN.

Layer 2 VPN (L2VPN)

PE routers provide connectivity to customer routers by creating a virtual circuit between two nodes. Packets are received on an interface, labeled with a circuit ID, then labeled for the remote PE.

Virtual Private LAN Service (VPLS) is an evolution of L2VPN that provides multisite access to the same bridge domain. A VPLS includes logic to maintain MAC address table mappings to a specific PE router and provides the capability of LAN switching to devices spread across large geographic areas.

An L2VPN can be set up with topologies of point-to-point, point-to-multipoint, or full-mesh. This technology allows for all packets to be exchanged between PE routers.

Layer 3 VPN (L3VPN)

A PE router uses a virtual context known as *Virtual Route Forwarding (VRF)* for each customer. In order to provide segmentation and maintain exclusive communications between customer sites, every VRF context provides a method for routers to maintain a separate routing and forwarding table for each customer.

The PE routers must contain all the routes for a particular customer (VRF), whereas the CE routers require only a default route to the PE router. The route table on the PE routers can be programmed via a static route at the local PE router or is advertised from the CE router. A PE router exchanges the VRF's routes with other PE routers using *Multiprotocol Border Gateway Protocol (MBGP)* using a special address family just for MPLS L3VPN networks. There are VPN labels associated to each of the VRF's routes to identify which VRF the routes belong to.

A CE router can use only a static default route toward the closest PE router, and the PE routers are responsible for maintaining and exchanging the complete routing table for that VRF using MBGP. But the CE router can use a routing protocol to dynamically advertise networks to the PE routers. This allows the PE routers to have a complete routing table that is dynamically learned versus statically configured. BGP is commonly used to exchange routes between CE and PE routers because of its scalability and capability for route policy manipulation.

MPLS is enabled in the SP network to forward traffic between PE routers. A packet is forwarded based upon the outermost label of the packet that references the destination PE router. Only the edge PE routers need to examine other portions of the packet (internal MPLS labels, packet headers, and payload) as they send traffic to the CE router.

Using MPLS VPNs provides reliable bandwidth because the SP owns the entire infrastructure. Deploying a new site requires connecting a new circuit between the CE router and the PE router. Minor configuration is needed on the PE router with the new circuit, and then that remote site has full connectivity to all other sites in the MPLS VPN.

The primary difference between L2VPNs and L3VPNs is how the CE-PE relationship is handled. L2VPNs are concerned only from the Layer 2 perspective, whereas L3VPNs participate in the routing table of the customer networks. Both models provide security to the customer networks through circuit/network segmentation, and the SP network is invisible to the customer. Each model has its advantages and disadvantages.

MPLS VPNs and Encryption

A common misunderstanding of network and information security engineers is whether the payload is encrypted for MPLS VPNs. MPLS VPNs do not typically encrypt packets that travel across them. MPLS labels are used to forward the packets and associate them to a VRF (for L3VPN) or to a circuit (for L2VPN). It is possible for data to leak between customers if there is a misconfiguration within the SP network.

Some organizations that are subject to stringent privacy regulations (such as healthcare, government, and financial) often require all traffic to be encrypted across all WAN links; other organizations encrypt traffic only where they feel vital data can be compromised. In instances like these, a second technology to encrypt and decrypt the traffic must be deployed.

Link Oversubscription on Multipoint Topologies

Routers are aware of the available bandwidth only for the circuit to which they are connected. They are not aware of the bandwidth on the remote link. Service providers size a circuit based on the total capacity to send and receive traffic. In point-to-point technologies, ensuring that a link is not oversubscribed is straightforward, because the SP keeps the bandwidth the same at both connection points. Network traffic is shaped outbound on both routers to match the rate identified by the SP.

Figure 2-4 illustrates this concept. The link connecting R1 and R2 allows 10 Mbps of traffic in each direction. R1 receives 20 Mbps of traffic locally, but R1 shapes the traffic down to a 10 Mbps traffic stream when sending on to R2. R2 receives a 15 Mbps traffic stream locally, but R2 shapes the traffic to a 10 Mbps traffic stream before sending it on to R1. If the QoS class buffers become full, packets should be dropped before traffic is sent across the link.

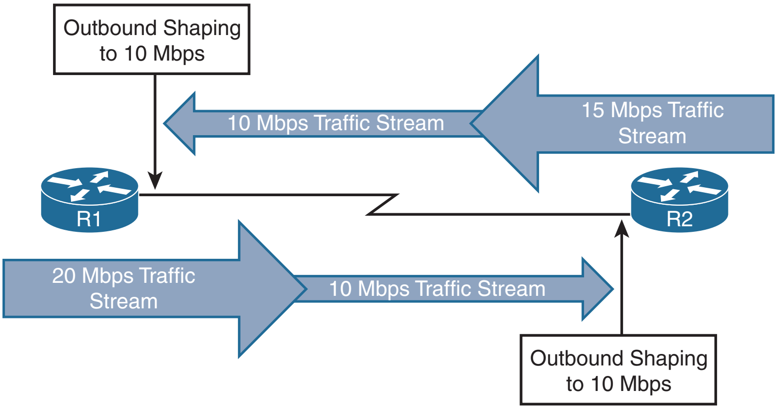


Figure 2-4 *Link Saturation*

The same logic may not apply to multipoint topologies because sites may have different bandwidth connectivity. It is possible for one router to send more traffic than the remote network can accept. When this happens, bandwidth is wasted on the transmitting router's link because those packets will ultimately be dropped. In addition, it is possible to prevent a recipient's router from receiving traffic from a different site.

Figure 2-5 illustrates the concept. Company ABC is using an MPLS VPN for connectivity between its headquarters (HQ) and branch sites. A majority of the network traffic is from the Los Angeles headquarters site to the Miami branch site. R1's circuit bandwidth has been sized to allow simultaneous communication with R2 and R3.

Unfortunately this causes problems instead. The problem is that R1 is not aware of R3's bandwidth and transmits up to a rate of 10 Mbps, whereas R3 can receive only 5 Mbps of traffic. The SP drops traffic that exceeds 5 Mbps as it is being sent to R3. In addition to wasting some of R1's bandwidth, R2 cannot communicate with R3 because R3's link is oversubscribed.

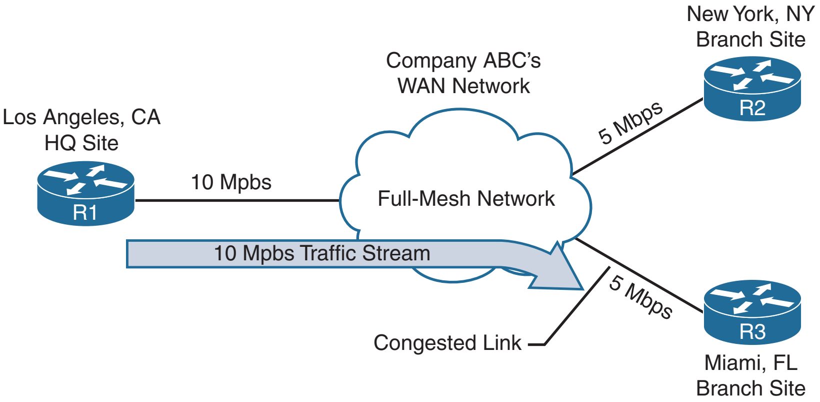


Figure 2-5 *Multipoint Link Saturation*

Dynamic Multipoint VPN (DMVPN)

Dynamic Multipoint VPN (DMVPN) is a Cisco overlay routing solution that addresses the deficiencies of site-to-site VPN tunnels. Remote locations (spokes) establish a static tunnel to a centralized location (hub) but are also capable of establishing connectivity to other remote locations with a dynamic spoke-to-spoke tunnel.

The dynamic spoke-to-spoke behavior allows full-mesh connectivity without the additional management of providing full-mesh connectivity. The spoke-to-spoke tunnels are removed after a certain period of inactivity, freeing up router memory and CPU. Because the site-to-site tunnels are dynamic, the spoke routers do not require as much memory or CPU as the hub routers.

Figure 2-6 illustrates a DMVPN tunnel where R1 is the hub and R2, R3, and R4 are spoke routers. R2 and R4 establish a dynamic spoke-to-spoke tunnel for direct communication, and eventually that dynamic tunnel is torn down after it is no longer needed.

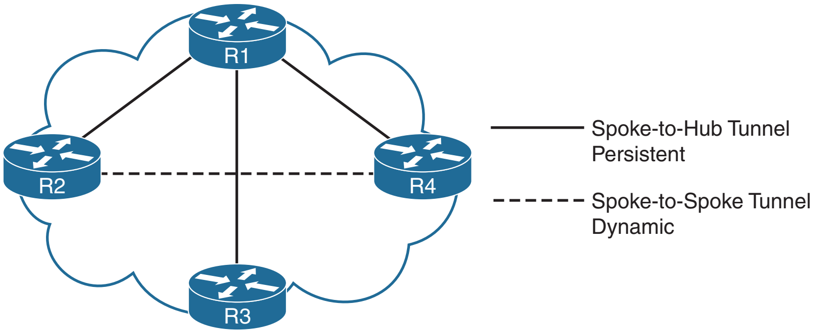


Figure 2-6 *Dynamic Multipoint VPN (DMVPN)*

DMVPN uses the following technologies:

- **Multipoint generic routing encapsulated (mGRE) tunnels:** mGRE is an overlay tunneling protocol that transports multiple protocols such as IPv4, IPv6, IPX (Internetwork Packet Exchange), and so on. Unlike IPsec VPN tunnels, GRE tunnels are assigned an actual interface and require addressing on the tunnel interface. Unlike traditional GRE tunnels which are point-to-point, mGRE supports more than two devices on the overlay network.
- **Next Hop Resolution Protocol (NHRP):** Because of the dynamic nature of DMVPN, the DMVPN spoke routers need a method to associate the tunnel endpoint IP address with the transport IP address for other DMVPN routers. NHRP provides IP resolution to *next-hop clients (NHCs)* by registering and querying a *next-hop server (NHS)*. Originally used for mapping IP addresses on .X25 networks, it is now used for resolving the tunnel endpoint addresses to the logical IP addresses of DMVPN network cloud. The protocol was defined in RFC 2332.
- **IPsec tunnel protection (optional):** IPsec uses cryptography to authenticate and encrypt IP network traffic across networks. It supports Internet Key Exchange (IKE) v1 and v2, and the Suite-B next-generation encryption technologies.

DMVPN provides the following benefits over traditional site-to-site IPsec VPN:

- **Zero-touch hub:** Adding more spoke routers does not require any additional configuration on the hub router. This simplifies the ongoing maintenance of the hub routers. Resolution of spoke endpoint IP addressing is accomplished with NHRP.
- **Spoke-to-spoke communication:** Phases 2 and 3 of NHRP support dynamic spoke-to-spoke network connectivity. All the DMVPN routers obtain the benefits of a full-mesh topology but conserve router resources by creating tunnels between VPN endpoints only as needed. Spoke-to-spoke communication reduces latency and jitter, while avoiding consumption of bandwidth at hub locations.
- **Tunnel IP addressing:** DMVPN uses GRE tunnels that use IP addressing in the overlay network. IPv4 or IPv6 addresses can be used in the overlay or to provide

connectivity in the underlay. Routing protocols can run on top of the DMVPN tunnel, thereby allowing dynamic network updates versus manually updating the routing tables when using IPsec VPN tunnels.

Note Some networks use DMVPN as a method of providing IPv6 connectivity across an IPv4 network.

- **Encryption is optional:** The benefits of overlay routing can be obtained with MPLS L3VPN networks without unnecessary consumption of router resources for encryption. DMVPN tunnels operating on insecure networks (such as the Internet) typically encrypt the payload with IPsec technologies.
- **Multicast support:** Native IPsec VPN tunnels do not support multicast traffic, which complicates the ability to transmit multicast traffic from branch sites to headquarters sites. DMVPN provides multicast support between hub-and-spoke devices for hub-to-spoke traffic flows.
- **Per-tunnel QoS:** DMVPN supports the ability of the DMVPN hub to set different QoS and bandwidth policies for each tunnel to a spoke router based on the site's connectivity model.

Benefits of Transport Independence

All the traditional transports (dial-up, leased circuits, MPLS, and IPsec VPNs) have advantages and disadvantages. One transport may prefer one routing protocol whereas a different transport may prefer a different routing protocol. Some transports are not always available or cost-effective at all locations. Intermixing multiple transports into a native WAN architecture can result in a complex WAN environment.

Ensuring that the network is always available for business purposes requires that there not be any SPOFs. Network architects plan for backup circuits that are active only when the primary circuit fails, or dual circuits that can be used at the same time. Network architects must also decide on the number of routers at each site, whether a router is dedicated to each circuit, or if both circuits should connect to the same router.

Using a single router with a single transport provides "*three nines*" (99.9%) of availability, which correlates to about four to nine hours of downtime a year. Using a single router with two transports provides "*four nines*" (99.995%) of availability, correlating to 26 minutes of downtime a year. Using two routers, each with its own transport, provides "*five nines*" (99.999%) of availability, which correlates to five minutes of downtime a year.

Figure 2-7 depicts a typical scenario for Company ABC, which uses two different providers and transports because it cannot locate two SPs that provide the same service at both the branch and headquarters. ABC uses MPLS L3VPN from one SP and creates an IPsec tunnel across the Internet for a backup circuit. R1 and R2 form an external BGP (EBGP) session to the MPLS L3VPN provider, and the routers use static routes and reverse-route injection to send traffic across the IPsec tunnel.

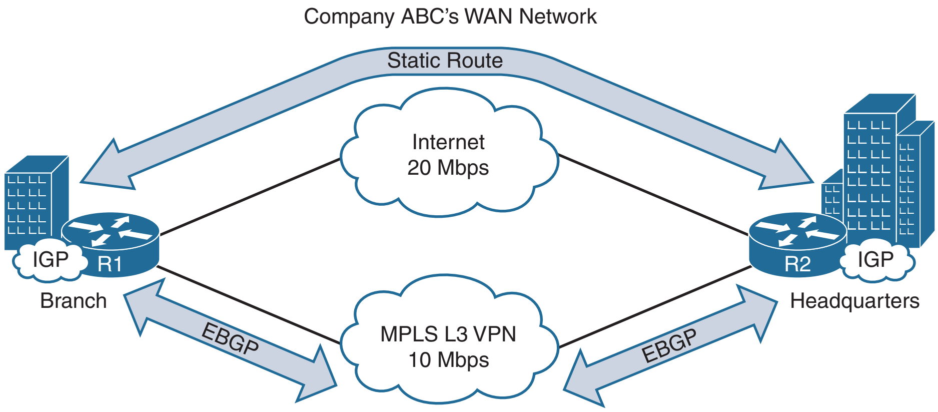


Figure 2-7 *Complexities with Multiple WAN Transports*

In scenarios like the one shown in Figure 2-7, one transport acts as a primary and the other as a backup, and there are multiple routing protocols on the WAN routers that require advanced configuration to ensure that the correct path is taken. The complexity increases when routes are exchanged between IGP (Enhanced IGRP [EIGRP], Open Shortest Path First [OSPF], and so forth) at each site and the WAN routing protocols, which could lead to suboptimal routing or routing loops.

A well-designed network architecture should take into account the operational support staff and reduce complexity where possible. It should account for junior-level engineers working in the *network operations center (NOC)* performing day-to-day management of the network. Redistribution between protocols should be kept to a minimum to prevent confusion and routing loops. The design should be modular to support future growth and allow for a simple migration between SPs to address business needs.

DMVPN provides transport independence through the use of full-mesh overlay routing. This technology allows organizations to use multiple WAN transports, because the transport type is associated to the underlay network and is irrelevant to the overlay network. The overlay network is consistent and normalized to the DMVPN tunnel.

Transport independence allows operational consistency for WAN architectures by providing the following:

- **Single routing domain:** Traditional routing protocols were designed to solve the endpoint reachability problem in a hop-by-hop destination-only forwarding environment of unknown topology. The routing protocols choose only the best path based on statically assigned cost. Because DMVPN presents a flat topology, it provides a consistent topology that allows ECMP load balancing across DMVPN tunnels.
- **Consistent troubleshooting:** The same process can be used for troubleshooting connectivity for the DMVPN tunnel and the underlay network. Even though the underlay transport changes, it relies on basic IP connectivity between tunnel endpoints.

- **Consistent topology:** There is a consistent topology and methodology for deploying other services such as performance routing.

Figure 2-8 illustrates the same topology as before except that R1 and R2 use a DMVPN tunnel for each different transport (path). Company ABC can use the same routing protocol across both paths while maintaining a consistent topology. In addition, it can change one of the WAN transports at a later time without impacting the overall WAN design or operational support model.

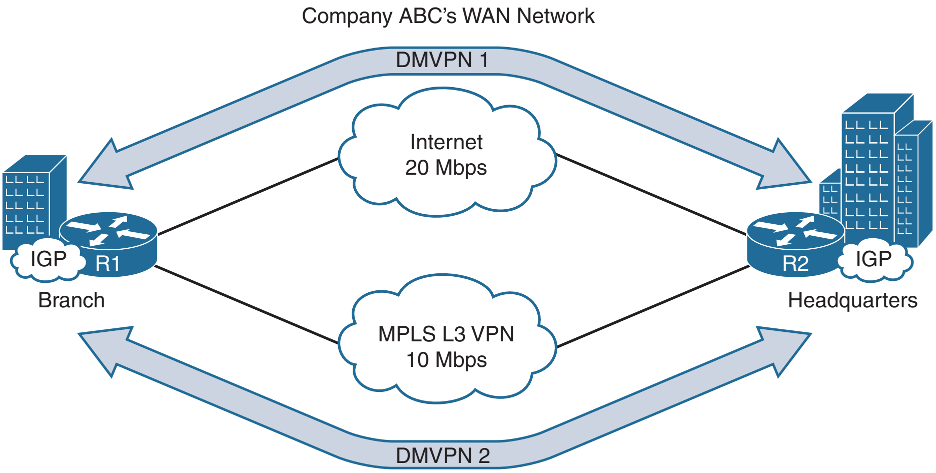


Figure 2-8 *Simplification with Transport Independence*

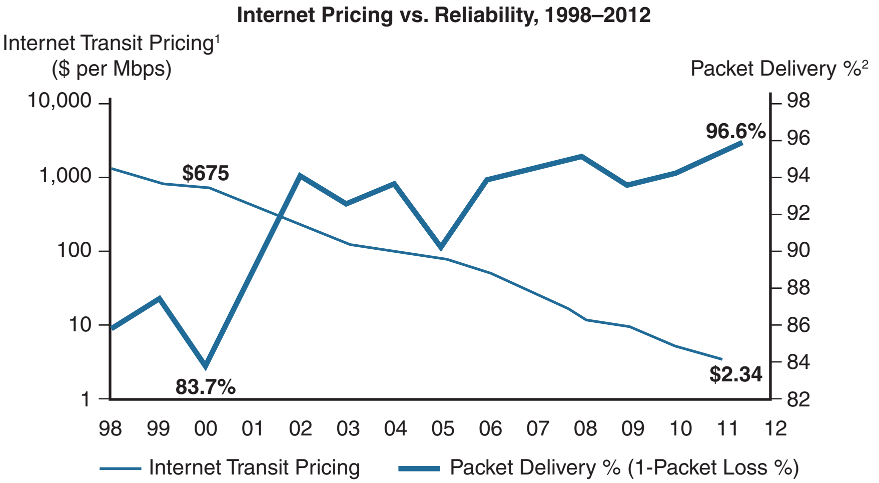
> **Note** An example use case for transport independence is when a company deploys a new branch. Most SPs have a 60- to 90-day lead time for new circuit installation. A company can immediately deploy a router using a cellular data plan and then switch to the ordered circuit at a later time with minimal changes to the configuration. The operational support model remains the same regardless of which WAN transport is used.

Managing Bandwidth Cost

Three common solutions for providing additional bandwidth are

- Increasing bandwidth to existing primary paths, which can be expensive
- Deploying additional network circuits, sometimes using low-cost residential circuits
- Deploying application optimization technologies such as Cisco WAAS
- Deploying intelligent load-balancing technologies such as PfR

Additional WAN bandwidth improves aggregate throughput but does not improve end-to-end delay or packet loss for critical applications. Furthermore, additional bandwidth can be expensive, especially in rural areas where faster connections may not be available. At the same time, the cost of Internet connectivity is generally decreasing while reliability is increasing, as shown in Figure 2-9. Enterprises are now qualifying the Internet transport as an alternative business-class WAN circuit.



[1] Internet Transit Pricing based on surveys and informal data collection primarily from Internet Operations Forums—"street pricing" estimates
[2] Packet delivery based on 15 years of ping data from PingER for WORLD (global server sample) from EDU.STANFORD.SLAC in California
Source: William Norton (DrPeering.net); Standford ping end-to-end reporting (PingER)

Figure 2-9 *Traditional WAN, Active/Backup*

Leveraging the Internet

Enterprises face a big bandwidth challenge as guaranteed access bandwidth becomes more costly; they need to determine how to create a secure, reliable, and optimized network. According to Nemertes Research ("Benchmark 2012–2013 Emerging WAN Trends: The Internet Arises," July 1, 2013), nearly half (46%) of all businesses are migrating, or are planning to migrate, their WAN to the Internet for transport. The Internet has become a much more stable platform, and the price-to-performance gains are compelling.

Offloading network traffic to the Internet can help load-balance best-effort traffic (that is, lower-priority traffic) across both links to help deal with traffic from business VPN access. In addition, administrators can use local Internet access for a distributed Internet access model to offload employee traffic that goes directly to public cloud services (such as Google Apps, Salesforce.com, Office 365, and so on) from the private WAN altogether.

This approach uses the Internet connection not just as a backup but as a real component in dealing with WAN workloads. With the right network technologies to optimize the flows, administrators can reduce overall WAN transport bandwidth requirements and improve application performance.

Intelligent WAN Transport Models

Internet- and MPLS-based VPNs are the most common WAN transports. Taking into account each transport type and the need for redundancy, there are three typical WAN deployment models:

- **Dual MPLS:** The first transport is an MPLS VPN provided by one SP, and a second MPLS VPN provided by a second SP. Assuming that both providers use the same type of MPLS VPN (all MPLS L3VPN or all MPLS L2VPN), the same routing protocol can be used.
- **Dual hybrid:** The first transport is an MPLS VPN provided by one SP, and Internet connectivity is a second transport. The Internet transport can be provided by the same or a different SP. Ideally the SPs would use different "last mile" circuits and routers to eliminate SPOFs. The Internet circuit is used to establish a VPN service to another site.

  This model provides connectivity for a distributed Internet model for all or select Internet-based traffic. It is possible to allow Internet access only for IT-approved *cloud applications.*
- **Dual Internet:** The first transport is Internet connectivity provided by one SP, and Internet connectivity is provided by a different SP for resiliency purposes. This model provides connectivity for a distributed Internet model too and does not have to be restricted to only VPN tunnels.

Note It is important to understand the BGP best-path algorithm when using multiple Internet SPs that cannot completely provide connectivity to all your locations. Typically BGP AS_Path length determines the path a packet takes on the Internet. If you have to use multiple Internet SPs to provide connectivity for one transport, it may be possible for the paths to take the same route through a transit Internet SP.

In all three models, deploying a DMVPN tunnel for each transport provides transport independence and simplifies the routing domain, operational model, and troubleshooting.

Implementing transport independence on existing circuits provides flexibility and leverage during circuit renewals. Typically, using the Internet as a transport maintains a lower cost than using an MPLS VPN for a transport and may produce savings to your company.

Figure 2-10 illustrates three of the Cisco Intelligent WAN (IWAN) models. Notice the connectivity to the Internet and cloud-based applications for all three models. Internet connectivity is available only through the headquarters in dual MPLS, whereas the hybrid and dual Internet models can provide Internet and cloud connectivity directly at the branch.

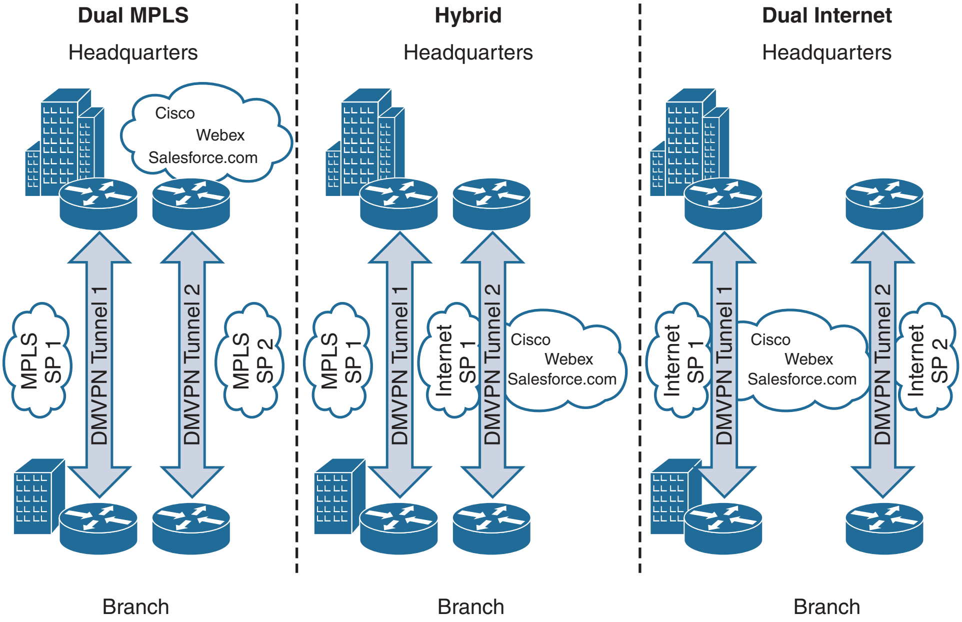


Figure 2-10 *Intelligent WAN (IWAN) Models*

Note The IWAN architecture does not limit the WAN design to only two transports. It is possible to use three or more transports to provide a custom solution. For example, you could use two different SPs providing MPLS VPN transports and a third SP providing Internet connectivity for a transport. All three transports would use DMVPN to keep the routing and topology consistent.

Summary

This chapter provided the history and overview of various WAN transports that are available today. Every WAN transport technology has advantages and disadvantages. Some remote locations cannot be serviced by an existing SP or provide the same time of transport as other sites.

Properly designed network architecture takes into account the skill level of the network engineers who support and maintain the network. The architecture should be consistent and scalable to ensure consistency in the environment regardless of the technology. Using DMVPN on top of the WAN transports provides a consistent operational model and provides transport independence while

- Simplifying the WAN with easy multihoming to SPs with a consistent design over all transports
- Providing scalable full-mesh connectivity among all DMVPN routers
- Providing a platform that supports proven robust security and cryptography
- Providing zero-touch configuration on DMVPN hub routers as branch sites are added

DMVPN is explained in greater detail in the following chapters.

Chapter 3

Dynamic Multipoint VPN

This chapter covers the following topics:

- Generic routing encapsulation (GRE) tunnels
- Next Hop Resolution Protocol (NHRP)
- Dynamic Multipoint VPN (DMVPN) tunnels
- Spoke-to-spoke communication
- DMVPN failure and detection and high availability
- DMVPN dual-hub and dual-cloud designs
- Sample IWAN DMVPN transport models

Dynamic Multipoint VPN (DMVPN) is a Cisco solution that provides a scalable VPN architecture. DMVPN uses *generic routing encapsulation (GRE)* for tunneling, *Next Hop Resolution Protocol (NHRP)* for on-demand forwarding and mapping information, and IPsec to provide a secure overlay network to address the deficiencies of site-to-site VPN tunnels while providing full-mesh connectivity. This chapter explains the underlying technologies and components of deploying DMVPN for IWAN.

DMVPN provides the following benefits to network administrators:

- **Zero-touch provisioning:** DMVPN hubs do not require additional configuration when additional spokes are added. DMVPN spokes can use a templated tunnel configuration.
- **Scalable deployment:** Minimal peering and minimal permanent state on spoke routers allow for massive scale. Network scale is not limited by device (physical, virtual, or logical).

- **Spoke-to-spoke tunnels:** DMVPN provides full-mesh connectivity while configuring only the initial spoke-to-hub tunnel. Dynamic spoke-to-spoke tunnels are created as needed and torn down when no longer needed. There is no packet loss while building dynamic on-demand spoke-to-spoke tunnels after the initial spoke-to-hub tunnels are established. A spoke maintains forwarding states only for spokes with which it is communicating.

- **Flexible network topologies:** DMVPN operation does not make any rigid assumptions about either the control plane or data plane overlay topologies. The DMVPN control plane can be used in a highly distributed and resilient model that allows massive scale and avoids a single point of failure or congestion. At the other extreme, it can also be used in a centralized model for a single point of control.

- **Multiprotocol support:** DMVPN supports IPv4, IPv6, and MPLS as the overlay or transport network protocol.

- **Multicast support:** DMVPN allows multicast traffic to flow on the tunnel interfaces.

- **Adaptable connectivity:** DMVPN routers can establish connectivity behind Network Address Translation (NAT). Spoke routers can use dynamic IP addressing such as Dynamic Host Configuration Protocol (DHCP).

- **Standardized building blocks:** DMVPN uses industry-standardized technologies (NHRP, GRE, and IPsec) to build an overlay network. This propagates familiarity while minimizing the learning curve and easing troubleshooting.

Generic Routing Encapsulation (GRE) Tunnels

A GRE tunnel provides connectivity to a wide variety of network-layer protocols by encapsulating and forwarding those packets over an IP-based network. The original use of GRE tunnels was to provide a transport mechanism for nonroutable legacy protocols such as DECnet, Systems Network Architecture (SNA), or IPX. GRE tunnels have been used as a quick workaround for bad routing designs, or as a method to pass traffic through a firewall or ACL. DMVPN uses *multipoint GRE (mGRE)* encapsulation and supports dynamic routing protocols, which eliminates many of the support issues associated with other VPN technologies. GRE tunnels are classified as an *overlay network* because the GRE tunnel is built on top of an existing transport network, also known as an *underlay network.*

Additional header information is added to the packet when the router encapsulates the packet for the GRE tunnel. The new header information contains the remote endpoint IP address as the destination. The new IP headers allow the packet to be routed between the two tunnel endpoints without inspection of the packet's payload. After the packet reaches the remote endpoint, the GRE headers are removed, and the original packet is forwarded out of the remote router.

Note GRE tunnels support IPv4 or IPv6 addresses as an overlay or transport network.

The following section explains the fundamentals of a GRE tunnel before explaining multipoint GRE tunnels that are a component of DMVPN. The process for configuring a GRE tunnel is described in the following sections.

GRE Tunnel Configuration

Figure 3-1 illustrates the configuration of a GRE tunnel. The 172.16.0.0/16 network range is the transport (underlay) network, and 192.168.100.0/24 is used for the GRE tunnel (overlay network).

In this topology, R11, R31, and the SP router have enabled Routing Information Protocol (RIP) on all the 10.0.0.0/8 and 172.16.0.0/16 network interfaces. This allows R11 and R31 to locate the remote router's encapsulating interface. R11 uses the SP router as a next hop to reach the 172.16.31.0/30 network, and R31 uses the SP router as a next hop toward the 172.16.11.0/30 network.

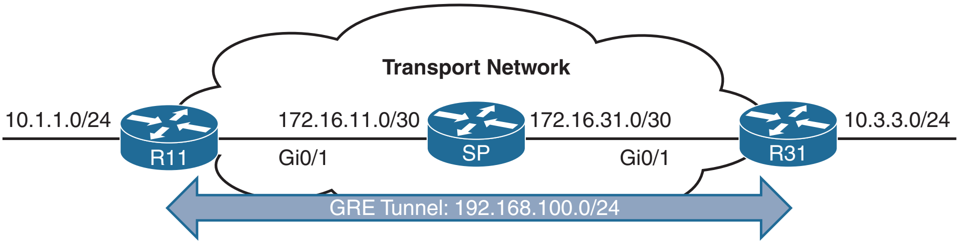


Figure 3-1 *GRE Tunnel Topology*

Note The RIP configuration does not include the 192.168.0.0/16 network range.

Example 3-1 shows the routing table of R11 before the GRE tunnel is created. Notice that the 10.3.3.0/24 network is reachable by RIP and is two hops away.

Example 3-1 *R11 Routing Table Without the GRE Tunnel*

```
R11# show ip route
! Output omitted for brevity
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C        10.1.1.0/24 is directly connected, GigabitEthernet0/2
R        10.3.3.0/24 [120/2] via 172.16.11.2, 00:00:01, GigabitEthernet0/1
      172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
C        172.16.11.0/30 is directly connected, GigabitEthernet0/1
R        172.16.31.0/30 [120/1] via 172.16.11.2, 00:00:10, GigabitEthernet0/1
```

```
R11# trace 10.3.3.3 source 10.1.1.1
Tracing the route to 10.3.3.3
  1 172.16.11.2 0 msec 0 msec 1 msec
  2 172.16.31.3 0 msec
```

The steps for configuring GRE tunnels are as follows:

Step 1. **Create the tunnel interface.**

Create the tunnel interface with the global configuration command **interface tunnel** *tunnel-number*.

Step 2. **Identify the tunnel source.**

Identify the local source of the tunnel with the interface parameter command **tunnel source** {*ip-address* | *interface-id*}. The tunnel source interface indicates the interface that will be used for encapsulation and decapsulation of the GRE tunnel. The tunnel source can be a physical interface or a loopback interface. A loopback interface can provide reachability if one of the transport interfaces were to fail.

Step 3. **Identify the remote destination IP address.**

Identify the tunnel destination with the interface parameter command **tunnel destination** *ip-address*. The tunnel destination is the remote router's underlay IP address toward which the local router sends GRE packets.

Step 4. **Allocate an IP address to the tunnel interface.**

An IP address is allocated to the interface with the command **ip address** *ip-address subnet-mask*.

Step 5. **Define the tunnel bandwidth (optional).**

Virtual interfaces do not have the concept of latency and need to have a reference bandwidth configured so that routing protocols that use bandwidth for best-path calculation can make an intelligent decision. Bandwidth is also used for QoS configuration on the interface. Bandwidth is defined with the interface parameter command **bandwidth** [*1-10000000*], which is measured in kilobits per second.

Step 6. **Specify a GRE tunnel keepalive (optional).**

Tunnel interfaces are GRE *point-to-point (P2P)* by default, and the line protocol enters an *up* state when the router detects that a route to the tunnel destination exists in the routing table. If the tunnel destination is not in the routing table, the tunnel interface (line protocol) enters a *down* state.

Tunnel keepalives ensure that bidirectional communication exists between tunnel endpoints to keep the line protocol up. Otherwise the router must rely upon routing protocol timers to detect a dead remote endpoint.

Keepalives are configured with the interface parameter command **keepalive** [*seconds* [*retries*]]. The default timer is 10 seconds and three retries.

Step 7. **Define the IP maximum transmission unit (MTU) for the tunnel interface (optional).**

The GRE tunnel adds a minimum of 24 bytes to the packet size to accommodate the headers that are added to the packet. Specifying the IP MTU on the tunnel interface has the router perform the fragmentation in advance of the host having to detect and specify the packet MTU. IP MTU is configured with the interface parameter command **ip mtu** *mtu*.

Table 3-1 displays the amount of encapsulation overhead for various tunnel techniques. The header size may change based upon the configuration options used. For all of our examples, the IP MTU is set to 1400.

Table 3-1 *Encapsulation Overhead for Tunnels*

Tunnel Type	Tunnel Header Size
GRE without IPsec	24 bytes
DES/3DES IPsec (transport mode)	18–25 bytes
DES/3DES IPsec (tunnel mode)	38–45 bytes
GRE/DMVPN + DES/3DES	42–49 bytes
GRE/DMVPN + AES + SHA-1	62–77 bytes

GRE Example Configuration

Example 3-2 provides the GRE tunnel configuration for R11 and R31. EIGRP is enabled on the LAN (10.0.0.0/8) and GRE tunnel (192.168.100.0/24) networks. RIP is enabled on the LAN (10.0.0.0/8) and transport (172.16.0.0/16) networks but is not enabled on the GRE tunnel. R11 and R31 become direct EIGRP peers on the GRE tunnel because all the network traffic is encapsulated between them.

EIGRP has a lower administrative distance (AD), 90, and the routers use the route learned via the EIGRP connection (using the GRE tunnel) versus the route learned via RIP (120) that came from the transport network. Notice that the EIGRP configuration uses named mode. *EIGRP named mode* provides clarity and keeps the entire EIGRP configuration in one centralized location. EIGRP named mode is the only method of EIGRP configuration that supports some of the newer features such as stub site.

Example 3-2 *GRE Configuration*

```
R11
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.11 255.255.255.0
 ip mtu 1400
 keepalive 5 3
 tunnel source GigabitEthernet0/1
 tunnel destination 172.16.31.1
!
router eigrp GRE-OVERLAY
 address-family ipv4 unicast autonomous-system 100
  topology base
  exit-af-topology
  network 10.0.0.0
  network 192.168.100.0
 exit-address-family
!
router rip
 version 2
 network 172.16.0.0
 no auto-summary
```

```
R31
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.31 255.255.255.0
 ip mtu 1400
 keepalive 5 3
 tunnel source GigabitEthernet0/1
 tunnel destination 172.16.11.1
!
router eigrp GRE-OVERLAY
 address-family ipv4 unicast autonomous-system 100
  topology base
  exit-af-topology
  network 10.0.0.0
  network 192.168.100.0
 exit-address-family
!
router rip
 version 2
 network 172.16.0.0
 no auto-summary
```

Now that the GRE tunnel is configured, the state of the tunnel can be verified with the command **show interface tunnel** *number*. Example 3-3 displays output from the command. Notice that the output includes the tunnel source and destination addresses, keepalive values (if any), and the tunnel line protocol state, and that the tunnel is a GRE/IP tunnel.

Example 3-3 *Display of GRE Tunnel Parameters*

```
R11# show interface tunnel 100
! Output omitted for brevity
Tunnel100 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 192.168.100.1/24
  MTU 17916 bytes, BW 400 Kbit/sec, DLY 50000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive set (5 sec), retries 3
  Tunnel source 172.16.11.1 (GigabitEthernet0/1), destination 172.16.31.1
   Tunnel Subblocks:
      src-track:
         Tunnel100 source tracking subblock associated with GigabitEthernet0/1
          Set of tunnels with source GigabitEthernet0/1, 1 member (includes
          iterators), on interface <OK>
  Tunnel protocol/transport GRE/IP
    Key disabled, sequencing disabled
    Checksumming of packets disabled
  Tunnel TTL 255, Fast tunneling enabled
  Tunnel transport MTU 1476 bytes
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Last input 00:00:02, output 00:00:02, output hang never
```

Example 3-4 displays the routing table of R11 after it has become an EIGRP neighbor with R31. Notice that R11 learns the 10.3.3.0/24 network directly from R31 via tunnel 100.

Example 3-4 *R11 Routing Table with GRE Tunnel*

```
R11# show ip route
! Output omitted for brevity
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

Gateway of last resort is not set
      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C        10.1.1.0/24 is directly connected, GigabitEthernet0/2
D        10.3.3.0/24 [90/38912000] via 192.168.100.31, 00:03:35, Tunnel100
      172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
C        172.16.11.0/30 is directly connected, GigabitEthernet0/1
R        172.16.31.0/30 [120/1] via 172.16.11.2, 00:00:03, GigabitEthernet0/1
      192.168.100.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
```

Example 3-5 verifies that traffic from 10.1.1.1 takes tunnel 100 (192.168.100.0/24) to reach the 10.3.3.3 network.

Example 3-5 *Verification of the Path from R11 to R31*

```
R11# traceroute 10.3.3.3 source 10.1.1.1
Tracing the route to 10.3.3.3
  1 192.168.100.31 1 msec *  0 msec
```

> **Note** Notice that from R11's perspective, the network is only one hop away. The traceroute does not display all the hops in the underlay. In the same fashion, the packet's *time to live (TTL)* is encapsulated as part of the payload. The original TTL decreases by only one for the GRE tunnel regardless of the number of hops in the transport network.

Next Hop Resolution Protocol (NHRP)

Next Hop Resolution Protocol (NHRP) is defined in RFC 2332 as a method to provide address resolution for hosts or networks (ARP-like capability) for *non-broadcast multi-access (NBMA)* networks such as Frame Relay and ATM. NHRP provides a method for devices to learn the protocol and NBMA network, thereby allowing them to directly communicate with each other.

NHRP is a client-server protocol that allows devices to register themselves over directly connected or disparate networks. NHRP *next-hop servers (NHSs)* are responsible for registering addresses or networks, maintaining an NHRP repository, and replying to any queries received by *next-hop clients (NHCs)*. The NHC and NHS are transactional in nature.

DMVPN uses multipoint GRE tunnels, which requires a method of mapping tunnel IP addresses to the transport (underlay) IP address. NHRP provides the technology for mapping those IP addresses. DMVPN spokes (NHCs) are *statically* configured with the IP address of the hubs (NHSs) so that they can register their tunnel and NBMA (transport) IP address with the hubs (NHSs). When a spoke-to-spoke tunnel is established, NHRP messages provide the necessary information for the spokes to locate each other so that they can build a spoke-to-spoke DMVPN tunnel. The NHRP messages also allow a spoke to locate a remote network. Cisco has added additional NHRP message types to those defined in RFC 2332 to provide some of the recent enhancements in DMVPN.

All NHRP packets must include the *source NBMA address*, *source protocol address*, *destination protocol address*, and NHRP message type. The NHRP message types are explained in Table 3-2.

Note The NBMA address refers to the transport network, and the protocol address refers to the IP address assigned to the overlay network (tunnel IP address or a network/host address).

Table 3-2 *NHRP Message Types*

Message Type	Description
Registration	Registration messages are sent by the NHC (DMVPN spokes) toward the NHS (DMVPN hubs). Registration allows the hubs to know about the spoke's NBMA information. The NHC also specifies the amount of time that the registration should be maintained by the NHS along with other attributes.
Resolution	Resolution messages are NHRP messages to locate and provide the address resolution information of the egress router toward the destination. A resolution request is sent during the actual query, and a resolution reply provides the tunnel IP address and the NBMA IP address of the remote spoke.
Redirect	Redirect messages are an essential component of DMVPN Phase 3. They allow an intermediate router to notify the encapsulator (a router) that a specific network can be reached by a more optimal path (spoke-to-spoke tunnel). The encapsulator may send a redirect suppress message to suppress redirect requests for a specified period of time. This is typically done if a more optimal path is not feasible or the policy does not allow it.

(*Continued*)

Table 3-2 *Continued*

Message Type	Description
Purge	Purge messages are sent to remove a cached NHRP entry. Purge messages notify routers of the loss of a route used by NHRP. Purges are typically sent by an NHS to NHCs (which it answered) to indicate that the mapping for an address/network that it answered is not valid anymore (for example, if the network is unreachable from the original station or has moved). Purge messages take the most direct path (spoke-to-spoke tunnel) if feasible. If a spoke-to-spoke tunnel is not established, purge messages are forwarded via the hub.
Error	Error messages are used to notify the sender of an NHRP packet that an error has occurred.

NHRP messages can contain additional information that is included in the extension part of a message. Table 3-3 lists the common NHRP message extensions.

Table 3-3 *NHRP Message Extensions*

NHRP Extension	Description
Responder address	This is used to determine the address of the responding node for reply messages.
Forward transit NHS record	This contains a list of NHSs that the NHRP request packet may have traversed.
Reverse transit NHS record	This contains a list of NHSs that the NHRP reply packet may have traversed.
Authentication	This conveys authentication information between NHRP speakers. Authentication is done pairwise on a hop-by-hop basis. This field is transmitted in plaintext.
Vendor private	This conveys vendor private information between NHRP speakers.
NAT	DMVPN works when a hub or spoke resides behind a device that performs NAT and when the tunnel is encapsulated in IPsec. This NHRP extension is able to detect the *claimed NBMA address* (inside local address) using the source protocol address of the NHRP packet, and the inside global IP address from the IP headers of the NHRP packet itself.

Dynamic Multipoint VPN (DMVPN)

DMVPN provides complete connectivity while simplifying configuration as new sites are deployed. It is considered a zero-touch technology because no configuration is needed on the DMVPN hub routers as new spokes are added to the DMVPN network. This facilitates a consistent configuration where all spokes can use identical tunnel configuration (that is, can be templatized) to simplify support and deployment with network provisioning systems like Cisco Prime Infrastructure.

Spoke sites initiate a persistent VPN connection to the hub router. Network traffic between spoke sites does not have to travel through the hubs. DMVPN dynamically builds a VPN tunnel between spoke sites on an as-needed basis. This allows network traffic, such as for VoIP, to take a direct path, which reduces delay and jitter without consuming bandwidth at the hub site.

DMVPN was released in three phases, and each phase was built on the previous one with additional functions. All three phases of DMVPN need only one tunnel interface on a router, and the DMVPN network size should accommodate all the endpoints associated to that tunnel network. DMVPN spokes can use DHCP or static addressing for the transport and overlay networks. They locate the other spokes' IP addresses (protocols and NBMA) through NHRP.

Phase 1: Spoke-to-Hub

DMVPN Phase 1 was the first DMVPN implementation and provides a zero-touch deployment for VPN sites. VPN tunnels are created only between spoke and hub sites. Traffic between spokes must traverse the hub to reach the other spoke.

Phase 2: Spoke-to-Spoke

DMVPN Phase 2 provides additional capability from DMVPN Phase 1 and allows spoke-to-spoke communication on a dynamic basis by creating an on-demand VPN tunnel between the spoke devices. DMVPN Phase 2 does not allow summarization (next-hop preservation). As a result, it also does not support spoke-to-spoke communication between different DMVPN networks (multilevel hierarchical DMVPN).

Phase 3: Hierarchical Tree Spoke-to-Spoke

DMVPN Phase 3 refines spoke-to-spoke connectivity by enhancing the NHRP messaging and interacting with the routing table. With DMVPN Phase 3 the hub sends an NHRP redirect message to the spoke that originated the packet flow. The NHRP redirect message provides the necessary information so that the originator spoke can initiate a resolution of the destination host/network. Cisco PfRv3 adds API support for DMVPN Phase 3 as well.

In DMVPN Phase 3, NHRP installs paths in the routing table for the shortcuts it creates. NHRP shortcuts modify the next-hop entry for existing routes or add a more explicit route entry to the routing table. Because NHRP shortcuts install more explicit routes in the routing table, DMVPN Phase 3 supports summarization of networks at the hub while providing optimal routing between spoke routers. NHRP shortcuts allow a hierarchical tree topology so that a regional hub is responsible for managing NHRP traffic and subnets within that region, but spoke-to-spoke tunnels can be established outside of that region.

Figure 3-2 illustrates the differences in traffic patterns for all three DMVPN phases. All three models support direct spoke-to-hub communication as shown by R1 and R2. Spoke-to-spoke packet flow in DMVPN Phase 1 is different from the packet flow in DMVPN Phases 2 and 3. Traffic between R3 and R4 must traverse the hub for Phase 1 DMVPN, whereas a dynamic spoke-to-spoke tunnel is created for DMVPN Phase 2 and Phase 3 that allows direct communication.

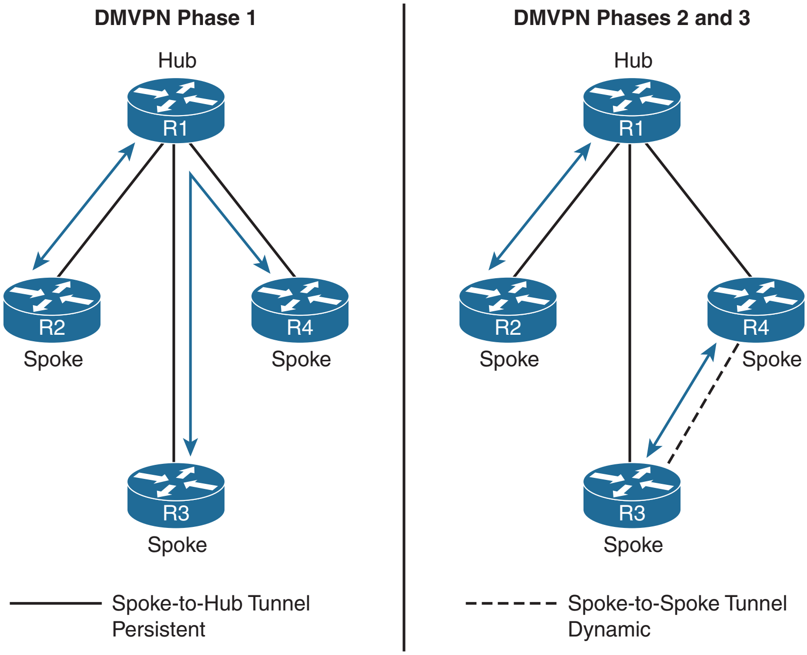


Figure 3-2 *DMVPN Traffic Patterns in the Different DMVPN Phases*

Figure 3-3 illustrates the difference in traffic patterns between Phase 2 and Phase 3 DMVPN with hierarchical topologies (multilevel). In this two-tier hierarchical design, R2 is the hub for DMVPN tunnel 20, and R3 is the hub for DMVPN tunnel 30. Connectivity between DMVPN tunnels 20 and 30 is established by DMVPN tunnel 10. All three DMVPN tunnels use the same DMVPN tunnel ID even though they use different tunnel interfaces. For Phase 2 DMVPN tunnels, traffic from R5 must flow to the hub R2, where it is sent to R3 and then back down to R6. For Phase 3 DMVPN tunnels, a spoke-to-spoke tunnel is established between R5 and R6, and the two routers can communicate directly.

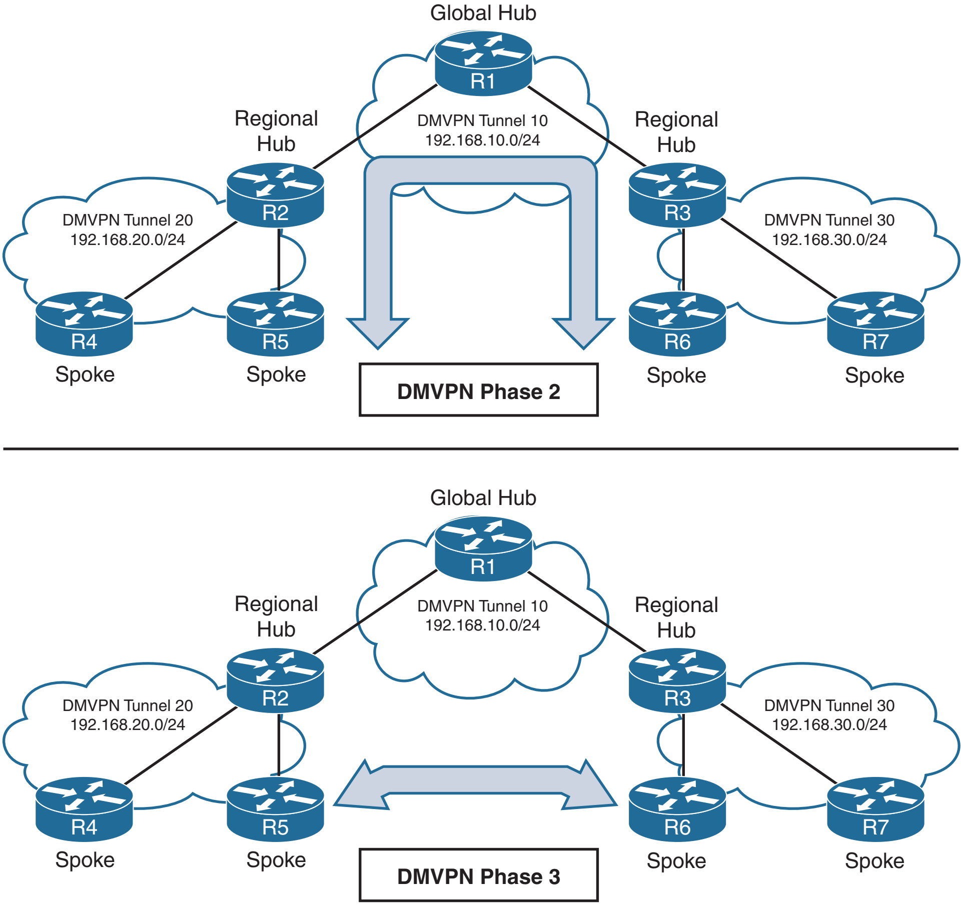


Figure 3-3 *Comparison of DMVPN Phase 2 and Phase 3*

Note Each DMVPN phase has its own specific configuration. Intermixing DMVPN phases on the same tunnel network is not recommended. If you need to support multiple DMVPN phases for a migration, a second DMVPN network (subnet and tunnel interface) should be used.

This book explains the DMVPN fundamentals with DMVPN Phase 1 and then explains DMVPN Phase 3. It does not cover DMVPN Phase 2. DMVPN Phase 3 is part of the prescriptive IWAN validated design and is explained thoroughly. At the time of writing this book, two-level hierarchical DMVPN topologies are not supported as part of the prescriptive IWAN validated design.

DMVPN Configuration

There are two types of DMVPN configurations (hub or spoke), which vary depending on a router's role. The DMVPN hub is the NHRP NHS, and the DMVPN spoke is the NHRP NHC. The spokes should be preconfigured with the hub's static IP address, but a spoke's NBMA IP address can be static or assigned from DHCP.

Note In this book, the terms "spoke router" and "branch router" are interchangeable, as are the terms "hub router" and "headquarters/data center router."

Figure 3-4 shows the first topology used to explain DMVPN configuration and functions. R11 acts as the DMVPN hub, and R31 and R41 are the DMVPN spokes. All three routers use a static default route to the SP router that provides connectivity for the NBMA (transport) networks in the 172.16.0.0/16 network range. EIGRP has been configured to operate on the DMVPN tunnel and to advertise the local LAN networks. Specific considerations for configuring EIGRP are addressed in Chapter 4, "Intelligent WAN (IWAN) Routing."

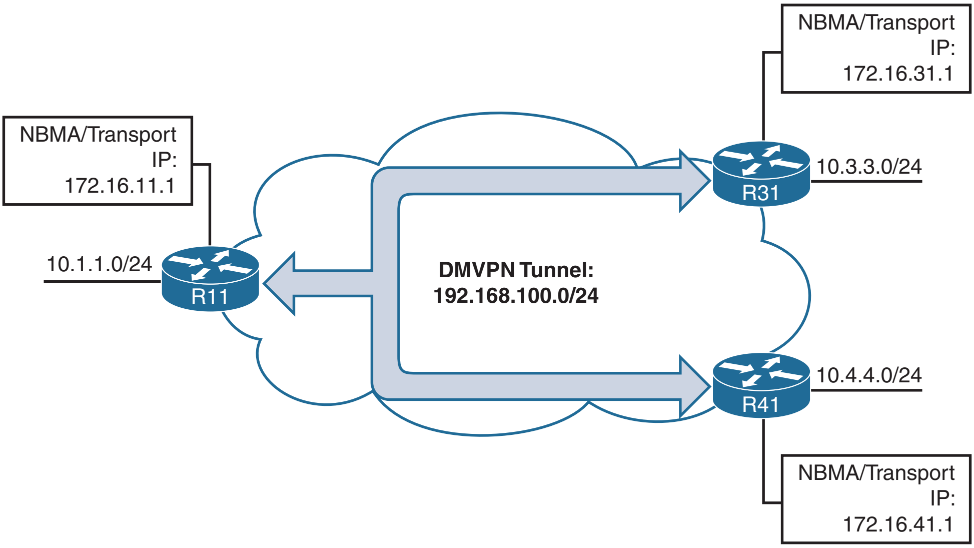

NBMA/Transport IP: 172.16.31.1
NBMA/Transport IP: 172.16.11.1
10.3.3.0/24
R31
10.1.1.0/24
R11
DMVPN Tunnel: 192.168.100.0/24
10.4.4.0/24
R41
NBMA/Transport IP: 172.16.41.1

Figure 3-4 *Simple DMVPN Topology*

DMVPN Hub Configuration

The steps for configuring DMVPN on a hub router are as follows:

Step 1. **Create the tunnel interface.**

Create the tunnel interface with the global configuration command **interface tunnel** *tunnel-number*.

Step 2. **Identify the tunnel source.**

Identify the local source of the tunnel with the interface parameter command **tunnel source** {*ip-address* | *interface-id*}. The tunnel source depends on the transport type. The encapsulating interface can be a logical interface such as a loopback or a subinterface.

Note QoS problems can occur with the use of loopback interfaces when there are multiple paths in the forwarding table to the decapsulating router. The same problems occur automatically with port channels, which are not recommended at the time of this writing.

Step 3. **Convert the tunnel to a GRE multipoint interface.**

Configure the DMVPN tunnel as a GRE multipoint tunnel with the interface parameter command **tunnel mode gre multipoint.**

Step 4. **Allocate an IP address for the DMVPN network (tunnel).**

An IP address is configured to the interface with the command **ip address** *ip-address subnet-mask.*

Note The subnet mask or size of the network should accommodate the total number of routers that are participating in the DMVPN tunnel. All the DMVPN tunnels in this book use /24, which accommodates 254 routers. Depending on the hardware used, the DMVPN network can scale much larger to 2000 or more devices.

Step 5. **Enable NHRP on the tunnel interface.**

Enable NHRP and uniquely identify the DMVPN tunnel for the virtual interface with the interface parameter command **ip nhrp network-id** *1-4294967295*.

The NHRP network ID is locally significant and is used to identify a DMVPN cloud on a router because multiple tunnel interfaces can belong to the same DMVPN cloud. It is recommended that the NHRP network ID match on all routers participating in the same DMVPN network.

Step 6. **Define the tunnel key (optional).**

The tunnel key helps identify the DMVPN virtual tunnel interface if multiple tunnel interfaces use the same tunnel source interfaces as defined in Step 3. Tunnel keys, if configured, must match for a DMVPN tunnel to establish between two routers. The tunnel key adds 4 bytes to the DMVPN header.

The tunnel key is configured with the command **tunnel key** *0-4294967295*.

Note There is no technical correlation between the NHRP network ID and the tunnel interface number; however, keeping them the same helps from an operational support aspect.

Step 7. **Enable multicast support for NHRP (optional).**

NHRP provides a mapping service of the protocol (tunnel IP) address to the NBMA (transport) address for multicast packets too. In order to support multicast or routing protocols that use multicast, this must be enabled on DMVPN hub routers with the tunnel command **ip nhrp map multicast dynamic**. This feature is explained further in Chapter 4.

Step 8. **Enable NHRP redirect (used only for Phase 3).**

Enable NHRP redirect functions with the command **ip nhrp redirect**.

Step 9. **Define the tunnel bandwidth (optional).**

Virtual interfaces do not have the concept of latency and need to have a reference bandwidth configured so that routing protocols that use bandwidth for best-path calculation can make an intelligent decision. Bandwidth is also used for QoS configuration on the interface. Bandwidth is defined with the interface parameter command **bandwidth** [*1-10000000*], which is measured in kilobits per second.

Step 10. **Define the IP MTU for the tunnel interface (optional).**

The IP MTU is configured with the interface parameter command **ip mtu** *mtu*. Typically an MTU of 1400 is used for DMVPN tunnels to account for the additional encapsulation overhead.

Step 11. **Define the TCP maximum segment size (MSS) (optional).**

The TCP Adjust MSS feature ensures that the router will edit the payload of a TCP three-way handshake if the MSS exceeds the configured value. The command is **ip tcp adjust-mss** *mss-size*. Typically DMVPN interfaces use a value of 1360 to accommodate IP, GRE, and IPsec headers.

Note Multipoint GRE tunnels do not support the option for using a keepalive.

DMVPN Spoke Configuration for DMVPN Phase 1 (Point-to-Point)

Configuration of DMVPN Phase 1 spokes is similar to the configuration for a hub router except:

- It does not use a multipoint GRE tunnel. Instead, the tunnel destination is specified.
- The NHRP mapping points to at least one active NHS.

The process for configuring a DMVPN Phase 1 spoke router is as follows:

Step 1. **Create the tunnel interface.**

Create the tunnel interface with the global configuration command **interface tunnel** *tunnel-number*.

Step 2. **Identify the remote destination IP address.**

Identify the tunnel destination with the interface parameter command **tunnel destination** *ip-address*.

Step 3. **Identify the tunnel source.**

Identify the local source of the tunnel with the interface parameter command **tunnel source** {*ip-address* | *interface-id*}.

Step 4. **Define the tunnel destination (hub).**

Identify the tunnel destination with the interface parameter command **tunnel destination** *ip-address*. The tunnel destination is the DMVPN hub IP (NBMA) address that the local router uses to establish the DMVPN tunnel.

Step 5. **Allocate an IP address for the DMVPN network (tunnel).**

An IP address is configured to the interface with the command **ip address** {*ip-address subnet-mask* | **dhcp**} or with the command **ipv6 address** *ipv6-address/prefix-length*. At the time of writing this book, DHCP is not supported for tunnel IPv6 address allocation.

Step 6. **Enable NHRP on the tunnel interface.**

Enable NHRP and uniquely identify the DMVPN tunnel for the virtual interface with the interface parameter command **ip nhrp network-id** *1-4294967295*.

Step 7. **Define the NHRP tunnel key (optional).**

The NHRP tunnel key helps identify the DMVPN virtual tunnel interface if multiple tunnels terminate on the same interface as defined in Step 3. Tunnel keys must match for a DMVPN tunnel to establish between two routers. The tunnel key adds 4 bytes to the DMVPN header.

The tunnel key is configured with the command **tunnel key** *0-4294967295*.

Note If the tunnel key is defined on the hub router, it must be defined on all the spoke routers.

Step 8. **Specify the NHRP NHS, NBMA address, and multicast mapping.**

Specify the address of one or more NHRP NHS servers with the command **ip nhrp nhs** *nhs-address* **nbma** *nbma-address* [**multicast**]. The **multicast** keyword provides multicast mapping functions in NHRP and is required to support the following routing protocols: RIP, EIGRP, and OSPF.

This command is the simplest method of defining the NHRP configuration. Table 3-4 lists the alternative NHRP mapping commands, which are needed only in cases where a static unicast or multicast map is needed for a node that is not an NHS.

Table 3-4 *Alternative NHRP Mapping Commands*

Command	Function
ip nhrp nhs *nhs-address*	Creates an NHS entry and assigns it to the tunnel IP address
ip nhrp map *ip-address nbma-address*	Maps the NBMA address to the tunnel IP address
ip nhrp map multicast [*nbma-address* | **dynamic**]	Maps NBMA addresses used as destinations for broadcast or multicast packets to be sent across the network

Note Remember that the NBMA address is the transport IP address, and the NHS address is the protocol address for the DMVPN hub. This is the hardest concept for most network engineers to remember.

Step 9. **Define the tunnel bandwidth (optional).**

Virtual interfaces do not have the concept of latency and need to have a reference bandwidth configured so that routing protocols that use bandwidth for best-path calculation can make an intelligent decision. Bandwidth is also used for QoS configuration on the interface. Bandwidth is defined with the interface parameter command **bandwidth** [*1-10000000*], which is measured in kilobits per second.

Step 10. **Define the IP MTU for the tunnel interface (optional).**

The IP MTU is configured with the interface parameter command **ip mtu** *mtu*. Typically an MTU of 1400 is used for DMVPN tunnels to account for the additional encapsulation overhead.

Step 11. **Define the TCP MSS (optional).**

The TCP Adjust MSS feature ensures that the router will edit the payload of a TCP three-way handshake if the MSS exceeds the configured value. The command is **ip tcp adjust-mss** *mss-size*. Typically DMVPN interfaces use a value of 1360 to accommodate IP, GRE, and IPsec headers.

Example 3-6 provides a sample configuration for R11 (hub), R31 (spoke), and R41 (spoke). Notice that R11 uses the **tunnel mode gre multipoint** configuration, whereas R31 and R41 use **tunnel destination 172.16.11.1** (R11's transport endpoint IP address). All three routers have set the appropriate MTU, bandwidth, and TCP MSS values.

Note R31's NHRP settings are configured with the single multivalue NHRP command, whereas R41's configuration uses three NHRP commands to provide identical functions. This configuration has been highlighted and should demonstrate the complexity it may add for typical uses.

Example 3-6 *Phase 1 DMVPN Configuration*

```
R11-Hub
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.11 255.255.255.0
 ip mtu 1400
 ip nhrp map multicast dynamic
 ip nhrp network-id 100
 ip tcp adjust-mss 1360
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel key 100
```

```
R31-Spoke (Single Command NHRP Configuration)
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.31 255.255.255.0
 ip mtu 1400
 ip nhrp network-id 100
 ip nhrp nhs 192.168.100.11 nbma 172.16.11.1 multicast
 ip tcp adjust-mss 1360
 tunnel source GigabitEthernet0/1
 tunnel destination 172.16.11.1
 tunnel key 100
```

```
R41-Spoke (Multi-Command NHRP Configuration)
interface Tunnel100
 bandwidth 40000
 ip address 192.168.100.41 255.255.255.0
 ip mtu 1400
 ip nhrp map 192.168.100.1 172.16.11.1
 ip nhrp map multicast 172.16.11.1
 ip nhrp network-id 100
 ip nhrp nhs 192.168.100.11
 ip tcp adjust-mss 1360
 tunnel source GigabitEthernet0/1
 tunnel destination 172.16.11.1
 tunnel key 100
```

Viewing DMVPN Tunnel Status

Upon configuring a DMVPN network, it is a good practice to verify that the tunnels have been established and that NHRP is functioning properly.

The command **show dmvpn** [**detail**] provides the tunnel interface, tunnel role, tunnel state, and tunnel peers with uptime. When the DMVPN tunnel interface is administratively shut down, there are no entries associated to that tunnel interface. The tunnel states are, in order of establishment:

- **INTF:** The line protocol of the DMVPN tunnel is down.
- **IKE:** DMVPN tunnels configured with IPsec have not yet successfully established an IKE session.
- **IPsec:** An IKE session is established but an IPsec security association (SA) has not yet been established.
- **NHRP:** The DMVPN spoke router has not yet successfully registered.
- **Up:** The DMVPN spoke router has registered with the DMVPN hub and received an ACK (positive registration reply) from the hub.

Example 3-7 provides sample output of the command **show dmvpn**. The output displays that R31 and R41 have defined one tunnel with one NHS (R11). This entry is in a static state because of the static NHRP mappings in the tunnel interface. R11 has two tunnels that were learned dynamically when R31 and R41 registered and established a tunnel to R11.

Example 3-7 *Viewing the DMVPN Tunnel Status for DMVPN Phase 1*

```
R11-Hub# show dmvpn
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        T1 - Route Installed, T2 - Nexthop-override
        C - CTS Capable
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
        UpDn Time --> Up or Down Time for a Tunnel
==========================================================================

Interface: Tunnel100, IPv4 NHRP Details
Type:Hub, NHRP Peers:2,

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb
 ----- --------------- -------------- ----- -------- -----
     1 172.16.31.1      192.168.100.31    UP 00:05:26     D
     1 172.16.41.1      192.168.100.41    UP 00:05:26     D
```

```
R31-Spoke# show dmvpn
! Output omitted for brevity
Interface: Tunnel100, IPv4 NHRP Details
Type:Spoke, NHRP Peers:1,

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb
 ----- --------------- --------------- ----- -------- -----
     1 172.16.11.1       192.168.100.11    UP 00:05:26     S
```

```
R41-Spoke# show dmvpn
! Output omitted for brevity
Interface: Tunnel100, IPv4 NHRP Details
Type:Spoke, NHRP Peers:1,

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb
 ----- --------------- --------------- ----- -------- -----
     1 172.16.11.1       192.168.100.11    UP 00:05:26     S
```

Note Both routers must maintain an *up* NHRP state with each other for data traffic to flow successfully between them.

Example 3-8 provides output of the command **show dmvpn detail**. Notice that the **detail** keyword provides the local tunnel and NBMA IP addresses, tunnel health monitoring, and VRF contexts. In addition, IPsec crypto information (if configured) is displayed.

Example 3-8 *Viewing the DMVPN Tunnel Status for Phase 1 DMVPN*

```
R11-Hub# show dmvpn detail
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        T1 - Route Installed, T2 - Nexthop-override
        C - CTS Capable
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
        UpDn Time --> Up or Down Time for a Tunnel
==========================================================================

Interface Tunnel100 is up/up, Addr. is 192.168.100.11, VRF ""
   Tunnel Src./Dest. addr: 172.16.11.1/MGRE, Tunnel VRF ""
   Protocol/Transport: "multi-GRE/IP", Protect ""
   Interface State Control: Disabled
   nhrp event-publisher : Disabled
Type:Hub, Total NBMA Peers (v4/v6): 2
```

```
# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1 172.16.31.1      192.168.100.31    UP 00:01:05     D  192.168.100.31/32
    1 172.16.41.1      192.168.100.41    UP 00:01:06     D  192.168.100.41/32
```

```
R31-Spoke# show dmvpn detail
! Output omitted for brevity

Interface Tunnel100 is up/up, Addr. is 192.168.100.31, VRF ""
   Tunnel Src./Dest. addr: 172.16.31.1/172.16.11.1, Tunnel VRF ""
   Protocol/Transport: "GRE/IP", Protect ""
   Interface State Control: Disabled
   nhrp event-publisher : Disabled

IPv4 NHS:
192.168.100.11  RE NBMA Address: 172.16.11.1 priority = 0 cluster = 0
Type:Spoke, Total NBMA Peers (v4/v6): 1

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Ne
----- --------------- --------------- ----- -------- ----- ------------
    1 172.16.11.1      192.168.100.11    UP 00:00:28     S  192.168.100
```

```
R41-Spoke# show dmvpn detail
! Output omitted for brevity

Interface Tunnel100 is up/up, Addr. is 192.168.100.41, VRF ""
   Tunnel Src./Dest. addr: 172.16.41.1/172.16.11.1, Tunnel VRF ""
   Protocol/Transport: "GRE/IP", Protect ""
   Interface State Control: Disabled
   nhrp event-publisher : Disabled

IPv4 NHS:
192.168.100.11  RE NBMA Address: 172.16.11.1 priority = 0 cluster = 0
Type:Spoke, Total NBMA Peers (v4/v6): 1

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1 172.16.11.1      192.168.100.11    UP 00:02:00     S  192.168.100.11/32
```

Viewing the NHRP Cache

The information that NHRP provides is a vital component of the operation of DMVPN. Every router maintains a cache of requests that it receives or is processing. The command **show ip nhrp** [**brief**] displays the local NHRP cache on a router. The NHRP cache contains the following fields:

- Network entry for hosts (IPv4: /32 or IPv6: /128) or for a network /x and the tunnel IP address to NBMA (transport) IP address.
- The interface number, duration of existence, and when it will expire (*hours:minutes:seconds*). Only dynamic entries expire.
- The NHRP mapping entry type. Table 3-5 provides a list of NHRP mapping entries in the local cache.

Table 3-5 *NHRP Mapping Entries*

NHRP Mapping Entry	Description
static	An entry created statically on a DMVPN interface.
dynamic	An entry created dynamically. In DMVPN Phase 1, an entry created from a spoke that registered with an NHS server with an NHRP registration request.
incomplete	A temporary entry placed locally while an NHRP resolution request is processing. An incomplete entry prevents repetitive NHRP requests for the same entry, avoiding unnecessary consumption of router resources. Eventually this will time out and permit another NHRP resolution request for the same network.
local	Displays local mapping information. One typical entry represents a local network that was advertised for an NHRP resolution reply. This entry records which nodes received this local network mapping via an NHRP resolution reply.
(no-socket)	These mapping entries do not have an associated IPsec socket and encryption is not triggered.
NBMA address	Non-broadcast multi-access address, or the transport IP address where the entry was received.

NHRP message flags specify attributes of an NHRP cache entry or of the peer for which the entry was created. Table 3-6 provides a listing of the NHRP message flags and their meanings.

Table 3-6 *NHRP Message Flags*

NHRP Message Flag	Description
used	Indication that this NHRP mapping entry was used to forward data packets within the last 60 seconds.
implicit	Indicates that the NHRP mapping entry was learned implicitly. Examples of such entries would be the source mapping information gleaned from an NHRP resolution request received by the local router, or from an NHRP resolution packet that was forwarded through the router.

(*Continued*)

Table 3-6 *Continued*

NHRP Message Flag	Description
unique	Indicates that this NHRP mapping entry must be unique, and that it cannot be overwritten with a mapping entry that has the same tunnel IP address but a different NBMA address.
router	Indicates that this NHRP mapping entry is from a remote router that provides access to a network or host behind the remote router.
rib	Indicates that this NHRP mapping entry has a corresponding routing entry in the routing table. This entry has an associated '*H*' route.
nho	Indicates that this NHRP mapping entry has a corresponding path overriding the next hop for a remote network as installed by another routing protocol.
nhop	Indicates an NHRP mapping entry for a remote next-hop address (for example, a remote tunnel interface) and its associated NBMA address.

The command **show ip nhrp** [**brief** | **detail**] displays the local NHRP cache on a router. Example 3-9 displays the local NHRP cache for the various routers in the sample topology. R11 contains only dynamic registrations for R31 and R41. In the event that R31 and R41 cannot maintain connectivity to R11's transport IP address, eventually the tunnel mapping will be removed on R11. The NHRP message flags on R11 indicate that R31 and R41 successfully registered with the unique registration to R11, and that traffic has recently been forwarded to both routers.

Example 3-9 *Local NHRP Cache for DMVPN Phase 1*

```
R11-Hub# show ip nhrp
192.168.100.31/32 via 192.168.100.31
   Tunnel100 created 23:04:04, expire 01:37:26
   Type: dynamic, Flags: unique registered used nhop
   NBMA address: 172.16.31.1
192.168.100.41/32 via 192.168.100.41
   Tunnel100 created 23:04:00, expire 01:37:42
   Type: dynamic, Flags: unique registered used nhop
   NBMA address: 172.16.41.1
```

```
R31-Spoke# show ip nhrp
192.168.100.11/32 via 192.168.100.11
   Tunnel100 created 23:02:53, never expire
   Type: static, Flags:
   NBMA address: 172.16.11.1
```

```
R41-Spoke# show ip nhrp
192.168.100.11/32 via 192.168.100.11
   Tunnel100 created 23:02:53, never expire
   Type: static, Flags:
   NBMA address: 172.16.11.1
```

Note Using the optional **detail** keyword provides a list of routers that submitted an NHRP resolution request and its request ID.

Example 3-10 provides the output for the **show ip nhrp brief** command. Some information such as the *used* and *nhop* NHRP message flags are not shown with the **brief** keyword.

Example 3-10 *Sample Output from the* **show ip nhrp brief** *Command*

```
R11-Hub# show ip nhrp brief
****************************************************************************
    NOTE: Link-Local, No-socket and Incomplete entries are not displayed
****************************************************************************
Legend: Type --> S - Static, D - Dynamic
        Flags --> u - unique, r - registered, e - temporary, c - claimed
        a - authoritative, t - route
============================================================================

Intf     NextHop Address                                    NBMA Address
         Target Network                              T/Flag
-------- ------------------------------------------- ------ ----------------
Tu100    192.168.100.31                                     172.16.31.1
         192.168.100.31/32                           D/ur
Tu100    192.168.100.41                                     172.16.41.1
         192.168.100.41/32                           D/ur
```

```
R31-Spoke# show ip nhrp brief
! Output omitted for brevity
Intf     NextHop Address                                    NBMA Address
         Target Network                              T/Flag
-------- ------------------------------------------- ------ ----------------
Tu100    192.168.100.11                                     172.16.11.1
         192.168.100.11/32                           S/
```

```
R41-Spoke# show ip nhrp brief
! Output omitted for brevity
```

```
Intf     NextHop Address                                    NBMA Address
         Target Network                              T/Flag
-------- ------------------------------------------- ------ ----------------
Tu100    192.168.100.11                                     172.16.11.1
         192.168.100.11/32                           S/
```

Example 3-11 displays the routing tables for R11, R31, and R41. All three routers maintain connectivity to the 10.1.1.0/24, 10.3.3.0/24, and 10.4.4.0/24 networks. Notice that the next-hop address between spoke routers is 192.168.100.11 (R11).

Example 3-11 *DMVPN Phase 1 Routing Table*

```
R11-Hub# show ip route
! Output omitted for brevity
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

Gateway of last resort is 172.16.11.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.11.2
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C        10.1.1.0/24 is directly connected, GigabitEthernet0/2
D        10.3.3.0/24 [90/27392000] via 192.168.100.31, 23:03:53, Tunnel100
D        10.4.4.0/24 [90/27392000] via 192.168.100.41, 23:03:28, Tunnel100
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.11.0/30 is directly connected, GigabitEthernet0/1
      192.168.100.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
```

```
R31-Spoke# show ip route
! Output omitted for brevity
Gateway of last resort is 172.16.31.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.31.2
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
D        10.1.1.0/24 [90/26885120] via 192.168.100.11, 23:04:48, Tunnel100
C        10.3.3.0/24 is directly connected, GigabitEthernet0/2
D        10.4.4.0/24 [90/52992000] via 192.168.100.11, 23:04:23, Tunnel100
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.31.0/30 is directly connected, GigabitEthernet0/1
      192.168.100.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
```

```
R41-Spoke# show ip route
! Output omitted for brevity
Gateway of last resort is 172.16.41.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.41.2
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
D        10.1.1.0/24 [90/26885120] via 192.168.100.11, 23:05:01, Tunnel100
D        10.3.3.0/24 [90/52992000] via 192.168.100.11, 23:05:01, Tunnel100
C        10.4.4.0/24 is directly connected, GigabitEthernet0/2
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.41.0/24 is directly connected, GigabitEthernet0/1
      192.168.100.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
```

Example 3-12 verifies that R31 can connect to R41, but network traffic must still pass through R11.

Example 3-12 *Phase 1 DMVPN Traceroute from R31 to R41*

```
R31-Spoke# traceroute 10.4.4.1 source 10.3.3.1
Tracing the route to 10.4.4.1
  1 192.168.100.11 0 msec 0 msec 1 msec
  2 192.168.100.41 1 msec *  1 msec
```

DMVPN Configuration for Phase 3 DMVPN (Multipoint)

The Phase 3 DMVPN configuration for the hub router adds the interface parameter command **ip nhrp redirect** on the hub router. This command checks the flow of packets on the tunnel interface and sends a redirect message to the source spoke router when it detects packets hairpinning out of the DMVPN cloud. Hairpinning is when traffic is received and sent out of an interface in the same cloud (identified by the NHRP network ID). For instance, packets coming in and going out of the same tunnel interface is a case of hairpinning.

The Phase 3 DMVPN configuration for spoke routers uses the multipoint GRE tunnel interface and uses the command **ip nhrp shortcut** on the tunnel interface.

Note There are no negative effects of placing **ip nhrp shortcut** and **ip nhrp redirect** on the same DMVPN tunnel interface.

The process for configuring a DMVPN Phase 3 spoke router is as follows:

Step 1. **Create the tunnel interface.**

Create the tunnel interface with the global configuration command **interface tunnel** *tunnel-number*.

Step 2. **Identify the tunnel source.**

Identify the local source of the tunnel with the interface parameter command **tunnel source** {*ip-address* | *interface-id*}.

Step 3. **Convert the tunnel to a GRE multipoint interface.**

Configure the DMVPN tunnel as a GRE multipoint tunnel with the interface parameter command **tunnel mode gre multipoint.**

Step 4. **Allocate an IP address for the DMVPN network (tunnel).**

An IP address is configured to the interface with the command **ip address** *ip-address subnet-mask*.

Step 5. **Enable NHRP on the tunnel interface.**

Enable NHRP and uniquely identify the DMVPN tunnel for the virtual interface with the interface parameter command **ip nhrp network-id** *1-4294967295*.

Step 6. **Define the tunnel key (optional).**

The tunnel key is configured with the command **tunnel key** *0-4294967295*. Tunnel keys must match for a DMVPN tunnel to establish between two routers.

Step 7. **Enable NHRP shortcut.**

Enable the NHRP shortcut function with the command **ip nhrp shortcut**.

Step 8. **Specify the NHRP NHS, NBMA address, and multicast mapping.**

Specify the address of one or more NHRP NHSs with the command **ip nhrp nhs** *nhs-address* **nbma** *nbma-address* [**multicast**].

Step 9. **Define the IP MTU for the tunnel interface (optional).**

MTU is configured with the interface parameter command **ip mtu** *mtu*. Typically an MTU of 1400 is used for DMVPN tunnels.

Step 10. **Define the TCP MSS (optional).**

The TCP Adjust MSS feature ensures that the router will edit the payload of a TCP three-way handshake if the MSS exceeds the configured value. The command is **ip tcp adjust-mss** *mss-size*. Typically DMVPN interfaces use a value of 1360 to accommodate IP, GRE, and IPsec headers.

Example 3-13 provides a sample configuration for R11 (hub), R21 (spoke), and R31 (spoke) configured with Phase 3 DMVPN. Notice that all three routers have **tunnel mode gre multipoint** and have set the appropriate MTU, bandwidth, and TCP MSS values too. R11 uses the command **ip nhrp redirect** and R31 and R41 use the command **ip nhrp shortcut**.

Example 3-13 *DMVPN Phase3 Configuration for Spokes*

```
R11-Hub
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.11 255.255.255.0
 ip mtu 1400
 ip nhrp map multicast dynamic
 ip nhrp network-id 100
 ip nhrp redirect
 ip tcp adjust-mss 1360
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel key 100
```

```
R31-Spoke
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.31 255.255.255.0
 ip mtu 1400
 ip nhrp network-id 100
 ip nhrp nhs 192.168.100.11 nbma 172.16.11.1 multicast
 ip nhrp shortcut
 ip tcp adjust-mss 1360
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel key 100
```

```
R41-Spoke
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.41 255.255.255.0
 ip mtu 1400
 ip nhrp network-id 100
 ip nhrp nhs 192.168.100.12 nbma 172.16.11.1
 ip nhrp shortcut
 ip tcp adjust-mss 1360
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel key 100
```

Spoke-to-Spoke Communication

After the configuration on R11, R31, and R41 has been modified to support DMVPN Phase 3, the tunnels are established. All the DMVPN, NHRP, and routing tables look exactly like they did in Examples 3-7 through 3-11. Please note that no traffic is exchanged between R31 and R41 at this time.

This section focuses on the underlying mechanisms used to establish spoke-to-spoke communication. In DMVPN Phase 1, the spoke devices rely upon the configured **tunnel destination** to identify where to send the encapsulated packets. Phase 3 DMVPN uses multipoint GRE tunnels and thereby relies upon NHRP redirect and resolution request messages to identify the NBMA address for any destination networks.

Packets flow through the hub in a traditional hub-and-spoke manner until the spoke-to-spoke tunnel has been established in both directions. As packets flow across the hub, the hub engages NHRP redirection to start the process of finding a more optimal path with spoke-to-spoke tunnels.

In Example 3-14, R31 initiates a traceroute to R41. Notice that the first packet travels across R11 (hub), but by the time a second stream of packets is sent, the spoke-to-spoke tunnel has been initialized so that traffic flows directly between R31 and R41 on the transport and overlay networks.

Example 3-14 *Initiation of Traffic Between Spoke Routers*

```
! Initial Packet Flow
R31-Spoke# traceroute 10.4.4.1 source 10.3.3.1
Tracing the route to 10.4.4.1
  1 192.168.100.11 5 msec 1 msec 0 msec  <- This is the Hub Router (R11-Hub)
  2 192.168.100.41 5 msec *  1 msec
```

```
! Packetflow after Spoke-to-Spoke Tunnel is Established
R31-Spoke# traceroute 10.4.4.1 source 10.3.3.1
Tracing the route to 10.4.4.1
  1 192.168.100.41 1 msec *  0 msec
```

Forming Spoke-to-Spoke Tunnels

This section explains in detail how a spoke-to-spoke DMVPN tunnel is formed. Figure 3-5 illustrates the packet flow among all three devices to establish a bidirectional spoke-to-spoke DMVPN tunnel; the numbers in the figure correspond to the steps in the following list:

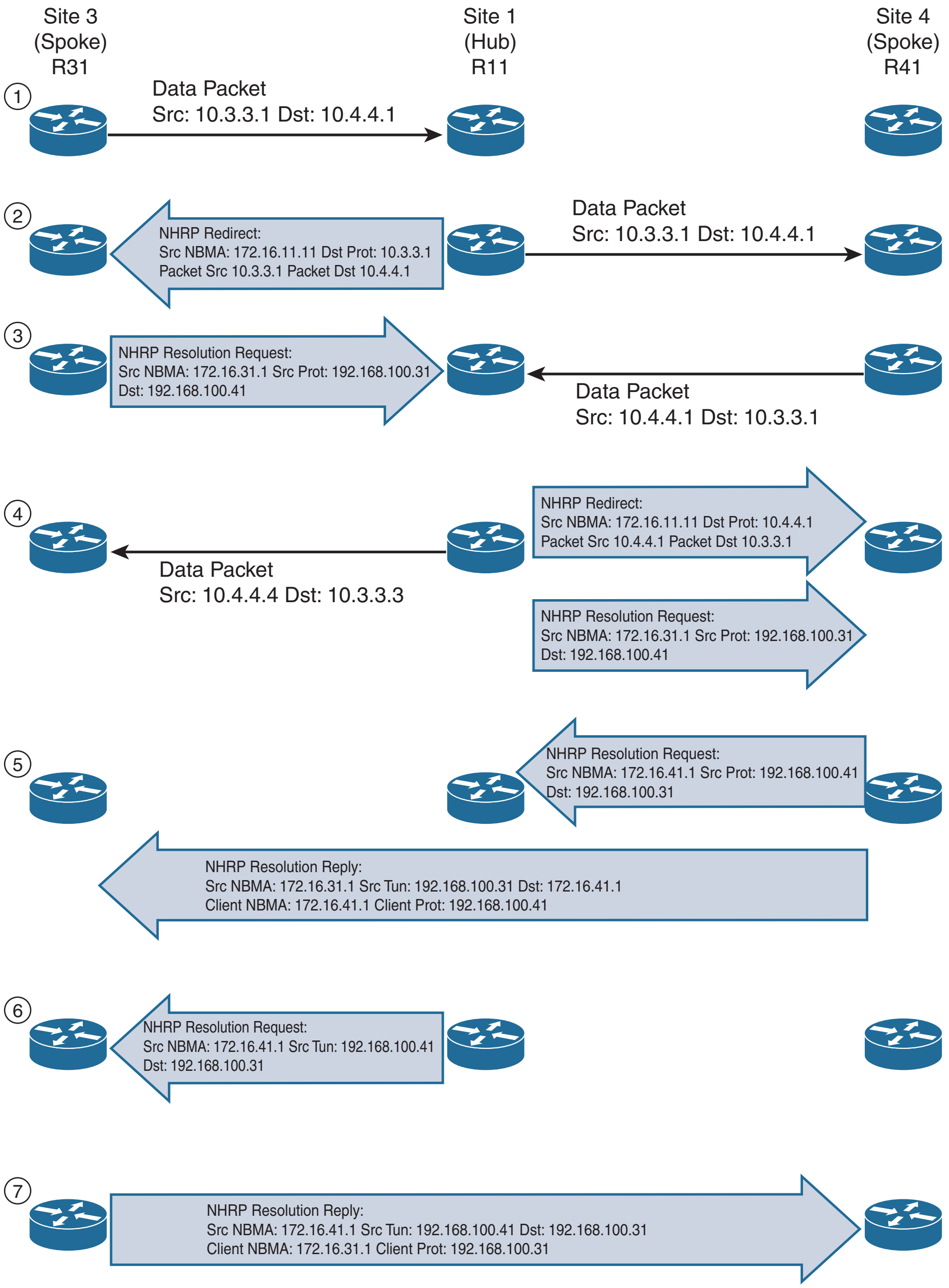

Site 3
(Spoke)
R31
Site 1
(Hub)
R11
Site 4
(Spoke)
R41
1
Data Packet
Src: 10.3.3.1 Dst: 10.4.4.1
2
NHRP Redirect:
Src NBMA: 172.16.11.11 Dst Prot: 10.3.3.1
Packet Src 10.3.3.1 Packet Dst 10.4.4.1
Data Packet
Src: 10.3.3.1 Dst: 10.4.4.1
3
NHRP Resolution Request:
Src NBMA: 172.16.31.1 Src Prot: 192.168.100.31
Dst: 192.168.100.41
Data Packet
Src: 10.4.4.1 Dst: 10.3.3.1
4
NHRP Redirect:
Src NBMA: 172.16.11.11 Dst Prot: 10.4.4.1
Packet Src 10.4.4.1 Packet Dst 10.3.3.1
Data Packet
Src: 10.4.4.4 Dst: 10.3.3.3
NHRP Resolution Request:
Src NBMA: 172.16.31.1 Src Prot: 192.168.100.31
Dst: 192.168.100.41
5
NHRP Resolution Request:
Src NBMA: 172.16.41.1 Src Prot: 192.168.100.41
Dst: 192.168.100.31
NHRP Resolution Reply:
Src NBMA: 172.16.31.1 Src Tun: 192.168.100.31 Dst: 172.16.41.1
Client NBMA: 172.16.41.1 Client Prot: 192.168.100.41
6
NHRP Resolution Request:
Src NBMA: 172.16.41.1 Src Tun: 192.168.100.41
Dst: 192.168.100.31
7
NHRP Resolution Reply:
Src NBMA: 172.16.41.1 Src Tun: 192.168.100.41 Dst: 192.168.100.31
Client NBMA: 172.16.31.1 Client Prot: 192.168.100.31

Figure 3-5 *Phase 3 DMVPN Spoke-to-Spoke Traffic Flow and Tunnel Creation*

Step 1 (on R31).

R31 performs a route lookup for 10.4.4.1 and finds the entry 10.4.4.0/24 with a next-hop IP address of 192.168.100.11. R31 encapsulates the packet destined for 10.4.4.1 and forwards it to R11 out of the tunnel 100 interface.

Step 2 (on R11).

R11 receives the packet from R31 and performs a route lookup for the packet destined for 10.4.4.1. R11 locates the 10.4.4.0/24 network with a next-hop IP address of 192.168.100.41. R11 checks the NHRP cache and locates the entry for the 192.168.100.41/32 address. R11 forwards the packet to R41 using the NBMA IP address 172.16.41.1 found in the NHRP cache. The packet is then forwarded out of the same tunnel interface.

R11 has **ip nhrp redirect** configured on the tunnel interface and recognizes that the packet received from R31 hairpinned out of the tunnel interface. R11 sends an NHRP redirect to R31 indicating the packet source of 10.3.3.1 and destination of 10.4.4.1. The NHRP redirect indicates to R31 that the traffic is using a suboptimal path.

Step 3

(On R31). R31 receives the NHRP redirect and sends an NHRP resolution request to R11 for the 10.4.4.1 address. Inside the NHRP resolution request, R31 provides its protocol (tunnel IP) address, 192.168.100.31, and source NBMA address, 172.16.31.1.

(On R41). R41 performs a route lookup for 10.3.3.1 and finds the entry 10.3.3.0/24 with a next-hop IP address of 192.168.100.11. R41 encapsulates the packet destined for 10.4.4.1 and forwards it to R11 out of the tunnel 100 interface.

Step 4 (on R11).

R11 receives the packet from R41 and performs a route lookup for the packet destined for 10.3.3.1. R11 locates the 10.3.3.0/24 network with a next-hop IP address of 192.168.100.31. R11 checks the NHRP cache and locates an entry for 192.168.100.31/32. R11 forwards the packet to R31 using the NBMA IP address 172.16.31.1 found in the NHRP cache. The packet is then forwarded out of the same tunnel interface.

R11 has **ip nhrp redirect** configured on the tunnel interface and recognizes that the packet received from R41 hairpinned out of the tunnel interface. R11 sends an NHRP redirect to R41 indicating the packet source of 10.4.4.1 and a destination of 10.3.3.1 The NHRP redirect indicates to R41 that the traffic is using a suboptimal path.

R11 forwards R31's NHRP resolution requests for the 10.4.4.1 address.

Step 5 (on R41).

R41 sends an NHRP resolution request to R11 for the 10.3.3.1 address and provides its protocol (tunnel IP) address, 192.168.100.41, and source NBMA address, 172.16.41.1.

R41 sends an NHRP resolution reply directly to R31 using the source information from R31's NHRP resolution request. The NHRP resolution reply contains the original source information in R31's NHRP resolution request as a method of verification and contains the client protocol address of 192.168.100.41 and the client NBMA address of 172.16.41.1. (If IPsec protection is configured, the IPsec tunnel is set up before the NHRP reply is sent.)

Note The NHRP reply is for the entire subnet rather than the specified host address.

Step 6 (on R11).

R11 forwards R41's NHRP resolution requests for the 192.168.100.31 and 10.4.4.1 entries.

Step 7 (on R31).

R31 sends an NHRP resolution reply directly to R41 using the source information from R41's NHRP resolution request. The NHRP resolution reply contains the original source information in R41's NHRP resolution request as a method of verification and contains the client protocol address of 192.168.100.31 and the client NBMA address of 172.16.31.1. (Again, if IPsec protection is configured, the tunnel is set up before the NHRP reply is sent back in the other direction.)

A spoke-to-spoke DMVPN tunnel is established in both directions after Step 7 has completed. This allows traffic to flow across the spoke-to-spoke tunnel instead of traversing the hub router.

Example 3-15 displays the status of DMVPN tunnels on R31 and R41 where there are two new spoke-to-spoke tunnels (highlighted). The *DLX* entries represent the local (no-socket) routes. The original tunnel to R11 remains as a static tunnel.

Example 3-15 *Detailed NHRP Mapping with Spoke-to-Hub Traffic*

```
R31-Spoke# show dmvpn detail
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        T1 - Route Installed, T2 - Nexthop-override
        C - CTS Capable
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
        UpDn Time --> Up or Down Time for a Tunnel
============================================================================
Interface Tunnel100 is up/up, Addr. is 192.168.100.31, VRF ""
   Tunnel Src./Dest. addr: 172.16.31.1/MGRE, Tunnel VRF ""
   Protocol/Transport: "multi-GRE/IP", Protect ""
   Interface State Control: Disabled
   nhrp event-publisher : Disabled

IPv4 NHS:
192.168.100.11  RE NBMA Address: 172.16.11.1 priority = 0 cluster = 0
Type:Spoke, Total NBMA Peers (v4/v6): 3

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1 172.16.31.1      192.168.100.31    UP 00:00:10   DLX        10.3.3.0/24
    2 172.16.41.1      192.168.100.41    UP 00:00:10   DT2        10.4.4.0/24
      172.16.41.1      192.168.100.41    UP 00:00:10   DT1  192.168.100.41/32
    1 172.16.11.1      192.168.100.11    UP 00:00:51     S  192.168.100.11/32
```

```
R41-Spoke# show dmvpn detail
! Output omitted for brevity
IPv4 NHS:
192.168.100.11  RE NBMA Address: 172.16.11.1 priority = 0 cluster = 0
Type:Spoke, Total NBMA Peers (v4/v6): 3

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    2 172.16.31.1      192.168.100.31    UP 00:00:34   DT2        10.3.3.0/24
      172.16.31.1      192.168.100.31    UP 00:00:34   DT1  192.168.100.31/32
    1 172.16.41.1      192.168.100.41    UP 00:00:34   DLX        10.4.4.0/24
    1 172.16.11.1      192.168.100.11    UP 00:01:15     S  192.168.100.11/32
```

Example 3-16 displays the NHRP cache for R31 and R41. Notice the NHRP mappings: *router*, *rib*, *nho*, and *nhop*. The flag *rib nho* indicates that the router has found an identical route in the routing table that belongs to a different protocol. NHRP has overridden the other protocol's next-hop entry for the network by installing a *next-hop shortcut* in the routing table. The flag *rib nhop* indicates that the router has an explicit method to reach the tunnel IP address via an NBMA address and has an associated route installed in the routing table.

Example 3-16 *NHRP Mapping with Spoke-to-Hub Traffic*

```
R31-Spoke# show ip nhrp detail
10.3.3.0/24 via 192.168.100.31
   Tunnel100 created 00:01:44, expire 01:58:15
   Type: dynamic, Flags: router unique local
   NBMA address: 172.16.31.1
   Preference: 255
    (no-socket)
   Requester: 192.168.100.41 Request ID: 3
10.4.4.0/24 via 192.168.100.41
   Tunnel100 created 00:01:44, expire 01:58:15
   Type: dynamic, Flags: router rib nho
   NBMA address: 172.16.41.1
   Preference: 255
192.168.100.11/32 via 192.168.100.11
   Tunnel100 created 10:43:18, never expire
   Type: static, Flags: used
   NBMA address: 172.16.11.1
   Preference: 255
192.168.100.41/32 via 192.168.100.41
   Tunnel100 created 00:01:45, expire 01:58:15
   Type: dynamic, Flags: router used nhop rib
   NBMA address: 172.16.41.1
   Preference: 255
```

```
R41-Spoke# show ip nhrp detail
10.3.3.0/24 via 192.168.100.31
   Tunnel100 created 00:02:04, expire 01:57:55
   Type: dynamic, Flags: router rib nho
   NBMA address: 172.16.31.1
   Preference: 255
10.4.4.0/24 via 192.168.100.41
   Tunnel100 created 00:02:04, expire 01:57:55
   Type: dynamic, Flags: router unique local
   NBMA address: 172.16.41.1
   Preference: 255
    (no-socket)
   Requester: 192.168.100.31 Request ID: 3
192.168.100.11/32 via 192.168.100.11
   Tunnel100 created 10:43:42, never expire
   Type: static, Flags: used
   NBMA address: 172.16.11.1
   Preference: 255
192.168.100.31/32 via 192.168.100.31
   Tunnel100 created 00:02:04, expire 01:57:55
   Type: dynamic, Flags: router used nhop rib
   NBMA address: 172.16.31.1   Preference: 255
```

Note Example 3-16 uses the optional **detail** keyword for viewing the NHRP cache information. The 10.4.4.0/24 entry on R31 and the 10.3.3.0/24 entry on R41 display a list of devices to which the router responded to resolution request packets and the request ID that they received.

NHRP Route Table Manipulation

NHRP tightly interacts with the routing/forwarding tables and installs or modifies routes in the *routing information base (RIB)*, also known as the routing table, as necessary. In the event that an entry exists with an exact match for the network and prefix length, NHRP overrides the existing next hop with a shortcut. The original protocol is still responsible for the prefix, but overwritten next-hop addresses are indicated in the routing table by the percent sign (%).

Example 3-17 provides the routing tables for R31 and R41. The next-hop IP address for the EIGRP remote network (highlighted) still shows 192.168.100.11 as the next-hop address but includes a percent sign (%) to indicate a next-hop override. Notice that R31 installs the NHRP route to 192.168.10.41/32 and that R41 installs the NHRP route to 192.18.100.31/32 into the routing table as well.

Example 3-17 *NHRP Routing Table Manipulation*

```
R31-Spoke# show ip route
! Output omitted for brevity
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is 172.16.31.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.31.2
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
D        10.1.1.0/24 [90/26885120] via 192.168.100.11, 10:44:45, Tunnel100
C        10.3.3.0/24 is directly connected, GigabitEthernet0/2
D   %    10.4.4.0/24 [90/52992000] via 192.168.100.11, 10:44:45, Tunnel100
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.31.0/30 is directly connected, GigabitEthernet0/1
      192.168.100.0/24 is variably subnetted, 3 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
H        192.168.100.41/32 is directly connected, 00:03:21, Tunnel100
```

```
R41-Spoke# show ip route
! Output omitted for brevity
Gateway of last resort is 172.16.41.2 to network 0.0.0.0
```

```
S*    0.0.0.0/0 [1/0] via 172.16.41.2
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
D        10.1.1.0/24 [90/26885120] via 192.168.100.11, 10:44:34, Tunnel100
D   %    10.3.3.0/24 [90/52992000] via 192.168.100.11, 10:44:34, Tunnel100
C        10.4.4.0/24 is directly connected, GigabitEthernet0/2
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.41.0/24 is directly connected, GigabitEthernet0/1
      192.168.100.0/24 is variably subnetted, 3 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
H        192.168.100.31/32 is directly connected, 00:03:10, Tunnel100
```

The command **show ip route next-hop-override** displays the routing table with the explicit NHRP shortcuts that were added. Example 3-18 displays the command's output for our topology. Notice that the NHRP shortcut is indicated by the *NHO* marking and shown underneath the original entry with the correct next-hop IP address.

Example 3-18 *Next-Hop Override Routing Table*

```
R31-Spoke# show ip route next-hop-override
! Output omitted for brevity
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       + - replicated route, % - next hop override

Gateway of last resort is 172.16.31.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.31.2
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
D        10.1.1.0/24 [90/26885120] via 192.168.100.11, 10:46:38, Tunnel100
C        10.3.3.0/24 is directly connected, GigabitEthernet0/2
D   %    10.4.4.0/24 [90/52992000] via 192.168.100.11, 10:46:38, Tunnel100
                     [NHO][90/255] via 192.168.100.41, 00:05:14, Tunnel100
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.31.0/30 is directly connected, GigabitEthernet0/1
      192.168.100.0/24 is variably subnetted, 3 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
H        192.168.100.41/32 is directly connected, 00:05:14, Tunnel100
```

```
R41-Spoke# show ip route next-hop-override
! Output omitted for brevity
Gateway of last resort is 172.16.41.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.41.2
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
```

```
D        10.1.1.0/24 [90/26885120] via 192.168.100.11, 10:45:44, Tunnel100
D   %    10.3.3.0/24 [90/52992000] via 192.168.100.11, 10:45:44, Tunnel100
                     [NHO][90/255] via 192.168.100.31, 00:04:20, Tunnel100
C        10.4.4.0/24 is directly connected, GigabitEthernet0/2
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.41.0/24 is directly connected, GigabitEthernet0/1
      192.168.100.0/24 is variably subnetted, 3 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
H        192.168.100.31/32 is directly connected, 00:04:20, Tunnel100
```

Note Review the output from Example 3-15 again. Notice that the *DT2* entries represent the networks that have had the next-hop IP address overwritten.

NHRP Route Table Manipulation with Summarization

Summarizing routes on WAN links provides stability by hiding network convergence and thereby adding scalability. This section demonstrates NHRP's interaction on the routing table when the exact route does not exist there. R11's EIGRP configuration now advertises the 10.0.0.0/8 summary prefix out of tunnel 100. The spoke routers use the summary route for forwarding traffic until the NHRP establishes the spoke-to-spoke tunnel. The more explicit entries from NHRP install into the routing table after the spoke-to-spoke tunnels have initialized.

Example 3-19 displays the change to R11's EIGRP configuration for summarizing the 10.0.0.0/8 networks out of the tunnel 100 interface.

Example 3-19 *R11's Summarization Configuration*

```
R11-Hub
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 100
  af-interface Tunnel100
   summary-address 10.0.0.0 255.0.0.0
   hello-interval 20
   hold-time 60
   no split-horizon
  exit-af-interface
  !
  topology base
  exit-af-topology
  network 10.0.0.0
  network 192.168.100.0
 exit-address-family
```

The NHRP cache is cleared on all routers with the command **clear ip nhrp** which removes any NHRP entries. Example 3-20 provides the routing table for R11, R31, and R41. Notice that only the 10.0.0.0/8 summary route provides initial connectivity among all three routers.

Example 3-20 *Routing Table with Summarization*

```
R11-Hub# show ip route
! Output omitted for brevity
Gateway of last resort is 172.16.11.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.11.2
      10.0.0.0/8 is variably subnetted, 5 subnets, 3 masks
D        10.0.0.0/8 is a summary, 00:28:44, Null0
C        10.1.1.0/24 is directly connected, GigabitEthernet0/2
D        10.3.3.0/24 [90/27392000] via 192.168.100.31, 11:18:13, Tunnel100
D        10.4.4.0/24 [90/27392000] via 192.168.100.41, 11:18:13, Tunnel100
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.11.0/30 is directly connected, GigabitEthernet0/1
      192.168.100.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
```

```
R31-Spoke# show ip route
! Output omitted for brevity
Gateway of last resort is 172.16.31.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.31.2
      10.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
D        10.0.0.0/8 [90/26885120] via 192.168.100.11, 00:29:28, Tunnel100
C        10.3.3.0/24 is directly connected, GigabitEthernet0/2
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.31.0/30 is directly connected, GigabitEthernet0/1
      192.168.100.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
```

```
R41-Spoke# show ip route
! Output omitted for brevity
Gateway of last resort is 172.16.41.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.41.2
      10.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
D        10.0.0.0/8 [90/26885120] via 192.168.100.11, 00:29:54, Tunnel100
C        10.4.4.0/24 is directly connected, GigabitEthernet0/2
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.41.0/24 is directly connected, GigabitEthernet0/1
      192.168.100.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
```

Traffic was re-initiated from 10.3.3.1 to 10.4.4.1 to initialize the spoke-to-spoke tunnels. R11 still sends the NHRP redirect for hairpinned traffic, and the pattern would complete as shown earlier except that NHRP would install a more specific route (10.3.3.0/24) into the routing table on R31 and R4. The NHRP injected route is indicated by the '*H*' entry as shown in Example 3-21.

Example 3-21 *Routing Table with Summarization and Spoke-to-Spoke Traffic*

```
R31-Spoke# show ip route
! Output omitted for brevity
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP

Gateway of last resort is 172.16.31.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.31.2
      10.0.0.0/8 is variably subnetted, 4 subnets, 3 masks
D        10.0.0.0/8 [90/26885120] via 192.168.100.11, 00:31:06, Tunnel100
C        10.3.3.0/24 is directly connected, GigabitEthernet0/2
H        10.4.4.0/24 [250/255] via 192.168.100.41, 00:00:22, Tunnel100
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.31.0/30 is directly connected, GigabitEthernet0/1
      192.168.100.0/24 is variably subnetted, 3 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
H        192.168.100.41/32 is directly connected, 00:00:22, Tunnel100
```

```
R41-Spoke# show ip route
! Output omitted for brevity
Gateway of last resort is 172.16.41.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.41.2
      10.0.0.0/8 is variably subnetted, 4 subnets, 3 masks
D        10.0.0.0/8 [90/26885120] via 192.168.100.11, 00:31:24, Tunnel100
H        10.3.3.0/24 [250/255] via 192.168.100.31, 00:00:40, Tunnel100
C        10.4.4.0/24 is directly connected, GigabitEthernet0/2
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.41.0/24 is directly connected, GigabitEthernet0/1
      192.168.100.0/24 is variably subnetted, 3 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
H        192.168.100.31/32 is directly connected, 00:00:40, Tunnel100
```

Example 3-22 displays the DMVPN tunnels after R31 and R41 have initialized the spoke-to-spoke tunnel with summarization on R11. Notice that both of the new spoke-to-spoke tunnel entries are *DT1* because they are new routes in the RIB. If the routes were more explicit (as shown in Example 3-17), NHRP would have overridden the next-hop address and used a *DT2* entry.

Example 3-22 *Detailed DMVPN Tunnel Output*

```
R31-Spoke# show dmvpn detail
! Output omitted for brevity
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        T1 - Route Installed, T2 - Nexthop-override
        C - CTS Capable
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
        UpDn Time --> Up or Down Time for a Tunnel
==========================================================================
IPv4 NHS:
192.168.100.11  RE NBMA Address: 172.16.11.1 priority = 0 cluster = 0
Type:Spoke, Total NBMA Peers (v4/v6): 3

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1 172.16.31.1      192.168.100.31    UP 00:01:17   DLX       10.3.3.0/24
    2 172.16.41.1      192.168.100.41    UP 00:01:17   DT1       10.4.4.0/24
      172.16.41.1      192.168.100.41    UP 00:01:17   DT1 192.168.100.41/32
    1 172.16.11.1      192.168.100.11    UP 11:21:33     S 192.168.100.11/32
```

```
R41-Spoke# show dmvpn detail
! Output omitted for brevity
IPv4 NHS:
192.168.100.11  RE NBMA Address: 172.16.11.1 priority = 0 cluster = 0
Type:Spoke, Total NBMA Peers (v4/v6): 3

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    2 172.16.31.1      192.168.100.31    UP 00:01:56   DT1       10.3.3.0/24
      172.16.31.1      192.168.100.31    UP 00:01:56   DT1 192.168.100.31/32
    1 172.16.41.1      192.168.100.41    UP 00:01:56   DLX       10.4.4.0/24
    1 172.16.11.1      192.168.100.11    UP 11:22:09     S 192.168.100.11/32
```

This section demonstrated the process for establishing spoke-to-spoke DMVPN tunnels and the methods by which NHRP interacts with the routing table. Phase 3 DMVPN fully supports summarization, which should be used to minimize the number of prefixes advertised across the WAN.

Problems with Overlay Networks

There are two common problems that are frequently found with tunnel or overlay networks: recursive routing and outbound interface selection. The following section explains these problems and provides a solution to them.

Recursive Routing Problems

Explicit care must be taken when using a routing protocol on a network tunnel. If a router tries to reach the remote router's encapsulating interface (transport IP address) via the tunnel (overlay network), problems will occur. This is a common issue if the transport network is advertised into the same routing protocol that runs on the overlay network.

Figure 3-6 demonstrates a simple GRE tunnel between R11 and R31. R11, R31, and the SP routers are running OSPF on the 100.64.0.0/16 transport networks. R11 and R31 are running EIGRP on the 10.0.0.0/8 LAN and 192.168.100.0/24 tunnel network.

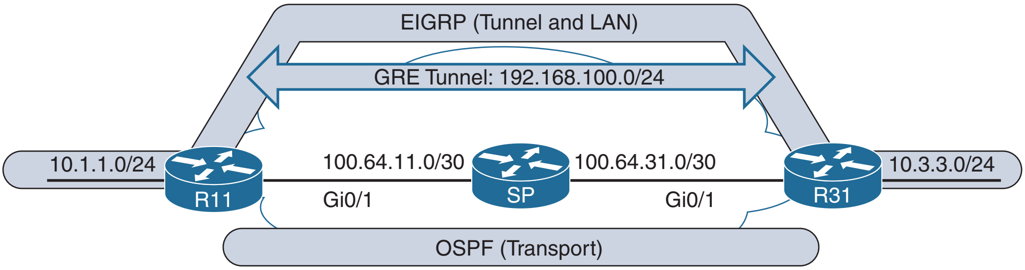


Figure 3-6 *Typical LAN Network*

Example 3-23 provides R11's routing table with everything working properly.

Example 3-23 *R11 Routing Table with GRE Tunnel*

```
R11# show ip route
! Output omitted for brevity
      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C        10.1.1.0/24 is directly connected, GigabitEthernet0/2
D        10.3.3.0/24 [90/25610240] via 192.168.100.31, 00:02:35, Tunnel0
      100.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C        100.64.11.0/24 is directly connected, GigabitEthernet0/1
O        100.64.31.0/24 [110/2] via 100.64.11.2, 00:03:11, GigabitEthernet0/1
      192.168.100.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
```

A junior network administrator has accidentally added the 100.64.0.0/16 network interfaces to EIGRP on R11 and R31. The SP router is not running EIGRP, so an adjacency does not form, but R11 and R31 add the transport network to EIGRP which has a lower

AD than OSPF. The routers will then try to use the tunnel to reach the tunnel endpoint address, which is not possible. This scenario is known as "recursive routing."

The router detects recursive routing and provides an appropriate syslog message as shown in Example 3-24. The tunnel is brought down, which terminates the EIGRP neighbors, and then R11 and R31 find each other using OSPF again. The tunnel is reestablished, EIGRP forms a relationship, and the problem repeats over and over again.

Example 3-24 *Recursive Routing Syslog Messages on R11 for GRE Tunnels*

```
00:49:52: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 192.168.100.31 (Tunnel100)
            is up: new adjacency
00:49:52: %ADJ-5-PARENT: Midchain parent maintenance for IP midchain out of
            Tunnel100 - looped chain attempting to stack
00:49:57: %TUN-5-RECURDOWN: Tunnel100 temporarily disabled due recursive routing
00:49:57: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel100, changed
            state to down
00:49:57: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 192.168.30.3 (Tunnel100) is
            down: interface down
00:50:12: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel100, changed
            state to up
00:50:15: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 192.168.100.31 (Tunnel100)
            is up: new adjacency
```

Note Only point-to-point GRE tunnels provide the syslog message "*temporarily disabled due to recursive routing.*" Both DMVPN and GRE tunnels use "*looped chained attempting to stack.*"

Recursive routing problems are remediated by preventing the tunnel endpoint address from being advertised across the tunnel network. Removing EIGRP on the transport network stabilizes this topology.

Outbound Interface Selection

In certain scenarios, it is difficult for a router to properly identify the outbound interface for encapsulating packets for a tunnel. Typically a branch site uses multiple transports (one DMVPN tunnel per transport) for network resiliency. Imagine that R31 is connected to an MPLS provider and the Internet. Both transports use DHCP to assign IP addresses to the encapsulating interfaces. R31 would have only two default routes for providing connectivity to the transport networks as shown in Example 3-25.

How would R31 know which interface to use to send packets for tunnel 100? How does the decision process change when R31 sends packets for tunnel 200? If the router picks the correct interface, the tunnel will come up; but if it picks the wrong interface, the tunnel will never come up.

Example 3-25 *Two Default Routes and Path Selection*

```
R31-Spoke# show ip route
! Output omitted for brevity
Gateway of last resort is 172.16.31.2 to network 0.0.0.0

S*    0.0.0.0/0 [254/0] via 172.16.31.2
                [254/0] via 100.64.31.2
      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C        10.3.3.0/24 is directly connected, GigabitEthernet1/0
      100.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        100.64.31.0/30 is directly connected, GigabitEthernet0/2
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.31.0/30 is directly connected, GigabitEthernet0/1
      192.168.100.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
      192.168.200.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.200.0/24 is directly connected, Tunnel200
```

Note The problem can be further exacerbated if the hub routers need to advertise a default route across the DMVPN tunnel.

Front-Door Virtual Route Forwarding (FVRF)

Virtual Route Forwarding (VRF) contexts create unique logical routers on a physical router so that router interfaces, routing tables, and forwarding tables are completely isolated from other VRFs. This means that the routing table of one transport network is isolated from the routing table of the other transport network, and that the routing table of the LAN interfaces is separate from that of all the transport networks. All router interfaces belong to the *global VRF* (also known as the *default VRF*) until they are specifically assigned to a different VRF. The global VRF is identical to the regular routing table and configuration without any VRFs defined.

DMVPN tunnels are VRF aware in the sense that the tunnel source or destination can be associated to a different VRF from the DMVPN tunnel itself. This means that the interface associated to the transport network can be associated to a transport VRF while the DMVPN tunnel is associated to a different VRF. The VRF associated to the transport network is known as the *front-door VRF (FVRF)*.

Using a front-door VRF for every DMVPN tunnel prevents route recursion because the transport and overlay networks remain in separate routing tables. Using a unique front-door VRF for each transport and associating it to the correlating DMVPN tunnel ensures that packets will always use the correct interface.

Note VRFs are locally significant, but the configuration/naming should be consistent to simplify the operational aspects.

Configuring Front-Door VRF (FVRF)

The following steps are required to create a front-door VRF, assign it to the transport interface, and make the DMVPN tunnel aware of the front-door VRF:

Step 1. **Create the front-door VRF.**

The VRF instance is created with the command **vrf definition** *vrf-name*.

Step 2. **Identify the address family.**

Initialize the appropriate address family for the transport network with the command **address-family** {**ipv4** | **ipv6**}. The address family can be IPv4, IPv6, or both.

Step 3. **Associate the front-door VRF to the interface.**

Enter interface configuration submode and specify the interface to be associated with the VRF with the command **interface** *interface-id*.

The VRF is linked to the interface with the interface parameter command **vrf forwarding** *vrf-name*.

Note If an IP address is already configured on the interface, when the VRF is linked to the interface, the IP address is removed from that interface.

Step 4. **Configure an IP address on the interface or subinterface.**

Configure an IPv4 address with the command **ip address** *ip-address subnet-mask* or an IPv6 address with the command **ipv6 address** *ipv6-address/prefix-length*.

Step 5. **Make the DMVPN tunnel VRF aware.**

Associate the front-door VRF to the DMVPN tunnel with the interface parameter command **tunnel vrf** *vrf-name* on the DMVPN tunnel.

Example 3-26 shows how the FVRFs named INET01 and MPLS01 are created on R31. Notice that when the FVRFs are associated, the IP addresses are removed from the interfaces. The IP addresses are reconfigured and the FVRFs are associated to the DMVPN tunnels.

Example 3-26 *FVRF Configuration Example*

```
R31-Spoke(config)# vrf definition INET01
R31-Spoke(config-vrf)# address-family ipv4
R31-Spoke(config-vrf-af)# vrf definition MPLS01
R31-Spoke(config-vrf)# address-family ipv4
R31-Spoke(config-vrf-af)# interface GigabitEthernet0/1
R31-Spoke(config-if)# vrf forwarding MPLS01
% Interface GigabitEthernet0/1 IPv4 disabled and address(es) removed due to
    enabling VRF MPLS01
R31-Spoke(config-if)# ip address 172.16.31.1 255.255.255.252
R31-Spoke(config-if)# interface GigabitEthernet0/2
R31-Spoke(config-if)# vrf forwarding INET01
% Interface GigabitEthernet0/2 IPv4 disabled and address(es) removed due to
    enabling VRF INET01
R31-Spoke(config-if)# ip address dhcp
R31-Spoke(config-if)# interface tunnel 100
R31-Spoke(config-if)# tunnel vrf MPLS01
R31-Spoke(config-if)# interface tunnel 200
R31-Spoke(config-if)# tunnel vrf INET01
```

FVRF Static Routes

FVRF interfaces that are assigned an IP address via DHCP automatically install a default route with an AD of 254. FVRF interfaces with static IP addressing require only a static default route in the FVRF context. This is accomplished with the command **ip route vrf** *vrf-name* **0.0.0.0 0.0.0.0** *next-hop-ip*. Example 3-27 shows the configuration for R31 for the MPLS01 FVRF. The INET01 FVRF does not need a static default route because it gets the route from the DHCP server.

Example 3-27 *FVRF Static Default Route Configuration*

```
R31-Spoke
ip route vrf MPLS01 0.0.0.0 0.0.0.0 172.16.41.2
```

Verifying Connectivity on an FVRF Interface

An active part of troubleshooting DMVPN tunnels is to ensure connectivity between tunnel endpoints with the command **ping vrf** *vrf-name ip-address* or with the command **traceroute vrf** *vrf-name ip-address*. Example 3-28 demonstrates the use of both commands from R31.

Example 3-28 *VRF Configuration Example*

```
R31-Spoke# ping vrf MPLS01 172.16.11.1
Sending 5, 100-byte ICMP Echos to 172.16.11.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

```
R31-Spoke# traceroute vrf MPLS01 172.16.11.1
Tracing the route to 172.16.11.1
VRF info: (vrf in name/id, vrf out name/id)
  1 172.16.31.2 0 msec 0 msec 1 msec
  2 172.16.11.1 0 msec *  1 msec
```

Note DMVPN tunnels can be associated to a VRF while using an FVRF. Both of the commands **vrf forwarding** *vrf-name* and **tunnel vrf** *vrf-name* are used on the tunnel interface. Different VRF names would need to be selected for it to be effective.

Viewing the VRF Routing Table

A specific VRF's routing table can be viewed with the command **show ip route vrf** *vrf-name*. Example 3-29 demonstrates the use of the command for the MPLS01 and INET01 VRFs on R31.

Example 3-29 *VRF Configuration Example*

```
R31-Spoke# show ip route vrf MPLS01
! Output omitted for brevity
Routing Table: MPLS01
Gateway of last resort is 172.16.31.2 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 172.16.31.2
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.31.0/30 is directly connected, GigabitEthernet0/1
```

```
R31-Spoke# show ip route vrf INET01
! Output omitted for brevity
Routing Table: INET01
Gateway of last resort is 100.64.31.2 to network 0.0.0.0

S*    0.0.0.0/0 [254/0] via 100.64.31.2
      100.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C        100.64.31.0/30 is directly connected, GigabitEthernet0/2
S        100.64.31.2/32 [254/0] via 100.64.31.2, GigabitEthernet0/2
```

IP NHRP Authentication

The NHRP protocol does include an authentication capability. This authentication is weak because the password is stored in plaintext. Most network administrators use NHRP authentication as a method to ensure that two different tunnels do not accidentally form. NHRP authentication is enabled with the interface parameter command **ip nhrp authentication** *password*.

Unique IP NHRP Registration

When an NHC registers with an NHS, it provides the protocol (tunnel IP) address and the NBMA (transport IP) address. By default, an NHC requests that the NHS keep the NBMA address assigned to the protocol address unique so that the NBMA address cannot be overwritten with a different IP address. The NHS server maintains a local cache of these settings. This capability is indicated by the NHRP message flag *unique* on the NHS as shown in Example 3-30.

Example 3-30 *Unique NHRP Registration*

```
R11-Hub# show ip nhrp 192.168.100.31
192.168.100.31/32 via 192.168.100.31
   Tunnel100 created 00:11:24, expire 01:48:35
   Type: dynamic, Flags: unique registered used nhop
   NBMA address: 172.16.31.1
```

If an NHC client attempts to register with the NHS using a different NBMA address, the registration process fails. Example 3-31 demonstrates this concept by disabling the DMVPN tunnel interface, changing the IP address on the transport interface, and reenabling the DMVPN tunnel interface. Notice that the DMVPN hub denies the NHRP registration because the protocol address is registered to a different NBMA address.

Example 3-31 *Failure to Connect Because of Unique Registration*

```
R31-Spoke(config)# interface tunnel 100
R31-Spoke(config-if)# shutdown
00:17:48.910: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 192.168.100.11
      (Tunnel100) is down: interface down
00:17:50.910: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel100,
  changed state to down
00:17:50.910: %LINK-5-CHANGED: Interface Tunnel100, changed state to
  administratively down
R31-Spoke(config-if)# interface GigabitEthernet0/1
R31-Spoke(config-if)# ip address 172.16.31.31 255.255.255.0
R31-Spoke(config-if)# interface tunnel 100
```

```
R31-Spoke(config-if)# no shutdown
00:18:21.011: %NHRP-3-PAKREPLY: Receive Registration Reply packet with error -
  unique address registered already(14)
00:18:22.010: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel100, changed
  state to up
```

This can cause problems for sites with transport interfaces that connect via DHCP where they could be assigned a different IP address before the NHRP cache times out. If a router were to lose connectivity and be assigned a different IP address, it would not be able to register with the NHS router until that router's entry is flushed from the NHRP cache because of its age.

The interface parameter command **ip nhrp registration no-unique** stops routers from placing the *unique* NHRP message flag in registration request packets sent to the NHS. This allows clients to reconnect to the NHS even if the NBMA address changes. This should be enabled on all DHCP-enabled spoke interfaces. However, placing this on all spoke tunnel interfaces keeps the configuration consistent for all tunnel interfaces and simplifies verification of settings from an operational perspective. The configurations in this book place it on all interfaces.

Example 3-32 demonstrates the configuration for R31.

Example 3-32 no-unique *NHRP Registration Configuration*

```
R31-Spoke
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.31 255.255.255.0
 ip mtu 1400
 ip nhrp network-id 100
 ip nhrp nhs 192.168.100.11 nbma 172.16.11.1 multicast
 ip nhrp registration no-unique
 ip nhrp shortcut
 ip tcp adjust-mss 1360
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
```

Now that the change has been made, the *unique* flag is no longer seen on R11's NHRP cache as shown in Example 3-33.

Example 3-33 *NHRP Table of Client Without Unique Registration*

```
R11-Hub# show ip nhrp 192.168.100.31
192.168.100.31/32 via 192.168.100.31
   Tunnel100 created 00:00:14, expire 01:59:48
   Type: dynamic, Flags: registered used nhop
   NBMA address: 172.16.31.31
```

Note The NHC (spoke) has to register for this change to take effect on the NHS. This happens during the normal NHRP expiration timers or can be accelerated by resetting the tunnel interface on the spoke router before its transport IP address changes.

DMVPN Failure Detection and High Availability

An NHRP mapping entry stays in the NHRP cache for a finite amount of time. The entry is valid based upon the *NHRP holdtime* period, which defaults to 7200 seconds (2 hours). The NHRP holdtime can be modified with the interface parameter command **ip nhrp holdtime** *1-65535* and should be changed to the recommended value of 600 seconds.

A secondary function of the NHRP registration packets is to verify that connectivity is maintained to the NHS (hubs). NHRP registration messages are sent every *NHRP timeout* period, and if the NHRP registration reply is not received for a request, the NHRP registration request is sent again with the first packet delayed for 1 second, the second packet delayed for 2 seconds, and the third packet delayed for 4 seconds. The NHS is declared *down* if the NHRP registration reply has not been received after the third retry attempt.

Note To further clarify, the spoke-to-hub registration is taken down and shows as the *NHRP* state when examined with the **show dmvpn** command. The actual tunnel interface still has a line protocol state of *up*.

During normal operation of the spoke-to-hub tunnels, the spoke continues to send periodic NHRP registration requests refreshing the NHRP timeout entry and keeping the spoke-to-hub tunnel up. However, in spoke-to-spoke tunnels, if a tunnel is still being used within 2 minutes of the expiration time, an NHRP request refreshes the NHRP timeout entry and keeps the tunnel. If the tunnel is not being used, it is torn down.

The NHRP timeout period defaults to one-third of the NHRP holdtime, which equates to 2400 seconds (40 minutes). The NHRP timeout period can be modified with the interface parameter command **ip nhrp registration timeout** *1-65535*.

Note When an NHS is declared *down*, NHCs still attempt to register with the down NHS. This is known as the *probe* state. The delay between retry packets increments between iterations and uses the following delay pattern: 1, 2, 4, 8, 16, 32, 64 seconds. The delay never exceeds 64 seconds, and after a registration reply is received, the NHS (hub) is declared *up* again.

NHRP Redundancy

Connectivity from a DMVPN spoke to a hub is essential to maintain connectivity. If the hub fails, or if a spoke loses connectivity to a hub, that DMVPN tunnel loses its ability to transport packets. Deploying multiple DMVPN hubs for the same DMVPN tunnel provides redundancy and eliminates an SPOF.

Figure 3-7 illustrates NHRP NHS redundancy. Routers R11, R12, R21, and R22 are DMVPN hub routers, and R31 and R41 are spoke routers. No connectivity (backdoor links) is established between R11, R12, R21, and R22.

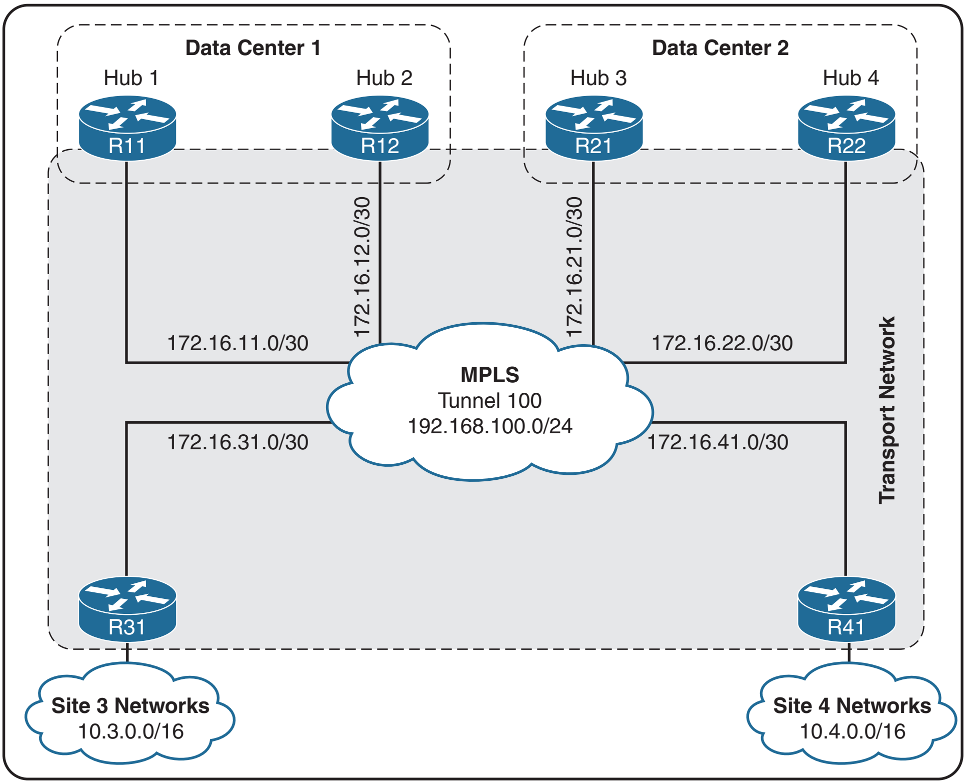


Figure 3-7 *DMVPN Multihub Topology*

Additional DMVPN hubs are added simply by adding NHRP mapping commands to the tunnel interface. All active DMVPN hubs participate in the routing domain for exchanging routes. DMVPN spoke routers maintain multiple NHRP entries (one per DMVPN hub). No additional configuration is required on the hubs.

In our sample topology, R31's and R41's configurations use R11, R12, R21, and R22 as the DMVPN hubs for tunnel 100. Example 3-34 provides R31's tunnel configuration.

Example 3-34 *Configuration for NHRP Redundancy*

```
R31-Spoke
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.31 255.255.255.0
 no ip redirects
 ip mtu 1400
 ip nhrp network-id 100
 ip nhrp holdtime 60
 ip nhrp nhs 192.168.100.11 nbma 172.16.11.1 multicast
 ip nhrp nhs 192.168.100.12 nbma 172.16.12.1 multicast
 ip nhrp nhs 192.168.100.21 nbma 172.16.21.1 multicast
 ip nhrp nhs 192.168.100.22 nbma 172.16.22.1 multicast
 ip nhrp shortcut
 ip tcp adjust-mss 1360
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
```

Example 3-35 provides verification that R31 has successfully registered and established a DMVPN tunnel to all four hub routers. Notice that all four NHS devices are assigned to cluster 0 and a priority of 0. These are the default values if the priority or cluster is not defined with the NHS mapping.

Example 3-35 *Verification of NHRP Redundancy*

```
R31-Spoke# show dmvpn detail
! Output omitted for brevity
IPv4 NHS:
192.168.100.11  RE NBMA Address: 172.16.11.1 priority = 0 cluster = 0
192.168.100.12  RE NBMA Address: 172.16.12.1 priority = 0 cluster = 0
192.168.100.21  RE NBMA Address: 172.16.21.1 priority = 0 cluster = 0
192.168.100.22  RE NBMA Address: 172.16.22.1 priority = 0 cluster = 0
Type:Spoke, Total NBMA Peers (v4/v6): 4

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1 172.16.11.1      192.168.100.11    UP 00:00:07     S  192.168.100.11/32
    1 172.16.12.1      192.168.100.12    UP 00:00:07     S  192.168.100.12/32
    1 172.16.21.1      192.168.100.21    UP 00:00:07     S  192.168.100.21/32
    1 172.16.22.1      192.168.100.22    UP 00:00:07     S  192.168.100.22/32
```

The command **show ip nhrp nhs redundancy** displays the current NHS state. Example 3-36 displays the output where R31 is connected with all four NHS routers.

Example 3-36 *Viewing NHRP NHS Redundancy*

```
R31-Spoke# show ip nhrp nhs redundancy
Legend: E=Expecting replies, R=Responding, W=Waiting
No. Interface Clus       NHS     Prty  Cur-State  Cur-Queue Prev-State Prev-Queue
  1 Tunnel100  0  192.168.100.22   0         RE    Running         E    Running
  2 Tunnel100  0  192.168.100.21   0         RE    Running         E    Running
  3 Tunnel100  0  192.168.100.12   0         RE    Running         E    Running
  4 Tunnel100  0  192.168.100.11   0         RE    Running         E    Running

No. Interface Clus Status Max-Con Totl-NHS Register/UP  Expecting  Waiting Fallbk
  1 Tunnel100  0  Disable Not Set       4          4          0        0       0
```

R31 and R41 have established EIGRP neighborship with all four NHS routers. This is confirmed by the fact that R31 has established an EIGRP adjacency with all four hub routers and learned about the 10.4.4.0/24 network from all of them as shown in Example 3-37. Notice that all four paths are installed into the routing table with equal cost (52,992,000).

Example 3-37 *Routing Table for Redundancy of DMVPN Hubs*

```
R31-Spoke# show ip route eigrp
! Output omitted for brevity
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
D        10.4.4.0/24 [90/52992000] via 192.168.100.11, 00:19:51, Tunnel100
                     [90/52992000] via 192.168.100.12, 00:19:51, Tunnel100
                     [90/52992000] via 192.168.100.21, 00:19:51, Tunnel100
                     [90/52992000] via 192.168.100.22, 00:19:51, Tunnel100
```

Traffic flow or convergence issues may arise when multiple hubs are configured. An active session is established with all hub routers, and the hub is randomly chosen based on a Cisco Express Forwarding hash for new data flows. It is possible that initial interspoke traffic will forward to a suboptimal hub. For example, the hub may be located far away from both spokes, resulting in an increase in latency and jitter. At this point, the spoke is one hop away from the overlay network perspective. There is no way to dynamically detect delay or latency yet. In addition, each session with an NHS consumes router resources for NHRP registrations and routing protocol configuration.

The number of active NHS routers can be limited for an NHS cluster with the interface parameter command **ip nhrp nhs cluster** *cluster-number* **max-connections** *0-255*. Configuring this setting allows multiple NHS routers to be configured, but only a subset of them would be active at a time. This reduces the number of site-to-site tunnels and neighbors for each routing protocol.

If there are more NHSs than the maximum connections support, the NHSs are selected by priority. A lower priority is preferred over a higher priority. The priority for an NHS server can be configured in the NHS mapping. The priority for an NHS server can be specified with the command **ip nhrp nhs** *nhs-address* **priority** *0-255.*

A cluster group represents a collection of NHS routers in a similar geographic area such as a DC. NHS routers can be associated to a cluster with the command **ip nhrp nhs** *nhs-address* **cluster** *0-10*.

The preferred method for setting priority and NHS cluster grouping is to add the **priority** and **cluster** keywords to the NHS command **ip nhrp nhs** *nhs-address* **nbma** *nbma-address* [**multicast**] [**priority** *0-255*] [**cluster** *0-10*].

NHRP redundancy is always configured from the perspective of the NHC (spoke).

Note Additional information on DMVPN cluster models can be found in Appendix A, "DMVPN Cluster Models."

NHRP Traffic Statistics

The command **show ip nhrp nhs detail** provides a listing of the NHS routers for a specific tunnel, the priority, cluster number, and counts of various NHRP requests, replies, and failures. This information is helpful for troubleshooting and is shown in Example 3-38.

Example 3-38 *NHRP Traffic Statistics per Hub*

```
R31-Spoke# show ip nhrp nhs detail
Legend: E=Expecting replies, R=Responding, W=Waiting
Tunnel100:
192.168.100.11  RE NBMA Address: 172.16.11.1 priority = 1 cluster = 1  req-sent
     3265  req-failed 0  repl-recv 3263 (00:00:12 ago)
192.168.100.12  W NBMA Address: 172.16.12.1 priority = 2 cluster = 1  req-sent
     2  req-failed 3254  repl-recv 2 (18:05:14 ago)
192.168.100.21  RE NBMA Address: 172.16.21.1 priority = 1 cluster = 2  req-sent
     3264  req-failed 0  repl-recv 3263 (00:00:12 ago)
192.168.100.22  W NBMA Address: 172.16.22.1 priority = 2 cluster = 2  req-sent
     2  req-failed 3254  repl-recv 2 (18:05:14 ago)
```

The command **show ip nhrp traffic** classifies and displays counts for the various NHRP message types on a per-tunnel basis. Example 3-39 demonstrates the output of this command. This is another helpful command for troubleshooting.

Example 3-39 *NHRP Traffic Statistics per Tunnel*

```
R31-Spoke# show ip nhrp traffic
Tunnel100: Max-send limit:100Pkts/10Sec, Usage:0%
   Sent: Total 41574
         9102 Resolution Request  9052 Resolution Reply  23411 Registration Request
         0 Registration Reply  8 Purge Request  1 Purge Reply
         0 Error Indication  0 Traffic Indication  0 Redirect Suppress
   Rcvd: Total 41542
         9099 Resolution Request  9051 Resolution Reply  0 Registration Request
         23374 Registration Reply  1 Purge Request  8 Purge Reply
         0 Error Indication  9 Traffic Indication  0 Redirect Suppress
```

DMVPN Tunnel Health Monitoring

The line protocol for the DMVPN tunnel interface remains in an *up* state regardless of whether it can connect to an NHS (DMVPN hub). The interface parameter command **if-state nhrp** changes the behavior, so that the line protocol for a DMVPN tunnel changes to *down* if it cannot maintain active registration with at least one NHS. This command should be added to DMVPN spoke tunnel interfaces but should not be added to DMVPN hub routers.

The configuration of DMVPN tunnel health monitoring is shown when examining the DMVPN tunnel. DMVPN tunnel health monitoring is enabled on tunnel 100 in Example 3-40.

Example 3-40 *Identification of DMVPN Tunnel Health Monitoring*

```
R31-Spoke# show dmvpn detail
! Output omitted for brevity
==========================================================================
Interface Tunnel100 is up/up, Addr. is 192.168.100.31, VRF ""
   Tunnel Src./Dest. addr: 172.16.31.1/MGRE, Tunnel VRF ""
   Protocol/Transport: "multi-GRE/IP", Protect ""
   Interface State Control: Enabled
   nhrp event-publisher : Disabled
```

DMVPN Dual-Hub and Dual-Cloud Designs

When network engineers build and design highly available networks, they always place devices in pairs. Look at campus designs; very few networks are built with a single core device. The WAN is no different. Just as one DMVPN cloud has redundant hubs, a WAN design should accommodate transport failures to reduce network downtime and have a second DMVPN cloud on a different transport. Providing a second transport increases the resiliency of the WAN for a variety of failures and provides a second path for network traffic.

In a dual-hub and dual-cloud model, there are two separate WAN transports. The transports can be the same technology provided by two different SPs, or two different transport technologies provided by the same SP. A DMVPN hub router contains only one DMVPN tunnel, to ensure that the proper spoke-to-spoke tunnel forms.

Typically there is only one transport per hub router. In other words, in a dual-MPLS model, there are two MPLS SPs, MPLS SP1 and MPLS SP2. Assuming that both MPLS SP1 and MPLS SP2 can reach all the locations where the hub and spoke routers are located, a hub router is dedicated to MPLS SP1 and a different hub router is dedicated to MPLS SP2 within the same DC. Redundancy is provided within each cloud by duplicating the design in a second DC. In this topology, a tunnel is assigned for each transport, and there are two hubs for every DMVPN tunnel to provide resiliency for that DMVPN tunnel.

Note A DMVPN hub should have only one DMVPN tunnel. If a DMVPN hub contains multiple DMVPN tunnels, a packet from a spoke could be forwarded out of a different tunnel interface from the one on which it was received. The hub would not send an NHRP redirect to the originating spoke, and a spoke-to-spoke tunnel would not form. NHRP redirect messages are sent only if a packet hairpins out of a tunnel interface.

Figure 3-8 illustrates a dual-hub and dual-cloud topology that is frequently referenced throughout this book. R11 and R21 reside in different DCs and are the hub routers for DMVPN tunnel 100 (MPLS transport). R12 and R22 reside in different DCs and are the hub routers for DMVPN tunnel 200 (Internet transport).

Site 3 and Site 4 do not have redundant routers, so R31 and R41 are connected to both transports via DMVPN tunnels 100 and 200. However, at Site 5, redundant routers have been deployed. R51 connects to the MPLS transport with DMVPN tunnel 100, and R52 connects to the Internet transport with DMVPN tunnel 200.

At remote sites that use two DMVPN spoke routers for redundancy, a dedicated network link (or logical VLAN) is established for exchanging routes and cross-router traffic. Access to the LAN segments uses a separate network link from the cross-router link.

The DMVPN spoke routers use R11 and R21 for the NHS server for DMVPN tunnel 100 and use R12 and R22 for the NHS server for DMVPN tunnel 200.

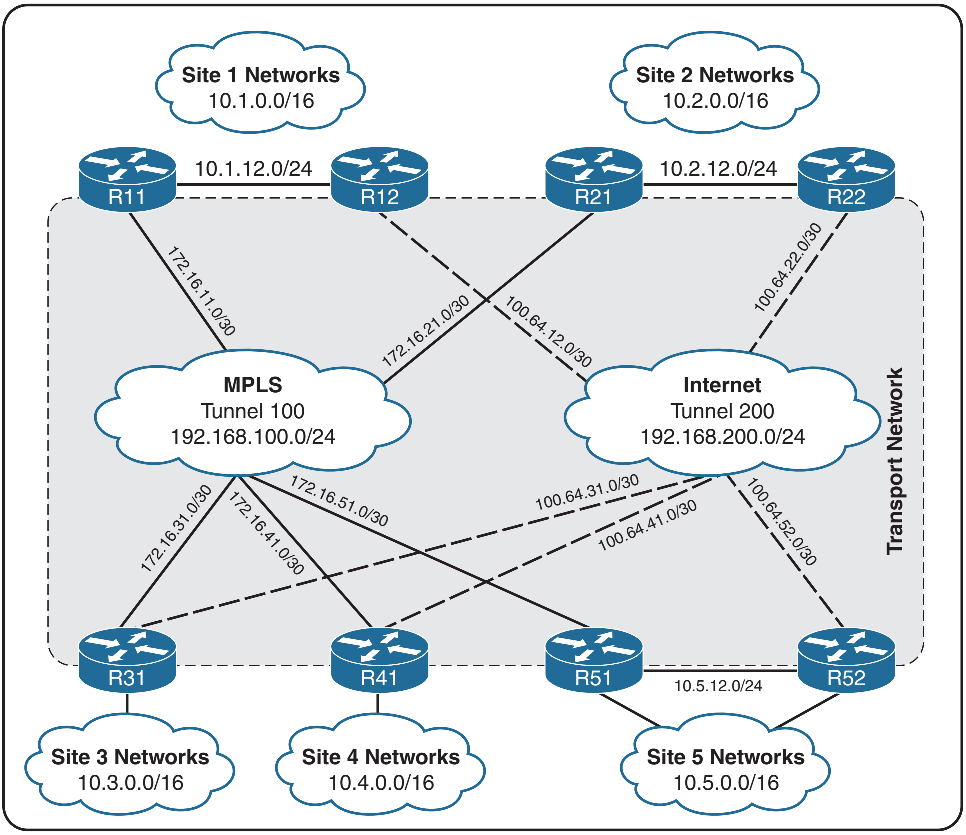


Figure 3-8 *DMVPN Multihub Topology*

Note In some more advanced designs, a DMVPN hub may use more advanced routing in the transport network and connect to multiple CE networks. This allows a hub router to have multiple paths within the transport network for its DMVPN tunnel. The simplest use case is if MPLS SP1 provides an active and backup CE device; then only one DMVPN hub is needed for that environment. Spoke routers still have full-mesh connectivity in the MPLS SP1 network and can establish a spoke-to-spoke tunnel.

Associating multiple transports/SPs to the same DMVPN hub router allows the selection of only one path between the hub and the spoke. If both paths between a hub and spoke for PfR are to be measured independently, stick to one transport per DMVPN hub router.

IWAN DMVPN Sample Configurations

This section explains the components of DMVPN Phase 3 and various NHRP features. Example 3-41 provides a complete DMVPN configuration for the DMVPN hub routers in Figure 3-8. Notice that the configuration for the MPLS routers (R11 and R21) or the Internet (R12 and R22) is the same for each transport except for different IP addresses.

Example 3-41 *DMVPN Hub Configuration on R11 and R12*

```
R11-Hub
vrf definition MPLS01
 address-family ipv4
 exit-address-family
!
interface GigabitEthernet0/1
 description MPLS01-TRANSPORT
 vrf forwarding MPLS01
 ip address 172.16.11.1 255.255.255.252
interface GigabitEthernet0/3
 description Cross-Link to R12
 ip address 10.1.12.11 255.255.255.0
!
interface Tunnel100
 description DMVPN-MPLS
 bandwidth 4000
 ip address 192.168.100.11 255.255.255.0
 ip mtu 1400
 ip nhrp authentication CISCO
 ip nhrp map multicast dynamic
 ip nhrp network-id 100
 ip nhrp holdtime 600
ip nhrp redirect
 ip tcp adjust-mss 1360
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel key 100
 tunnel vrf MPLS01
!
ip route vrf MPLS01 0.0.0.0 0.0.0.0 172.16.11.2
```

```
R12-Hub
vrf definition INET01
 address-family ipv4
 exit-address-family
!
```

```
interface GigabitEthernet0/2
 description INET01-TRANSPORT
 vrf forwarding INET01
 ip address 100.64.12.1 255.255.255.252
interface GigabitEthernet0/3
 description Cross-Link to R11
 ip address 10.1.12.12 255.255.255.0
!
interface Tunnel200
 description DMVPN-Internet
 bandwidth 4000
 ip address 192.168.200.12 255.255.255.0
 ip mtu 1400
 ip nhrp authentication CISCO2
 ip nhrp map multicast dynamic
 ip nhrp network-id 200
 ip nhrp holdtime 600
ip nhrp redirect
 ip tcp adjust-mss 1360
 tunnel source GigabitEthernet0/2
 tunnel mode gre multipoint
 tunnel key 200
 tunnel vrf INET01
!
ip route vrf INET01 0.0.0.0 0.0.0.0 100.64.12.2
```

```
R21-Hub
vrf definition MPLS01
 address-family ipv4
 exit-address-family
!
interface GigabitEthernet0/1
 description MPLS01-TRANSPORT
 vrf forwarding MPLS01
 ip address 172.16.21.1 255.255.255.252
interface GigabitEthernet0/3
 description Cross-Link to R22
 ip address 10.2.12.21 255.255.255.0
!
interface Tunnel100
 description DMVPN-MPLS
 bandwidth 4000
 ip address 192.168.100.21 255.255.255.0
 ip mtu 1400
 ip nhrp authentication CISCO
 ip nhrp map multicast dynamic
 ip nhrp network-id 100
 ip nhrp holdtime 600
ip nhrp redirect
 ip tcp adjust-mss 1360
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel key 100
 tunnel vrf MPLS01
!
ip route vrf MPLS01 0.0.0.0 0.0.0.0 172.16.21.2
```

```
R22-Hub
vrf definition INET01
 address-family ipv4
 exit-address-family
!
interface GigabitEthernet0/2
 description INET01-TRANSPORT
 vrf forwarding INET01
 ip address 100.64.22.1 255.255.255.252
interface GigabitEthernet0/3
 description Cross-Link to R21
 ip address 10.2.12.22 255.255.255.0
!
interface Tunnel200
 description DMVPN-Internet
 bandwidth 4000
 ip address 192.168.200.22 255.255.255.0
 ip mtu 1400
 ip nhrp authentication CISCO2
 ip nhrp map multicast dynamic
 ip nhrp network-id 200
 ip nhrp holdtime 600
ip nhrp redirect
 ip tcp adjust-mss 1360
 tunnel source GigabitEthernet0/2
 tunnel mode gre multipoint
 tunnel key 200
 tunnel vrf INET01
!
ip route vrf INET01 0.0.0.0 0.0.0.0 100.64.22.2
```

Example 3-42 provides the configuration for DMVPN spoke routers that are the only DMVPN router for that site. R31 and R41 are configured with both VRFs and DMVPN tunnels. Notice that both have only a static default route for the MPLS VRF. This is because the interfaces on the Internet VRF are assigned IP addresses via DHCP, which provides the default route to the routers.

Example 3-42 *DMVPN Configuration for R31 and R41 (Sole Router at Site)*

```
R31-Spoke
vrf definition INET01
 address-family ipv4
 exit-address-family
vrf definition MPLS01
 address-family ipv4
 exit-address-family
!
interface GigabitEthernet0/1
 description MPLS01-TRANSPORT
 vrf forwarding MPLS01
 ip address 172.16.31.1 255.255.255.252
interface GigabitEthernet0/2
 description INET01-TRANSPORT
 vrf forwarding INET01
 ip address dhcp
!
interface Tunnel100
 description DMVPN-MPLS
 bandwidth 4000
 ip address 192.168.100.31 255.255.255.0
 ip mtu 1400
 ip nhrp authentication CISCO
 ip nhrp network-id 100
 ip nhrp holdtime 600
 ip nhrp nhs 192.168.100.11 nbma 172.16.11.1 multicast
 ip nhrp nhs 192.168.100.21 nbma 172.16.21.1 multicast
! The following command keeps the tunnel configuration consistent across all
  tunnels.
 ip nhrp registration no-unique
 ip nhrp shortcut
 ip tcp adjust-mss 1360
 if-state nhrp
tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
tunnel key 100
 tunnel vrf MPLS01
!
```

```
interface Tunnel200
 description DMVPN-INET
 bandwidth 4000
 ip address 192.168.200.31 255.255.255.0
 ip mtu 1400
 ip nhrp authentication CISCO2
 ip nhrp network-id 200
 ip nhrp holdtime 600
 ip nhrp nhs 192.168.200.12 nbma 100.64.12.1 multicast
 ip nhrp nhs 192.168.200.22 nbma 100.64.22.1 multicast
 ip nhrp registration no-unique
 ip nhrp shortcut
 ip tcp adjust-mss 1360
 if-state nhrp
 tunnel source GigabitEthernet0/2
 tunnel mode gre multipoint
 tunnel key 200
 tunnel vrf INET01
!
ip route vrf MPLS01 0.0.0.0 0.0.0.0 172.16.31.2
```

```
R41-Spoke
vrf definition INET01
 address-family ipv4
 exit-address-family
vrf definition MPLS01
 address-family ipv4
 exit-address-family
!
interface GigabitEthernet0/1
 description MPLS01-TRANSPORT
 vrf forwarding MPLS01
 ip address 172.16.41.1 255.255.255.252
interface GigabitEthernet0/2
 description INET01-TRANSPORT
 vrf forwarding INET01
 ip address dhcp
!
interface Tunnel100
 description DMVPN-MPLS
 bandwidth 4000
 ip address 192.168.100.41 255.255.255.0
 ip mtu 1400
 ip nhrp authentication CISCO
 ip nhrp network-id 100
 ip nhrp holdtime 600
 ip nhrp nhs 192.168.100.11 nbma 172.16.11.1 multicast
 ip nhrp nhs 192.168.100.21 nbma 172.16.21.1 multicast
! The following command keeps the tunnel configuration consistent across all
  tunnels.
 ip nhrp registration no-unique
 ip nhrp shortcut
 ip tcp adjust-mss 1360
 if-state nhrp
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel key 100
 tunnel vrf MPLS01
!
interface Tunnel200
 description DMVPN-INET
 bandwidth 4000
 ip address 192.168.200.41 255.255.255.0
 ip mtu 1400
 ip nhrp authentication CISCO2
 ip nhrp network-id 200
 ip nhrp holdtime 600
 ip nhrp nhs 192.168.200.12 nbma 100.64.12.1 multicast
 ip nhrp nhs 192.168.200.22 nbma 100.64.22.1 multicast
 ip nhrp registration no-unique
 ip nhrp shortcut
 ip tcp adjust-mss 1360
 if-state nhrp
 tunnel source GigabitEthernet0/2
 tunnel mode gre multipoint
 tunnel key 200
 tunnel vrf INET01
!
ip route vrf MPLS01 0.0.0.0 0.0.0.0 172.16.41.2
```

Example 3-43 provides the configuration for both routers (R51 and R52) at Site 5. Notice that the cross-site link does not use a VRF.

Example 3-43 *DMVPN Configuration for R51 and R52 (Dual Routers at Site)*

```
R51-Spoke
vrf definition MPLS01
 address-family ipv4
 exit-address-family
!
interface GigabitEthernet0/1
 description MPLS01-TRANSPORT
 vrf forwarding MPLS01
 ip address 172.16.51.1 255.255.255.252
!
interface GigabitEthernet0/3
 description Cross-Link to R52
 ip address 10.5.12.51 255.255.255.0
!
interface Tunnel100
 description DMVPN-MPLS
 bandwidth 4000
 ip address 192.168.100.51 255.255.255.0
 ip mtu 1400
 ip nhrp authentication CISCO
 ip nhrp network-id 100
 ip nhrp holdtime 600
 ip nhrp nhs 192.168.100.11 nbma 172.16.11.1 multicast
 ip nhrp nhs 192.168.100.21 nbma 172.16.21.1 multicast
 ! The following command keeps the tunnel configuration consistent across all
   tunnels.
 ip nhrp registration no-unique
 ip nhrp shortcut
 ip tcp adjust-mss 1360
 if-state nhrp
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
 tunnel key 100
 tunnel vrf MPLS01
!
ip route vrf MPLS01 0.0.0.0 0.0.0.0 172.16.51.2
```

```
R52
vrf definition INET01
 !
 address-family ipv4
 exit-address-family
!
```

```
interface GigabitEthernet0/2
 description INET01-TRANSPORT
 vrf forwarding INET01
 ip address dhcp
!
interface GigabitEthernet0/3
 description R51
 ip address 10.5.12.52 255.255.255.0
!
interface Tunnel200
 description DMVPN-INET
 bandwidth 4000
 ip address 192.168.200.52 255.255.255.0
 ip mtu 1400
 ip nhrp authentication CISCO2
 ip nhrp network-id 200
 ip nhrp holdtime 600
 ip nhrp nhs 192.168.200.12 nbma 100.64.12.1 multicast
 ip nhrp nhs 192.168.200.22 nbma 100.64.22.1 multicast
 ip nhrp registration no-unique
 ip nhrp shortcut
 ip tcp adjust-mss 1360
 if-state nhrp
 tunnel source GigabitEthernet0/2
 tunnel mode gre multipoint
 tunnel key 200
 tunnel vrf INET01
```

Example 3-44 provides verification of the settings configured on R31. Tunnel 100 has been associated to the MPLS01 VRF, and tunnel 200 has been associated to the INET01 VRF. Both tunnel interfaces have enabled NHRP health monitoring and will bring down the line protocol for the DMVPN tunnels if all of the NHRP NHSs are not available for that tunnel. In addition, R31 has successfully registered with both hubs for tunnel 100 (R11 and R21) and for tunnel 200 (R12 and R22).

Example 3-44 *Verification of DMVPN Settings*

```
R31-Spoke# show dmvpn detail
! Output omitted for brevity
==========================================================================
Interface Tunnel100 is up/up, Addr. is 192.168.100.31, VRF ""
   Tunnel Src./Dest. addr: 172.16.31.1/MGRE, Tunnel VRF "MPLS01"
   Protocol/Transport: "multi-GRE/IP", Protect ""
   Interface State Control: Enabled
   nhrp event-publisher : Disabled
```

```
IPv4 NHS:
192.168.100.11  RE NBMA Address: 172.16.11.1 priority = 0 cluster = 0
192.168.100.21  RE NBMA Address: 172.16.21.1 priority = 0 cluster = 0
Type:Spoke, Total NBMA Peers (v4/v6): 3

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1 172.16.11.1      192.168.100.11    UP 00:09:59     S  192.168.100.11/32
    1 172.16.21.1      192.168.100.21    UP 00:09:31     S  192.168.100.21/32

Interface Tunnel200 is up/up, Addr. is 192.168.200.31, VRF ""
   Tunnel Src./Dest. addr: 100.64.31.1/MGRE, Tunnel VRF "INET01"
   Protocol/Transport: "multi-GRE/IP", Protect ""
   Interface State Control: Enabled
   nhrp event-publisher : Disabled

IPv4 NHS:
192.168.200.12  RE NBMA Address: 100.64.12.1 priority = 0 cluster = 0
192.168.200.22  RE NBMA Address: 100.64.22.1 priority = 0 cluster = 0
Type:Spoke, Total NBMA Peers (v4/v6): 2

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1 100.64.12.1      192.168.200.12    UP 00:12:08     S  192.168.200.12/32
    1 100.64.22.1      192.168.200.22    UP 00:11:38     S  192.168.200.22/32
```

Sample IWAN DMVPN Transport Models

Some network engineers do not fully understand the placement of DMVPN routers (hub or spoke) in a network topology. Combining the FVRF on the encapsulating interface drastically simplifies the concept because the transport network becomes a separate entity from the overlay and LAN networks.

As long as the transport network can deliver the DMVPN packets (unencrypted or encrypted) between the hub and spoke routers, the transport device topology is not relevant to the traffic flowing across the DMVPN tunnel.

Figure 3-9 provides some common deployment models for DMVPN routers in a network.

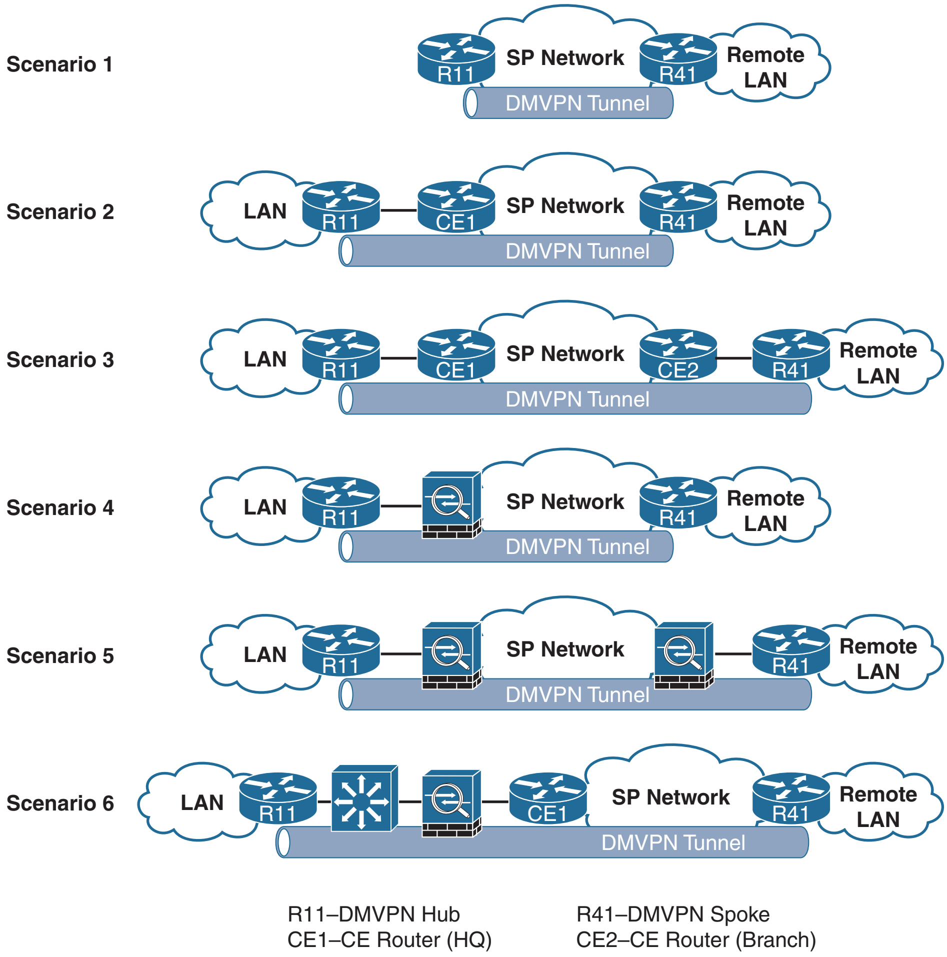


Figure 3-9 *Various DMVPN Deployment Scenarios*

- **Scenario 1:** The DMVPN hub router (R11) directly connects to the SP's PE router and is in essence the CE router. The DMVPN spoke router (R41) directly connects to the SP network at the branch site and is the CE device at the branch site.
- **Scenario 2:** The DMVPN hub router (R11) connects to the HQ CE router (CE1) which connects to the SP network. The DMVPN spoke router (R41) connects directly to the SP network at the branch site.
- **Scenario 3:** The DMVPN hub router (R11) connects to the HQ CE router (CE1) which connects to the SP network. The DMVPN spoke router (R41) connects to the branch CE router (CE2) which connects to the SP network at the branch site.

Note The SP may include the CE devices as part of a managed service. Scenario 3 reflects this type of arrangement, where the SP manages CE1 and CE2 and the DMVPN routers reside behind them. In this scenario, the managed CE devices should be thought of as the actual transport network.

- **Scenario 4:** The DMVPN hub router (R11) connects to a Cisco Adaptive Security Appliance (ASA) firewall which connects to the SP network. The DMVPN spoke router (R41) directly connects to the SP network at the branch site.

  Some organizations require a segmented or layered approach to securing their resources. The ASA creates a DMZ for the DMVPN hub router and is configured so that it allows only DMVPN packets to be forwarded to R11.

  The ASA can also provide static NAT services for R11. DMVPN traffic has to be encrypted for a DVMPN tunnel to form if either endpoint is behind an NAT device.

- **Scenario 5:** The DMVPN hub router (R11) connects to a Cisco ASA firewall which connects to the SP network at the central site. The DMVPN spoke router (R41) connects to an ASA firewall which connects to the SP network at the remote site. In this scenario, the ASAs can provide an additional level of segmentation.

Note In Scenario 5, it is possible for the two ASA firewalls to create a point-to-point IPsec tunnel as part of the transport network. The DMVPN tunnel would not need to be encrypted because the ASAs would encrypt all traffic. However, this would force the DMVPN network to operate in a hub-and-spoke manner because the spoke-to-spoke tunnels will not establish. Cisco ASAs do not support dynamic spoke-to-spoke tunnel creation.

Designs that require encrypted network traffic between branch sites should have the DMVPN routers perform encryption/decryption.

- **Scenario 6:** The DMVPN hub router (R11) connects to a multilayer switch that connects to a Cisco ASA firewall. The ASA firewall connects to the SP's CE router at its HQ site. The DMVPN spoke router (R41) connects directly to the SP network at the remote site.

  The significance of this scenario is that there are additional network hops in the central site that become a component of the transport network from the DMVPN router's perspective. The deeper the DMVPN router is placed into the network, the more design consideration is required to keep the transport network separated from the regular network.

Backup Connectivity via Cellular Modem

Some remote locations use only one physical transport because of their location or the additional cost of providing a second transport. Wireless phone carriers provide an alternative method of connectivity consumed in an on-demand method.

Wireless phone companies charge customers based on the amount of data transferred. Routing protocols consume data for hellos and keepalives on the cellular network, so companies configure the cellular network to be used only when all other networks fail, thus avoiding consumption of data when the primary links are available. This is achieved with the use of

- DMVPN tunnel health monitoring
- Creation of a backup DMVPN tunnel
- Enhanced object tracking (EOT)
- Cisco Embedded Event Manager (EEM)

After DMVPN tunnel health monitoring is enabled on the primary DMVPN interfaces, a dedicated DMVPN tunnel needs to be created for backup connectivity. The backup tunnel interface can belong to an existing DMVPN network but has to be separated from the primary DMVPN tunnel interfaces so that connectivity from the primary interface can be tracked. The FVRF should be different too; otherwise the primary tunnels attempt to use the cellular modem network to register with the hub routers.

For the example configurations in this section, assume that DMVPN tunnel 100 is for the MPLS transport, tunnel 200 is for the Internet transport, and tunnel 300 is the backup cellular modem. The cellular modem should be activated only in the event that DMVPN tunnels 100 and 200 are unavailable.

Enhanced Object Tracking (EOT)

The enhanced object tracking (EOT) feature provides separation between the objects to be tracked and the action to be taken by a client when a tracked object changes. This allows several clients to register their interest with the tracking process, track the same object, and take a different action when the object changes.

Tracked objects are identified by a unique number. The tracking process periodically polls tracked objects and indicates a change of a value. Values are reported as either *up* or *down*.

Example 3-45 provides the configuration for tracking DMVPN tunnels 100 and 200. Objects 100 and 200 track the individual DMVPN tunnel. Object 300 tracks the status of both DMVPN tunnels (nested) and reports a *down* status if both tunnels are down.

Example 3-45 *Configuration of EOT of DMVPN Tunnel Interfaces*

```
track 100 interface Tunnel100 line-protocol
track 200 interface Tunnel200 line-protocol
!
rrack 300 list Boolean or
  object 100
  object 200
  delay 20
```

Embedded Event Manager

The *Embedded Event Manager (EEM)* is a powerful and flexible feature that provides real-time event detection and automation. EEM supports a large number of event detectors that can trigger actions in response to network events. The policies can be programmed to take a variety of actions, but this implementation activates or deactivates the cellular modem.

Two EEM policies need to be created:

- A policy for detection of the failed primary DMVPN tunnel interfaces, which will activate the cellular modem
- A policy for detecting the restoration of service on the primary DMVPN tunnel interface

Example 3-46 displays the EEM policy for enabling the cellular modem upon failure of the primary DMVPN tunnels.

Example 3-46 *EEM Policy to Enable the Cellular Modem*

```
event manager applet ACTIVATE-LTE
event track 300 state down
action 10 cli command "enable"
action 20 cli command "configure terminal"
action 30 cli command "interface cellular0/1/0"
action 40 cli command "no shutdown"
action 50 cli command "end"
action 60 syslog msg "Both tunnels down - Activating Cellular Interface"
```

Example 3-47 displays the EEM policy for disabling the cellular modem upon restoration of service of the primary DMVPN tunnels.

Example 3-47 *EEM Policy to Disable the Cellular Modem*

```
event manager applet DEACTIVATE-LTE
event track 300 state up
action 10 cli command "enable"
action 20 cli command "configure terminal"
action 30 cli command "interface cellular0/1/0"
action 40 cli command "shutdown"
action 50 cli command "end"
action 60 syslog msg "Connectivity Restored - Deactivating Cellular"
```

Note PfRv3 has added the feature of *Last Resort* which may provide a more graceful solution. PfR configuration is explained in Chapter 8, "PfR Provisioning."

IWAN DMVPN Guidelines

The IWAN architecture is a prescriptive design with the following recommendations for DMVPN tunnels:

Design guidelines:

- As of the writing of this book, DMVPN hubs should be connected to only one DMVPN tunnel to ensure that NHRP redirect messages are processed properly for spoke-to-spoke tunnels. In essence, there is one path (transport) per DMVPN hub router. In future software releases such as 16.4.1, multiple transports per hub will be supported. Check with your local Cisco representative or partner for more information.
- DMVPN spokes can be connected to one or multiple transports.
- The DMVPN network should be sized appropriately to support all current devices and additional future locations.
- Ensure proper sizing of bandwidth at DMVPN hub router sites. Multicast network traffic increases the amount of bandwidth needed and should be accounted for. This topic is covered in Chapter 4.
- Use a front-door VRF (FVRF) for each transport. Only a static default route is required in that VRF. This prevents issues with route recursion or outbound interface selection.
- A DMVPN spoke router should connect to multiple active NHSs per tunnel at a time.
- Do not register Internet-based DMVPN endpoint IP addresses in DNS. This reduces visibility and the potential for a DDoS intrusion. Another option is to use a portion of the Internet SP's IP addressing to host the DMVPN hub routers.

- Internet-based DMVPN hub routers should be used solely to provide DMVPN connectivity. Internet edge functions should be provided by different routers or firewalls when possible.
- Use a different SP for each transport to increase failure domains and availability.
- If your SP provides a CE router as part of a managed service, the DMVPN hub or spoke routers are placed behind them. The CE routers should be thought of as part of the actual transport in the design.

Configuration guidelines:

- Use the command **ip nhrp nhs** *nhs-address* **nbma** *nbma-address* [**multicast**] instead of the three commands listed in Table 3-4 for mapping NHRP NHS.
- Enable Phase 3 DMVPN on the spokes with the command **ip nhrp shortcut** and with the command **ip nhrp redirect** on hub routers.
- Define the tunnel MTU, TCP maximum segment size, and tunnel bandwidth.
- Define the same MTU and TCP maximum segment size for all tunnels regardless of the transport used. Failing to do so can result in traffic flows being reset as packets change from one tunnel to a different tunnel.
- Use NHRP authentication with a different password for every tunnel to help detect misconfigurations.
- Remove unique NHRP registration on DMVPN tunnel interfaces with the command **ip nhrp registration no-unique** when connected to transports that are assigned IP addresses by DHCP. For consistency purposes, this command can be enabled on all spoke router tunnel interfaces.
- Maintain consistency in VRF names on the routers, keep the same tunnel interface numbering to transport, and correlate the tunnel ID to the tunnel number. This simplifies the configuration from an operational standpoint.
- Change the NHRP holdtime to 600 seconds.
- Enable NHRP health monitoring only on spoke routers with the command **if-state nhrp**. This brings down the line protocol which will notify the routing protocol.

Troubleshooting Tips

DMVPN can be an intimidating technology to troubleshoot when problems arise but is straightforward if you think about how it works. The following tips will help you troubleshoot basic DMVPN problems:

Tunnel establishment issues:

- Verify that the tunnel interface is not administratively shut down on both DMVPN hub and spoke routers. Then examine the status of the NHS entries on the spoke with the **show dmvpn detail** command. If the tunnel is missing, it is still shut down or not configured properly with NHS settings.

- If the tunnel is in an **NHRP** state, identify the DMVPN hub IP address and ping from the spoke's FVRF context. Example 3-28 demonstrates the verification of connectivity. If pings fail, verify that the packets can reach the gateway defined in the static IP address. The gateway can be identified as shown in Example 3-29.
- After connectivity to the DMVPN hub is confirmed, verify that the NHRP NHS mappings are correct in the tunnel address. The *nhs-address* and *nbma-address* must match what is configured on the DMVPN hub router.
- Then verify that the DMVPN spoke tunnel type is set to **tunnel mode gre multipoint** and that the correct interface is identified for encapsulating traffic.
- Examine NHRP traffic statistics as shown in Example 3-38 or 3-39, and look for NHRP registration requests and reply packets on the DMVPN hubs and spokes.
- Depending on the router's load, debugging NHRP with the command **debug nhrp packet** may provide confirmation of NHRP registration request and reply packets on the hub or spoke router.

Spoke-to-spoke forming issues:

- Verify bidirectional connectivity between spokes on the transport network. This can be accomplished with the **ping** or **traceroute** command from the FVRF context as shown in Example 3-42.
- Verify that traffic flowing from one spoke to another spoke travels through a DMVPN hub router that receives and sends the packets through the same interface. This is required for the hub to send an NHRP redirect message. This can be verified by looking at a traceroute on a spoke router from the global routing table.
- Verify that **ip nhrp redirect** is configured on the DMVPN hub, and that **ip nhrp shortcut** is configured on the DMVPN spoke.

Summary

DMVPN is a Cisco solution that addresses the deficiencies of site-to-site VPNs. It works off a centralized model where remote (spoke) routers connect to centralized (hub) routers. Through the use of multipoint GRE tunnels and NHRP, the spokes are able to establish spoke-to-spoke tunnels, providing full-mesh connectivity between all devices.

This chapter explained the NHRP protocol, multipoint GRE tunnels, and the process by which spoke-to-spoke DMVPN tunnels are established. Any portion of the network on top of which the DMVPN tunnel sends packets is considered the transport network. Any network device (router, switch, firewall, and so on) can reside in the path in the transport network as long as the mGRE packets are forwarded appropriately. Incorporating an FVRF eliminates problems with next-hop selection and route recursion in the transport network. Using multiple DMVPN hub routers for a transport and multiple transports provides resiliency and helps separate failure domains.

Chapter 4 describes the techniques for routing with transport independence, and Chapter 5, "Securing DMVPN Tunnels and Routers," encompasses IPsec encryption for DMVPN tunnels and methods to protect IWAN routers when connected to the Internet.

Further Reading

Cisco. "Cisco IOS Software Configuration Guides." www.cisco.com.

Cisco. "DMVPN Tunnel Health Monitoring and Recovery." www.cisco.com.

Cisco. "IPv6 over DMVPN." www.cisco.com.

Detienne, F., M. Kumar, and M. Sullenberger. Informational RFC, "Flexible Dynamic Mesh VPN." IETF, December 2013. http://tools.ietf.org/html/draft-detienne-dmvpn-01.

Hanks, S., T. Lee, D. Farianacci, and P. Traina. RFC 1702, "Generic Routing Encapsulation over IPv4 Networks." IETF, October 2004. http://tools.ietf.org/html/rfc1702.

Luciani, J., D. Katz, D. Piscitello, B. Cole, and N. Doraswamy. RFC 2332, "NBMA Next Hop Resolution Protocol (NHRP)." IETF, April 1998. http://tools.ietf.org/html/rfc2332.

Sullenberger, Mike. "Advanced Concepts of DMVPN (Dynamic Multipoint VPN)." Presented at Cisco Live, San Diego, 2015.

Chapter 4

Intelligent WAN (IWAN) Routing

This chapter covers the following topics:

- Routing protocol overview
- WAN routing principles
- EIGRP for IWAN
- Border Gateway Protocol (BGP)
- FVRF transport routing
- Multicast routing

The previous chapters described the benefits of a transport-independent architecture and the inner working components of DMVPN. An organization might use a single routing protocol for consistency, or split the routing domain into multiple routing protocols/processes for delineation of operational responsibility. Ensuring that the WAN design incorporates proper design of the routing protocols can simplify the operation and maintenance of the WAN. This chapter explains some of the fundamental WAN concepts that involve multiple transports and the factors for selecting the WAN routing protocol.

Experienced network engineers can often look at a routing protocol configuration and understand why a specific command is in the configuration, whereas others require diagrams or an explanation of the underlying logic. This chapter describes the full logic of the WAN routing protocol from initial configuration to final implementation. At the end of each routing protocol, a complete configuration is provided for all the routers.

Routing Protocol Overview

A routing protocol is classified as either an *Interior Gateway Protocol (IGP)* or an *Exterior Gateway Protocol (EGP)*. An *IGP is* designed and optimized for routing within a single administrative domain of control, referred to in networking as an *autonomous* *system (AS)*. An EGP, on the other hand, is typically used to exchange routes between different ASs. Every routing protocol uses a different logic for advertising, computing best path, and storing routes between routers. The three most common types are

- **Distance vector:** Routers advertise the routing information from their own perspective, modified from the original route that they received. Distance vector protocols do not have a map of the whole network. Their database reflects that a neighbor router knows how to reach the destination network and how far the neighbor router is from the destination network. Distance is typically measured by hop count.
- **Link state:** Link-state routing protocols advertise the link state and link metric for each of their connected links and directly connected routers to every router in the network. All routers maintain an identical copy of the link states so that all the routers in the network have an identical synchronized map of the network.

  Every router in the network computes the same best-path calculation against this map for the best and shortest loop-free path to all destinations.
- **Path vector:** A path vector protocol is similar to a distance vector protocol. The primary difference is that instead of looking at hop count to determine the best loop-free path, a path vector protocol looks at various path attributes to identify the best path.

WAN circuit bandwidth is always lower than what a LAN can provide and can create unique challenges because of the lower-speed transports compared to a LAN. WAN aggregation routers often terminate hundreds or thousands of neighbor adjacencies that consume memory and CPU cycle to bring up and maintain state. Some routing protocols are chatty and unnecessarily consume bandwidth by always flooding a copy of the routing table at set intervals. This consumes additional bandwidth and should be considered when selecting a routing protocol.

Table 4-1 provides a brief summary of the common routing protocols. It includes the protocol type, classification, flooding behavior, and recommendation for DMVPN.

Table 4-1 *Routing Protocol Summary*

Protocol	Protocol Type	Classification	Route Flooding	Recommended for DMVPN
RIP	Distance vector	IGP	Yes, every 30 seconds	No
EIGRP	Distance vector	IGP	No	Yes
OSPF	Link state	IGP	Yes, every 30 minutes	No
IS-IS	Link state	IGP	Yes, every 15 minutes	No
BGP	Path vector	EGP	No	Yes

Note EIGRP is an enhanced distance vector protocol in that it provides rapid convergence, sends updates only when a topology changes, uses hellos to establish and maintain neighbors, uses bandwidth and delay in lieu of hop count for path calculation, and supports unequal load balancing of traffic.

Specifying the routing protocol for the WAN is an important decision. For simplicity, it is best to use the same routing protocol as the rest of the network when possible. However, the following points should be considered:

- IS-IS (Intermediate System-to-Intermediate System) is not supported as an overlay routing protocol for DMVPN because it does not use TCP/IP as its fundamental basis for communications. IS-IS operates at a Layer 2 perspective. DMVPN (specifically NHRP) is IP based.
- RIP continuously floods all routes with every advertisement and has slow convergence. RIP uses only hop count for path selection and cannot identify the difference between a 56 Kbps circuit and a 100 Gbps circuit.
- OSPF is a link-state protocol in which all routers in the same area share an identical database. A topology change in one area requires that all routers in that area perform a complete SPF calculation. If a spoke router loses connectivity to the DMVPN hub, all routers attached to that area perform a complete SPF calculation. OSPF allows network summaries only between OSPF areas, so there is no way to limit the impact of one spoke's connectivity on the other spokes in that DMVPN network. In addition to topology changes, the inability to summarize at the DMVPN hub routers to the spokes in the same tunnel network requires that all the branch routers have the full routing table for all other branch routers. These factors limit the ability of an OSPF network to scale effectively on DMVPN.

A DMVPN network follows the traditional hub-and-spoke routing design model in which spoke routers always receive routing updates from hub routers. DMVPN relies on NHRP for direct routing of spoke-to-spoke network tunnels. This means that distance and path vector protocols work best for DMVPN networks.

The prescriptive IWAN architecture specifies that either EIGRP or IBGP be used for the DMVPN routing protocol. These protocols support summarization at any point of the network, do not flood the network with known routes, and provide interaction to Cisco PfR with their APIs.

Note A significant number of organizations use OSPF or IS-IS as their routing protocol for campus, DC, and/or the core of their network. These protocols can still be used on the LAN segments while using a DMVPN preferred protocol through redistribution. The BGP configuration section of this chapter demonstrates how BGP can interact with OSPF.

Topology

Figure 4-1 displays the dual-hub and dual-cloud topology used in this book to explain the configuration of EIGRP or BGP as the IWAN routing protocol.

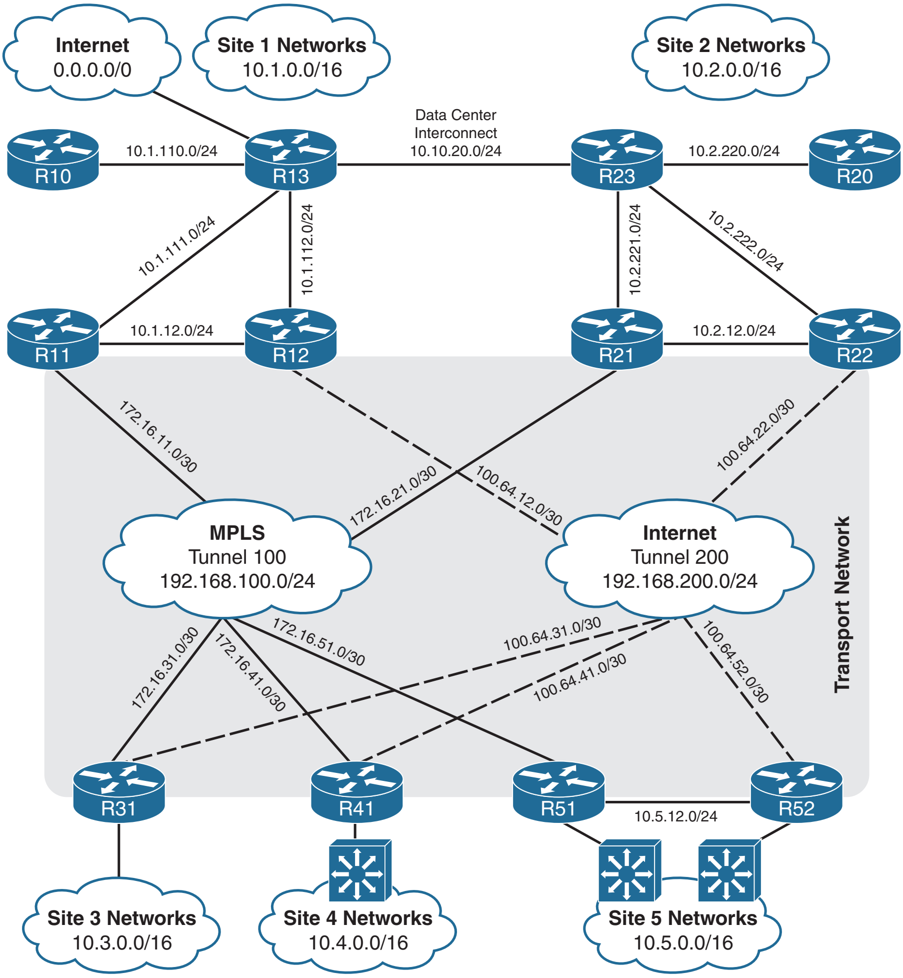


Figure 4-1 *Topology for Routing*

Table 4-2 provides the network site, site subnet, Loopback0 IP address, and transport connectivity for the routers. In the IP addressing scheme, all the LAN-based IP addresses are in the 10.0.0.0/8 network range. The second octet directly correlates to the site number to simplify summarization of network addresses. The last octet in the IP addresses correlates to the router number for LAN and DMVPN networks.

The DMVPN tunnels use 192.168.100.0/24 for tunnel 100 (MPLS), and tunnel 200 uses 192.168.200.0/24 (Internet). The MPLS underlay network uses 172.16.0.0/16, and the Internet underlay network uses 100.64.0.0/12 address space. For both underlay networks, the third octet refers to the router number.

Table 4-2 *Topology Table*

Router	Site Location	Primary Subnet	Loopback0 IP Address	MPLS Transport	Internet Transport
R10	Site 1, DC1	10.1.0.0/16	10.1.0.10		
R11	Site 1, DC1	10.1.0.0/16	10.1.0.11	✓	
R12	Site 1, DC1	10.1.0.0/16	10.1.0.12		✓
R13	Site 1, DC1	10.1.0.0/16	10.1.0.13		
R20	Site 2, DC2	10.2.0.0/16	10.2.0.20		
R21	Site 2, DC2	10.2.0.0/16	10.2.0.21	✓	
R22	Site 2, DC2	10.2.0.0/16	10.2.0.22		✓
R23	Site 2, DC2	10.2.0.0/16	10.2.0.23		
R31	Site 3, branch	10.3.0.0/16	10.3.0.31	✓	✓
R41	Site 4, branch	10.4.0.0/16	10.4.0.41	✓	✓
R51	Site 5, branch	10.5.0.0/16	10.5.0.51	✓	
R52	Site 5, branch	10.5.0.0/16	10.5.0.52		✓

R11 and R21 are the hub routers for DMVPN tunnel 100 (192.168.100.0/24) for the MPLS transport. R12 and R22 are the hub routers for DMVPN tunnel 200 (192.168.200.0/24) for the Internet transport.

R10 and R20 are the PfR hub master controllers (MCs) for Site 1 and Site 2 respectively. R13 and R23 provide connectivity within a DC and between DCs. In essence, they act as distribution routers. Another common solution is the use of multilayer switches (MLSs) for connectivity. In this topology, a dedicated network link (or logical VLAN) is established for exchanging routes and cross-router traffic between R11 and R12, R21 and R22, and R51 and R52. Access to the LAN segments uses a separate network link from the cross-router link.

Note In the topology shown, a dedicated network link is shown for the exchange of routes and cross-router traffic between IWAN routers (R11 and R12, R21 and R22, and R51 and R52). These links are not required, because connectivity can be shared with LAN users. Having a dedicated link (or logical VLAN) may simplify operational troubleshooting.

Three unique topologies are provided at the branch sites in this chapter's topology:

- Site 3 and Site 4 do not have a redundant WAN router, so spoke routers R31 and R41 are connected to both transports via DMVPN tunnels 100 and 200.
  - R31 acts as the Layer 2/Layer 3 boundary (L3 IP connectivity) for all the devices in Site 3.
  - R41 connects to a downstream Cisco Catalyst 4500 MLS. The Catalyst 4500 acts as the Layer 2/Layer 3 boundary for all the devices in Site 4. The Catalyst 4500 and R41 exchange routes via IGP.
- Site 5 has redundant WAN spoke routers: R51 and R52. R51 connects to the MPLS transport with DMVPN tunnel 100, and R52 connects to the Internet transport with DMVPN tunnel 200. R51 and R52 can act as the Layer 2/Layer 3 boundary for Site 5 or exchange routes with downstream devices. The routing configuration remains the same regardless of whether or not they act as the Layer 2/Layer 3 boundary.

WAN Routing Principles

This section explains some basic routing principles for WAN networks. Whether the WAN technology is DMVPN, MPLS VPN (L2 and L3), or point-to-point IPsec tunnels, network engineers and architects need to be aware of these concepts. Incorporating these concepts into the design can simplify the operational aspect of maintaining a WAN network.

Multihomed Branch Routing

Providing redundant connectivity at a remote location increases network availability from the remote user's perspective. It is common for a company to *multihome* a branch site. Multihoming is the connection of a network to two different transport networks to further isolate the failure domain from circuit failure to control plane failure at an SP.

When a site is multihomed, it does one of the following:

- **Load-balance traffic:** Traffic is sent out of both transports equally (active-active).
- **Load-share traffic:** Traffic is sent out of both transports (active-active), but not necessarily equally.
- **Use only one link:** Traffic is sent out through a primary circuit, and the other circuit acts as a backup (active-passive).

Proper network design should take traffic patterns into account to prevent suboptimal routing or routing loops. Figure 4-2 represents a multihomed design using multiple transports for all the sites. All the routers are configured so that they prefer the MPLS SP2 transport over the MPLS SP1 transport (active-passive). All the routers peer and advertise all the routes via EBGP to the SP routers. The routers do not filter any of the prefixes and set the local preference for MPLS SP2 to a higher value to route traffic through it.

When the network is working as intended, traffic between the sites uses the preferred SP network (MPLS SP2) in both directions. This simplifies troubleshooting when the traffic flow is symmetric (same path in both directions) as opposed to asymmetric forwarding (different paths for each direction) because the full path has to be discovered in both directions. The path is considered *deterministic* when the flow between sites is predetermined and predictable.

During a link failure within the SP network, there is the possibility of a branch router connecting to the destination branch router through an intermediary branch router. Figure 4-2 displays the failure scenario at the bottom. In this scenario R41 provides transit connectivity between Site 3 and Site 5.

Unplanned transit connectivity presents the following issues:

- The transit router's circuits can become oversaturated because they were sized only for that site's traffic and not the traffic crossing through it.
- The routing patterns can become unpredictable and nondeterministic. In this scenario, traffic from R31 may flow through R41, but the return traffic may take a different return path. The path can be very different if the traffic was sourced from a different router. This prevents deterministic routing, complicates troubleshooting, and can make your NOC staff feel as if they are playing *Whack-A-Mole* when troubleshooting network issues.

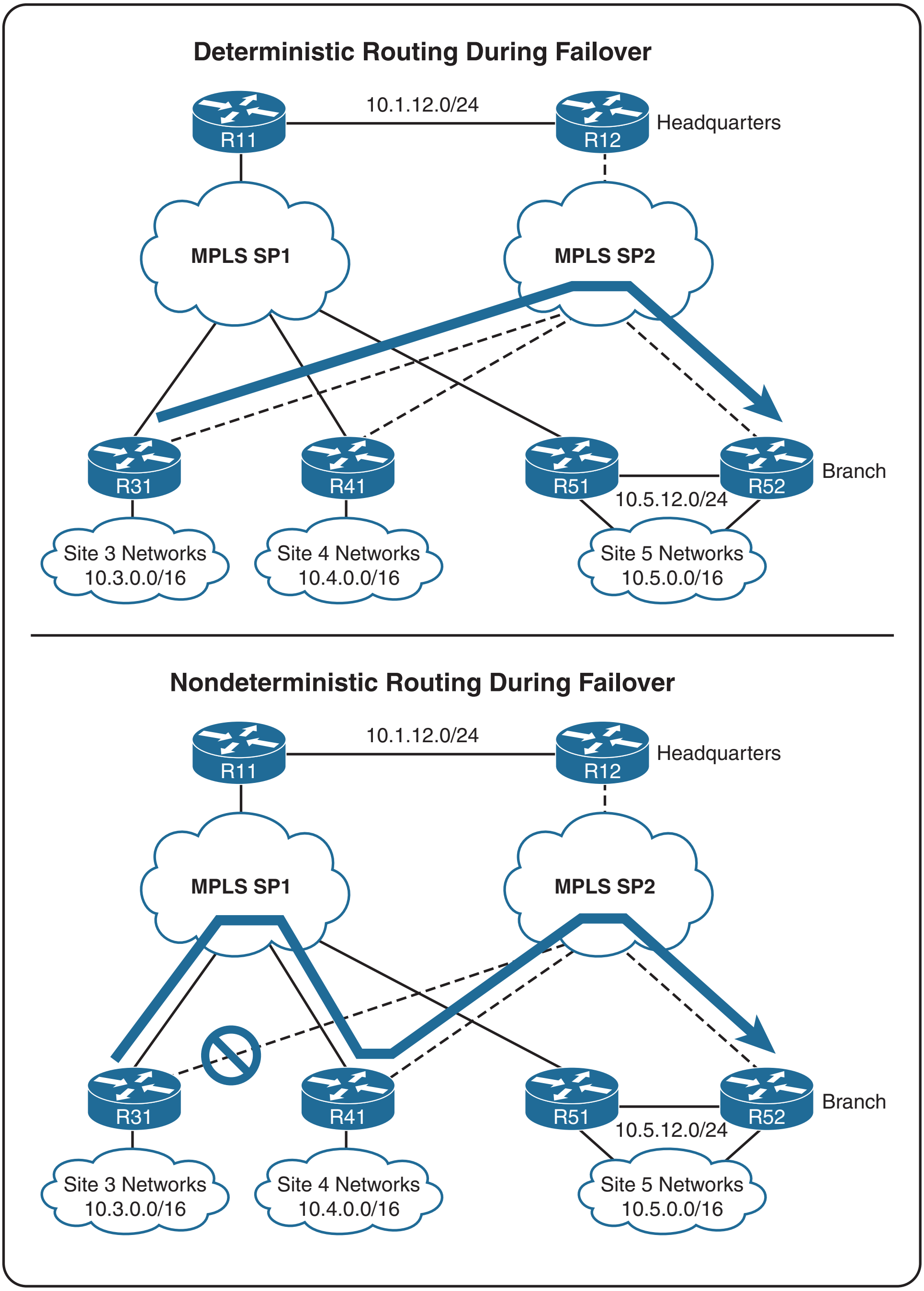


Figure 4-2 *Deterministic and Nondeterministic Routing*

Multihomed environments should be configured so that branch routers cannot act as transit routers. In most designs, transit routing of traffic from another branch or DC is undesirable, because WAN bandwidth may not be sized accordingly. Transit routing can be avoided by configuring outbound route filtering at each branch site. In essence, the branch sites do not advertise what they learn from the WAN but advertise only networks that face the LAN. If transit behavior is required, it is restricted to the hubs so that

- Proper routing design can accommodate outages
- Bandwidth can be sized accordingly
- The routing pattern is bidirectional and predictable

The bottom of Figure 4-2 demonstrates a consistent flow of traffic when a failure occurs.

Advertising only local LAN networks across the WAN allows sites like Site 5 to maintain connectivity when a link from one router fails, while maintaining a consistent routing policy regardless of whether single or multiple routers are deployed at a site.

Note Transit routing at branch sites can cause issues with PfR and automatic site prefix detection.

Route Summarization

Route summarization, also known as route aggregation, is the process of combining smaller, more specific network prefixes (/27) into a larger, less specific network prefix (/24). Route summarization occurs at various places in the network depending upon the routing protocol. EIGRP and BGP can summarize networks anywhere in the network because both are vector-based routing protocols.

Route summarization provides the following benefits:

- Topology changes are hidden from downstream routers. When a topology change occurs on the more specific network prefixes, any route withdrawals or additions are masked by the less specific summary route. This provides stability to the network because downstream routers do not constantly change the routing table.
- The hardware requirements of the routers are reduced because of the smaller routing table.
- Route lookups are faster, because there are fewer entries in the routing table when a packet needs to identify the next hop.

Routes should be summarized on the DMVPN hub routers to reduce the size of the routing table and number of routing updates and to increase convergence times for the branch routers. A proper IWAN routing design summarizes network advertisements (as large as possible) toward the WAN branch networks.

Note EIGRP and BGP allow for directed route summarization, so that some routers receive a summary range and other routers receive the full table. This ability reinforces their selection as routing protocols for DMVPN networks.

These summary routes provide connectivity to all the enterprise devices. An enterprise summary route consists of all the routes needed by a branch to reach other remote WAN sites, DCs, campus networks, and any other networks for business functions and excludes Internet connectivity. Enterprise summary routes typically include private networks defined in RFC 1918 (10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16) and can include any public IP addresses that are accessible internally. NHRP redirects still allow spoke-to-spoke tunnels to establish, and NHRP installs the more explicit routes into the routing table.

Figure 4-3 illustrates the summarization strategy for this book with the following logic:

- All four hub routers summarize the 10.0.0.0/8 network out of their tunnel interfaces. This provides connectivity to all the LAN and WAN sites. In essence, all four routers load-balance any of the initial spoke-to-spoke network traffic between branch routers.
- R11 and R12 advertise a second summary range (10.1.0.0/16 network) that directs traffic to the Site 1 networks to only these routers.
- R21 and R22 advertise a different second summary range (10.2.0.0/16 network) that directs traffic to the Site 2 networks to only these routers.
- A default route is advertised from all four hub routers to provide Internet connectivity. In the event that a branch router uses direct Internet access, the 10.0.0.0/8 summary route provides connectivity to the enterprise prefixes (LAN and WAN networks).

Note The branch routers (R31, R41, R51, and R52) use longest match routing to reach the hub router closest to the network prefix. In the event that an outage occurs with R11 and R12, the 10.1.0.0/16 route is removed from the routing table on the remote routers. The branch routers can connect to the 10.1.0.0/16 networks through the less specific 10.0.0.0/8 route advertised from R21 and R22, which can forward the packets through the backdoor network (10.10.20.0/24).

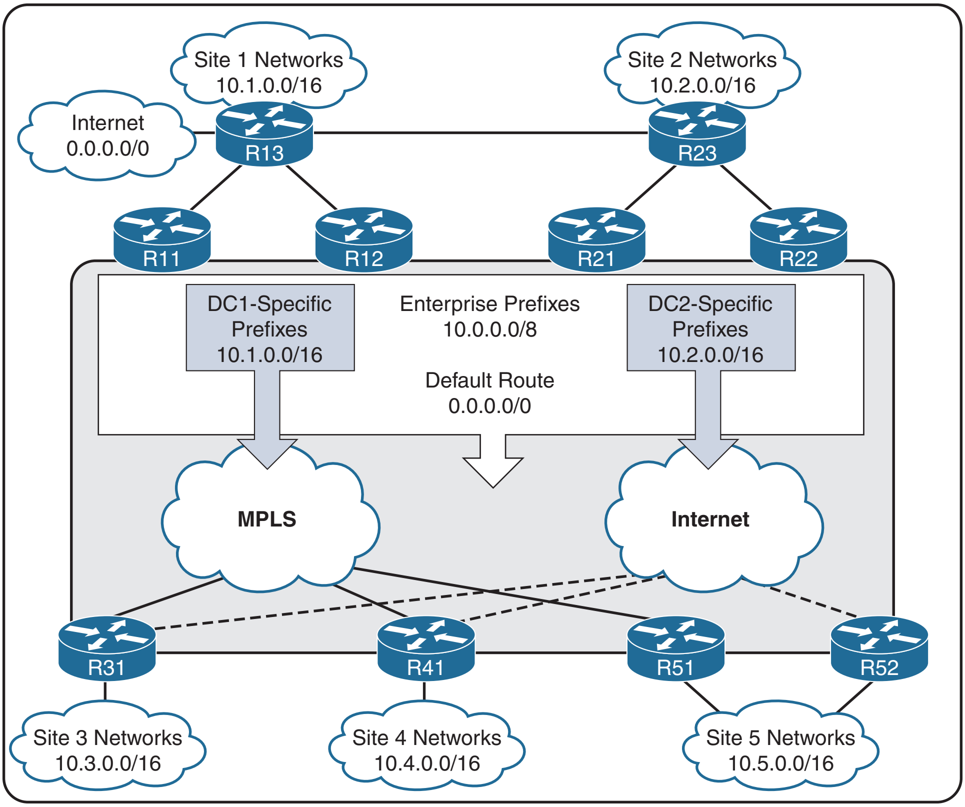


Figure 4-3 *Summarization of Topology*

Note The summary routes for the enterprise prefixes provide connectivity for the LAN and WAN networks. Advertising the default route from the DMVPN hub routers provides centralized Internet access. Although it may be tempting to send only a default route to branches to serve both purposes, this is not advised. When *direct Internet access (DIA)* is deployed, a static default route is placed on the branch router for Internet connectivity. The static route has a lower administrative distance than the routing protocol and is used as long as the router verifies that the direct Internet connection is alive. The enterprise summary prefixes (and DC-specific prefixes) must be present on the branch router connect back to the hubs for the enterprise prefixes while that static default route is installed on the branch router.

Traffic Engineering for DMVPN and PfR

BGP and EIGRP are vector-based routing protocols, and their best-path calculation is based on the information that they receive from their directly connected neighbors. Some network engineers call this *routing by rumor* because the routing decisions are made based only on the information relayed to them. They do not have the complete topology for calculating the best path. Although EIGRP and BGP use different algorithms to calculate the best path, they advertise their perspective of the best path only to their peer routers.

Earlier in Chapter 3, "Dynamic Multipoint VPN," the DMVPN design stated that only one transport should be associated to a DMVPN hub router to ensure that spoke-to-spoke tunnels form. The design also requires that the DMVPN hub router always prefer routes learned from tunnels associated to its transport. The DMVPN hub router should not prefer routes from a different DMVPN hub router when it can connect to the destination network from its own DMVPN tunnel network. This is required so that the spoke-to-spoke tunnel can form properly during the transmission of the initial packets between branch routers. The network traffic must hairpin on the same tunnel interface for the NHRP redirect message to be sent (thereby establishing spoke-to-spoke DMVPN tunnels).

The branch routers receive summary routes from the hub routers and really do not have a way to differentiate one summary prefix on one transport from another. Packets can be sent out of either interface while in an uncontrolled PfR state.

Figure 4-4 displays R11 connecting to the MPLS transport and R12 connecting to the Internet transport. The MPLS transport has a bandwidth of 100 Mbps for all sites, and the Internet transport has only 50 Mbps of bandwidth for all sites. There are four scenarios to help demonstrate the need for traffic engineering:

- **Scenario 1:** Only the MPLS DMVPN tunnel is available. R11 receives the 10.3.0.0/16 routes from R31's MPLS tunnel interface (192.168.100.31). It is the only path on R11. R11 advertises the 10.0.0.0/8 summary prefix to R41's MPLS tunnel interface (192.168.100.41).

  R41 uses R11's tunnel interface (192.168.100.11) to reach the 10.3.0.0/16 network through the 10.0.0.0/8 summary prefix. R11 has only one path, which is out of the same tunnel interface where the packet was received, and thereby sends an NHRP redirect to R41 so that a spoke-to-spoke tunnel can establish.

- **Scenario 2:** Only the Internet DMVPN tunnel is available. R12 receives the 10.3.0.0/16 routes from R31's Internet tunnel interface (192.168.200.31). It is the only path on R12. R12 advertises the 10.0.0.0/8 summary prefix to R41's Internet tunnel interface (192.168.200.41).

R41 uses R12's tunnel interface (192.168.200.12) to reach the 10.3.0.0/16 network through the 10.0.0.0/8 summary prefix. R12 has only one path, which is out of the same tunnel interface where the packet was received, and thereby sends an NHRP redirect to R41 so that a spoke-to-spoke tunnel can establish.

- **Scenario 3:** Both transports and their tunnels are available. R11 and R12 receive the 10.3.0.0/16 routes from both of R31's tunnels. R11 calculates the path to the 10.3.0.0/16 network through R31 as the best path because the MPLS transport has more bandwidth associated to it than the Internet transport. R12 calculates the path to the 10.3.0.0/16 network through R11 as the best path. Both advertise the 10.0.0.0/8 summary prefix. There is no indication by path attribute/metric in R12's summary prefix advertisement that R12 has selected the best path learned via R11.

  R41 receives the 10.0.0.0/8 summary prefix from R11 and R12. If R41 sends a packet to 10.3.0.0/16 through R11's 10.0.0.0/8 summary prefix, R11 sends the packet to R31 out of the same tunnel interface where the packet was received. This triggers the NHRP redirect to R41 so that a spoke-to-spoke tunnel can form.

  If R41 sends a packet to 10.3.0.0/16 through R12's 10.0.0.0/8 summary prefix, R12 sends the packet to R11 across the cross-link network. An NHRP redirect is not sent to R41, and a spoke-to-spoke tunnel is not created for the Internet transport.

  In this scenario, a spoke-to-spoke tunnel forms only for packets sent across the MPLS transport.

- **Scenario 4:** Both transports and their tunnels are available. R11 and R12 receive the 10.3.0.0/16 routes from both of R31's tunnels. R11 calculates the path to the 10.3.0.0/16 network through R31 as the best path. R12 calculates the path to the 10.3.0.0/16 network through R31 as the best path. Both advertise the 10.0.0.0/8 summary prefix.

  R41 receives the 10.3.0.0/16 network from R11 and R12. If R41 sends a packet to 10.3.0.0/16 through R11's 10.0.0.0/8 summary prefix, R11 sends the packet to R31 out of the same tunnel interface where the packet was received and sends out an NHRP redirect to R41 so that a spoke-to-spoke tunnel can form.

  If R41 sends a packet to 10.3.0.0/16 through R12's 10.0.0.0/8 summary prefix, R12 sends the packet to R31 out of the same tunnel interface where the packet was received and sends out an NHRP redirect to R41 so that a spoke-to-spoke tunnel can form.

Note In all four scenarios, traffic between R31 and R41 can occur without packet loss. The scenario emphasizes the ability for spoke-to-spoke DMVPN tunnel establishment.

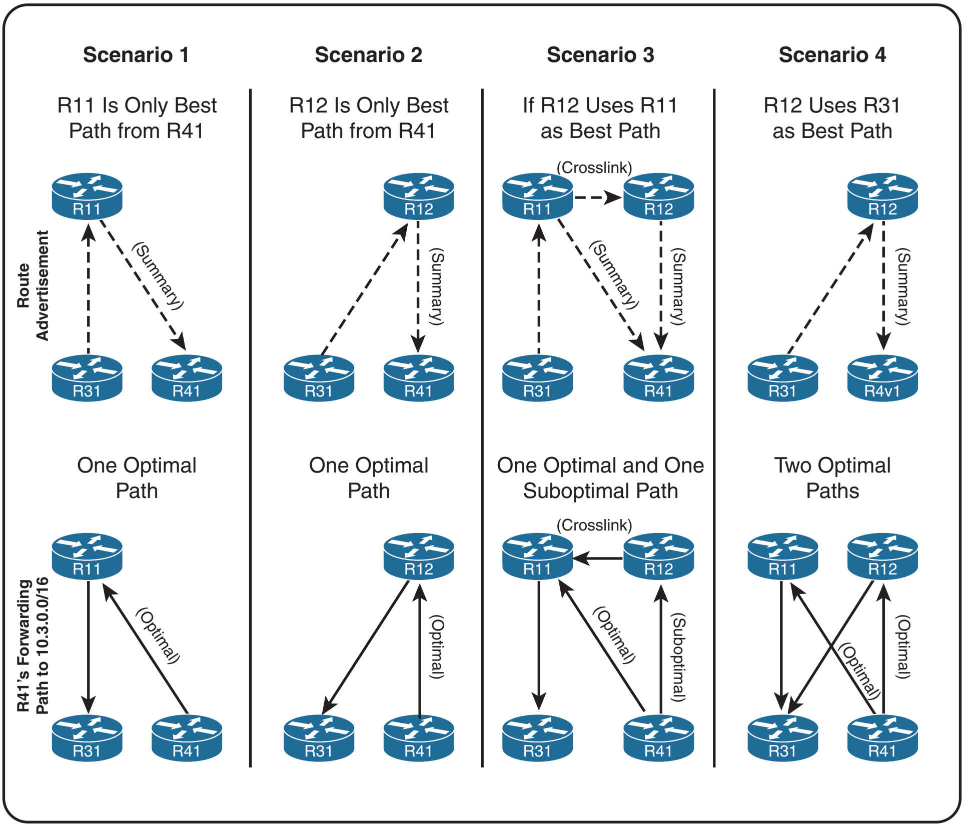

Scenario 1
Scenario 2
Scenario 3
Scenario 4
R11 Is Only Best Path from R41
R12 Is Only Best Path from R41
If R12 Uses R11 as Best Path
R12 Uses R31 as Best Path
Route Advertisement
(Crosslink)
R11
R12
R31
R41
R4v1
(Summary)
One Optimal Path
One Optimal Path
One Optimal and One Suboptimal Path
Two Optimal Paths
R41's Forwarding Path to 10.3.0.0/16
(Optimal)
(Suboptimal)

Figure 4-4 *Routing Protocol Best-Path Impact on Spoke-to-Spoke Tunnels*

The routing protocol design should ensure that the DMVPN hub router always prefers routes learned from the spoke router's interface attached to the same transport as the hub router.

Note It is important to note that when PfR is in an uncontrolled state, it cannot influence the path that network traffic takes. The IWAN design should account for this and direct network traffic toward the primary tunnel interface. When PfR controls the traffic, it then redirects or load-balances traffic across the tunnel interfaces (other paths).

EIGRP for IWAN

The EIGRP routing logic is simplified with every component (branch LAN, WAN, and headquarters LAN) of the network using EIGRP. There is no redistribution between routing protocols or loss of path visibility in the topology. R13 advertises the default route out of all its interfaces to provide connectivity to the Internet. The DMVPN hub routers advertise the summary routes as described before, and all the more specific routes still exist in the headquarters LAN, as shown in Figure 4-5.

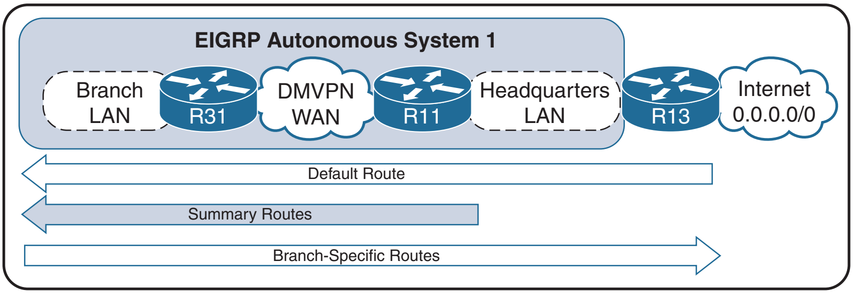


Figure 4-5 *EIGRP Routing Logic*

This section addresses the EIGRP configuration in logical sections to explain why portions of the configuration exist. The configuration is explained in the following sections:

- Base EIGRP configuration and verification of EIGRP neighbors
- EIGRP stub sites on branch routers to prevent transit routing
- EIGRP summarization and summarization metrics
- EIGRP traffic steering

After the last section, the complete EIGRP configuration is provided for all the routers.

Base Configuration

EIGRP has two configuration modes: classic AS and named mode. Named mode consolidates the EIGRP configuration to a centralized location, provides a clear scope of the commands, supports wide metrics, and is required for IWAN because of the newer features it contains. EIGRP wide metrics addresses the issue of scalability with higher-capacity interfaces (10 Gigabit Ethernet and higher).

It is assumed that the reader is familiar with EIGRP and can configure EIGRP on downstream routers in the network. This section focuses on the configuration of EIGRP on the IWAN routers.

Step 1. **Initialize the EIGRP process.**

The EIGRP process is initialized with the command **router eigrp** *process-name*.

Step 2. **Define the instance.**

The EIGRP instance is initialized for the appropriate address family with the command **address-family** {**ipv4**| **ipv6**} **unicast autonomous-system** *as-number*. The *autonomous system numbers (ASNs)* must match for EIGRP routers to become neighbors. This design uses the same ASN throughout the entire network.

Step 3. **Enable the interface.**

The network statement identifies the interfaces that EIGRP will use. The network statement uses a wildcard mask, which allows the configuration to be as specific or ambiguous as necessary. The syntax for the network statement is **network** *ip-address* [*wildcard-mask*] and exists under the EIGRP process. If the wildcard mask is omitted, the statement is entered by the closest *Classful* network address.

The network statement identifies an interface and adds the interface's connected network to the EIGRP topology table. EIGRP then advertises the topology table to other EIGRP routers.

Step 4. **Modify the hello and hold timers.**

The EIGRP hello interval is incremented to 20 seconds, and the hold timer is incremented to 60 seconds. Increasing the timers allows the DMVPN hub routers to handle a large number of remote sites. The hello and hold timers should match on the DMVPN hubs and spokes.

The DMVPN tunnel interface is specified with the command **af-interface tunnel** *tunnel-number*. Then the hello timer is set with the command **hello-interval 20** and the hold timer is set with the command **hold-time 60**.

Step 5. **Disable split horizon (hubs only).**

Most distance vector routing protocols use *split horizon* to prevent routing loops. Split horizon simply prevents a route from being advertised out of the same interface on which it was learned. If split horizon is enabled, the hub route does not advertise the networks learned from one spoke router (R31) to another spoke router (R41). Split horizon is enabled in the event that a branch site advertises networks that are not contained in the network summaries.

The DMVPN tunnel interface must be specified with the command **af-interface tunnel** *tunnel-number*. Then split horizon is disabled with the command **no split-horizon**.

Step 6. **Set the router ID.**

The *router ID (RID)* is a 32-bit number that uniquely identifies an EIGRP router. By default, Cisco IOS routers use the highest IP address of any up loopback interfaces. If there are no up loopback interfaces, the highest IPv4 address of any active up physical interfaces becomes the RID when the EIGRP process initializes. It is considered a best practice to statically configure the router ID. In this book, the router ID matches the loopback interface.

The command **eigrp router-id** *router-id* is used to set the RID.

Example 4-1 provides the basic EIGRP configuration for the DMVPN hub routers. Notice how the interface-based settings reside under the EIGRP routing process because of the use of EIGRP named mode.

Example 4-1 *EIGRP Configuration for DMVPN Hub Routers*

```
R11 and R21
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  af-interface Tunnel100
   hello-interval 20
   hold-time 60
   no split-horizon
  exit-af-interface
```

```
R12 and R22
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  af-interface Tunnel200
   hello-interval 20
   hold-time 60
   no split-horizon
  exit-af-interface
```

```
R11
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  eigrp router-id 10.1.0.11
  !
  network 10.1.0.0 0.0.255.255
  network 192.168.100.0
```

```
R12
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  eigrp router-id 10.1.0.12
  !
  network 10.1.0.0 0.0.255.255
  network 192.168.200.0
```

```
R21
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  eigrp router-id 10.2.0.21
  !
  network 10.2.0.0 0.0.255.255
  network 192.168.100.0
```

```
R22
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  eigrp router-id 10.2.0.22
  !
  network 10.2.0.0 0.0.255.255
  network 192.168.200.0
```

Example 4-2 provides the EIGRP configuration for the DMVPN spoke routers.

Example 4-2 *EIGRP Configuration for DMVPN Spoke Routers*

```
R31
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  af-interface Tunnel100
   hello-interval 20
   hold-time 60
  exit-af-interface
  af-interface Tunnel200
   hello-interval 20
   hold-time 60
  exit-af-interface
  !
  topology base
  exit-af-topology
  network 10.0.0.0
  network 192.168.100.0
  network 192.168.200.0
  eigrp router-id 10.3.0.31
```

```
R41
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  af-interface Tunnel100
   hello-interval 20
   hold-time 60
  exit-af-interface
  af-interface Tunnel200
   hello-interval 20
   hold-time 60
  exit-af-interface
  !
  topology base
  exit-af-topology
  network 10.0.0.0
  network 192.168.100.0
  network 192.168.200.0
  eigrp router-id 10.4.0.41
```

```
R51
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  af-interface Tunnel100
   hello-interval 20
   hold-time 60
  exit-af-interface
  !
  topology base
  exit-af-topology
  network 10.0.0.0
  network 192.168.100.0
  eigrp router-id 10.5.0.51
```

```
R52
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  af-interface Tunnel200
   hello-interval 20
   hold-time 60
  exit-af-interface
  !
  topology base
  exit-af-topology
  network 10.0.0.0
  network 192.168.200.0
  eigrp router-id 10.5.0.52
```

Verification of EIGRP Neighbor Adjacencies

Each EIGRP process maintains a table of neighbors to ensure that they are alive and processing updates accordingly. Without keeping track of a neighbor state, an autonomous system can contain incorrect data and potentially route traffic improperly. The EIGRP neighbor table is shown with the command **show ip eigrp neighbor**.

In the DMVPN architecture the spoke routers initialize static tunnels to the hub routers. The hub routers send NHRP redirect messages so that direct spoke-to-spoke communication can occur dynamically. Although the spokes can communicate directly with each other with spoke-to-spoke tunnels, spoke routers become EIGRP neighbors only with hub routers.

Example 4-3 provides the neighbor adjacencies for R11, R12, and R31. R11 forms an adjacency with the DMVPN tunnel 100 neighbors (R31, R41, and R51) and the other routers in Site 1 (R13 and R12). R12 forms an adjacency with the DMVPN tunnel 200 neighbors (R31, R41, and R52) and the other routers in Site 1 (R13 and R11).

Example 4-3 *EIGRP Neighbor Confirmation*

```
R11-Hub# show ip eigrp neighbors
EIGRP-IPv4 VR(IWAN) Address-Family Neighbors for AS(1)
H   Address                 Interface           Hold Uptime   SRTT   RTO  Q  Seq
                                                (sec)         (ms)       Cnt Num
4   192.168.100.51          Tu100                 53 00:17:44    8   100  0  121
3   192.168.100.41          Tu100                 54 00:18:40    6   100  0  64
2   192.168.100.31          Tu100                 58 00:19:00    3   100  0  70
1   10.1.12.12              Gi0/3                 14 00:36:56    1   100  0  142
0   10.1.111.10             Gi1/0                 12 00:36:56    1   100  0  149
```

```
R12-Hub# show ip eigrp neighbors
EIGRP-IPv4 VR(IWAN) Address-Family Neighbors for AS(1)
H   Address                 Interface           Hold Uptime   SRTT   RTO  Q  Seq
                                                (sec)         (ms)       Cnt Num
4   192.168.200.52          Tu200                 45 00:01:19    6   100  0  97
3   192.168.200.41          Tu200                 55 00:01:19    3   100  0  78
2   192.168.200.31          Tu200                 56 00:02:32    1   100  0  89
1   10.1.112.10             Gi1/0                 13 00:41:22    1   100  0  157
0   10.1.12.11              Gi0/3                 13 00:41:22    1   100  0  183
```

```
R31-Spoke# show ip eigrp neighbors
EIGRP-IPv4 VR(IWAN) Address-Family Neighbors for AS(1)
H   Address                 Interface           Hold Uptime   SRTT   RTO  Q  Seq
                                                (sec)         (ms)       Cnt Num
3   192.168.200.22          Tu200                 54 00:02:58    2   100  0  147
2   192.168.200.12          Tu200                 47 00:02:58    1   100  0  158
1   192.168.100.21          Tu100                 56 00:23:53    8   100  0  171
0   192.168.100.11          Tu100                 48 00:23:53    4   100  0  181
```

EIGRP Stub Sites on Spokes

EIGRP stub functions conserve router resources and improve network stability. When a link or router fails, EIGRP marks a route as active and sends out queries to locate an alternative route to that network. EIGRP does not send EIGRP queries to an EIGRP stub router. This provides faster convergence within an EIGRP autonomous system because it decreases the size of the query domain for that prefix. An EIGRP stub router announces itself as a stub within the EIGRP hello packet. Neighboring routers detect the stub field and update the EIGRP neighbor table to reflect the router's stub status so that it knows not to query a router.

EIGRP stub site functions build on EIGRP stub capabilities that allow a router to advertise itself as a stub to peers only on the specified WAN interfaces but allow it to exchange routes learned on LAN interfaces. EIGRP stub sites provide the following key benefits:

- EIGRP neighbors on WAN links do not send EIGRP queries to the remote site when a route becomes active.
- They allow downstream routers to receive and advertise network prefixes across the WAN.
- They prevent the EIGRP stub site from being a transit router.

The EIGRP stub site feature works by identifying the WAN interfaces and then setting an EIGRP stub site identifier. Routes received from a peer on the WAN interface are tagged with an EIGRP stub site identifier attribute. When EIGRP advertises network prefixes out of a WAN-identified interface, it checks for an EIGRP stub site identifier. If one is found, the route is not advertised; if one is not found, the route is advertised.

Figure 4-6 illustrates the concept further:

1. R11 advertises the 10.1.1.0/24 route to R41, and the 10.1.1.0/24 route is received on R41's WAN interface. R41 then is able to advertise that prefix to the downstream router R40.
2. R12 advertises the 10.1.2.0/24 route to R41, and the 10.1.1.0/24 route is received on R41's other WAN interface. R41 then is able to advertise that prefix to the downstream router R40.
3. R40 advertises the 10.4.4.0/24 network to R41. R41 checks the 10.4.4.0/24 route for the EIGRP stub site attribute before advertising that prefix out of either WAN interface. R41 is able to advertise the prefix because it does not contain an EIGRP stub site identifier attribute.

Notice that R41 does not advertise the 10.1.1.0/24 prefix to R12, and that it does not advertise the 10.1.2.0/24 prefix to R11. This is because the EIGRP stub site attribute was added upon receipt of the prefix and blocked during advertisement out of the other WAN interface.

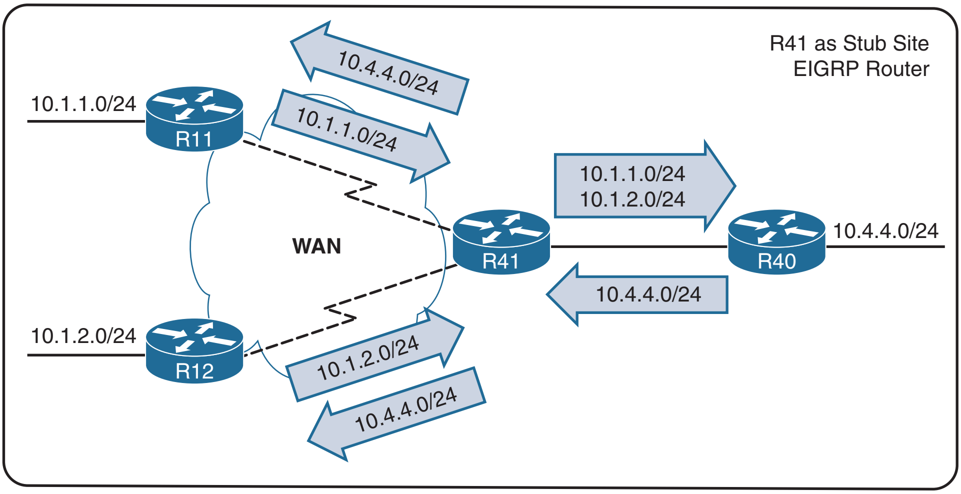


Figure 4-6 *EIGRP Stub Functions*

WAN interfaces are identified with the EIGRP **af-interface** command **stub-site wan-interface**. The stub site feature and identifier are enabled with the command **eigrp stub-site** *as-number:identifier*.

Note Configuring an EIGRP stub site resets the EIGRP neighbor(s) for that router.

The EIGRP stub site feature prevents transit routing at a branch site on sites that have one router (Site 3) or multiple routers (Site 5). Although not required, configuring the EIGRP stub site feature on all branch routers keeps the configuration consistent and allows the additional nondisruptive deployment of routers at that site in the future. Example 4-4 provides the EIGRP stub site configuration for R31, R41, R51, and R52. Although R31 and R41 do not have any LAN routers attached at the time, the stub site prevents transit routing.

Notice that the EIGRP stub site is set to 1:1 on all the routers. This is done because it has significance only behind the branch router. Keeping the number the same allows consistent configuration across all branches, simplifying template development with tools like Cisco Prime Infrastructure.

Example 4-4 *EIGRP Stub Site Configuration*

```
R31
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  af-interface Tunnel100
   stub-site wan-interface
  exit-af-interface
  !
  af-interface Tunnel200
   stub-site wan-interface
  exit-af-interface
  eigrp stub-site 1:1
 exit-address-family
```

```
R41
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  af-interface Tunnel100
   stub-site wan-interface
  exit-af-interface
  !
  af-interface Tunnel200
   stub-site wan-interface
  exit-af-interface
  eigrp stub-site 1:1
 exit-address-family
```

```
R51
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  af-interface Tunnel100
   stub-site wan-interface
  exit-af-interface
  !
  eigrp stub-site 1:1
 exit-address-family
```

```
R52
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  af-interface Tunnel200
   stub-site wan-interface
  exit-af-interface
  eigrp stub-site 1:1
 exit-address-family
```

Example 4-5 verifies that the 10.1.12.0/24 route learned via DMVPN tunnel 100 is identified as a route learned via a WAN interface. Notice that both entries (DMVPN tunnels 100 and 200) indicate the stub site identifier.

Example 4-5 *Verification of Routes Learned via the WAN Interface*

```
R31-Spoke# show ip eigrp topology 10.1.12.0/24
! Output omitted for brevity
EIGRP-IPv4 VR(IWAN) Topology Entry for AS(1)/ID(10.3.0.31) for 10.1.12.0/24
  Descriptor Blocks:
  192.168.100.11 (Tunnel100), from 192.168.100.11, Send flag is 0x0
        Originating router is 10.1.0.11
      Extended Community: StubSite:1:1
  192.168.200.12 (Tunnel200), from 192.168.200.12, Send flag is 0x0
        Originating router is 10.1.0.12
      Extended Community: StubSite:1:1
```

Example 4-6 verifies that R11 recognizes R31 as an EIGRP stub router and will not send it any queries when a route becomes active.

Example 4-6 *EIGRP Stub Router Flags*

```
R11-Hub# show ip eigrp neighbors detail tunnel 100
! Output omitted for brevity
EIGRP-IPv4 VR(IWAN) Address-Family Neighbors for AS(1)
H   Address                 Interface         Hold Uptime   SRTT   RTO  Q  Seq
                                              (sec)         (ms)       Cnt Num
2   192.168.100.31          Tu100                48 00:23:40    1   100  0  18
   Version 20.0/2.0, Retrans: 0, Retries: 0, Prefixes: 2
   Topology-ids from peer - 0
   Topologies advertised to peer:   base

   Stub Peer Advertising (CONNECTED STATIC SUMMARY REDISTRIBUTED ) Routes
   Suppressing queries
```

Note Branch sites with dual routers that do not support the EIGRP stub site feature should upgrade their software when feasible. If this cannot be done, routes learned via DMVPN tunnels need to be tagged inbound via a distribute list and then blocked out of the DMVPN tunnel to prevent transit routing.

EIGRP Summarization

Scalability of the EIGRP routing domain is dependent upon summarization. As the size of the EIGRP network increases, convergence may take longer. Summarizing multiple routes to an aggregate reduces the size of the routing table and creates a query boundary. Query boundaries stop EIGRP queries when a route becomes active, and they reduce convergence times.

EIGRP summarizes network prefixes on an interface basis. Prefixes within the summary aggregate are suppressed, and the summary aggregate prefix is advertised in lieu of the original prefixes. The summary aggregate prefix is not advertised until a prefix matches it. Interface-specific summarization can be performed at any portion of the network topology.

Summarization is configured under the **af-interface** with the command **summary-address** *network subnet-mask.*

Note Configuring an EIGRP summary aggregate on an interface resets the EIGRP neighbor(s) connected via that interface.

An inbound prefix list filter is applied on the DMVPN hub routers to prevent a DMVPN router from learning any of the summary routes that were advertised from a peer hub router. This includes the default route, enterprise summary, or DC-specific prefixes. Example 4-7 provides the summarization configuration for the hub routers.

Example 4-7 *EIGRP Summarization Commands*

```
R11, R12, R21, and R22
ip prefix-list EIGRPSUMMARYROUTES seq 10 deny 0.0.0.0/0
ip prefix-list EIGRPSUMMARYROUTES seq 20 deny 10.0.0.0/8
ip prefix-list EIGRPSUMMARYROUTES seq 30 deny 10.1.0.0/16
ip prefix-list EIGRPSUMMARYROUTES seq 40 deny 10.2.0.0/16
ip prefix-list EIGRPSUMMARYROUTES seq 50 permit 0.0.0.0/0 le 32
```

```
R11
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  af-interface Tunnel100
   summary-address 10.0.0.0 255.0.0.0
   summary-address 10.1.0.0 255.255.0.0
  exit-af-interface
  !
  topology base
   distribute-list prefix EIGRPSUMMARYROUTES in Tunnel100
```

```
R12
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  af-interface Tunnel200
   summary-address 10.0.0.0 255.0.0.0
   summary-address 10.1.0.0 255.255.0.0
  exit-af-interface
  !
  topology base
   distribute-list prefix EIGRPSUMMARYROUTES in Tunnel200
```

```
R21
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  af-interface Tunnel100
   summary-address 10.0.0.0 255.0.0.0
   summary-address 10.2.0.0 255.255.0.0
  exit-af-interface
  !
  topology base
   distribute-list prefix EIGRPSUMMARYROUTES in Tunnel100
```

```
R22
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  af-interface Tunnel200
   summary-address 10.0.0.0 255.0.0.0
   summary-address 10.2.0.0 255.255.0.0
  exit-af-interface
  !
  topology base
   distribute-list prefix EIGRPSUMMARYROUTES in Tunnel200
```

Note The configuration is not shown, but R13 and R23 have summarized only the 10.1.0.0/16 network and 10.2.0.0/16 network appropriately on the backdoor network between the two sites.

The summarizing router uses the lowest metric of the routes in the summary aggregate prefix. The path metric for the summary aggregate is based upon the path attributes of the lowest metric path. EIGRP path attributes such as total delay and minimum bandwidth are inserted into the summary route so that downstream routers can calculate the correct path metric for the summarized prefix.

Every time a matching prefix for the summary aggregate is added or removed, EIGRP must verify that the advertised path is still the lowest path metric. If it is not, a new summary aggregate is advertised with updated EIGRP attributes, and downstream routes must run the Diffusing Update Algorithm (DUAL) again. The summary aggregate hides the smaller prefixes from downstream routers, but downstream routers are still burdened with processing updates to the summary aggregate.

This behavior is overcome and consumption of router resources is reduced by using the topology command **summary-metric** *network* {*/prefix-length* | *subnet-mask*} *bandwidth delay reliability load mtu* [**distance** *ad*]. Bandwidth is in kilobits per second, delay is in 10-microsecond units, reliability and load are values between 1 and 255, and MTU is the maximum transmission unit for the interface. A summary metric is used to reduce computational load on the DMVPN hubs.

EIGRP uses the path's minimum bandwidth as part of the metric calculation. The path's minimum bandwidth is defined in a route advertisement in the minimum bandwidth path attribute. Setting the summary metric bandwidth to a low value (10 Mbps) essentially removes the ability to differentiate between a 10 Mbps tunnel (MPLS) and a 100 Mbps circuit (Internet) because both paths have a minimum bandwidth of 10 Mbps. Setting the summary metric bandwidth to 10 Gbps then allows the calculations on the branch router to differentiate tunnel bandwidth.

As part of the summarization process, EIGRP automatically installs a route for the summary aggregate prefix with a destination of Null0 as part of a loop prevention mechanism. By default the *administrative distance (AD)* for the Null0 route is 5, which may cause problems if a router further upstream advertises the same network to provide connectivity. It can potentially lead to blackholing of traffic at a hub site because the Null0 route has a lower AD than the upstream route in EIGRP. The use of the **distance** keyword with the summary metric allows the AD to be set to a value higher than the IGP with the valid route, which allows the valid IGP route to install in the RIB (also known as the routing table) on the hub routers.

Example 4-8 contains the summary metric configuration for the hub routers.

Example 4-8 *EIGRP Summarization Metric Commands*

```
R11 and R12
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  topology base
   summary-metric 10.1.0.0/16 10000000 1 255 0 1500 distance 250
   summary-metric 10.0.0.0/8 10000000 1 255 0 1500 distance 250
```

```
R21 and R22
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  topology base
   summary-metric 10.0.0.0/8 10000000 1 255 0 1500 distance 250
   summary-metric 10.2.0.0/16 10000000 1 255 0 1500 distance 250
```

Example 4-9 displays the routing table for the topology after the DMVPN hubs have enabled summarization for the branch networks. The routing table is smaller, route lookups should occur faster, and spoke-to-spoke routing will still occur with the NHRP redirects.

Example 4-9 *Routing Table After Summarization*

```
R31-Spoke# show ip route
! Output omitted for brevity
Gateway of last resort is 192.168.200.12 to network 0.0.0.0

D*    0.0.0.0/0 [90/522880] via 192.168.200.12, 01:39:29, Tunnel200
                [90/522880] via 192.168.100.11, 01:39:29, Tunnel100
      10.0.0.0/8 is variably subnetted, 6 subnets, 4 masks
D        10.0.0.0/8 [90/522240] via 192.168.200.22, 00:10:52, Tunnel200
                    [90/522240] via 192.168.200.12, 00:10:52, Tunnel200
                    [90/522240] via 192.168.100.21, 00:10:52, Tunnel100
                    [90/522240] via 192.168.100.11, 00:10:52, Tunnel100
D        10.1.0.0/16 [90/522240] via 192.168.200.12, 00:11:08, Tunnel200
                     [90/522240] via 192.168.100.11, 00:11:08, Tunnel100
D        10.2.0.0/16 [90/522240] via 192.168.200.22, 00:10:52, Tunnel200
                     [90/522240] via 192.168.100.21, 00:10:52, Tunnel100
C        10.3.0.31/32 is directly connected, Loopback0
C        10.3.3.0/24 is directly connected, GigabitEthernet1/0
      192.168.100.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
      192.168.200.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.200.0/24 is directly connected, Tunnel200
```

When the spoke-to-spoke tunnels initialize, NHRP installs more specific routes into the RIB while those tunnels remain active. Example 4-10 shows the 10.4.4.0/24 network installed in the RIB after R31 and R41 establish a spoke-to-spoke tunnel for the 10.4.4.0/24 network.

Example 4-10 *Routing Table After Summarization with NHRP Route Injection*

```
R31-Spoke# show ip route
! Output omitted for brevity
Gateway of last resort is 192.168.200.12 to network 0.0.0.0

D*    0.0.0.0/0 [90/522880] via 192.168.200.12, 01:42:38, Tunnel200
                [90/522880] via 192.168.100.11, 01:42:38, Tunnel100
      10.0.0.0/8 is variably subnetted, 7 subnets, 4 masks
D        10.0.0.0/8 [90/522240] via 192.168.200.22, 00:14:01, Tunnel200
                    [90/522240] via 192.168.200.12, 00:14:01, Tunnel200
                    [90/522240] via 192.168.100.21, 00:14:01, Tunnel100
                    [90/522240] via 192.168.100.11, 00:14:01, Tunnel100
```

```
D        10.1.0.0/16 [90/522240] via 192.168.200.12, 00:14:17, Tunnel200
                     [90/522240] via 192.168.100.11, 00:14:17, Tunnel100
D        10.2.0.0/16 [90/522240] via 192.168.200.22, 00:14:01, Tunnel200
                     [90/522240] via 192.168.100.21, 00:14:01, Tunnel100
C        10.3.0.31/32 is directly connected, Loopback0
C        10.3.3.0/24 is directly connected, GigabitEthernet1/0
H        10.4.4.0/24 [250/255] via 192.168.100.41, 00:00:03, Tunnel100
      192.168.100.0/24 is variably subnetted, 3 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
H        192.168.100.41/32 is directly connected, 00:00:03, Tunnel100
      192.168.200.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.200.0/24 is directly connected, Tunnel200
```

EIGRP Traffic Steering

It is important that each hub router identify its own DMVPN tunnel as the best path to reach any of the remote networks learned via the WAN. This allows the hub to advertise that path to other spoke routers.

The EIGRP path metric calculation (standard and wide) uses the path's minimum bandwidth and total delay. Here is the formula for EIGRP wide metrics with default K-value settings:

$$\text{METRIC} = 65{,}535 * ((10^7/\text{Min. Bandwidth}) + (\text{Total Delay}/10))$$

Modifying the EIGRP path attributes for total delay or minimum bandwidth provides a technique for traffic engineering with EIGRP. Lowering the bandwidth setting for an interface may impact QoS or other routing protocols; therefore, the best option for EIGRP traffic steering is to modify an interface's delay.

An interface's delay is modified with the interface parameter command **delay** *tens-of-microseconds*. Delay is verified with the command **show interface** *interface-id* and is displayed after the *DLY* field. Example 4-11 demonstrates how the delay for an interface can be verified.

Example 4-11 *Viewing Interface Delay Settings*

```
R13-DC1# show interfaces GigabitEthernet 0/1
! Output omitted for brevity
GigabitEthernet0/1 is administratively up, line protocol is up
  Hardware is AmdP2, address is aabb.cc00.6d10 (bia aabb.cc00.6d10)
  Description: R20
  Internet address is 10.10.20.10/24
  MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
```

On the DMVPN hub routers, adding a significant delay (24,000+) ensures that the routes learned via the DMVPN path are always identified as the best path. If a dedicated cross-link between DMVPN routers exists, more delay (+100, making 24,100) is added to the LAN-facing networks than to the cross-link to ensure that the cross-link is used appropriately. This concept is more applicable where the LAN-facing interfaces for both routers are connected via switches. Example 4-12 provides the configuration to ensure that the local DMVPN path is always identified as the best path.

> **Note** Most companies use a pair of MLSs (a switch Layer 3 routing) to provide connectivity to the WAN routers. A dedicated cross-link is not required between the DMVPN hub routers as shown in this book's topology. However, some network engineers find that using a dedicated cross-link simplifies troubleshooting. The MLS and DMVPN hub routers can participate in the same routing segment or provide connectivity from a straight L2 perspective (not using a switched virtual interface [SVI]). PfRv3 even allows connectivity across multiple L3 hops. Network engineers should not feel that they have to create a dedicated cross-link and do not need to purchase additional hardware.

Example 4-12 *Configuration to Ensure That the Local Tunnel Is Best Path on Hubs*

```
R11, R12, R21, R22
interface GigabitEthernet0/3
  description Cross-Link
  delay 24000
interface GigabitEthernet1/0
  description LAN Networks
  delay 24100
```

Example 4-13 verifies that R51 prefers DMVPN tunnel 100 for the 10.0.0.0/8, 10.1.0.0/16, and 10.2.0.0/16 networks and R52 prefers DMVPN tunnel 200 for the same networks.

Example 4-13 *Modifying Tunnel Metrics to Prefer MPLS over Internet*

```
R51-Spoke# show ip route
! Output omitted for brevity
Gateway of last resort is 192.168.100.11 to network 0.0.0.0
D*     0.0.0.0/0 [90/123909760] via 192.168.100.11, 00:01:02, Tunnel100
       10.0.0.0/8 is variably subnetted, 9 subnets, 4 masks
D         10.0.0.0/8 [90/522240] via 192.168.100.21, 00:01:02, Tunnel100
                     [90/522240] via 192.168.100.11, 00:01:02, Tunnel100
D         10.1.0.0/16 [90/522240] via 192.168.100.11, 00:01:02, Tunnel100
D         10.2.0.0/16 [90/522240] via 192.168.100.21, 00:01:02, Tunnel100
```

```
R52-Spoke# show ip route
! Output omitted for brevity
```

```
Gateway of last resort is 192.168.200.12 to network 0.0.0.0
D*    0.0.0.0/0 [90/123914880] via 192.168.200.12, 00:00:22, Tunnel200
      10.0.0.0/8 is variably subnetted, 9 subnets, 4 masks
D        10.0.0.0/8 [90/527360] via 192.168.200.22, 00:00:22, Tunnel200
                    [90/527360] via 192.168.200.12, 00:00:22, Tunnel200
D        10.1.0.0/16 [90/527360] via 192.168.200.12, 00:00:22, Tunnel200
D        10.2.0.0/16 [90/527360] via 192.168.200.22, 00:00:22, Tunnel200
```

The next step requires influencing the path selection while the router is in an uncontrolled PfR state. This is accomplished by adding a higher delay to the tunnel interfaces of the less preferred transport. On the DMVPN hub routers a delay of 1,000 is added to tunnel 100 and a delay of 2,000 to tunnel 200. On the spoke routers a delay of 1,000 is set on the primary path (tunnel 100) and a delay of 20,000 on all tertiary tunnel paths (tunnel 200) for the spoke routers.

Example 4-14 demonstrates the delay being added to the hub and branch routers.

Example 4-14 *Configuration to Direct Traffic in an Uncontrolled PfR State*

```
R11 and R21
interface Tunnel 100
  delay 1000
```

```
R12 and R22
interface Tunnel 200
  delay 2000
```

```
R31, R41, and R51
interface Tunnel 100
  delay 1000
```

```
R31, R41, and R52
interface Tunnel 200
  delay 20000
```

Just as delay is added to the LAN interfaces of the hub routers, delay should be added to the LAN interfaces of the branch routers. Realistically, the delay needs to be added only to sites with multiple IWAN routers, but it is added to all the LAN interfaces for consistency. The delay of the LAN interface should be set to match the secondary tunnel's delay of 20,000. If there is a dedicated cross-link network, the LAN network should have an elevated delay (+100). Example 4-15 demonstrates the additional delay being added to the branch site LAN interfaces.

Example 4-15 *Configuration to Direct Traffic in an Uncontrolled PfR State for Branch Routers*

```
R31 and R41
interface GigabitEthernet 1/0
  delay 20000
```

```
R51 and R52
interface GigabitEthernet 0/3
  delay 20000
interface GigabitEthernet 1/3
  delay 20100
```

Example 4-16 verifies that the MPLS transport is preferred for reaching the 10.0.0.0/8, 10.1.0.0/16, and 10.2.0.0/16 networks.

Example 4-16 *Modifying Tunnel Metrics to Prefer MPLS over Internet*

```
R31-Spoke# show ip route
! Output omitted for brevity
Gateway of last resort is 192.168.100.11 to network 0.0.0.0
D*    0.0.0.0/0 [90/ 129024640] via 192.168.100.11, 00:00:17, Tunnel100
      10.0.0.0/8 is variably subnetted, 6 subnets, 4 masks
D        10.0.0.0/8 [90/5637120] via 192.168.100.21, 00:00:17, Tunnel100
                    [90/5637120] via 192.168.100.11, 00:00:17, Tunnel100
D        10.1.0.0/16 [90/5637120] via 192.168.100.11, 00:00:17, Tunnel100
D        10.2.0.0/16 [90/5637120] via 192.168.100.21, 00:00:17, Tunnel100
```

Note Ensuring that the traffic is symmetric (uses the same transport in both directions) helps with application classification and WAAS. Multirouter sites such as Site 5 should use First-Hop Resiliency Protocols (FHRPs) such as Hot Standby Router Protocol (HSRP) which use the primary router that has the primary transport.

Complete EIGRP Configuration

This section explains EIGRP configuration in a step-by-step fashion to provide a thorough understanding of the configuration. Example 4-17 provides the complete configuration for the DMVPN hub routers.

Example 4-17 *EIGRP Hub Configuration*

R11-Hub

```
interface Tunnel100
 delay 1000
interface GigabitEthernet0/3
 description Site-Cross-link
 delay 24000
interface GigabitEthernet1/0
 description Site-LAN
 delay 24100
!
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  af-interface Tunnel100
   summary-address 10.0.0.0 255.0.0.0
   summary-address 10.1.0.0 255.255.0.0
   hello-interval 20
   hold-time 60
   no split-horizon
  exit-af-interface
  !
  topology base
   distribute-list prefix EIGRPSUMMARYROUTES in Tunnel100
   summary-metric 10.1.0.0/16 10000000 1 255 0 1500 distance 250
   summary-metric 10.0.0.0/8 10000000 1 255 0 1500 distance 250
  exit-af-topology
  network 10.1.0.0 0.0.255.255
  network 192.168.100.0
  eigrp router-id 10.1.0.11
 exit-address-family
!
ip prefix-list EIGRPSUMMARYROUTES seq 10 deny 0.0.0.0/0
ip prefix-list EIGRPSUMMARYROUTES seq 20 deny 10.0.0.0/8
ip prefix-list EIGRPSUMMARYROUTES seq 30 deny 10.1.0.0/16
ip prefix-list EIGRPSUMMARYROUTES seq 40 deny 10.2.0.0/16
ip prefix-list EIGRPSUMMARYROUTES seq 50 permit 0.0.0.0/0 le 32
```

R12-Hub

```
interface Tunnel200
 delay 2000
interface GigabitEthernet0/3
 description Site-Cross-link
 delay 24000
```

```
interface GigabitEthernet1/0
 description Site-LAN
 delay 24100
!
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  af-interface Tunnel200
   summary-address 10.0.0.0 255.0.0.0
   summary-address 10.1.0.0 255.255.0.0
   summary-address 10.2.0.0 255.255.0.0
   hello-interval 20
   hold-time 60
   no split-horizon
  exit-af-interface
  !
  topology base
   distribute-list prefix EIGRPSUMMARYROUTES in Tunnel200
   summary-metric 10.1.0.0/16 10000000 1 255 0 1500 distance 250
   summary-metric 10.0.0.0/8 10000000 1 255 0 1500 distance 250
  exit-af-topology
  network 10.1.0.0 0.0.255.255
  network 192.168.200.0
  eigrp router-id 10.1.0.12
 exit-address-family
!
ip prefix-list EIGRPSUMMARYROUTES seq 10 deny 0.0.0.0/0
ip prefix-list EIGRPSUMMARYROUTES seq 20 deny 10.0.0.0/8
ip prefix-list EIGRPSUMMARYROUTES seq 30 deny 10.1.0.0/16
ip prefix-list EIGRPSUMMARYROUTES seq 40 deny 10.2.0.0/16
ip prefix-list EIGRPSUMMARYROUTES seq 50 permit 0.0.0.0/0 le 32
```

R21-Hub

```
interface Tunnel100
 delay 1000
interface GigabitEthernet0/3
 description Site-Cross-link
 delay 24000
interface GigabitEthernet1/0
 description Site-LAN
 delay 24100
!
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  af-interface Tunnel100
```

```
   summary-address 10.0.0.0 255.0.0.0
   summary-address 10.2.0.0 255.255.0.0
   hello-interval 20
   hold-time 60
   no split-horizon
  exit-af-interface
  !
  topology base
   distribute-list prefix EIGRPSUMMARYROUTES in Tunnel100
   summary-metric 10.0.0.0/8 10000000 1 255 0 1500 distance 250
   summary-metric 10.2.0.0/16 10000000 1 255 0 1500 distance 250
  exit-af-topology
  network 10.2.0.0 0.0.255.255
  network 192.168.100.0
  eigrp router-id 10.2.0.21
 exit-address-family
!
ip prefix-list EIGRPSUMMARYROUTES seq 10 deny 0.0.0.0/0
ip prefix-list EIGRPSUMMARYROUTES seq 20 deny 10.0.0.0/8
ip prefix-list EIGRPSUMMARYROUTES seq 30 deny 10.1.0.0/16
ip prefix-list EIGRPSUMMARYROUTES seq 40 deny 10.2.0.0/16
ip prefix-list EIGRPSUMMARYROUTES seq 50 permit 0.0.0.0/0 le 32
```

R22-Hub

```
interface Tunnel200
 delay 2000
interface GigabitEthernet0/3
 description Site-Cross-link
 delay 24000
interface GigabitEthernet1/0
 description Site-LAN
 delay 24100
!
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  af-interface Tunnel200
   summary-address 10.0.0.0 255.0.0.0
   summary-address 10.2.0.0 255.255.0.0
   hello-interval 20
   hold-time 60
   no split-horizon
  exit-af-interface
  !
  topology base
```

```
   distribute-list prefix EIGRPSUMMARYROUTES in Tunnel200
   summary-metric 10.0.0.0/8 10000000 1 255 0 1500 distance 250
   summary-metric 10.2.0.0/16 10000000 1 255 0 1500 distance 250
  exit-af-topology
  network 10.2.0.0 0.0.255.255
  network 192.168.200.0
  eigrp router-id 10.2.0.22
exit-address-family
!
ip prefix-list EIGRPSUMMARYROUTES seq 10 deny 0.0.0.0/0
ip prefix-list EIGRPSUMMARYROUTES seq 20 deny 10.0.0.0/8
ip prefix-list EIGRPSUMMARYROUTES seq 30 deny 10.1.0.0/16
ip prefix-list EIGRPSUMMARYROUTES seq 40 deny 10.2.0.0/16
ip prefix-list EIGRPSUMMARYROUTES seq 50 permit 0.0.0.0/0 le 32
```

Example 4-18 provides the complete EIGRP configuration for the spoke routers.

Example 4-18 *EIGRP Spoke Configuration*

```
R31-Spoke
interface Tunnel100
 delay 1000
interface Tunnel200
 delay 20000
interface GigabitEthernet1/0
 description Site-LAN
 delay 20100
!
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  af-interface Tunnel100
   hello-interval 20
   hold-time 60
   stub-site wan-interface
  exit-af-interface
  af-interface Tunnel200
   hello-interval 20
   hold-time 60
   stub-site wan-interface
  exit-af-interface
  !
 network 10.0.0.0
  network 192.168.100.0
  network 192.168.200.0
  eigrp router-id 10.3.0.31
  eigrp stub-site 1:1
 exit-address-family
```

R41-Spoke

```
interface Tunnel100
 delay 1000
interface Tunnel200
 delay 20000
interface GigabitEthernet1/0
 description Site-LAN
 delay 20000
!
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  af-interface Tunnel100
   hello-interval 20
   hold-time 60
   stub-site wan-interface
  exit-af-interface
  af-interface Tunnel200
   hello-interval 20
   hold-time 60
   stub-site wan-interface
  exit-af-interface
  !
  network 10.0.0.0
  network 192.168.100.0
  network 192.168.200.0
  eigrp router-id 10.4.0.41
  eigrp stub-site 1:1
 exit-address-family
```

R51-Spoke

```
interface Tunnel100
 delay 1000
interface GigabitEthernet0/3
 description Site-Cross-link
 delay 20000
interface GigabitEthernet1/0
```

```
 description Site-LAN
 delay 20100
!
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  af-interface Tunnel100
   hello-interval 20
   hold-time 60
   stub-site wan-interface
  exit-af-interface
  !
  network 10.0.0.0
  network 192.168.100.0
  eigrp router-id 10.5.0.51
  eigrp stub-site 1:1
 exit-address-family
```

```
R52-Spoke
interface Tunnel200
 delay 20000
interface GigabitEthernet0/3
 description Site-Cross-link
 delay 20000
interface GigabitEthernet1/0
 description Site-LAN
 delay 20100
!
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  af-interface Tunnel200
   hello-interval 20
   hold-time 60
   stub-site wan-interface
  exit-af-interface
  !
  network 10.0.0.0
  network 192.168.200.0
  eigrp router-id 10.5.0.52
  eigrp stub-site 1:1
 exit-address-family
```

Advanced EIGRP Site Selection

In some scenarios, network engineers want to direct traffic to a preferred hub site. For example, assume that a company uses a centralized Internet access model. Internet connectivity is available only at Site 1, which is in Los Angeles, and at Site 2, which is located in London. It makes sense to direct the U.S. remote sites to Site 1 for Internet access and the European remote sites to Site 2.

The simplest method of accomplishing this task is to tag routes on the hub routers and influence the path metrics to prefer one site over another on the spoke routers. An outbound distribute list is placed on the hub router's DMVPN tunnel interfaces with the command **distribute-list route-map** *route-map-name* **out tunnel** *tunnel-number*. The **distribute-list route-map** contains only one **permit** sequence that sets a unique tag for that hub router.

Example 4-19 demonstrates the unique tagging of routes on the hub routers.

Example 4-19 *Configuration to Set EIGRP Tags on Advertised Routes*

```
R11-Hub
route-map EIGRP-TAG-ROUTE-HUB-ID permit 10
 description Tag all routes advertised from this hub
 set tag 11
!
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  topology base
   distribute-list route-map EIGRP-TAG-ROUTE-HUB-ID out Tunnel 100
```

```
R12-Hub
route-map EIGRP-TAG-ROUTE-HUB-ID permit 10
 description Tag all routes advertised from this hub
 set tag 12
!
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  topology base
   distribute-list route-map EIGRP-TAG-ROUTE-HUB-ID out Tunnel 200
```

```
R21-Hub
route-map EIGRP-TAG-ROUTE-HUB-ID permit 10
 description Tag all routes advertised from this hub
 set tag 21
!
```

```
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  topology base
   distribute-list route-map EIGRP-TAG-ROUTE-HUB-ID out Tunnel 100
```

```
R22-Hub
route-map EIGRP-TAG-ROUTE-HUB-ID permit 10
 description Tag all routes advertised from this hub
 set tag 22
!
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  topology base
   distribute-list route-map EIGRP-TAG-ROUTE-HUB-ID out Tunnel 200
```

The verification of the route tag is performed by viewing a network's entry in the EIGRP topology table with the command **show ip eigrp topology** *network/prefix-length*. Example 4-20 verifies that R11 (192.168.100.11) set the tag of 11, and that R21 (192.168.100.21) set the tag of 21. Notice that R31 identifies both paths via R11 and R21 as the best paths.

Example 4-20 *Verification of EIGRP Route Tagging*

```
R31-Spoke# show ip eigrp topology 0.0.0.0/0
! Output omitted for brevity
EIGRP-IPv4 VR(IWAN) Topology Entry for AS(1)/ID(10.3.0.31) for 0.0.0.0/0
  Descriptor Blocks:
  192.168.100.11 (Tunnel100), from 192.168.100.11, Send flag is 0x0
      Composite metric is (17476348586/16384737280), route is Internal
      Vector metric:
        Internal tag is 11
      Extended Community: StubSite:1:3
  192.168.100.21 (Tunnel100), from 192.168.100.21, Send flag is 0x0
      Composite metric is (17476348586/16449617920), route is Internal
      Vector metric:
        Internal tag is 21
      Extended Community: StubSite:1:3
```

```
R31-Spoke# show ip route
! Output omitted for brevity
Gateway of last resort is 192.168.100.21 to network 0.0.0.0
D*    0.0.0.0/0 [90/136533973] via 192.168.100.21, 00:13:18, Tunnel100
                [90/136533973] via 192.168.100.11, 00:13:18, Tunnel100
```

In order to set a preferred site, the metric must be increased on all routes (paths) that do not have the desired tag set. The spoke routers use an inbound distribute list with a route map for every DMVPN tunnel interface. The route map contains two sequences:

1. Match routes that contain the tag of the hub routers in the preferred site.
2. Add a metric to all the other routes learned from all other hub routers.

Example 4-21 provides the configuration for preferring routes in Site 1 or Site 2. Notice the plus sign (+) with the **set metric** command. It is required to increase the metric.

Example 4-21 *EIGRP Configuration for Hub Preference*

```
Spoke Prefers Site1 for Exit
route-map EIGRP-PREFER-SITE1 permit 10
 description Do not modify metric on routes from Site 1 with Tag 11 or 12
 match tag 11 12
route-map EIGRP-PREFER-SITE1 permit 20
 description Increase metric on all other non-matching routes
 set metric +100000000
!
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  topology base
   distribute-list route-map EIGRP-PREFER-SITE1 in Tunnel 100
   distribute-list route-map EIGRP-PREFER-SITE1 in Tunnel 200
```

```
Spoke Prefers Site2 for Exit
route-map EIGRP-PREFER-SITE1 permit 10
 description Do not modify metric on routes from Site 2 with Tag 21 or 22
 match tag 21 22
route-map EIGRP-PREFER-SITE1 permit 20
 description Increase metric on all other non-matching routes
 set metric +100000000
!
router eigrp IWAN
 address-family ipv4 unicast autonomous-system 1
  topology base
   distribute-list route-map EIGRP-PREFER-SITE2 in Tunnel 100
   distribute-list route-map EIGRP-PREFER-SITE2 in Tunnel 200
```

The configuration for preferring Site 1 for all routes was applied to R31. Example 4-22 verifies the effectiveness of the change. R31 installs only the path from R11 as the best path for the default route. Upon examining the EIGRP topology, the delay has been increased on the path from Site 2, making the path from Site 1 the preferred path.

Example 4-22 *Verification of Path Preference*

```
R31-Spoke# show ip route
! Output omitted for brevity
Gateway of last resort is 192.168.100.11 to network 0.0.0.0

D*    0.0.0.0/0 [90/136533973] via 192.168.100.11, 00:00:19, Tunnel100
      10.0.0.0/8 is variably subnetted, 6 subnets, 4 masks
D        10.0.0.0/8 [90/8538453] via 192.168.100.11, 00:00:19, Tunnel100
```

```
R31-Spoke# show ip eigrp topology 0.0.0.0/0
! Output omitted for brevity
EIGRP-IPv4 VR(IWAN) Topology Entry for AS(1)/ID(10.3.0.31) for 0.0.0.0/0
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 17476348586, RIB
    is 136533973
  Descriptor Blocks:
  192.168.100.11 (Tunnel100), from 192.168.100.11, Send flag is 0x0
      Composite metric is (17476348586/16384737280), route is Internal
      Vector metric:
        Minimum bandwidth is 1500 Kbit
        Total delay is 260001250000 picoseconds
        Internal tag is 11
      Extended Community: StubSite:1:3
  192.168.100.21 (Tunnel100), from 192.168.100.21, Send flag is 0x0
      Composite metric is (17576348586/16449617920), route is Internal
      Vector metric:
        Minimum bandwidth is 1500 Kbit
        Total delay is 261527128907 picoseconds
        Internal tag is 21
      Extended Community: StubSite:1:3
```

Note The path metric is calculated based on the path's minimum bandwidth path attribute and total delay. Interface delay cannot be set dynamically on a per-route basis. The use of a metric increase/offset list implicitly adds delay to the total path delay. The increase of a metric does not provide the same linear curve when bandwidth changes on interfaces. When the EIGRP metric is modified, the secondary site's primary transport is still used as the backup in the event of failure for non-Pfr-controlled traffic.

IOS XE 3.16.1 adds capabilities in PfR that allow the path selection to occur in the following order:

1. Primary site—primary transport
2. Primary site—secondary transport
3. Secondary site—primary transport
4. Secondary site—secondary transport.

Border Gateway Protocol (BGP)

BGP establishes sessions with other BGP routers. If the session is formed with a BGP router within the same AS, it is known as an IBGP (internal BGP) session. If the session is formed with a BGP router from a different AS, it is known as an EBGP (external BGP) session. The best-path calculations and behaviors in EBGP and IBGP sessions are slightly different.

BGP attaches *path attributes (PAs)* associated with each network path. The PAs provide BGP with granularity and control of routing policies. BGP communities provide additional capability for tagging routes and for modifying BGP routing policy on upstream and downstream routers. *BGP communities* can be appended, removed, or modified selectively on each attribute as the route travels from router to router. BGP communities are an optional transitive BGP attribute that can traverse from AS to AS and can be used to simplify BGP routing policy.

In the prescriptive IWAN design, IBGP is used instead of an EBGP session for the following reasons:

- BGP dynamic peers work only with an IBGP session that provides a zero-touch hub configuration for routing protocols. EBGP sessions require an explicit configuration for every branch router.
- IBGP provides built-in stub functions at branch sites, because IBGP peers do not advertise routes learned from an IBGP peer to a different IBGP peer. This behavior provides a default method of filtering and prevents transit routing.
- IBGP provides a centralized routing policy through BGP local preference which is not a transitive BGP community and requires an IBGP session. Changes can be made at the hub routers (small number of devices) without having to configure the spoke devices (large number of routers). PfR understands and can use local preference as well.

BGP Routing Logic

Running multiple routing protocols on a network is common. Using a proper design can simplify the operational aspects and prevent issues that can arise from routing loops. Figure 4-7 displays the logic for this book's BGP/OSPF design where BGP is used as the WAN transport for an OSPF network. The design uses the following logic:

- The hub routers are BGP route reflectors, and the spoke routers are route reflector clients.
- Hub routers redistribute BGP into OSPF that is running at the centralized sites.
- Within the centralized site a default route is advertised to provide connectivity for Internet traffic. The default route is advertised into BGP and then advertised into the spoke routers at the branch site.
- Hub routers advertise summary routes for enterprise prefixes (all DCs, campus, and remote office site routes) to the spoke routers. In addition, hub routers also advertise more specific DC summaries for the networks within their site, to simplify direct routing from branches.

- Hub routers should advertise networks only if they maintain connectivity to their LAN networks. This is to ensure that they are healthy and do not blackhole network traffic.
- Two options exist for branch connectivity. Smaller sites that have only directly attached LANs (one IWAN router) redistribute "connected" routes (tunnel networks excluded) into BGP. If multiple routers exist at a branch site, the spoke or spokes announce the default route into the IGP for Internet connectivity and mutually redistribute the IGP into BGP using a *tag-and-block* method to prevent routing loops.

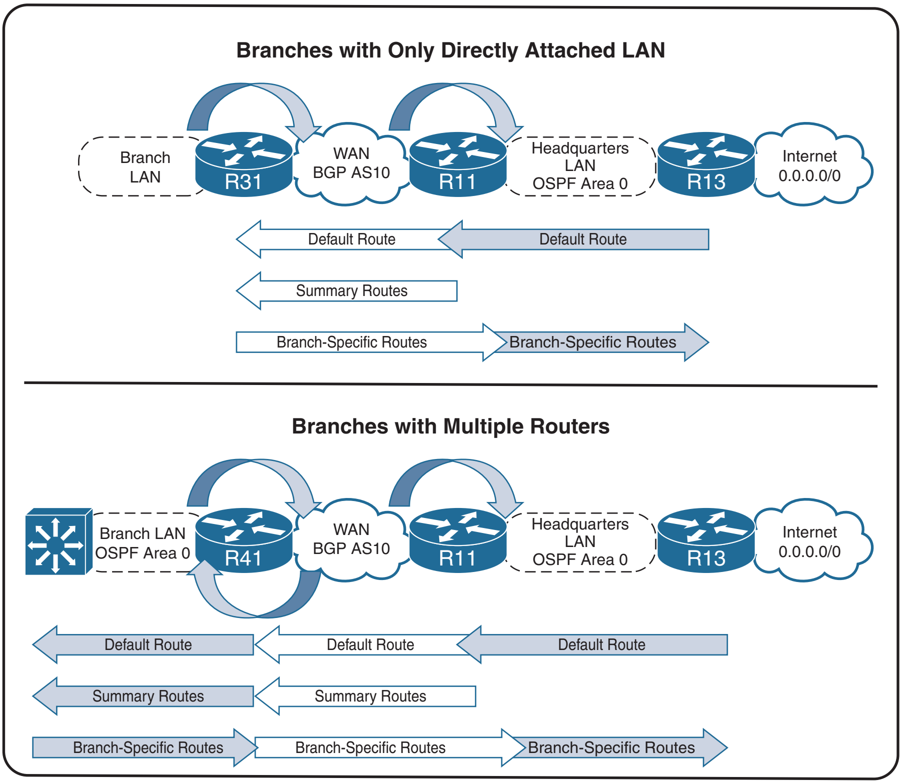


Figure 4-7 *Routing Logic for BGP WANs with OSPF LANs*

This chapter explains how BGP (IBGP specifically) can be used as the WAN routing protocol while using a different IGP (that is, OSPF) for the LAN. The routing protocol configuration requires the configuration of the following components:

- Base IBGP session configuration and verification of neighbors
- Default route advertisement
- Branch router advertisement (single-router sites and multiple-router branch sites)

- Changing of BGP AD
- Advertisement of routes at the hub routers
- Filtering of routes received and advertised at the hub routers
- Redistribution of BGP IWAN routes into the headquarters IGP
- Traffic steering

After the last section, the complete BGP configuration is provided for all the routers.

Base Configuration

When configuring the base IWAN BGP configuration, it is best to think of it from a modular perspective. BGP router configuration requires the following components:

- **BGP session parameters:** BGP session parameters provide settings that involve establishing communication to the remote BGP neighbor. Session settings include the ASN of the BGP peer, authentication, and keepalive timers.
- **Address family initialization and configuration:** The address family is initialized under the BGP router configuration mode. Network advertisement and summarization occur within the address family.
- **Activation of the address family on the BGP peer:** In order for a session to initiate, one address family for that neighbor must be activated. The router's IP address is added to the neighbor table, and BGP attempts to establish a BGP session or accepts a BGP session initiated from the peer router.

BGP Neighbor Sessions

Following are the steps for configuring BGP on a router. IOS activates the IPv4 address family by default. This can simplify the configuration in an IPv4 environment because Steps 6 and 7 are optional, but it may cause confusion when working with other address families.

Step 1. **Create the BGP routing process.**

Initialize the BGP process with the global command **router bgp** *as-number*.

Step 2. **Set the router ID.**

The RID is a 32-bit number that uniquely identifies a BGP router. Cisco IOS routers use the highest IP address of any up loopback interfaces. If there are no up loopback interfaces, the highest IPv4 address of any active up physical interfaces becomes the RID when the BGP initializes. It is considered a best practice to statically configure the router ID. In this book, the router ID matches the loopback interface.

The command **bgp router-id** *router-id* is used to set the RID.

Step 3. **Create a BGP peer group for BGP routers.**

Peer groups are a method of reducing BGP configuration by creation of a template configuration and then associating a router to the peer group (that is, the template).

The peer group is defined with the command **neighbor** *group-name* **peer-group.** All BGP parameters are configured using the peer group *group-name* in lieu of the neighbor's *ip-address*. BGP peer IP addresses are linked to the peer group with the command **neighbor** *ip-address* **peer-group** *group-name*.

Step 4. **Identify the BGP neighbor's IP address and ASN.**

Identify the BGP neighbor's IP address and ASN with the BGP router configuration command **neighbor** *ip-address* **remote-as** *as-number.*

BGP neighbors do not dynamically discover each other as with EIGRP, OSPF, or IS-IS. In order for DMVPN to support a zero-touch deployment, the hub routers do not require additional configuration when deploying more spoke routers. BGP provides a feature, *dynamic peers*, for peer addresses to be defined by a range of IP addresses and linked to a peer group. The command **bgp listen range** *network/length* **peer-group** *group-name* should be run only on the DMVPN hub routers with the IP range of the DMVPN tunnel of the client routers.

Step 5. **Configure BGP timers.**

BGP relies upon a stable network topology because of the size of the routing table. The default hold timer requires that a BGP update or keepalive packet be received every 180 seconds to maintain the BGP session. By default BGP sends a keepalive every 60 seconds to a BGP neighbor (allowing for three retries).

Three minutes is a long time span for convergence and is too slow for non-PfR-controlled network traffic. The BGP keepalive timer has been lowered to 20 seconds and a hold timer to 60 seconds to accommodate faster convergence of non-PfR-controlled network traffic. Lowering the timers impacts scalability within the topology to fewer than 1000 routers. Larger environments should change and test the numbers accordingly.

The command **neighbor** *ip-address* **timers** *keepalive holdtime* configures the neighbor session timers.

Step 6. **Initialize the address family.**

Initialize the address family with the BGP router configuration command **address-family** *address-family address-family-modifier*.

Step 7. **Activate the address family for the BGP neighbor.**

Activate the BGP neighbor for that address family with the command **neighbor** *ip-address* **activate.** Only BGP dynamic peers can be activated by peer group name. All others must be activated by the IP address.

Step 8. **Configure route reflectors on hub routers.**

The command **neighbor** *ip-address* **route-reflector-client** is used on IOS nodes.

Step 9. **Configure inbound soft reconfiguration (optional).**

BGP maintains three different tables to store routes. The first table, Adj-RIB-In, maintains only the raw unedited routes that were received from the neighbors and is purged after route policies are processed. Inbound soft reconfiguration is needed to view the raw unedited route after route policy processing, which is helpful in troubleshooting.

Enabling this feature can consume a significant amount of memory because the Adj-RIB-In table stays in memory. Inbound soft reconfiguration uses the command **neighbor** *ip-address* **soft-reconfiguration inbound.**

Step 10. **Enable BGP communities support (optional).**

IOS does not advertise BGP communities to peers by default. Communities are enabled on a neighbor-by-neighbor basis with the BGP address family configuration command **neighbor** *ip-address* **send-community** [**standard** | **extended** | **both**]. Standard communities are sent by default, unless the optional **extended** or **both** keyword is used.

Example 4-23 provides the basic BGP configuration for the DMVPN hub routers. The hub routers use BGP dynamic peer configuration for the spoke routers and associate to a peer group to simplify the configuration.

Note The BGP communities and inbound soft reconfiguration are not necessary components of the solution. However, adding them now prevents the necessary session reset if they are added at a later time.

The OSPF configuration for the hub routers is provided as part of this step. OSPF has been enabled on all interfaces, so that routers in the centralized site can have reachability information for the DMVPN tunnel networks (192.168.100.0/24 and 192.168.200.0/24). The **passive-interface default** command is used to prevent unwanted OSPF neighbor adjacencies from forming across the DMVPN tunnels. The LAN network interfaces have been made active with the **no passive-interface** *interface-id* command.

Example 4-23 *Base OPSF and BGP Configuration for DMVPN Hub Routers*

```
R11 and R21
router ospf 1
 passive-interface default
 no passive-interface GigabitEthernet0/3
 no passive-interface GigabitEthernet1/0
 network 0.0.0.0 255.255.255.255 area 0
!
router bgp 10
 bgp listen range 192.168.100.0/24 peer-group MPLS-SPOKES
 neighbor MPLS-SPOKES peer-group
 neighbor MPLS-SPOKES remote-as 10
 neighbor MPLS-SPOKES timers 20 60
 !
 address-family ipv4
  neighbor MPLS-SPOKES activate
  neighbor MPLS-SPOKES route-reflector-client
  neighbor MPLS-SPOKES send-community
  neighbor MPLS-SPOKES soft-reconfiguration inbound
```

```
R12 and R22
router ospf 1
 passive-interface default
 no passive-interface GigabitEthernet0/3
 no passive-interface GigabitEthernet1/0
 network 0.0.0.0 255.255.255.255 area 0
!
router bgp 10
 bgp listen range 192.168.200.0/24 peer-group INET-SPOKES
 neighbor INET-SPOKES peer-group
 neighbor INET-SPOKES remote-as 10
 neighbor INET-SPOKES timers 20 60
 !
 address-family ipv4
  neighbor INET-SPOKES activate
  neighbor INET-SPOKES route-reflector-client
  neighbor INET-SPOKES send-community
  neighbor INET-SPOKES soft-reconfiguration inbound
```

Example 4-24 provides the basic BGP configuration for the DMVPN spoke routers. Peer groups are used to reduce the configuration. Notice that the *peer-group-name* is used in lieu of specifying a neighbor's IP address.

Example 4-24 *BGP Configuration for DMVPN Spoke Routers*

```
R31, R41 and R51
router bgp 10
 neighbor MPLS-HUB peer-group
 neighbor MPLS-HUB remote-as 10
 neighbor MPLS-HUB timers 20 60
 neighbor 192.168.100.11 peer-group MPLS-HUB
 neighbor 192.168.100.21 peer-group MPLS-HUB
!
 address-family ipv4
  neighbor MPLS-HUB soft-reconfiguration inbound
  neighbor MPLS-HUB send-community
  neighbor 192.168.100.11 activate
  neighbor 192.168.100.21 activate
```

```
R31, R41 and R52
router bgp 10
 neighbor INET-HUB peer-group
 neighbor INET-HUB remote-as 10
 neighbor INET-HUB timers 20 60
 neighbor 192.168.200.12 peer-group INET-HUB
 neighbor 192.168.200.22 peer-group INET-HUB
 !
 address-family ipv4
  neighbor INET-HUB soft-reconfiguration inbound
  neighbor INET-HUB send-community
  neighbor 192.168.200.12 activate
  neighbor 192.168.200.22 activate
```

Note The configurations in Example 4-24 do not include the BGP or OSPF router IDs to save space. They should be configured as part of best-practice standards.

BGP maintains a table of neighbors to track current status. The command **show bgp** *address-family address-family-modifier* displays the neighbor state, messages sent and received, input and output queues, prefixes received, and count of routes in the BGP table.

Example 4-25 provides verification that R11 and R12 have established BGP sessions with the DMVPN spoke routers. This indicates that *State/PfxRcd* has a numerical value. All the BGP neighbors on the DMVPN hub routers are established dynamically.

Example 4-25 *BGP Neighbor Verification from DMVPN Hubs R11 and R12*

```
R11-Hub# show bgp ipv4 unicast summary
BGP router identifier 10.1.0.11, local AS number 10
BGP table version is 1, main routing table version 1

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
*192.168.100.31 4    10       8       7        1    0    0 00:01:52        0
*192.168.100.41 4    10       6       5        1    0    0 00:01:10        0
*192.168.100.51 4    10      22      22        1    0    0 00:05:40        0
* Dynamically created based on a listen range command
Dynamically created neighbors: 3, Subnet ranges: 1

BGP peergroup MPLS-SPOKES listen range group members:
  192.168.100.0/24

Total dynamically created neighbors: 3/(100 max), Subnet ranges: 1
```

```
R12-Hub# show bgp ipv4 unicast summary
! Output omitted for brevity
Neighbor        V   AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
*192.168.200.31 4   10      11      10        1    0    0 00:02:52        0
*192.168.200.41 4   10       9       9        1    0    0 00:02:19        0
*192.168.200.52 4   10      25      24        1    0    0 00:06:28        0
* Dynamically created based on a listen range command
Dynamically created neighbors: 3, Subnet ranges: 1

BGP peergroup INET-SPOKES listen range group members:
  192.168.200.0/24

Total dynamically created neighbors: 3/(100 max), Subnet ranges: 1
```

Note The default number of dynamic peers is 100 as shown in Example 4-25. This value should be changed to match the maximum number of devices that the DMVPN tunnel subnet can support. The limit is changed with the command **bgp listen limit** *1–2000* to match the size of the DMVPN network.

Example 4-26 verifies that R31 has established connectivity with all four DMVPN hub routers: two for MPLS and two for Internet.

Example 4-26 *BGP Neighbor Verification from DMVPN Spoke R31*

```
R31-Spoke# show bgp ipv4 unicast summary
BGP router identifier 10.3.0.31, local AS number 10
BGP table version is 1, main routing table version 1

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.100.11  4    10      12      13       1     0    0 00:03:27         0
192.168.100.21  4    10      12      13       1     0    0 00:03:29         0
192.168.200.12  4    10      12      13       1     0    0 00:03:23         0
192.168.200.22  4    10      12      13       1     0    0 00:03:22         0
```

Default Route Advertisement into BGP

R13 advertises the default route into OSPF to provide Internet connectivity. Example 4-27 provides verification that R11 and R21 contain the default route from R13.

Example 4-27 *Verification of R13's Default Route for Internet Connectivity*

```
R11-Hub# show ip route
! Output omitted for brevity
O*E2  0.0.0.0/0 [110/1] via 10.1.111.13, 00:37:40, GigabitEthernet1/0
      10.0.0.0/8 is variably subnetted, 21 subnets, 2 masks
```

```
R21-Hub# show ip route
! Output omitted for brevity
O*E2  0.0.0.0/0 [210/1] via 10.2.221.23, 00:38:00, GigabitEthernet1/0
      10.0.0.0/8 is variably subnetted, 22 subnets, 2 masks
```

```
R11-Hub# show ip ospf database external
! Output omitted for brevity
            OSPF Router with ID (10.1.0.22) (Process ID 1)

                Type-5 AS External Link States
  LS Type: AS External Link
  Link State ID: 0.0.0.0 (External Network Number)
  Advertising Router: 10.1.0.13
  Network Mask: /0
        Metric Type: 2 (Larger than any link state path)
        Metric: 1
        Forward Address: 0.0.0.0
```

The hub routers must be configured to advertise the default route into BGP so that the spoke routers have connectivity to the Internet via R13.

The command **network** *network* **mask** *subnet-mask* [**route-map** *route-map-name*] is used to advertise IPv4 networks into BGP. The optional **route-map** parameter provides a method to set specific BGP PAs when the prefix installs into the BGP table.

Example 4-28 provides R11's configuration for advertising the default route into BGP. The advertisement was received on R31, but it could not find a best path as indicated by >. Upon further examination, the next-hop IP address 10.1.111.10 is not accessible and is the reason the best path has not been identified.

Example 4-28 *Viewing R31's Default Route Information*

```
R11
router bgp 10
 address-family ipv4
  network 0.0.0.0
```

```
R31-Spoke# show bgp ipv4 unicast
! Output omitted for brevity
     Network          Next Hop            Metric LocPrf Weight Path
 * i 0.0.0.0          10.1.111.10              1    100      0 i
```

```
R31-Spoke# show bgp ipv4 unicast 0.0.0.0
BGP routing table entry for 0.0.0.0/0, version 0
Paths: (1 available, no best path)
  Not advertised to any peer
  Refresh Epoch 1
  Local
    10.1.111.13 (inaccessible) from 192.168.100.11 (10.1.0.11)
      Origin IGP, metric 1, localpref 100, valid, internal
```

Routes learned from IBGP peers do not change the next-hop IP address by default. Because the route was learned by OSPF on R11, the next-hop BGP attribute is set to the next-hop IP address in the routing table. Also, the BGP origin is set to *internal*, the weight is set to 32,768 on the originating routers, and the BGP multi-exit discriminator (MED) is set to the IGP metric.

The BGP next-hop IP address should be modified to ensure that it passes the next-hop address check on all the receiving routers. The next-hop IP address can be modified for all routes as they are advertised to the hub router with the BGP configuration command **neighbor** *ip-address* **next-hop-self** [**all**]. The **next-hop-self** feature does not modify the next-hop address for IBGP prefixes by default. Appending the optional **all** keyword modifies the next-hop address on IBGP prefixes too. This ensures that the next-hop IP address is reachable from any of the routers.

Example 4-29 provides the proper configuration for advertising the default route into BGP on the hub routers.

Example 4-29 *Configuration for Advertising the Default Route with Accessible Next Hop*

```
R11 and R21
router bgp 10
 address-family ipv4
  network 0.0.0.0
  neighbor MPLS-SPOKES next-hop-self all
```

```
R12 and R22
router bgp 10
 address-family ipv4
  network 0.0.0.0
  neighbor INET-SPOKES next-hop-self all
```

Example 4-30 provides verification that the hub routers set the next-hop IP address to their tunnel IP address, which is reachable by the spoke routers.

Example 4-30 *Verification of Reachable Next Hop for the Default Route*

```
R11-Hub# show bgp ipv4 unicast
! Output omitted for brevity
     Network          Next Hop            Metric LocPrf Weight Path
 *>  0.0.0.0          192.168.100.11           1         32768 i
```

```
R12-Hub# show bgp ipv4 unicast
! Output omitted for brevity
     Network          Next Hop            Metric LocPrf Weight Path
 *>  0.0.0.0          192.168.200.12           1         32768 i
```

```
R31-Spoke# show bgp ipv4 unicast
! Output omitted for brevity
     Network          Next Hop            Metric LocPrf Weight Path
 *>i 0.0.0.0          192.168.100.11           1    100      0 i
 * i                  192.168.200.12           1    100      0 i
 * i                  192.168.100.21           1    100      0 i
 * i                  192.168.200.22           1    100      0 i
```

Routes Learned via DMVPN Tunnel Are Always Preferred

It is vital for the DMVPN hub router to prefer the direct DMVPN next-hop interfaces over transit DC LAN interfaces for routes learned from spoke routers. When a BGP router advertises or redistributes routes, it sets the weight to 32,768. Weight is the first path attribute compared when calculating the best path. There are certain race conditions that can occur with BGP and redistribution that are easily resolved by setting the weight to a value higher than 32,768. When the weight is set to 50,000 for all BGP peers, the possibility of a race condition is removed, and routes learned via a DMVPN tunnel will always be preferred.

Example 4-31 provides the configuration for setting the BGP weight on the DMVPN routers.

Example 4-31 *BGP Configuration to Set the Weight on Hub and Spoke Peers*

```
R11 and R21
router bgp 10
 address-family ipv4
  neighbor MPLS-SPOKES weight 50000
```

```
R12 and R22
router bgp 10
 address-family ipv4
  neighbor INET-SPOKES weight 50000
```

```
R31, R41 and R51
router bgp 10
 address-family ipv4
  neighbor MPLS-HUB weight 50000
```

```
R31, R41 and R52
router bgp 10
 address-family ipv4
  neighbor INET-HUB weight 50000
```

Example 4-32 verifies that the BGP weight has changed from the remote site routers after a soft reset was performed.

Example 4-32 *Verification of the Change in BGP Weight*

```
R31-Spoke# show bgp ipv4 unicast
! Output omitted for brevity
     Network          Next Hop            Metric LocPrf Weight Path
 * i 0.0.0.0          192.168.200.22           1    100  50000 i
 * i                  192.168.100.21           1    100  50000 i
 * i                  192.168.200.12           1    100  50000 i
 *>i                  192.168.100.11           1    100  50000 i
```

```
R41-Spoke# show bgp ipv4 unicast
     Network          Next Hop            Metric LocPrf Weight Path
 * i 0.0.0.0          192.168.200.22           1    100  50000 i
 * i                  192.168.100.21           1    100  50000 i
 * i                  192.168.200.12           1    100  50000 i
 *>i                  192.168.100.11           1    100  50000 i
```

Branch Router Configuration

All the remote site LAN networks need to be advertised into BGP. Network prefixes can be statically advertised with the **network** command, but dynamic redistribution from the connected interface database or from an IGP (such as the OSPF database) is more scalable.

The configuration for a branch router depends on the number of routers that are located in a branch site. A primary goal for branch routers is to provide a consistent and templatable configuration that can be deployed with network management tools. Therefore, there are two basic configurations: single-router sites and multirouter sites.

If there is only one router at a site, only the connected interfaces need to be redistributed into the BGP WAN protocol. If there are multiple routers, the default route should be advertised into the LAN IGP (OSPF). Then BGP and IGP (OSPF) should be mutually redistributed with controlled filtering. A route map is used at each redistribution point to prevent routing loops using a tag-and-block method.

The redistributed routes populate the next-hop IP address with the IP address from the advertising router. The next-hop IP address should be modified to ensure that it passes the next-hop address check on all the receiving routers. The next-hop IP address can be modified for all routes as they are advertised to the hub router with the BGP configuration command **neighbor** *ip-address* **next-hop-self** [**all**].

Note The **next-hop-self** feature does not modify the next-hop address for IBGP prefixes by default. Appending the optional **all** keyword modifies the next-hop address on IBGP prefixes too.

Single-Router Branch Sites

As stated before, single-router sites receive all the routes from DMVPN hubs and do not need to run a routing protocol in the branch network. Single-site routers only redistribute the connected LAN routes into BGP. It is important to note that the DMVPN tunnel networks should not be redistributed into BGP. Redistribution is accomplished by using a simple route map with the command **redistribute connected route-map** *route-map-name*.

Example 4-33 provides the configuration for single-router sites (Site 3) that use BGP.

Example 4-33 *Configuration for Route Advertisement at Single-Router Sites*

```
R31
router bgp 10
 address-family ipv4
  redistribute connected route-map REDIST-CONNECTED-TO-BGP
  neighbor MPLS-HUB next-hop-self all
  neighbor INET-HUB next-hop-self all
!
```

```
route-map REDIST-CONNECTED-TO-BGP deny 10
 description Block redistribution of DMVPN Tunnel Interfaces
 match interface Tunnel100 Tunnel200
route-map REDIST-CONNECTED-TO-BGP permit 20
 description Redistribute all other prefixes
```

Example 4-34 verifies that R31 has successfully advertised its networks into BGP. The locally advertised routes have a weight of 32,768, whereas the routes learned from the DMVPN hub routers have a weight of 50,000. Notice that the DMVPN networks (192.168.100.0/24 and 192.168.200.0/24) are not redistributed into BGP.

Example 4-34 *Verification of Route Advertisements into BGP*

```
R31-Spoke# show bgp ipv4 unicast
! Output omitted for brevity
     Network          Next Hop            Metric LocPrf Weight Path
 * i 0.0.0.0          192.168.200.22           1    100  50000 i
 * i                  192.168.100.21           1    100  50000 i
 * i                  192.168.200.12           1    100  50000 i
 *>i                  192.168.100.11           1    100  50000 i
 *>  10.3.0.31/32     0.0.0.0                  0         32768 ?
 *>  10.3.3.0/24      0.0.0.0                  0         32768 ?
 *>  192.168.100.0    0.0.0.0                  0         32768 ?
 *>  192.168.200.0    0.0.0.0                  0         32768 ?
```

Multiple-Router Branch Sites

For sites with multiple routers, the configuration accommodates sites that have only downstream routers (Site 4) or multiple IWAN routers (Site 5). The multirouter site routing configuration uses mutual redistribution between the IGP and BGP so that it can accommodate both scenarios. Multisite routing design includes this logic:

- Establish an IGP routing protocol for the LAN networks such as OSPF for the book's sample scenario. The IGP should not be enabled on or advertise the DMVPN tunnel networks at the branch sites.
- The BGP learned default route is advertised into the IGP routing protocol. This provides a default route for Internet traffic for any routers at the branch site.
- Redistribute BGP routes into the IGP. In essence, only the BGP summary routes are redistributed into the IGP. During this process the routes are tagged as the first step of a loop prevention process. The command **bgp redistribute-internal** is required to redistribute the IBGP learned network prefixes into the IGP.
- Selectively redistribute IGP routes into BGP. Any route in the IGP that was not tagged from an earlier step (therefore indicating that the route originated in the IGP) is redistributed into BGP.

Note The DMVPN tunnel networks should not be redistributed into BGP at the branch routers.

There are two things to consider when using OSPF as the IGP and how it interacts with BGP:

- OSPF does not redistribute OSPF external routes into BGP by default and requires the explicit route identification for this to happen. External routes can be selected in a route map or by adding them to the **redistribute** command under the BGP protocol.
- The router must have a default route in the routing table to inject the 0.0.0.0/0 link-state advertisement (LSA) into the OSPF database. The default route is an external Type-2 LSA by default. (External routes are classified as Type-1 or Type-2 with a Type-1 route preferred over a Type-2 route.)

The command **default-information originate** [**always**] [**metric** *metric-value*] [**metric-type** *type-value*] advertises the default route into OSPF. In essence, this command redistributes the default route from one protocol into OSPF. The **always** optional keyword removes the requirement for the default route to be present on the advertising router. By default, BGP does not redistribute internal routes (routes learned via an IBGP peer) into an IGP protocol (OSPF) as a safety mechanism. The command **bgp redistribute-internal** allows IBGP routes to be redistributed into the IGP and is required for advertising the default route into OSPF in this scenario.

Example 4-35 displays the multirouter site configuration for R41, R51, and R52. OSPF uses a passive interface default to prevent an OSPF neighborship from forming across the DMVPN tunnel in case OSPF is accidentally enabled.

Example 4-35 *Configuration for Downstream OSPF Routers*

```
R41
router bgp 10
 address-family ipv4
  neighbor MPLS-HUB next-hop-self all
  neighbor INET-HUB next-hop-self all
```

```
R51
router bgp 10
 address-family ipv4
  neighbor MPLS-HUB next-hop-self all
```

```
R52
router bgp 10
 address-family ipv4
  neighbor INET-HUB next-hop-self all
```

```
R41, R51 and R52
router ospf 1
 passive-interface default
 no passive-interface GigabitEthernet0/3
 no passive-interface GigabitEthernet1/0
 redistribute bgp 10 subnets route-map REDIST-BGP-TO-OSPF
 network 10.0.0.0 0.255.255.255 area 0
 default-information originate
!
router bgp 10
 address-family ipv4
  bgp redistribute-internal
  redistribute ospf 1 route-map REDIST-OSPF-TO-BGP
!
route-map REDIST-BGP-TO-OSPF permit 10
 description Set a route tag to identify routes redistributed from BGP
 set tag 1
!
route-map REDIST-OSPF-TO-BGP deny 10
 description Block all routes redistributed from BGP
 match tag 1
!
route-map REDIST-OSPF-TO-BGP deny 15
 match ip address prefix-list TUNNEL-DMVPN
!
route-map REDIST-OSPF-TO-BGP permit 30
 description Redistribute all other traffic
 match route-type internal
 match route-type external type-1
 match route-type external type-2
!
ip prefix-list TUNNEL-DMVPN seq 10 permit 192.168.100.0/24
ip prefix-list TUNNEL-DMVPN seq 20 permit 192.168.200.0/24
```

Example 4-36 demonstrates that OSPF is enabled only on the LAN and loopback interfaces, and that the OSPF networks were redistributed into BGP. Notice that the DMVPN networks were not redistributed into BGP.

Example 4-36 *Verification of OSPF Interfaces and Route Advertisements into BGP*

```
R51-Spoke# show ip ospf interface brief
Interface    PID   Area            IP Address/Mask    Cost  State Nbrs F/C
Lo0          1     0               10.5.0.51/32       1     LOOP  0/0
Gi1/0        1     0               10.5.5.51/24       1     DR    1/1
Gi0/3        1     0               10.5.12.51/24      1     DR    1/1
```

```
R51-Spoke# show bgp ipv4 unicast
BGP table version is 408, local router ID is 10.5.0.51
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 * i 0.0.0.0          192.168.100.21           1    100  50000 i
 *>i                  192.168.100.11           1    100  50000 i
 * i 10.3.0.31/32     192.168.100.21           0    100  50000 ?
 *>i                  192.168.100.11           0    100  50000 ?
 * i 10.3.3.0/24      192.168.100.21           0    100  50000 ?
 *>i                  192.168.100.11           0    100  50000 ?
 r>i 10.4.0.41/32     192.168.100.11           0    100  50000 ?
 r i                  192.168.100.21           0    100  50000 ?
 r>i 10.4.4.0/24      192.168.100.11           0    100  50000 ?
 r i                  192.168.100.21           0    100  50000 ?
 *>  10.5.0.51/32     0.0.0.0                  0         32768 ?
 *>  10.5.0.52/32     10.5.12.52               2         32768 ?
 *>  10.5.5.0/24      0.0.0.0                  0         32768 ?
 *>  10.5.12.0/24     0.0.0.0                  0         32768 ?
```

Example 4-37 verifies that the 10.3.3.0/24 network was redistributed from BGP into OSPF and the route tag of 1 was set to prevent the network prefix from being redistributed back into BGP.

Example 4-37 *Verification of IGP Route Tagging to Prevent Routing Loops*

```
R51-Spoke# show ip ospf database external 10.3.3.0
            OSPF Router with ID (10.5.0.51) (Process ID 1)
                Type-5 AS External Link States
  LS age: 528
  Options: (No TOS-capability, DC, Upward)
  LS Type: AS External Link
  Link State ID: 10.3.3.0 (External Network Number)
  Advertising Router: 10.5.0.51
  LS Seq Number: 800000D5
  Checksum: 0x1477
  Length: 36
  Network Mask: /24
        Metric Type: 2 (Larger than any link state path)
        MTID: 0
        Metric: 1
        Forward Address: 0.0.0.0
        External Route Tag: 1
```

Changing BGP Administrative Distance

A review of the BGP table on R51 in Example 4-36 indicates that there are BGP RIB failures (indicated by the '*r*'). The command **show bgp** *afi safi* **rib-failure** provides more detailed information on the RIB failure. This behavior happens because OSPF has a lower AD, 110, than a route learned from an IBGP peer, 200. This is confirmed in Example 4-38.

Example 4-38 *Identifying the Reason for BGP RIB Failure*

```
R51-Spoke# show bgp ipv4 unicast rib-failure
  Network            Next Hop                      RIB-failure   RIB-NH Matches
10.4.0.41/32       192.168.100.11      Higher admin distance              n/a
10.4.4.0/24        192.168.100.11      Higher admin distance              n/a
```

BGP differentiates between routes learned from IBGP peers, routes learned from EBGP peers, and routes learned locally. The AD needs to be modified on all the DMVPN routers to ensure that they will always prefer IBGP to OSPF. The AD for EBGP learned routes was elevated in case of route leaking to provide connectivity from other SP services so that IWAN learned routes always take precedence. The default AD values can be modified with the address family command **distance bgp** *external-ad internal-ad local-routes* to set the AD for each BGP network type.

Note Locally learned routes are from aggregate (summary) or backdoor networks. Routes advertised via the **network** statement use the AD setting for IBGP routes.

Example 4-39 displays the configuration to modify the AD so that IBGP routes from the WAN are preferred over OSPF paths. This configuration is deployed to all the routers (branch and hub) to ensure a consistent routing policy.

Example 4-39 *Modification of BGP Administrative Distance*

```
R11, R12, R21, R22, R31, R41, R51 and R52
router bgp 10
 address-family ipv4 unicast
  distance bgp 201 19  19
```

> **Note** The changes to the BGP AD are not seen until the route is reintroduced to the RIB. The command **clear ip route** * forces the RIB to reload from all routing protocol databases.

Example 4-40 verifies that the AD has changed for BGP and is now 19 for IBGP-based routes. R12 now uses the BGP route learned via tunnel 200 to reach the 10.3.3.0/24 network. Now R12 redistributes the route into OSPF too. R13 now has two equal-cost paths to reach the 10.3.3.0/24 network.

Example 4-40 *Verification of AD Change*

```
R51-Spoke# show bgp ipv4 unicast
! Output omitted for brevity
     Network          Next Hop            Metric LocPrf Weight Path
*>i 10.4.0.41/32     192.168.100.11           0    100  50000 ?
 * i                  192.168.100.21           0    100  50000 ?
 *>i 10.4.4.0/24      192.168.100.11           0    100  50000 ?
 * i                  192.168.100.21           0    100  50000 ?
```

```
R51-Spoke# show bgp ipv4 unicast rib-failure
  Network            Next Hop                      RIB-failure   RIB-NH Matches
```

```
R51-Spoke# show ip route bgp
! Output omitted for brevity
B*    0.0.0.0/0 [19/1] via 192.168.100.11, 00:13:57
      10.0.0.0/8 is variably subnetted, 10 subnets, 2 masks
B        10.3.0.31/32 [19/0] via 192.168.100.11, 00:13:57
B        10.3.3.0/24 [19/0] via 192.168.100.11, 00:13:57
B        10.4.0.41/32 [19/0] via 192.168.100.11, 00:13:57
B        10.4.4.0/24 [19/0] via 192.168.100.11, 00:13:57
```

Route Advertisement on DMVPN Hub Routers

The DMVPN hub routers play a critical role in route advertisement and are responsible for

- Redistributing BGP network prefixes into the IGP (OSPF).
- Reflecting branch site network prefixes to other branch sites.

- Advertising network prefixes that reside in the local DC.
- Advertising network prefixes that reside elsewhere in the organization.
- Summarizing network prefixes where feasible to reduce the size of the routing table on the branches. NHRP can inject more specific routes where spoke-to-spoke DMVPN tunnels have been established.
- Advertising a default route for a centralized Internet access model (this task was accomplished earlier in the BGP section).

The DMVPN hub router advertises the enterprise (LAN and WAN) network prefixes to the branch routers and the WAN prefixes to the headquarters LAN via redistribution. The router needs to be healthy to advertise network prefixes to the branches so that it can be an active transit to the headquarters LAN. The first check is to make sure that the WAN is healthy, which is easily accomplished with the branch router's ability to establish a tunnel with the hub. It is equally important that the DMVPN hub router maintain LAN connectivity to avoid blackholing network traffic. PfR identifies the optimal path across the WAN transport but does not go deeper into the network infrastructure like the DC for end-to-end verification.

Network prefixes can be injected into BGP dynamically and then summarized before they are advertised to the branch sites. However, in some failure scenarios, it is possible that a branch site can still trigger the enterprise prefix summary route to be generated. PfR would see that the WAN transport to the impaired hub router is still viable and preferred, but traffic would blackhole at the DMVPN hub router.

The solution is to place floating static routes (pointed at Null0) for the enterprise prefixes that use object tracking. The appropriate BGP network statements match the floating static route, are advertised into BGP, and then are advertised to the branch sites. In the event that the route is present in an IGP routing table, the AD of the IGP will be lower than that of the floating static route.

The redistribution of BGP into OSPF is dynamic in nature and does not require any health checks. As any routes are removed from the BGP table, they are withdrawn from the OSPF link-state database (LSDB).

DMVPN Hub LAN Connectivity Health Check

The DMVPN hub assesses its own health by evaluating its connectivity to the LAN. It does this by confirming whether it can reach the following things:

- The local site's loopback address of the PfR MC, which is either a Hub MC or a Transit MC
- The loopback address of the upstream WAN distribution switches, or, if there are no WAN distribution switches, the next-hop routers that lead toward the core network

All three loopback addresses are combined into a consolidated tracked object. Only when the hub router loses connectivity to the local MC and both upstream distribution switches is the router deemed unhealthy. At this time, the router withdraws the floating static routes, which then withdraw the routes from BGP. Multiple devices are used to check the health to prevent false positives when maintenance is performed on a network device.

The logic uses the track object feature, which allows multiple types of objects to be tracked. Specifically in this instance, the route to the loopback (a /32 network prefix) is tracked. If the route is present, it returns an *up* value, and if the route is not present in the RIB, it returns a *down*. The logic is based upon the hub router having these routes in its RIB, if it maintains connectivity to establish an OSPF neighbor adjacency. When all the loopback addresses have been removed from the RIB (presumably because of no OSPF neighborship), the tracked monitor reports a value of *down*.

Before the floating static routes are created, the tracking must be created using the following process:

Step 1. **Create child tracked objects.**

Every one of the individual routes (loopback addresses) needs to be configured as a child tracked object with the command **track** *track-number* **ip route** *network subnet-mask* **reachability**. Every loopback address has a unique number.

Step 2. **Create a track list monitor for all the child objects.**

The command **track** *track-number* **list boolean or** defines the master tracking entity.

Step 3. **Link the child tracked objects.**

The child tracked objects for each one of the loopback interfaces are added with the command **object** *track-number*. The tracked monitor reports a value of *up* when any of the loopback addresses that were learned from OSPF are in the RIB.

Example 4-41 demonstrates the configuration for creating the tracked objects so that a router can verify their health. The first tracked object is the local PfR MC. The second tracked object is the upstream router, and the third tracked object is the other DMVPN hub router at that site.

Example 4-41 *Configuration to Check DMVPN Health with a LAN Network*

```
R11
track 1 ip route 10.1.0.10 255.255.255.255 reachability
track 2 ip route 10.1.0.13 255.255.255.255 reachability
track 3 ip route 10.1.0.12 255.255.255.255 reachability
```

```
R12
track 1 ip route 10.1.0.10 255.255.255.255 reachability
track 2 ip route 10.1.0.13 255.255.255.255 reachability
track 3 ip route 10.1.0.11 255.255.255.255 reachability
```

```
R21
track 1 ip route 10.2.0.20 255.255.255.255 reachability
track 2 ip route 10.2.0.23 255.255.255.255 reachability
track 3 ip route 10.2.0.22 255.255.255.255 reachability
```

```
R22
track 1 ip route 10.2.0.20 255.255.255.255 reachability
track 2 ip route 10.2.0.23 255.255.255.255 reachability
track 3 ip route 10.2.0.21 255.255.255.255 reachability
```

```
R11, R12, R21, and R22
track 100 list boolean or
 object 1
 object 2
 object 3
```

The status of the health check can be acquired by using the command **show track**, as shown in Example 4-42. The output provides the object, what is being tracked, the number of changes, and the last state change. If the object is a child object, the output indicates the parent object that is using it.

Example 4-42 *Verification of Object Tracking*

```
R11-DC1-Hub1# show track
Track 1
  IP route 10.1.0.10 255.255.255.255 reachability
  Reachability is Up (OSPF)
    2 changes, last change 04:32:32
  First-hop interface is GigabitEthernet1/0
  Tracked by:
    Track List 100
Track 2
  IP route 10.1.0.13 255.255.255.255 reachability
  Reachability is Up (OSPF)
    3 changes, last change 04:31:47
  First-hop interface is GigabitEthernet1/0
  Tracked by:
    Track List 100
Track 3
  IP route 10.1.0.12 255.255.255.255 reachability
  Reachability is Up (OSPF)
    1 change, last change 04:35:17
  First-hop interface is GigabitEthernet0/3
  Tracked by:
    Track List 100
Track 100
  List boolean or
  Boolean OR is Up
    2 changes, last change 04:34:12
    object 1 Up
    object 2 Up
    object 3 Up
```

BGP Route Advertisement on Hub Routers

Now that the LAN health check has been configured for the hub router, it is time to create the static routes. The static routes are used to create an entry into the global RIB, which is required for the route to be installed into BGP when using **network** statements.

The static route uses the syntax **ip route** *network subnet-mask outbound-interface* [*administrative distance*] [**track** *track-number*]. The static route uses the Null0 interface to drop network packets. It is set with a high administrative distance (254) so that if the route is advertised by an IGP, the IGP path can be used. Otherwise, the static route always has a lower route and causes the router to drop traffic even if a valid route exists in an IGP. Because the static route is not always used if the route is present in an IGP, it is called a *floating static route*. The floating static route is linked to the hub health check by using the **track** option with the parent object. As long as the tracked object is in an *up* state, the static route attempts to install into the RIB.

A static route needs to be created for the following:

- **The enterprise prefix summary networks:** These should include all the networks in use in the LAN and WAN. Generally, these are the networks in the RFC 1918 space (10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16). If the organization has public IP addresses and uses them for internal connectivity, they should be added as well. For the sake of brevity, this book uses only the 10.0.0.0/8 range.

- **The local DC's networks:** This ensures that traffic is directed toward these hub routers instead of other hub routers.

After those static routes have been defined, the network prefixes need to be configured in BGP. The BGP configuration command **network** *network* **mask** *subnet-mask* notifies BGP to search the global RIB for that exact prefix. When a match is found, the route is installed into the BGP table for advertisement. The command needs to be run for the enterprise prefix summary networks, the local DC networks, the local PfR MCs loopback interface, and the default route (done earlier in this chapter).

Note PfR does not require the MC loopback address in BGP, but this can be very helpful when troubleshooting. A static route for the local MC is not required as it is a /32 address and learned via IGP.

Example 4-43 displays the configuration of the floating static routes and the BGP network statements. Notice that the configuration is grouped by DC site location.

Example 4-43 *Configuration of Floating Static Routes and BGP Network Statements*

```
R11 and R12
ip route 10.0.0.0 255.0.0.0 Null0 254 track 100
ip route 10.1.0.0 255.255.0.0 Null0 254 track 100
!
router bgp 10
 address-family ipv4
  network 10.0.0.0 mask 255.0.0.0
  network 10.1.0.0 mask 255.255.0.0
  network 10.1.0.10 mask 255.255.255.255
```

```
R21 and R22
ip route 10.0.0.0 255.0.0.0 Null0 254 track 100
ip route 10.2.0.0 255.255.0.0 Null0 254 track 100
!
router bgp 10
 address-family ipv4
  network 10.0.0.0 mask 255.0.0.0
  network 10.2.0.0 mask 255.255.0.0
  network 10.2.0.20 mask 255.255.255.255
```

Example 4-44 verifies that all the routes that were advertised on the hub routers were received at the branch routers. Notice that the DC-specific prefixes are advertised out of the appropriate hub router.

Example 4-44 *Verification of Routes Advertised into BGP*

```
R31-Spoke# show bgp ipv4 unicast
! Output omitted for brevity
    Network          Next Hop            Metric LocPrf Weight Path
 * i 0.0.0.0          192.168.200.12           1    100  50000 i
 * i                  192.168.200.22           1    100  50000 i
 * i                  192.168.100.21           1    100  50000 i
 *>i                  192.168.100.11           1    100  50000 i
 * i 10.0.0.0         192.168.200.12           0    100  50000 i
 * i                  192.168.200.22           0    100  50000 i
 * i                  192.168.100.21           0    100  50000 i
 *>i                  192.168.100.11           0    100  50000 i
 * i 10.1.0.0/16      192.168.200.12           0    100  50000 i
 *>i                  192.168.100.11           0    100  50000 i
 * i 10.1.0.10/32     192.168.200.12           3    100  50000 i
 *>i                  192.168.100.11           3    100  50000 i
 * i 10.2.0.0/16      192.168.200.22           0    100  50000 i
 *>i                  192.168.100.21           0    100  50000 i
 * i 10.2.0.20/32     192.168.200.22           3    100  50000 i
 *>i                  192.168.100.21           3    100  50000 i
 *>  10.3.0.31/32     0.0.0.0                  0         32768 ?
 *>  10.3.3.0/24      0.0.0.0                  0         32768 ?
 * i 10.4.0.41/32     192.168.200.22           0    100  50000 ?
 * i                  192.168.200.12           0    100  50000 ?
 * i                  192.168.100.21           0    100  50000 ?
 *>i                  192.168.100.11           0    100  50000 ?
 * i 10.4.4.0/24      192.168.200.22           0    100  50000 ?
 * i                  192.168.200.12           0    100  50000 ?
 * i                  192.168.100.21           0    100  50000 ?
 *>i                  192.168.100.11           0    100  50000 ?
```

BGP Route Filtering

The hub routers currently advertise all the routes that they have learned to other branches. Now that the BGP table has been injected with network summaries, the advertisement of network prefixes can be restricted to only the summary prefixes.

As a safety precaution, the same prefixes that the hub routers advertise to the branches should be prohibited from being received from a branch router. In addition to those prefixes, the hub routers should not accept the DMVPN tunnel networks.

The best approach is to create multiple prefix lists, so that a prefix list correlates with a certain function: default route, enterprise prefix, local DC segment, local MC, or DMVPN tunnel network.

Example 4-45 demonstrates the process for creating a prefix list for a specific function. An outbound route map is added to the branches that contains only the approved prefixes (default route, enterprise prefix list, DC-specific networks, and local MC). A BGP community can be added to the prefix to assist with additional routing logic at a later time (if desired). The inbound route map denies all the summary routes and the DMVPN tunnel networks.

Example 4-45 *Configuration for Outbound and Inbound BGP Filtering*

```
R11 and R12
ip prefix-list BGP-ENTERPRISE-PREFIX seq 10 permit 10.0.0.0/8
ip prefix-list BGP-LOCALDC-PREFIX seq 10 permit 10.1.0.0/16
ip prefix-list BGP-LOCALMC seq 10 permit 10.1.0.10/32
ip prefix-list DEFAULT-ROUTE seq 10 permit 0.0.0.0/0
ip prefix-list TUNNEL-DMVPN seq 10 permit 192.168.100.0/24
ip prefix-list TUNNEL-DMVPN seq 20 permit 192.168.200.0/24
```

```
R21 and R22
ip prefix-list BGP-ENTERPRISE-PREFIX seq 10 permit 10.0.0.0/8
ip prefix-list BGP-LOCALDC-PREFIX seq 10 permit 10.2.0.0/16
ip prefix-list BGP-LOCALMC seq 10 permit 10.2.0.20/32
ip prefix-list DEFAULT-ROUTE seq 10 permit 0.0.0.0/0
ip prefix-list TUNNEL-DMVPN seq 10 permit 192.168.100.0/24
ip prefix-list TUNNEL-DMVPN seq 20 permit 192.168.200.0/24
```

```
R11 and R21
router bgp 10
 address-family ipv4
  neighbor MPLS-SPOKES route-map BGP-MPLS-SPOKES-IN in
  neighbor MPLS-SPOKES route-map BGP-MPLS-SPOKES-OUT out
!
route-map BGP-MPLS-SPOKES-OUT permit 10
 match ip address prefix-list DEFAULT-ROUTE
route-map BGP-MPLS-SPOKES-OUT permit 20
 match ip address prefix-list BGP-ENTERPRISE-PREFIX
route-map BGP-MPLS-SPOKES-OUT permit 30
 match ip address prefix-list BGP-LOCALDC-PREFIX
route-map BGP-MPLS-SPOKES-OUT permit 40
match ip address prefix-list BGP-LOCALMC
!
! The first five sequences of the route-map deny network prefixes
! that can cause suboptimal routing or routing loops
route-map BGP-MPLS-SPOKES-IN deny 10
 match ip address prefix-list DEFAULT-ROUTE
route-map BGP-MPLS-SPOKES-IN deny 20
 match ip address prefix-list BGP-ENTERPRISE-PREFIX
```

```
route-map BGP-MPLS-SPOKES-IN deny 30
 match ip address prefix-list BGP-LOCALDC-PREFIX
route-map BGP-MPLS-SPOKES-IN deny 40
 match ip address prefix-list BGP-LOCALMC
route-map BGP-MPLS-SPOKES-IN deny 50
 match ip address prefix-list TUNNEL-DMVPN
route-map BGP-MPLS-SPOKES-IN permit 60
 description Allow Everything Else
```

R12 and R22

```
router bgp 10
 address-family ipv4
  neighbor INET-SPOKES route-map BGP-INET-SPOKES-IN in
  neighbor INET-SPOKES route-map BGP-INET-SPOKES-OUT out
!
route-map BGP-INET-SPOKES-OUT permit 10
 match ip address prefix-list DEFAULT-ROUTE
route-map BGP-INET-SPOKES-OUT permit 20
 match ip address prefix-list BGP-ENTERPRISE-PREFIX
route-map BGP-INET-SPOKES-OUT permit 30
 match ip address prefix-list BGP-LOCALDC-PREFIX
route-map BGP-INET-SPOKES-OUT permit 40
match ip address prefix-list BGP-LOCALMC
!
! The first five sequences of the route-map deny network prefixes
! that can cause suboptimal routing or routing loops
route-map BGP-INET-SPOKES-IN deny 10
 match ip address prefix-list DEFAULT-ROUTE
route-map BGP-INET-SPOKES-IN deny 20
 match ip address prefix-list BGP-ENTERPRISE-PREFIX
route-map BGP-INET-SPOKES-IN deny 30
 match ip address prefix-list BGP-LOCALDC-PREFIX
route-map BGP-INET-SPOKES-IN deny 40
 match ip address prefix-list BGP-LOCALMC
route-map BGP-INET-SPOKES-IN deny 50
 match ip address prefix-list TUNNEL-DMVPN
route-map BGP-INET-SPOKES-IN permit 60
description Allow Everything Else
```

Note The number of sequence numbers in the route maps can be reduced by adding multiple conditional matches of the same type (prefix list); however, this also creates an SPOF if a sequence is accidentally deleted when making changes.

Example 4-46 confirms that only the appropriate routes are being advertised from the DMVPN hub routers. Filtering the other routes reduces the amount of memory needed to maintain the BGP table. NHRP injects more specific routes for spoke-to-spoke tunnels when they are established.

Example 4-46 *Verification of Route Filtering on DMVPN Hub Routers*

```
R31-Spoke# show bgp ipv4 unicast
! Output omitted for brevity
     Network          Next Hop            Metric LocPrf Weight Path
 * i 0.0.0.0          192.168.200.12           1    100  50000 i
 * i                  192.168.200.22           1    100  50000 i
 * i                  192.168.100.21           1    100  50000 i
 *>i                  192.168.100.11           1    100  50000 i
 * i 10.0.0.0         192.168.200.12           0    100  50000 i
 * i                  192.168.200.22           0    100  50000 i
 * i                  192.168.100.21           0    100  50000 i
 *>i                  192.168.100.11           0    100  50000 i
 * i 10.1.0.0/16      192.168.200.12           0    100  50000 i
 *>i                  192.168.100.11           0    100  50000 i
 * i 10.1.0.10/32     192.168.200.12           3    100  50000 i
 *>i                  192.168.100.11           3    100  50000 i
 * i 10.2.0.0/16      192.168.200.22           0    100  50000 i
 *>i                  192.168.100.21           0    100  50000 i
 * i 10.2.0.20/32     192.168.200.22           3    100  50000 i
 *>i                  192.168.100.21           3    100  50000 i
 *>  10.3.0.31/32     0.0.0.0                  0         32768 ?
 *>  10.3.3.0/24      0.0.0.0                  0         32768 ?
```

Redistribution of BGP into OSPF

The last component of route advertisement on the hub routers is the process of redistributing routes from BGP into OSPF. The BGP configuration command **bgp redistribute-internal** is required because all the branch site routes were learned via an IBGP session.

A route map is required during redistribution to prevent the static network statements from being redistributed into OSPF. The route map can reuse the prefix lists that were used to filter routes.

Example 4-47 displays the configuration for R11, R12, R21, and R22 that redistributes the BGP network prefixes into OSPF. Notice that the first three sequences of the REDIST-BGP-TO-OSPF route map block the default route, enterprise summary, and local DC prefixes from being redistributed.

Example 4-47 *BGP Route Advertisement into OSPF*

```
R11, R12, R21, and R22
router bgp 10
 address-family ipv4 unicast
 bgp redistribute-internal
!
router ospf 1
 redistribute bgp 10 subnets route-map REDIST-BGP-TO-OSPF
!
route-map REDIST-BGP-TO-OSPF deny 10
 match ip address prefix-list BGP-ENTERPRISE-PREFIX
route-map REDIST-BGP-TO-OSPF deny 20
 match ip address prefix-list BGP-LOCALDC-PREFIX
route-map REDIST-BGP-TO-OSPF deny 30
 match ip address prefix-list DEFAULT-ROUTE
route-map REDIST-BGP-TO-OSPF permit 40
 description Modify Metric to Prefer MPLS over Internet
set metric-type type-1
```

Example 4-48 verifies that the redistribution was successful and that the routes are populating appropriately in the headquarters LAN. Notice that R13 can take either path to reach the branch network sites.

Example 4-48 *Verification of Branch Network Prefixes at the Headquarters LAN*

```
R13# show ip route ospf
! Output omitted for brevity
O E1     10.3.0.31/32 [110/2] via 10.1.112.12, 00:01:12, GigabitEthernet1/1
                      [110/2] via 10.1.111.11, 00:01:21, GigabitEthernet1/0
O E1     10.3.3.0/24 [110/2] via 10.1.112.12, 00:01:12, GigabitEthernet1/1
                     [110/2] via 10.1.111.11, 00:01:21, GigabitEthernet1/0
O E1     10.4.0.41/32 [110/2] via 10.1.112.12, 00:01:12, GigabitEthernet1/1
                      [110/2] via 10.1.111.11, 00:01:21, GigabitEthernet1/0
O E1     10.4.4.0/24 [110/2] via 10.1.112.12, 00:01:12, GigabitEthernet1/1
                     [110/2] via 10.1.111.11, 00:01:21, GigabitEthernet1/0
O E1     10.5.0.51/32 [110/2] via 10.1.112.12, 00:01:12, GigabitEthernet1/1
                      [110/2] via 10.1.111.11, 00:01:21, GigabitEthernet1/0
O E1     10.5.0.52/32 [110/2] via 10.1.112.12, 00:01:12, GigabitEthernet1/1
                      [110/2] via 10.1.111.11, 00:01:21, GigabitEthernet1/0
O E1     10.5.5.0/24 [110/2] via 10.1.112.12, 00:01:12, GigabitEthernet1/1
                     [110/2] via 10.1.111.11, 00:01:21, GigabitEthernet1/0
O E1     10.5.12.0/24 [110/2] via 10.1.112.12, 00:01:12, GigabitEthernet1/1
                      [110/2] via 10.1.111.11, 00:01:21, GigabitEthernet1/0
```

Traffic Steering

An examination of Example 4-46 reveals that the BGP path attributes look the same for all the network prefixes, leaving the BGP best path nondeterministic. As explained earlier, the routing protocol design should accommodate the situation when PfR is in an uncontrolled state and direct traffic across the preferred transport.

Local preference is the second step in identifying the best path in BGP and can be set locally or remotely. Setting the BGP local preference on the hubs allows the routing policy to be set on all the branch devices.

R11 and R21 are the DMVPN hubs for the MPLS transport, which is the preferred transport. R12 and R22 are the DMVPN hubs for the Internet transport, which is the secondary transport. Branch routers should prefer Site 1 over Site 2 when PfR is in an uncontrolled state for establishing connectivity to other branches.

R11 advertises routes with a local preference of 100,000, R21 with a value of 20,000, R12 with a value of 3000, and R22 with a value of 400. All these values are above the default setting of 100 and easily show the first, second, third, and fourth order of preference.

Example 4-49 provides the necessary configuration to obtain the results described above. The local preference must be set for every sequence number.

Example 4-49 *Hub Configuration to Set the BGP Path Preference on Branch Routers*

```
R11
! This router should be selected first
route-map BGP-MPLS-SPOKES-OUT permit 10
 set local-preference 100000
route-map BGP-MPLS-SPOKES-OUT permit 20
 set local-preference 100000
route-map BGP-MPLS-SPOKES-OUT permit 30
 set local-preference 100000
route-map BGP-MPLS-SPOKES-OUT permit 40
 set local-preference 100000
```

```
R12
! This router should be selected third
route-map BGP-INET-SPOKES-OUT permit 10
 set local-preference 3000
route-map BGP-INET-SPOKES-OUT permit 20
 set local-preference 3000
route-map BGP-INET-SPOKES-OUT permit 30
 set local-preference 3000
route-map BGP-INET-SPOKES-OUT permit 40
 set local-preference 3000
```

```
R21
! This router should be selected second
route-map BGP-MPLS-SPOKES-OUT permit 10
 set local-preference 20000
route-map BGP-MPLS-SPOKES-OUT permit 20
 set local-preference 20000
route-map BGP-MPLS-SPOKES-OUT permit 30
 set local-preference 20000
route-map BGP-MPLS-SPOKES-OUT permit 40
 set local-preference 20000
```

```
R22
! This router should be selected last
route-map BGP-INET-SPOKES-OUT permit 10
 set local-preference 400
route-map BGP-INET-SPOKES-OUT permit 20
 set local-preference 400
route-map BGP-INET-SPOKES-OUT permit 30
 set local-preference 400
route-map BGP-INET-SPOKES-OUT permit 40
 set local-preference 400
```

Example 4-50 displays R31's BGP table after making the changes on the hub routers. The path priorities are easy to identify with the technique shown in the preceding example. Notice that four paths are still shown for the default route and the 10.0.0.0/8 network. There are only two paths for the 10.1.0.0/16 and the 10.2.0.0/16 networks.

Example 4-50 *BGP Table Demonstrating Path Preference*

```
R31-Spoke# show bgp ipv4 unicast
! Output omitted for brevity
     Network          Next Hop            Metric LocPrf Weight Path
 * i 0.0.0.0          192.168.200.22           1    400  50000 i
 * i                  192.168.100.21           1  20000  50000 i
 * i                  192.168.200.12           1   3000  50000 i
 *>i                  192.168.100.11           1 100000  50000 i
 * i 10.0.0.0         192.168.200.22           0    400  50000 i
 * i                  192.168.100.21           0  20000  50000 i
 * i                  192.168.200.12           0   3000  50000 i
 *>i                  192.168.100.11           0 100000  50000 i
 * i 10.1.0.0/16      192.168.200.12           0   3000  50000 i
 *>i                  192.168.100.11           0 100000  50000 i
 * i 10.2.0.0/16      192.168.200.22           0    400  50000 i
 *>i                  192.168.100.21           0  20000  50000 i
```

Ensuring that the network traffic takes a symmetric path (both directions) simplifies troubleshooting. Setting the local preference on the hub routers ensures the path taken from the branch routers but does not influence the return traffic. Example 4-51 provides R13's routing table, which shows that traffic can go through R11 (MPLS) or through R12 (Internet) on the return path.

Example 4-51 *R13 Path Preference*

```
R13# show ip route ospf
! Output omitted for brevity
O E1     10.3.0.31/32 [110/2] via 10.1.112.12, 00:01:12, GigabitEthernet1/1
                      [110/2] via 10.1.111.11, 00:01:21, GigabitEthernet1/0
```

Setting a higher metric on the OSPF routes as they are redistributed on the Internet routers ensures that the paths are symmetric. Example 4-52 demonstrates the additional configuration to the existing route maps to influence path selection. OSPF prefers a lower-cost path to a higher-cost path.

Example 4-52 *Modification to the Route Map to Influence Return Path Traffic*

```
R11 and R21
route-map REDIST-BGP-TO-OSPF permit 40
 description Modify Metric to Prefer MPLS over Internet
 set metric 1000
```

```
R12 and R22
route-map REDIST-BGP-TO-OSPF permit 40
 description Modify Metric to Prefer MPLS over Internet
 set metric 2000
```

Example 4-53 verifies that the changes made to the route map have removed the asymmetric routing. Traffic leaving Site 1 will always take the path through R11 (MPLS).

Example 4-53 *Verification of the Path Preference on Internal Routers*

```
R13# show ip route ospf
! Output omitted for brevity
O E1     10.3.3.0/24 [110/1001] via 10.1.111.11, 00:00:09, GigabitEthernet1/0
```

Note Ensuring that the traffic is symmetric (uses the same transport in both directions) helps with application classification and WAAS. Multirouter sites like Site 5 should use FHRPs like HSRP that use the primary router that has the primary transport.

Complete BGP Configuration

The preceding sections explained the logic for deploying BGP for the WAN (DMVPN overlay) with redistribution into OSPF at the hub routers (centralized sites). The components were explained in a step-by-step fashion to provide a thorough understanding of the configuration. Example 4-54 shows the complete routing configuration for the DMVPN hub routers.

Example 4-54 *IBGP Hub Router Configuration*

```
R11-Hub
router ospf 1
 redistribute bgp 10 subnets route-map REDIST-BGP-TO-OSPF
 passive-interface default
 no passive-interface GigabitEthernet0/3
 no passive-interface GigabitEthernet1/0
 network 0.0.0.0 255.255.255.255 area 0
!
track 1 ip route 10.1.0.10 255.255.255.255 reachability
track 2 ip route 10.1.0.13 255.255.255.255 reachability
track 3 ip route 10.1.0.12 255.255.255.255 reachability
track 100 list boolean or
 object 1
 object 2
 object 3
!
ip route 10.0.0.0 255.0.0.0 Null0 254 track 100
ip route 10.1.0.0 255.255.0.0 Null0 254 track 100
!
router bgp 10
 bgp router-id 10.1.0.11
 bgp listen range 192.168.100.0/24 peer-group MPLS-SPOKES
 bgp listen limit 254
 neighbor MPLS-SPOKES peer-group
 neighbor MPLS-SPOKES remote-as 10
 neighbor MPLS-SPOKES timers 20 60
 !
 address-family ipv4
  bgp redistribute-internal
  network 0.0.0.0
  network 10.0.0.0
  network 10.1.0.0 mask 255.255.0.0
  network 10.1.0.10 mask 255.255.255.255
  neighbor MPLS-SPOKES activate
  neighbor MPLS-SPOKES send-community
 neighbor MPLS-SPOKES route-reflector-client
  neighbor MPLS-SPOKES next-hop-self all
  neighbor MPLS-SPOKES weight 50000
  neighbor MPLS-SPOKES soft-reconfiguration inbound
  neighbor MPLS-SPOKES route-map BGP-MPLS-SPOKES-IN in
  neighbor MPLS-SPOKES route-map BGP-MPLS-SPOKES-OUT out
  distance bgp 201 19 19
 exit-address-family
!
ip prefix-list BGP-ENTERPRISE-PREFIX seq 10 permit 10.0.0.0/8
ip prefix-list BGP-LOCALDC-PREFIX seq 10 permit 10.1.0.0/16
ip prefix-list BGP-LOCALMC seq 10 permit 10.1.0.10/32
ip prefix-list DEFAULT-ROUTE seq 10 permit 0.0.0.0/0
ip prefix-list TUNNEL-DMVPN seq 10 permit 192.168.100.0/24
ip prefix-list TUNNEL-DMVPN seq 20 permit 192.168.200.0/24
!
route-map BGP-MPLS-SPOKES-OUT permit 10
 match ip address prefix-list DEFAULT-ROUTE
 set local-preference 100000
route-map BGP-MPLS-SPOKES-OUT permit 20
 match ip address prefix-list BGP-ENTERPRISE-PREFIX
 set local-preference 100000
route-map BGP-MPLS-SPOKES-OUT permit 30
 match ip address prefix-list BGP-LOCALDC-PREFIX
 set local-preference 100000
route-map BGP-MPLS-SPOKES-OUT permit 40
 match ip address prefix-list BGP-LOCALMC
 set local-preference 100000
!
route-map BGP-MPLS-SPOKES-IN deny 10
 match ip address prefix-list DEFAULT-ROUTE
route-map BGP-MPLS-SPOKES-IN deny 20
 match ip address prefix-list BGP-ENTERPRISE-PREFIX
route-map BGP-MPLS-SPOKES-IN deny 30
 match ip address prefix-list BGP-LOCALDC-PREFIX
route-map BGP-MPLS-SPOKES-IN deny 40
 match ip address prefix-list BGP-LOCALMC
route-map BGP-MPLS-SPOKES-IN deny 50
 match ip address prefix-list TUNNEL-DMVPN
route-map BGP-MPLS-SPOKES-IN permit 60
 description Allow Everything Else
!
route-map REDIST-BGP-TO-OSPF deny 10
 match ip address prefix-list BGP-ENTERPRISE-PREFIX
```

```
route-map REDIST-BGP-TO-OSPF deny 20
 match ip address prefix-list BGP-LOCALDC-PREFIX
route-map REDIST-BGP-TO-OSPF deny 30
 match ip address prefix-list DEFAULT-ROUTE
route-map REDIST-BGP-TO-OSPF permit 40
 description Modify Metric to Prefer MPLS over Internet
 set metric 1000
 set metric-type type-1
```

R12-Hub

```
router ospf 1
 redistribute bgp 10 subnets route-map REDIST-BGP-TO-OSPF
 passive-interface default
 no passive-interface GigabitEthernet0/3
 no passive-interface GigabitEthernet1/0
 network 0.0.0.0 255.255.255.255 area 0
!
track 1 ip route 10.1.0.10 255.255.255.255 reachability
track 2 ip route 10.1.0.13 255.255.255.255 reachability
track 3 ip route 10.1.0.11 255.255.255.255 reachability
track 100 list boolean or
 object 1
 object 2
 object 3
!
ip route 10.0.0.0 255.0.0.0 Null0 254 track 100
ip route 10.1.0.0 255.255.0.0 Null0 254 track 100
!
router bgp 10
 bgp router-id 10.1.0.12
 bgp listen range 192.168.200.0/24 peer-group INET-SPOKES
 bgp listen limit 254
 neighbor INET-SPOKES peer-group
 neighbor INET-SPOKES remote-as 10
 neighbor INET-SPOKES timers 20 60
 !
 address-family ipv4
  bgp redistribute-internal
  network 0.0.0.0
  network 10.0.0.0
  network 10.1.0.0 mask 255.255.0.0
  network 10.1.0.10 mask 255.255.255.255
  neighbor INET-SPOKES activate
  neighbor INET-SPOKES send-community
  neighbor INET-SPOKES route-reflector-client
  neighbor INET-SPOKES next-hop-self all
  neighbor INET-SPOKES weight 50000
  neighbor INET-SPOKES soft-reconfiguration inbound
  neighbor INET-SPOKES route-map BGP-INET-SPOKES-IN in
  neighbor INET-SPOKES route-map BGP-INET-SPOKES-OUT out
  distance bgp 201 19 19
 exit-address-family
!
ip prefix-list BGP-ENTERPRISE-PREFIX seq 10 permit 10.0.0.0/8
ip prefix-list BGP-LOCALDC-PREFIX seq 10 permit 10.1.0.0/16
ip prefix-list BGP-LOCALMC seq 10 permit 10.1.0.10/32
ip prefix-list DEFAULT-ROUTE seq 10 permit 0.0.0.0/0
ip prefix-list TUNNEL-DMVPN seq 10 permit 192.168.100.0/24
ip prefix-list TUNNEL-DMVPN seq 20 permit 192.168.200.0/24
!
route-map BGP-INET-SPOKES-OUT permit 10
 match ip address prefix-list DEFAULT-ROUTE
 set local-preference 3000
route-map BGP-INET-SPOKES-OUT permit 20
 match ip address prefix-list BGP-ENTERPRISE-PREFIX
 set local-preference 3000
route-map BGP-INET-SPOKES-OUT permit 30
 match ip address prefix-list BGP-LOCALDC-PREFIX
 set local-preference 3000
route-map BGP-INET-SPOKES-OUT permit 40
 match ip address prefix-list BGP-LOCALMC
 set local-preference 3000
!
route-map REDIST-BGP-TO-OSPF deny 10
 match ip address prefix-list BGP-ENTERPRISE-PREFIX
route-map REDIST-BGP-TO-OSPF deny 20
 match ip address prefix-list BGP-LOCALDC-PREFIX
route-map REDIST-BGP-TO-OSPF deny 30
 match ip address prefix-list DEFAULT-ROUTE
route-map REDIST-BGP-TO-OSPF permit 40
 description Modify Metric to Prefer MPLS over Internet
 set metric 2000
 set metric-type type-1
!
route-map BGP-INET-SPOKES-IN deny 10
 match ip address prefix-list DEFAULT-ROUTE
route-map BGP-INET-SPOKES-IN deny 20
 match ip address prefix-list BGP-ENTERPRISE-PREFIX
```

```
route-map BGP-INET-SPOKES-IN deny 30
 match ip address prefix-list BGP-LOCALDC-PREFIX
route-map BGP-INET-SPOKES-IN deny 40
 match ip address prefix-list BGP-LOCALMC
route-map BGP-INET-SPOKES-IN deny 50
 match ip address prefix-list TUNNEL-DMVPN
route-map BGP-INET-SPOKES-IN permit 60
 description Allow Everything Else
```

R21-Hub

```
router ospf 1
 redistribute bgp 10 subnets route-map REDIST-BGP-TO-OSPF
 passive-interface default
 no passive-interface GigabitEthernet0/3
 no passive-interface GigabitEthernet1/0
 network 0.0.0.0 255.255.255.255 area 0
!
track 1 ip route 10.2.0.20 255.255.255.255 reachability
track 2 ip route 10.2.0.23 255.255.255.255 reachability
track 3 ip route 10.2.0.22 255.255.255.255 reachability
track 100 list boolean or
 object 1
 object 2
 object 3
!
ip route 10.0.0.0 255.0.0.0 Null0 254 track 100
ip route 10.2.0.0 255.255.0.0 Null0 254 track 100
!
router bgp 10
 bgp router-id 10.2.0.21
 bgp listen range 192.168.100.0/24 peer-group MPLS-SPOKES
 bgp listen limit 254
 neighbor MPLS-SPOKES peer-group
 neighbor MPLS-SPOKES remote-as 10
 neighbor MPLS-SPOKES timers 20 60
 !
 address-family ipv4
  bgp redistribute-internal
  network 0.0.0.0
  network 10.0.0.0
  network 10.2.0.0 mask 255.255.0.0
  network 10.2.0.20 mask 255.255.255.255
  neighbor MPLS-SPOKES activate
  neighbor MPLS-SPOKES send-community
  neighbor MPLS-SPOKES route-reflector-client
  neighbor MPLS-SPOKES next-hop-self all
  neighbor MPLS-SPOKES weight 50000
  neighbor MPLS-SPOKES soft-reconfiguration inbound
  neighbor MPLS-SPOKES route-map BGP-MPLS-SPOKES-IN in
  neighbor MPLS-SPOKES route-map BGP-MPLS-SPOKES-OUT out
  distance bgp 201 19 19
 exit-address-family
!
ip prefix-list BGP-ENTERPRISE-PREFIX seq 10 permit 10.0.0.0/8
ip prefix-list BGP-LOCALDC-PREFIX seq 10 permit 10.2.0.0/16
ip prefix-list BGP-LOCALMC seq 10 permit 10.2.0.20/32
ip prefix-list DEFAULT-ROUTE seq 10 permit 0.0.0.0/0
ip prefix-list TUNNEL-DMVPN seq 10 permit 192.168.100.0/24
ip prefix-list TUNNEL-DMVPN seq 20 permit 192.168.200.0/24
!
route-map BGP-MPLS-SPOKES-OUT permit 10
 match ip address prefix-list DEFAULT-ROUTE
 set local-preference 20000
route-map BGP-MPLS-SPOKES-OUT permit 20
 match ip address prefix-list BGP-ENTERPRISE-PREFIX
 set local-preference 20000
route-map BGP-MPLS-SPOKES-OUT permit 30
 match ip address prefix-list BGP-LOCALDC-PREFIX
 set local-preference 20000
route-map BGP-MPLS-SPOKES-OUT permit 40
 match ip address prefix-list BGP-LOCALMC
 set local-preference 20000
!
route-map BGP-MPLS-SPOKES-IN deny 10
 match ip address prefix-list DEFAULT-ROUTE
route-map BGP-MPLS-SPOKES-IN deny 20
 match ip address prefix-list BGP-ENTERPRISE-PREFIX
route-map BGP-MPLS-SPOKES-IN deny 30
 match ip address prefix-list BGP-LOCALDC-PREFIX
route-map BGP-MPLS-SPOKES-IN deny 40
 match ip address prefix-list BGP-LOCALMC
route-map BGP-MPLS-SPOKES-IN deny 50
 match ip address prefix-list TUNNEL-DMVPN
route-map BGP-MPLS-SPOKES-IN permit 60
 description Allow Everything Else
!
route-map REDIST-BGP-TO-OSPF deny 10
 match ip address prefix-list BGP-ENTERPRISE-PREFIX
```

```
route-map REDIST-BGP-TO-OSPF deny 20
 match ip address prefix-list BGP-LOCALDC-PREFIX
route-map REDIST-BGP-TO-OSPF deny 30
 match ip address prefix-list DEFAULT-ROUTE
route-map REDIST-BGP-TO-OSPF permit 40
 description Modify Metric to Prefer MPLS over Internet
 set metric 1000
 set metric-type type-1
```

R22-Hub

```
router ospf 1
 redistribute bgp 10 subnets route-map REDIST-BGP-TO-OSPF
 passive-interface default
 no passive-interface GigabitEthernet0/3
 no passive-interface GigabitEthernet1/0
 network 0.0.0.0 255.255.255.255 area 0
!
!
track 1 ip route 10.2.0.20 255.255.255.255 reachability
track 2 ip route 10.2.0.23 255.255.255.255 reachability
track 3 ip route 10.2.0.21 255.255.255.255 reachability
track 100 list boolean or
 object 1
 object 2
 object 3
!
ip route 10.0.0.0 255.0.0.0 Null0 254 track 100
ip route 10.2.0.0 255.255.0.0 Null0 254 track 100
!
router bgp 10
 bgp router-id 10.2.0.22
 bgp listen range 192.168.200.0/24 peer-group INET-SPOKES
 bgp listen limit 254
 neighbor INET-SPOKES peer-group
 neighbor INET-SPOKES remote-as 10
 neighbor INET-SPOKES timers 20 60
 !
 address-family ipv4
  bgp redistribute-internal
  network 0.0.0.0
  network 10.0.0.0
  network 10.2.0.0 mask 255.255.0.0
  network 10.2.0.20 mask 255.255.255.255
  neighbor INET-SPOKES activate
  neighbor INET-SPOKES send-community
  neighbor INET-SPOKES route-reflector-client
  neighbor INET-SPOKES next-hop-self all
  neighbor INET-SPOKES weight 50000
  neighbor INET-SPOKES soft-reconfiguration inbound
  neighbor INET-SPOKES route-map BGP-INET-SPOKES-IN in
  neighbor INET-SPOKES route-map BGP-INET-SPOKES-OUT out
  distance bgp 201 19 19
 exit-address-family
!
ip prefix-list BGP-ENTERPRISE-PREFIX seq 10 permit 10.0.0.0/8
ip prefix-list BGP-LOCALDC-PREFIX seq 10 permit 10.2.0.0/16
ip prefix-list BGP-LOCALMC seq 10 permit 10.2.0.20/32
ip prefix-list DEFAULT-ROUTE seq 10 permit 0.0.0.0/0
ip prefix-list TUNNEL-DMVPN seq 10 permit 192.168.100.0/24
ip prefix-list TUNNEL-DMVPN seq 20 permit 192.168.200.0/24
!
route-map BGP-INET-SPOKES-OUT permit 10
 match ip address prefix-list DEFAULT-ROUTE
 set local-preference 400
route-map BGP-INET-SPOKES-OUT permit 20
 match ip address prefix-list BGP-ENTERPRISE-PREFIX
 set local-preference 400
route-map BGP-INET-SPOKES-OUT permit 30
 match ip address prefix-list BGP-LOCALDC-PREFIX
 set local-preference 400
route-map BGP-INET-SPOKES-OUT permit 40
 match ip address prefix-list BGP-LOCALMC
 set local-preference 400
!
route-map REDIST-BGP-TO-OSPF deny 10
 match ip address prefix-list BGP-ENTERPRISE-PREFIX
route-map REDIST-BGP-TO-OSPF deny 20
 match ip address prefix-list BGP-LOCALDC-PREFIX
route-map REDIST-BGP-TO-OSPF deny 30
 match ip address prefix-list DEFAULT-ROUTE
route-map REDIST-BGP-TO-OSPF permit 40
 description Modify Metric to Prefer MPLS over Internet
 set metric 2000
 set metric-type type-1
!
route-map BGP-INET-SPOKES-IN deny 10
 match ip address prefix-list DEFAULT-ROUTE
```

```
route-map BGP-INET-SPOKES-IN deny 20
 match ip address prefix-list BGP-ENTERPRISE-PREFIX
route-map BGP-INET-SPOKES-IN deny 30
 match ip address prefix-list BGP-LOCALDC-PREFIX
route-map BGP-INET-SPOKES-IN deny 40
 match ip address prefix-list BGP-LOCALMC
route-map BGP-INET-SPOKES-IN deny 50
 match ip address prefix-list TUNNEL-DMVPN
route-map BGP-INET-SPOKES-IN permit 60
description Allow Everything Else
```

Example 4-55 provides the BGP configuration for the BGP spoke routers.

Example 4-55 *IBGP Spoke Router Configuration*

```
R31-Spoke (Directly Attached Sites Only)
router bgp 10
 neighbor MPLS-HUB peer-group
 neighbor MPLS-HUB remote-as 10
 neighbor MPLS-HUB timers 20 60
 neighbor INET-HUB peer-group
 neighbor INET-HUB remote-as 10
 neighbor INET-HUB timers 20 60
 neighbor 192.168.100.11 peer-group MPLS-HUB
 neighbor 192.168.100.21 peer-group MPLS-HUB
 neighbor 192.168.200.12 peer-group INET-HUB
 neighbor 192.168.200.22 peer-group INET-HUB
 !
 address-family ipv4
  redistribute connected route-map REDIST-CONNECTED-TO-BGP
  neighbor MPLS-HUB send-community
  neighbor MPLS-HUB next-hop-self all
  neighbor MPLS-HUB weight 50000
  neighbor MPLS-HUB soft-reconfiguration inbound
  neighbor INET-HUB send-community
  neighbor INET-HUB next-hop-self all
  neighbor INET-HUB weight 50000
  neighbor INET-HUB soft-reconfiguration inbound
  neighbor 192.168.100.11 activate
  neighbor 192.168.100.21 activate
  neighbor 192.168.200.12 activate
  neighbor 192.168.200.22 activate
  distance bgp 201 19 19
 exit-address-family
!
```

```
route-map REDIST-CONNECTED-TO-BGP deny 10
 description Block redistribution of DMVPN Tunnel Interfaces
 match interface Tunnel100 Tunnel200
route-map REDIST-CONNECTED-TO-BGP permit 20
 description Redistribute all other prefixes
```

R41-Spoke (Multiple Routers – Downstream Only)

```
router ospf 1
 redistribute bgp 10 subnets route-map REDIST-BGP-TO-OSPF
 passive-interface default
 no passive-interface GigabitEthernet0/3
 no passive-interface GigabitEthernet1/0
 network 10.0.0.0 0.255.255.255 area 0
 default-information originate
!
router bgp 10
 neighbor MPLS-HUB peer-group
 neighbor MPLS-HUB remote-as 10
 neighbor MPLS-HUB timers 20 60
 neighbor INET-HUB peer-group
 neighbor INET-HUB remote-as 10
 neighbor INET-HUB timers 20 60
 neighbor 192.168.100.11 peer-group MPLS-HUB
 neighbor 192.168.100.21 peer-group MPLS-HUB
 neighbor 192.168.200.12 peer-group INET-HUB
 neighbor 192.168.200.22 peer-group INET-HUB
 !
address-family ipv4
  bgp redistribute-internal
  redistribute ospf 1 route-map REDIST-OSPF-TO-BGP
  neighbor MPLS-HUB send-community
  neighbor MPLS-HUB next-hop-self all
  neighbor MPLS-HUB weight 50000
  neighbor MPLS-HUB soft-reconfiguration inbound
  neighbor INET-HUB send-community
  neighbor INET-HUB next-hop-self all
  neighbor INET-HUB weight 50000
  neighbor INET-HUB soft-reconfiguration inbound
  neighbor 192.168.100.11 activate
  neighbor 192.168.100.21 activate
  neighbor 192.168.200.12 activate
  neighbor 192.168.200.22 activate
  distance bgp 201 19 19
 exit-address-family
```

```
!
ip prefix-list TUNNEL-DMVPN seq 10 permit 192.168.100.0/24
ip prefix-list TUNNEL-DMVPN seq 20 permit 192.168.200.0/24
!
route-map REDIST-BGP-TO-OSPF permit 10
 description Set a route tag to identify routes redistributed from BGP
 set tag 1
!
route-map REDIST-OSPF-TO-BGP deny 10
 description Block all routes redistributed from BGP
 match tag 1
route-map REDIST-OSPF-TO-BGP deny 15
 match ip address prefix-list TUNNEL-DMVPN
route-map REDIST-OSPF-TO-BGP permit 30
 description Redistribute all other traffic
 match route-type internal
 match route-type external type-1
 match route-type external type-2
```

R51-Spoke (Multiple Routers – Multiple Transport)

```
router ospf 1
 redistribute bgp 10 subnets route-map REDIST-BGP-TO-OSPF
 passive-interface default
 no passive-interface GigabitEthernet0/3
 no passive-interface GigabitEthernet1/0
 network 10.0.0.0 0.255.255.255 area 0
 default-information originate
!
router bgp 10
 bgp log-neighbor-changes
 neighbor MPLS-HUB peer-group
 neighbor MPLS-HUB remote-as 10
 neighbor MPLS-HUB timers 20 60
 neighbor 192.168.100.11 peer-group MPLS-HUB
 neighbor 192.168.100.21 peer-group MPLS-HUB
 !
 address-family ipv4
  bgp redistribute-internal
  redistribute ospf 1 route-map REDIST-OSPF-TO-BGP
  neighbor MPLS-HUB send-community
  neighbor MPLS-HUB weight 50000
  neighbor MPLS-HUB soft-reconfiguration inbound
  neighbor 192.168.100.11 activate
  neighbor 192.168.100.21 activate
  distance bgp 201 19 19
 exit-address-family
```

```
!
ip prefix-list TUNNEL-DMVPN seq 10 permit 192.168.100.0/24
ip prefix-list TUNNEL-DMVPN seq 20 permit 192.168.200.0/24
!
route-map REDIST-BGP-TO-OSPF permit 10
 description Set a route tag to identify routes redistributed from BGP
 set tag 1
!
route-map REDIST-OSPF-TO-BGP deny 10
 description Block all routes redistributed from BGP
 match tag 1
route-map REDIST-OSPF-TO-BGP deny 15
 match ip address prefix-list TUNNEL-DMVPN
route-map REDIST-OSPF-TO-BGP permit 30
 description Redistribute all other traffic
 match route-type internal
 match route-type external type-1
 match route-type external type-2
```

R52-Spoke (Multiple Routers – Multiple Transport)

```
router ospf 1
 redistribute bgp 10 subnets route-map REDIST-BGP-TO-OSPF
 passive-interface default
 no passive-interface GigabitEthernet0/3
 no passive-interface GigabitEthernet1/0
 network 10.0.0.0 0.255.255.255 area 0
default-information originate
!
router bgp 10
 bgp log-neighbor-changes
 neighbor INET-HUB peer-group
 neighbor INET-HUB remote-as 10
 neighbor INET-HUB timers 20 60
 neighbor 192.168.200.12 peer-group INET-HUB
 neighbor 192.168.200.22 peer-group INET-HUB
 !
 address-family ipv4
  bgp redistribute-internal
  redistribute ospf 1 route-map REDIST-OSPF-TO-BGP
  neighbor INET-HUB send-community
  neighbor INET-HUB weight 50000
  neighbor INET-HUB soft-reconfiguration inbound
  neighbor 192.168.200.12 activate
  neighbor 192.168.200.22 activate
  distance bgp 201 19 19
 exit-address-family
!
ip prefix-list TUNNEL-DMVPN seq 10 permit 192.168.100.0/24
ip prefix-list TUNNEL-DMVPN seq 20 permit 192.168.200.0/24
!
route-map REDIST-BGP-TO-OSPF permit 10
 description Set a route tag to identify routes redistributed from BGP
 set tag 1
!
route-map REDIST-OSPF-TO-BGP deny 10
 description Block all routes redistributed from BGP
 match tag 1
route-map REDIST-OSPF-TO-BGP deny 15
 match ip address prefix-list TUNNEL-DMVPN
route-map REDIST-OSPF-TO-BGP permit 30
 description Redistribute all other traffic
 match route-type internal
 match route-type external type-1
 match route-type external type-2
```

Advanced BGP Site Selection

For scenarios that require a spoke router to prefer one site over another site, the hub routers include a BGP community with all the network prefixes. The spoke routers then change their routing policy based upon the BGP community to override the local preference that was advertised with the network prefix.

Assume that a spoke router should prefer Site 2-MPLS (R21), then Site 2-Internet (R22), then Site 1-MPLS (R11), then Site 1-Internet (R12). Example 4-56 demonstrates how to set the BGP community on the outbound route map.

Example 4-56 *Configuration for Setting BGP Communities on Prefix Advertisement*

```
R11
route-map BGP-MPLS-SPOKES-OUT permit 10
 match ip address prefix-list DEFAULT-ROUTE
 set local-preference 100000
 set community 10:11
route-map BGP-MPLS-SPOKES-OUT permit 20
 match ip address prefix-list BGP-ENTERPRISE-PREFIX
 set local-preference 100000
 set community 10:11
route-map BGP-MPLS-SPOKES-OUT permit 30
 match ip address prefix-list BGP-LOCALDC-PREFIX
 set local-preference 100000
 set community 10:11
```

```
route-map BGP-MPLS-SPOKES-OUT permit 40
 match ip address prefix-list BGP-LOCALMC
 set local-preference 100000
 set community 10:11
```

R12

```
route-map BGP-INET-SPOKES-OUT permit 10
 match ip address prefix-list DEFAULT-ROUTE
 set local-preference 3000
 set community 10:12
route-map BGP-INET-SPOKES-OUT permit 20
 match ip address prefix-list BGP-ENTERPRISE-PREFIX
 set local-preference 3000
 set community 10:12
route-map BGP-INET-SPOKES-OUT permit 30
 match ip address prefix-list BGP-LOCALDC-PREFIX
 set local-preference 3000
 set community 10:12
route-map BGP-INET-SPOKES-OUT permit 40
 match ip address prefix-list BGP-LOCALMC
 set local-preference 3000
 set community 10:12
```

R21

```
route-map BGP-MPLS-SPOKES-OUT permit 10
 match ip address prefix-list DEFAULT-ROUTE
 set local-preference 20000
 set community 10:21
route-map BGP-MPLS-SPOKES-OUT permit 20
 match ip address prefix-list BGP-ENTERPRISE-PREFIX
 set local-preference 20000
 set community 10:21
route-map BGP-MPLS-SPOKES-OUT permit 30
 match ip address prefix-list BGP-LOCALDC-PREFIX
 set local-preference 20000
 set community 10:21
route-map BGP-MPLS-SPOKES-OUT permit 40
 match ip address prefix-list BGP-LOCALMC
 set local-preference 20000
 set community 10:21
```

R22

```
route-map BGP-INET-SPOKES-OUT permit 10
 match ip address prefix-list DEFAULT-ROUTE
 set local-preference 400
 set community 10:22
route-map BGP-INET-SPOKES-OUT permit 20
 match ip address prefix-list BGP-ENTERPRISE-PREFIX
 set local-preference 400
 set community 10:22
route-map BGP-INET-SPOKES-OUT permit 30
 match ip address prefix-list BGP-LOCALDC-PREFIX
 set local-preference 400
 set community 10:22
route-map BGP-INET-SPOKES-OUT permit 40
 match ip address prefix-list BGP-LOCALMC
 set local-preference 400
set community 10:22
```

In order to set a preferential site, the local preference is increased on the paths with the desired tag. A different local preference is used to differentiate between the MPLS and Internet transport. The route map contains four sequences:

1. Match routes that have the hub's BGP community from the primary transport in the primary site. The local preference is set to 123,456, which exceeds the highest local preference for the primary transport.
2. Match routes that have the hub's BGP community from the secondary transport in the primary site. The local preference is set to 23,456, which exceeds the highest local preference for the secondary transport.
3. Match routes that have the hub's BGP community from the primary transport in the secondary site. The local preference is set to 3456.
4. Allow all other routes to pass.

Example 4-57 provides the configuration for preferring routes in Site 1 or Site 2.

Example 4-57 *BGP Configuration for Hub Preference*

```
Prefer Site1
router bgp 10
 address-family ipv4
  neighbor MPLS-HUB route-map BGP-DEFAULT-ROUTE-PREFER-SITE1 in
  neighbor INET-HUB route-map BGP-DEFAULT-ROUTE-PREFER-SITE1 in
!
route-map BGP-DEFAULT-ROUTE-PREFER-SITE1 permit 10
 match community R11
 set local-preference 123456
```

```
route-map BGP-DEFAULT-ROUTE-PREFER-SITE1 permit 20
 match community R12
 set local-preference 23456
route-map BGP-DEFAULT-ROUTE-PREFER-SITE1 permit 30
 match community R12
 set local-preference 3456
route-map BGP-DEFAULT-ROUTE-PREFER-SITE1 permit 40
!
ip community-list standard R11 permit 10:11
ip community-list standard R12 permit 10:12
ip community-list standard R21 permit 10:21
```

Prefer Site2

```
router bgp 10
 address-family ipv4
  neighbor MPLS-HUB route-map BGP-DEFAULT-ROUTE-PREFER-SITE2 in
  neighbor INET-HUB route-map BGP-DEFAULT-ROUTE-PREFER-SITE2 in
!
route-map BGP-DEFAULT-ROUTE-PREFER-SITE2 permit 10
 match community R21
 set local-preference 123456
route-map BGP-DEFAULT-ROUTE-PREFER-SITE2 permit 20
 match community R22
 set local-preference 23456
route-map BGP-DEFAULT-ROUTE-PREFER-SITE2 permit 30
 match community R11
 set local-preference 3456
route-map BGP-DEFAULT-ROUTE-PREFER-SITE2 permit 40
!
ip community-list standard R11 permit 10:11
ip community-list standard R21 permit 10:21
ip community-list standard R22 permit 10:22
```

Note Matching on BGP communities requires defining a standard community list.

Example 4-58 verifies that R21 (MPLS primary site) is identified as the best path. Examining the local preference in the BGP table verifies that the DMVPN hubs will be preferred in the desired order.

Example 4-58 *Verification of BGP Path Preference*

```
R31-Spoke# show ip route
! Output omitted for brevity
Gateway of last resort is 192.168.100.21 to network 0.0.0.0

B*    0.0.0.0/0 [19/1] via 192.168.100.21, 00:01:44
```

```
R31-Spoke# show bgp ipv4 unicast
! Output omitted for brevity
     Network          Next Hop            Metric LocPrf Weight Path
 * i 0.0.0.0          192.168.200.12           1   3000  50000 i
 * i                  192.168.200.22           1  23456  50000 i
 * i                  192.168.100.11           1   3456  50000 i
 *>i                  192.168.100.21           1 123456  50000 i
```

FVRF Transport Routing

In Chapter 3, a simple fully specified default static route was used in the FVRF to provide connectivity between the DMVPN encapsulating interfaces. Some scenarios may require additional routing configuration in the FVRF scenario.

These designs incorporate routing in the FVRF context. Most of the routing protocol configuration is exactly the same, with the exception that a VRF is associated in the routing protocol configuration.

The most common use case is that the SP needs to establish a BGP session with the CE device as part of its capability to monitor the circuit. In this situation, MBGP is used. The peering with the SP is established under the VRF address family specifically for FVRF with the command **address-family ipv4 vrf** *vrf-name. This* places the router into the VRF address family configuration submode where the neighbor session is defined, and the networks are advertised into the FVRF.

Note A *route distinguisher (RD)* must be defined in the VRF configuration to enter into the VRF address family configuration mode. The command **rd** *asn:rd-identifier* configures the RD in the VRF definition.

If the IWAN design uses BGP, and the ASN for the routing architecture is different from what the SP uses, a change request needs to be submitted to the SP, which may take a long time to process. An alternative is to use the **neighbor** *ip-address* **local-as** *sp-peering-asn* **no-prepend replace-as dual-as** command, which keeps the original ASN from the BGP process but lets the SP peer as if the router were using the existing ASN. Example 4-59 demonstrates the configuration of R41 to peer with the SP BGP router (which expects it to be ASN 41).

Example 4-59 *MBGP VRF Address Family Configuration for the FVRF Network*

```
R41
vrf definition INET01
 rd 10:41
 address-family ipv4
 exit-address-family
!
router bgp 10
 address-family ipv4 vrf INET01
  redistribute connected
  neighbor 172.16.41.1 remote-as 65000
  neighbor 172.16.41.1 local-as 41 no-prepend replace-as dual-as
  neighbor 172.16.41.1 activate
 exit-address-family
```

Multicast Routing

Multicast communication is a technology that optimizes network bandwidth utilization and allows for one-to-many or many-to-many communication. Only one data packet is sent on a link as needed and is replicated as the data forks (splits) on a network device. IP multicast is much more efficient than multiple individual unicast streams or a broadcast stream that propagates everywhere.

Multicast packets are referred to as a *stream* that uses a special multicast *group address*. Multicast group addresses are in the IP address range from 224.0.0.0 to 239.255.255.255. Clients of the multicast stream are called *receivers*.

Note Multicast routing is a significant topic unto itself. The focus of this section is to provide basic design guidelines for multicast in the IWAN architecture.

Multicast Distribution Trees

A multicast router creates a distribution tree to define the path for the stream to reach the receivers. Two types of multicast distribution trees are source shortest path trees and shared trees.

Source Trees

A source tree is a multicast tree where the root is the source of the tree and the branches form a distribution tree through the network all the way down to the receivers. When this tree is built, it uses the path calculated by the network from the leaves to the source of the tree. The leaves use the routing table to locate the source. This is the reason why it is also referred to as a *shortest path tree (SPT)*. The forwarding state of the SPT uses the notation (*S*,*G*). *S* is the source of the multicast stream (server) and *G* is the multicast group address.

Shared Trees

A shared tree is a multicast tree where the root of the shared tree is a router designated as the *rendezvous point (RP)*. Multicast traffic is forwarded down the shared tree according to the group address *G* to which the packets are addressed, regardless of the source address. The forwarding state on the shared tree uses the notation (*,*G*).

Rendezvous Points

An RP is a single common root placed at a chosen point on a shared distribution tree as described in the previous sections in this chapter. An RP can be either configured statically in each router or learned through a dynamic mechanism.

Protocol Independent Multicast (PIM)

A multicast routing protocol is necessary to route multicast traffic throughout the network so that routers can locate and request multicast streams from other routers. *Protocol Independent Multicast (PIM)* is a multicast routing protocol that routes multicast traffic between network segments. PIM uses any of the unicast routing protocols to identify the path between the source and receivers.

PIM sparse mode (PIM SM) uses the unicast routing table to perform *reverse path forwarding (RPF)* checks and does not care which routing protocol (including static routes) populates the unicast routing table; therefore, it is protocol independent. The RPF interface is the interface with the path selected by the unicast routing protocols toward the IP address of the source or the RP.

PIM sparse mode uses an explicit join model where upon receipt of an IGMP Join, the IGMP Join is converted into a PIM Join. The PIM Join is sent to the root of the tree, either the RP for shared trees or the router attached to the multicast source for an SPT tree.

Then the multicast stream transmits from the source to the RP and from the RP to the receiver's router and finally to the receiver. This is a simplified view of how PIM SM achieves multicast forwarding.

Source Specific Multicast (SSM)

In earlier and traditional PIM sparse mode/dense mode (DM) networks, receivers use IGMPv2 Joins to signal that they would like to receive multicast traffic from a specific multicast group (*G*). The IGMPv2 Joins include only the multicast group *G* the receiver wants to join, but they do not specify the source (*S*) for the multicast traffic. Because the source is unknown to the receiver, the receiver can accept traffic from any source transmitting to the group. This type of multicast service model is known as *Any Source Multicast (ASM)*.

One of the problems with ASM is that it is possible for a receiver to receive multicast traffic from different sources transmitting to the same group. Even though the application on the receivers typically can filter out the unwanted traffic, network bandwidth and resources are wasted.

PIM SSM provides granularity and allows clients to specify the source of a multicast stream. SSM operates in conjunction with IGMPv3 and requires IGMPv3 support on the multicast routers, the receiver where the application is running, and the application itself.

As one of the operating modes of PIM, IGMPv3 membership reports (joins) allow a receiver to specify the source *S* and the group *G* from which it would like to receive multicast traffic. Because the IGMPv3 join includes the (*S*,*G*), referred to as a *channel* in SSM, the designated router (DR) builds a source tree (SPT) by sending an (*S*,*G*) PIM Join directly to the source. SSM is source tree based, so RPs are not required.

Note The *Internet Assigned Number Authority (IANA)* assigned the 232.0.0.0/8 multicast range to SSM for default use. SSM is allowed to use any other multicast group in the 224.0.0.0/4 multicast range as long as it is not reserved.

Unless explicitly stated, this chapter discusses multicast in the context of ASM.

Multicast Routing Table

The logic for multicast routing of traffic varies from that of unicast routing. A router forwards packets away from the source down the distribution tree. A router organizes the multicast forwarding table based on the reverse path (receivers to the root of the distribution tree).

To avoid routing loops, incoming packets are accepted only if the outgoing interface matches the interface pointed toward the source of the packet. The process of checking the inbound interface to the source of the packet is known as *RPF check*. The multicast routing table is used for RPF checking.

Note If a multicast packet fails the RPF check, the multicast packet is dropped.

The unicast routing table is assembled from the unicast routing protocol databases. The multicast routing table is blank by default. In the event that a route cannot be matched in the multicast routing table, it is looked up in the unicast routing table.

IWAN Multicast Configuration

The following steps are used to configure multicast routing for all routers in the environment:

Step 1. **Enable multicast support for NHRP.**

NHRP provides a mapping service of the protocol (tunnel IP) address to the NBMA (that is, transport) address for unicast packets. The same capability is required for multicast traffic. DMVPN hub routers enable multicast NHRP support with the tunnel command **ip nhrp map multicast dynamic**.

On the DMVPN spoke routers, the **multicast** keyword is necessary to enable multicast NHRP functions when using the command **ip nhrp nhs** *nhs-address* **nbma** *nbma-address* **multicast**.

Step 2. **Enable multicast routing.**

Multicast routing is enabled with the global command **ip multicast-routing**.

Step 3. **Enable PIM and IGMP.**

PIM and IGMPv2 are enabled by entering the command **ip pim sparse-mode** on all participating LAN and DMVPN tunnel interfaces (including the ones facing the receivers). Enabling PIM SM on receiver-facing interfaces enables IGMPv2 by default.

Step 4. **Configure an RP.**

The RP is a control plane operation that should be placed in the core of the network or close to the multicast sources on a pair of routers. Proper multicast design provides multiple RPs for redundancy purposes. This book uses static RP assignment with an Anycast RP address that resides on two DMVPN hub routers.

The RP is statically assigned with the command **ip pim rp-address** *ip-address*.

Step 5. **Enable PIM NBMA mode on hub DMVPN tunnel interfaces.**

By default, all PIM messages are transmitted between the spoke and the hub. Spokes do not see each other's PIM messages. This can cause problems when a router (R11) is forwarding multicast traffic to multiple spokes (R31, R41, and R51). This is fine as long as R31, R41, and R51 have multicast subscribers; they all build multicast trees across R11. At R11, all the trees converge to a single link.

When a spoke router (R31) stops the stream with a PIM Prune message to R11, R11 does not receive a PIM Prune override message from R41 or R51 and declares the tunnel free of any multicast receivers. R11 stops and prunes the stream state, leaving R41 and R51 without multicast traffic. Neither R41 nor R51 sees the PIM Prune message so that they can override it.

PIM NBMA mode treats every spoke connection as its own link, preventing scenarios like this from happening. PIM NBMA mode is enabled with the tunnel command **ip pim nbma-mode.**

Step 6. **Disable PIM designated router functions on spoke routers.**

The PIM designated router resides on a multi-access link and registers active sources with the RP. It is essential for the PIM DR to receive and send packets to all routers on the multi-access link. Considering that multicast traffic works only spoke to hub, it is essential that only the hubs be PIM DRs. A spoke router can be removed from the PIM DR election by setting its priority to zero on DMVPN tunnel interfaces with the command **ip pim dr-priority 0**.

Example 4-60 displays R11's multicast configuration as a reference configuration for the other DMVPN hub routers. Notice that the LAN interface GigabitEthernet 1/0 has PIM enabled on it.

Example 4-60 *R11 Multicast Configuration*

```
R11-Hub
ip multicast-routing
!
interface Loopback0
 ip pim sparse-mode
interface Tunnel100
 description DMVPN Tunnel
 ip pim nbma-mode
 ip pim sparse-mode
 ip nhrp map multicast dynamic
interface GigabitEthernet0/1
 description MPLS Transport transport
interface GigabitEthernet0/3
 description Cross-Link to R12
 ip pim sparse-mode
interface GigabitEthernet1/0
 description LAN interface
 ip pim sparse-mode
!
ip pim rp-address 192.168.1.1
```

Example 4-61 displays R31's multicast configuration as a reference configuration for the other DMVPN hub routers.

Example 4-61 *R31 Multicast Configuration*

```
R31
ip multicast-routing
!
interface Loopback0
 ip pim sparse-mode
interface Tunnel100
 description DMVPN Tunnel for MPLS transport
 ip pim sparse-mode
 ip pim dr-priority 0
 ip nhrp nhs 192.168.100.11 nbma 172.16.11.1 multicast
 ip nhrp nhs 192.168.100.21 nbma 172.16.21.1 multicast
interface Tunnel200
 description DMVPN Tunnel for Internet transport
 ip pim sparse-mode
 ip pim dr-priority 0
 ip nhrp nhs 192.168.200.12 nbma 100.64.12.1 multicast
 ip nhrp nhs 192.168.200.22 nbma 100.64.22.1 multicast
interface GigabitEthernet1/0
 description LAN interface
 ip pim sparse-mode
!
ip pim rp-address 192.168.1.1
```

Example 4-62 displays the PIM interfaces with the command **show ip pim interface** and verifies PIM neighbors with the command **show ip pim neighbor.** Notice that the spoke PIM neighbors have a DR priority of zero.

Example 4-62 *Verification of PIM Interfaces and Neighbors*

```
R11-Hub# show ip pim interface
Address          Interface          Ver/   Nbr    Query  DR         DR
                                    Mode   Count  Intvl  Prior
10.1.111.11      GigabitEthernet1/0 v2/S   1      30     1       10.1.111.11
192.168.100.11   Tunnel100          v2/S   3      30     1       192.168.100.11
10.1.12.11       GigabitEthernet0/3 v2/S   1      30     1       10.1.12.11
10.1.0.11        Loopback0          v2/S   0      30     1       10.1.0.11
```

```
R11-Hub# show ip pim neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      P - Proxy Capable, S - State Refresh Capable, G - GenID Capable,
      L - DR Load-balancing Capable
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.111.13       GigabitEthernet1/0       18:24:55/00:01:20 v2    1 / S P G
10.1.12.12        GigabitEthernet0/3       18:24:15/00:01:20 v2    1 / S P G
192.168.100.51    Tunnel100                18:23:40/00:01:36 v2    0 / S P G
192.168.100.41    Tunnel100                18:23:42/00:01:21 v2    0 / S P G
192.168.100.31    Tunnel100                18:23:53/00:01:17 v2    0 / S P G
```

Hub-to-Spoke Multicast Stream

This section describes the behaviors of multicast traffic flowing from a hub to a spoke router. The most significant change in behavior with multicast traffic and DMVPN is that multicast traffic flows only between the devices that use the multicast NHRP map statements. This means that multicast traffic always travels through the hub and does not travel across a spoke-to-spoke tunnel.

Figure 4-8 displays a multicast server (10.1.1.1) transmitting a multicast video stream to the group address of 225.1.1.1. R31 has a client (10.3.3.3) that wants to watch the video stream.

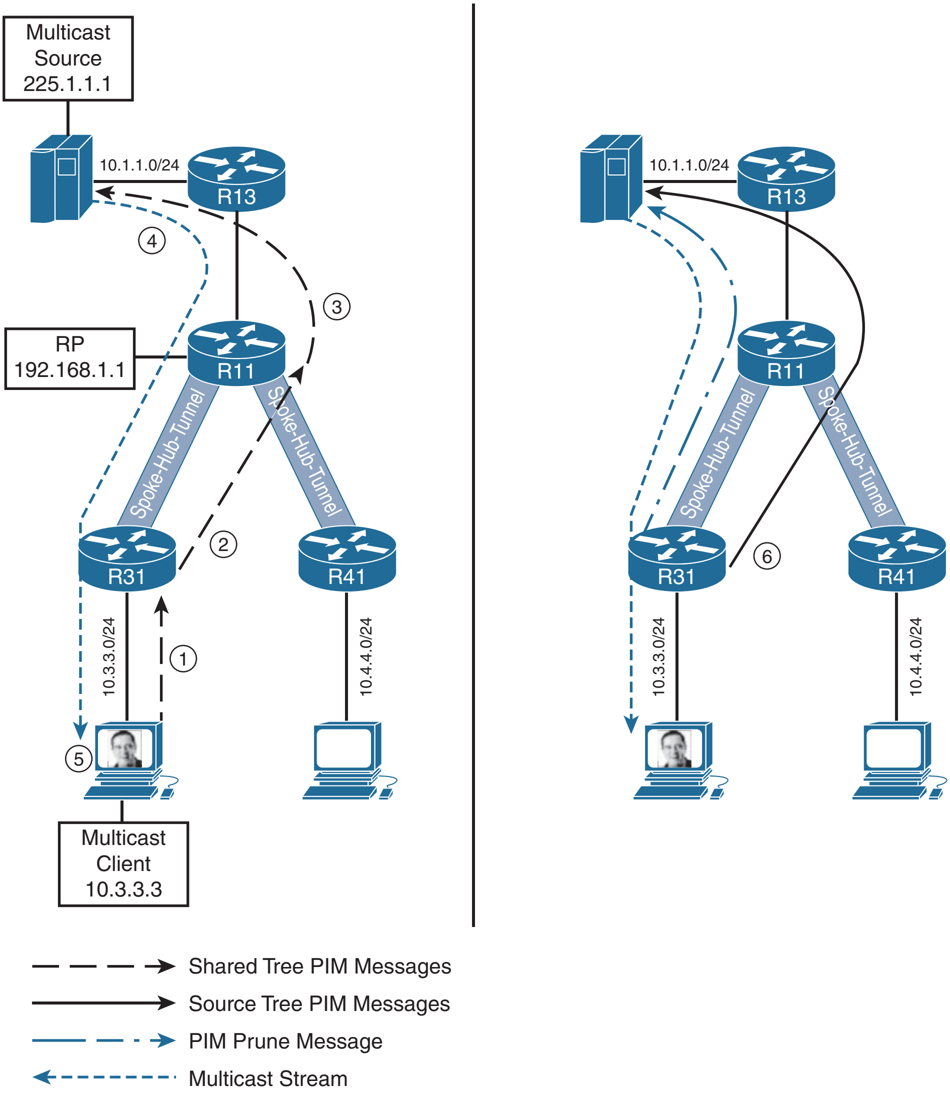


Figure 4-8 *Hub-to-Spoke Multicast Stream*

The following events correspond to Figure 4-8 when a receiver subscribes to a multicast stream:

1. The receiver (10.3.3.3) attached to R31 sends an IGMP Join for the group address 225.1.1.1.
2. R31 creates an entry in the multicast routing table for 225.1.1.1 and identifies the RP (R11) for this group. R31 then performs an RPF check for the RP's address

(192.168.1.1) and resolves 192.168.100.11 as the RPF neighbor. The PIM Join is sent on the shared tree to the PIM neighbor R11 (192.168.100.11) via the tunnel interface, where it is processed on R11.

3. R11 has shared tree (*, 225.1.1.1) and source tree (10.1.1.1, 225.1.1.1) entries in the multicast routing table, both of which are pruned because there are no multicast clients to the 225.1.1.1 stream at this time. R11 registers the PIM Join, removes the prune entries on the shared tree and source tree entries, and resets the z-flag (multicast tunnel). R11 sends a PIM Join to R13 for the 225.1.1.1 stream.

4. R13 removes the prune on its shared tree (*, 225.1.1.1) and starts to forward packets toward R11. R11 then forwards the packets to R31, which then forwards the packets to the 10.3.3.0/24 LAN segment.

5. The receiver displays the video stream on a PC.

6. As soon as R31 receives a multicast packet on the shared tree (*, 225.1.1.1), it attempts to optimize the multicast path because it is directly attached to a receiver. R31 sends a PIM Join message to the source tree (10.1.1.1, 225.1.1.1). In order to prevent packets from being sent from both streams, R31 sends a PIM Prune message for the shared tree. At this time, both trees use the DMVPN hub as the next hop and have the same RPF neighbor.

Note The scenario described here assumes that there are no active multicast clients to the 225.1.1.1 stream. If there were other active clients (such as R51), the shared tree (*, 225.1.1.1) and source tree (10.1.1.1, 225.1.1.1) would not be pruned on R11. R11 would use the source tree for forwarding decisions. R31 would use the shared tree initially until optimized with the source tree.

The command **show ip mroute** [*group-address*] displays the multicast routing table as shown in Example 4-63. The first entry in the multicast routing table is for the shared path tree (*,225.1.1.1). The use of the asterisk (*) for the source placement indicates any source belonging to that group address. This entry represents the shared tree, which is the path on which multicast data arrives initially from a source. Notice that the source tree has an incoming interface identified and has the '*T*' flag set, whereas the shared path tree does not have an incoming interface and has the '*P*' flag set for pruning.

The second entry (10.1.1.1, 255.1.1.1) displays the source tree for the multicast stream 225.1.1.1 from the source of 10.1.1.1.

Example 4-63 *R13's Multicast Routing Table for 225.1.1.1*

```
R13# show ip mroute 225.1.1.1
! Output omitted for brevity
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
Outgoing interface flags: H - Hardware switched, A - Assert winner, p - PIM Join
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 225.1.1.1), 01:00:29/00:01:05, RP 192.168.1.1, flags: SPF
  Incoming interface: GigabitEthernet1/0, RPF nbr 10.1.111.11
  Outgoing interface list: Null

(10.1.1.1, 225.1.1.1), 01:00:29/00:02:49, flags: FT
  Incoming interface: GigabitEthernet0/0, RPF nbr 10.1.1.1
  Outgoing interface list:
    GigabitEthernet1/0, Forward/Sparse, 00:01:49/00:02:50
```

Note An active multicast stream has only one incoming interface and one or more outgoing interfaces.

Example 4-64 displays the multicast routing table for 225.1.1.1 on R31 and R11. Notice that R31 has the '*C*' flag that indicates a receiver is directly attached to it.

Example 4-64 *R11's and R31's Multicast Routing Table for 255.1.1.1*

```
R31-Spoke# show ip mroute 225.1.1.1
! Output omitted for brevity

(*, 225.1.1.1), 00:11:46/stopped, RP 192.168.1.1, flags: SJC
  Incoming interface: Tunnel100, RPF nbr 192.168.100.11
  Outgoing interface list:
    GigabitEthernet1/0, Forward/Sparse, 00:08:47/00:02:32

(10.1.1.1, 225.1.1.1), 00:02:20/00:00:39, flags: JT
  Incoming interface: Tunnel100, RPF nbr 192.168.100.11
  Outgoing interface list:
    GigabitEthernet1/0, Forward/Sparse, 00:02:20/00:02:32
```

```
R11-Hub# show ip mroute 225.1.1.1
! Output omitted for brevity
(*, 225.1.1.1), 01:08:33/00:02:49, RP 192.168.1.1, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Tunnel100, 192.168.100.31, Forward/Sparse, 00:07:32/00:02:49

(10.1.1.1, 225.1.1.1), 00:10:53/00:03:20, flags: TA
  Incoming interface: GigabitEthernet1/0, RPF nbr 10.1.111.10
  Outgoing interface list:
    Tunnel100, 192.168.100.31, Forward/Sparse, 00:07:32/00:03:23
```

Spoke-to-Spoke Multicast Traffic

A router's multicast forwarding behavior operates normally when the source is located behind the hub, but it causes issues when the source is located behind a spoke router, specifically when the receiver is behind a different spoke router and a spoke-to-spoke tunnel exists between the two routers.

Figure 4-9 displays a multicast server (10.4.4.4) transmitting a multicast video stream to the group address of 225.4.4.4. R31 has a client (10.3.3.3) that wants to watch the video stream.

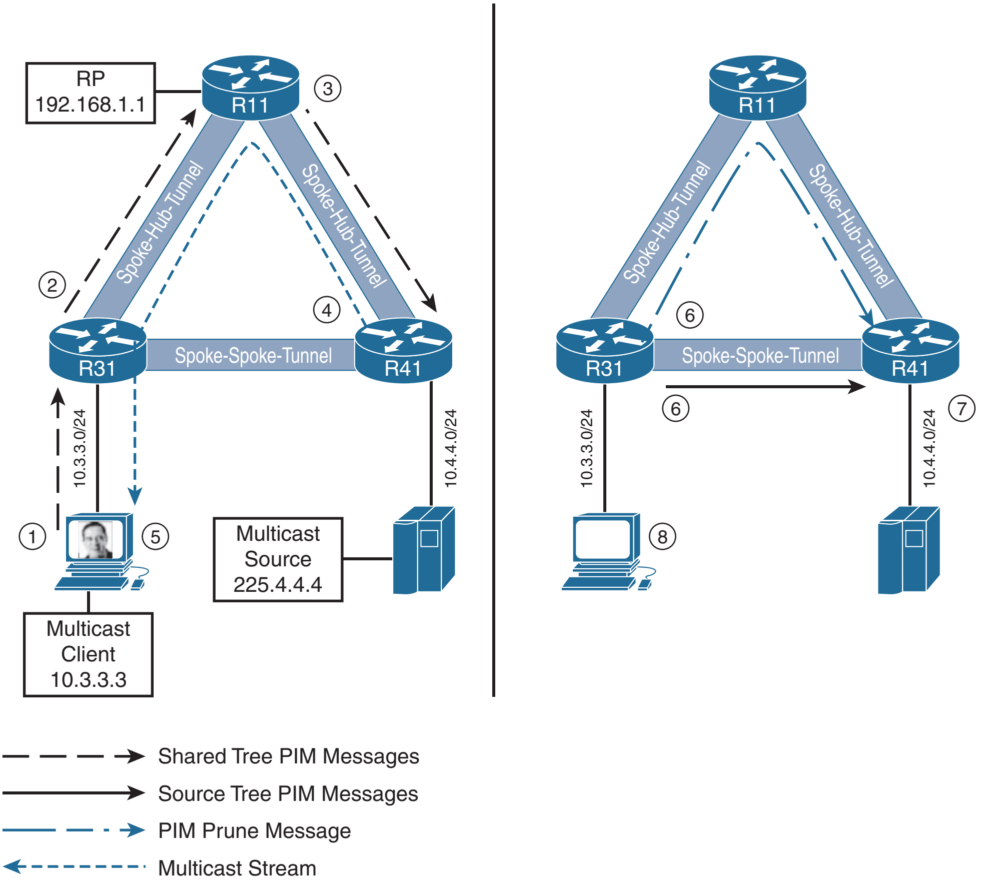


Figure 4-9 *Spoke-to-Spoke Multicast Stream*

The following events correspond to Figure 4-9 when a receiver subscribes to a multicast stream:

1. The receiver (10.3.3.3) attached to R31 sends an IGMP Join for the group address 225.4.4.4.
2. R31 creates an entry in the multicast routing table for 225.4.4.4 and identifies the RP (R11) for this group. R31 then performs an RPF check for the RP's address

(192.168.1.1) and resolves 192.168.100.11 as the RPF neighbor. The PIM Join is sent on the shared tree (*, 225.4.4.4) to the PIM neighbor R11 (192.168.100.11) via the tunnel interface, where it is processed on R11.

3. R11 has shared tree (*, 225.4.4.4) and source tree (10.4.4.4, 225.4.4.4) entries in the multicast routing table, both of which are pruned. R11 removes the prune entry on the shared tree and sends a PIM Join to R41 for the 225.4.4.4 stream.

Note Just as in the previous scenario, there are no active multicast clients to 225.4.4.4. If there were other active clients (such as R13), the shared tree (*, 225.4.4.4) and source tree (10.4.4.4, 225.4.4.4) would not be pruned on R11. Step 4 would be skipped and R11 would forward packets toward R31 using the source tree for forwarding decisions.

4. R41 removes the prune on its shared tree and starts to forward packets toward R11, which are then forwarded to R31 and then forwarded to the 10.3.3.0/24 LAN segment.

5. The receiver connected to R31 displays the video stream on a PC.

6. As soon as R31 receives a multicast packet on the shared tree, it attempts to optimize the multicast path because a receiver is attached to it.

   A spoke-to-spoke tunnel is established between R31 and R41, and NHRP injects a route for 10.4.4.0/24 with a next hop of 192.168.100.41 on R31.

   R31 tries to send a PIM Join message to the source tree via the spoke-to-spoke tunnel. R31 and R41 cannot directly exchange PIM messages or become PIM neighbors with each other. At the same time that R31 tries to send a PIM Join message to the source tree, it sends a PIM Prune message for the shared tree to prevent duplicate packets.

Note Multicast packets travel across the DMVPN network only where there is an explicit NHRP mapping that correlates to the spoke-to-hub configuration. PIM messages operate using the multicast group 224.0.0.13.

7. The PIM Prune message succeeds because it was sent through the hub router, and the PIM Join message fails because multicast traffic does not travel across spoke-to-spoke tunnels. R41 receives the PIM Prune message on the shared tree and stops sending multicast packets on the shared tree toward R11.

   From R41's perspective, R31 does not want to receive the multicast stream anymore and has stopped sending the multicast stream down the shared tree.

8. The receiver stops displaying the video stream on the PC.

Example 4-65 displays R31's unicast routing table. Notice that the 10.4.4.0/24 network was installed by NHRP. A more explicit route for the 10.4.4.4 source does not exist in the multicast routing table, so the NHRP entry from the unicast routing table is used. Notice that the 10.4.4.0/24 entry indicates the spoke-to-spoke tunnel for multicast traffic to and from R41.

Example 4-65 *R31's Routing Table and DMVPN Tunnels*

```
R31-Spoke# show ip route
! Output omitted for brevity
B*    0.0.0.0/0 [19/1] via 192.168.100.11, 00:10:10
      10.0.0.0/8 is variably subnetted, 7 subnets, 4 masks
B        10.0.0.0/8 [19/0] via 192.168.100.11, 00:10:10
B        10.1.0.0/16 [19/0] via 192.168.100.11, 00:10:10
B        10.2.0.0/16 [19/0] via 192.168.100.21, 00:10:10
C        10.3.0.31/32 is directly connected, Loopback0
C        10.3.3.0/24 is directly connected, GigabitEthernet1/0
H        10.4.4.0/24 [250/255] via 192.168.100.41, 00:03:44, Tunnel100
```

```
R31-Spoke# show dmvpn detail
! Output omitted for brevity
# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1 172.16.31.1      192.168.100.31   IKE 00:04:51   DLX        10.3.3.0/24
    2 172.16.41.1      192.168.100.41    UP 00:04:51   DT1        10.4.4.0/24
      172.16.41.1      192.168.100.41    UP 00:04:51   DT1  192.168.100.41/32
    1 172.16.11.1      192.168.100.11    UP 00:11:32     S  192.168.100.11/32
    1 172.16.21.1      192.168.100.21    UP 00:11:32     S  192.168.100.21/32
```

Example 4-66 displays R31's multicast routing table for the 225.4.4.4 group. Notice the difference in RPF neighbors on the shared tree versus the source tree.

Example 4-66 *R31's Multicast Routing Table for the 225.4.4.4 Group*

```
R31-Spoke# show ip mroute 225.4.4.4
! Output omitted for brevity
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

 (*, 225.4.4.4), 00:02:35/stopped, RP 192.168.1.1, flags: SJC
  Incoming interface: Tunnel100, RPF nbr 192.168.100.11
 Outgoing interface list:
    GigabitEthernet1/0, Forward/Sparse, 00:02:35/00:02:38

(10.4.4.4, 225.4.4.4), 00:02:35/00:00:24, flags: JT
  Incoming interface: Tunnel100, RPF nbr 192.168.100.41
  Outgoing interface list:
    GigabitEthernet1/0, Forward/Sparse, 00:02:35/00:02:38
```

The problem occurs because PIM tries to optimize the multicast traffic flow over the shortest path, which is a source tree. Routes in the multicast routing table that have the '*T*' flag set indicate that PIM has tried to move the stream from a shared path tree to a source tree.

There are two solutions to the problem:

- **Change the SPT threshold:** Cisco routers try to initiate a changeover to the source tree upon receipt of the first packet. This behavior can be disabled so that the router never tries to switch to a source tree.
- **Modify the multicast routing table:** Creating a multicast route for the source's IP address allows a different next-hop IP address to be used for multicast traffic versus unicast network traffic. The route needs to be as specific as the NHRP route that is injected for the source LAN network.

Modify the SPT Threshold

Disabling PIM's ability to switch over from a shared tree to a source tree forces multicast traffic to always flow through the RP. Because the RP is placed behind the DMVPN hub router, multicast traffic never tries to flow across the spoke-to-spoke DMVPN tunnel.

The command **ip pim spt-threshold infinity** ensures that the source-based distribution tree is never used. The configuration needs to be placed on all routers in the remote LANs that have clients attached, as shown in Example 4-67. If the command is missed on a router with an attached receiver, the DMVPN spoke router tries to use a source-based tree and stop traffic for all routers behind that spoke router.

Example 4-67 *Disabling the SPT Threshold Configuration*

```
R31, R41, R51 and R52
ip pim spt-threshold infinity
```

Example 4-68 displays the multicast routing table for R31, R11, and R41 for the 225.4.4.4 group address. Only the shared tree exists on R31. R41 transmits out of both distribution trees. R11 receives packets on the source tree but transmits from the source and shared distribution trees.

Example 4-68 *Shared Tree Routing Table for the 255.4.4.4 Stream*

```
R31-Spoke# show ip mroute 225.4.4.4
! Output omitted for brevity
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry, E - Extranet,
       Z - Multicast Tunnel, z - MDT-data group sender,
IP Multicast Routing Table
 (*, 225.4.4.4), 00:00:31/00:02:28, RP 192.168.1.1, flags: SC
  Incoming interface: Tunnel100, RPF nbr 192.168.100.11
  Outgoing interface list:
    GigabitEthernet1/0, Forward/Sparse, 00:00:31/00:02:28
```

```
R11-Hub# show ip mroute 225.4.4.4
! Output omitted for brevity
(*, 225.4.4.4), 00:00:55/00:02:56, RP 192.168.1.1, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Tunnel100, 192.168.100.31, Forward/Sparse, 00:00:33/00:02:56

(10.4.4.4, 225.4.4.4), 00:00:55/00:02:04, flags: TA
  Incoming interface: Tunnel100, RPF nbr 192.168.100.41
  Outgoing interface list:
    Tunnel100, 192.168.100.31, Forward/Sparse, 00:00:33/00:02:56
```

```
R41-Spoke# show ip mroute 225.4.4.4
! Output omitted for brevity
(*, 225.4.4.4), 00:01:57/stopped, RP 192.168.1.1, flags: SPF
  Incoming interface: Tunnel100, RPF nbr 192.168.100.11
  Outgoing interface list: Null

(10.4.4.4, 225.4.4.4), 00:01:57/00:01:02, flags: FT
  Incoming interface: GigabitEthernet1/0, RPF nbr 10.4.4.4
  Outgoing interface list:
    Tunnel100, Forward/Sparse, 00:01:34/00:02:31
```

The pitfalls of disabling the SPT threshold are the following:

- It is not applicable to SSM.
- It must be configured on all multicast-enabled routers on spoke LANs. It is the last-hop router that tries to join the source tree.

- It applies to all multicast traffic and is not selective based on the stream.
- It prevents the creation of (*S*,*G*) entries, which reduces the granularity of show commands for troubleshooting.

Modify the Multicast Routing Table

The other solution is to modify the multicast routing table so that the multicast stream's source network is reached via the DMVPN hub. Modifying the multicast routing table does not impact the traffic flow of the unicast routing table. The route for the source LAN address must be added to the multicast routing table.

Static multicast routes can be placed on all the spoke routers, but it is not a scalable solution. The best solution is to use the multicast address family of BGP. Establishing a BGP session was explained earlier; the only difference is that the **address-family ipv4 multicast** command initializes the multicast address family.

The hub routers act as a route reflector for the spoke routers. Just as in unicast routing, the spoke and hub routers set the next hop for all multicast traffic to ensure that the hub's IP address is the next-hop address. Last, the spoke router that hosts the source advertises the source's LAN network into BGP.

Example 4-69 displays the multicast BGP configuration for the hub routers.

Example 4-69 *Hub Multicast BGP Configuration*

```
R11 and R21
router bgp 10
 address-family ipv4 multicast
  neighbor MPLS-SPOKES activate
  neighbor MPLS-SPOKES next-hop-self all
  neighbor MPLS-SPOKES route-reflector-client
```

```
R12 and R22
router bgp 10
 address-family ipv4 multicast
  neighbor INET-SPOKES activate
  neighbor INET-SPOKES next-hop-self all
  neighbor INET-SPOKES route-reflector-client
```

Example 4-70 displays the multicast configuration for the spoke routers.

Example 4-70 *Spoke Multicast BGP Configuration*

```
R31, R41 and R51
router bgp 10
 address-family ipv4 multicast
  neighbor 192.168.100.11 activate
  neighbor 192.168.100.21 activate
  neighbor MPLS-SPOKES next-hop-self all
```

```
R31, R41 and R52
router bgp 10
 address-family ipv4 multicast
  neighbor 192.168.200.12 activate
  neighbor 192.168.200.22 activate
  neighbor INET-SPOKES next-hop-self all
```

Example 4-71 displays R41's multicast BGP advertisement of the 10.4.4.0/24 network.

Example 4-71 *R41's Advertisement of the 10.4.4.0/24 Network in the Multicast BGP Table*

```
R41
router bgp 10
 address-family ipv4 multicast
  network 10.4.4.0 mask 255.255.255.0
```

Example 4-72 verifies that all four hub routers are advertising the 10.4.4.0/24 network into the multicast BGP table.

Example 4-72 *Verification of the Multicast BGP Table*

```
R31-Spoke# show bgp ipv4 multicast
BGP table version is 21, local router ID is 10.3.0.31
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 * i 10.4.4.0/24      192.168.200.22           0    100      0 i
 * i                  192.168.200.12           0    100      0 i
 * i                  192.168.100.21           0    100      0 i
 *>i                  192.168.100.11           0    100      0 i
```

Example 4-73 displays the multicast routing table for R11 and R41. Notice the MBGP indication in the multicast routing table entry for 225.4.4.4.

Example 4-73 *225.4.4.4 Multicast Routing Table After Multicast BGP*

```
R31-Spoke# show ip mroute 225.4.4.4
! Output omitted for brevity
 (*, 225.4.4.4), 00:49:28/stopped, RP 192.168.1.1, flags: SJC
  Incoming interface: Tunnel100, RPF nbr 192.168.100.11
  Outgoing interface list:
    GigabitEthernet1/0, Forward/Sparse, 00:49:28/00:02:33

(10.4.4.4, 225.4.4.4), 00:24:23/00:02:55, flags: JT
  Incoming interface: Tunnel100, RPF nbr 192.168.100.11, Mbgp
  Outgoing interface list:
    GigabitEthernet1/0, Forward/Sparse, 00:24:23/00:02:33
```

```
R11-Hub# show ip mroute 225.4.4.4
! Output omitted for brevity
(*, 225.4.4.4), 00:48:43/00:03:23, RP 192.168.1.1, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Tunnel100, 192.168.100.31, Forward/Sparse, 00:13:55/00:03:23

(10.4.4.4, 225.4.4.4), 00:48:43/00:02:45, flags: TA
  Incoming interface: Tunnel100, RPF nbr 192.168.100.41, Mbgp
  Outgoing interface list:
    Tunnel100, 192.168.100.31, Forward/Sparse, 00:10:44/00:03:23
```

Example 4-74 verifies that the advertisement in BGP has no impact on the multicast routing entry for the server connected to R13 on the 10.1.1.0/24 network.

Example 4-74 *225.1.1.1 Multicast Routing Table After Multicast BGP*

```
R31-Spoke# show ip mroute 225.1.1.1
! Output omitted for brevity
(*, 225.1.1.1), 00:00:16/stopped, RP 192.168.1.1, flags: SJC
  Incoming interface: Tunnel100, RPF nbr 192.168.100.11
  Outgoing interface list:
    GigabitEthernet1/0, Forward/Sparse, 00:00:16/00:02:43

(10.1.1.1, 225.1.1.1), 00:00:16/00:02:43, flags: JT
  Incoming interface: Tunnel100, RPF nbr 192.168.100.11
  Outgoing interface list:
    GigabitEthernet1/0, Forward/Sparse, 00:00:16/00:02:43
```

Summary

This chapter focused on the routing protocol design principles and their deployment for EIGRP or BGP to exchange routes across the Intelligent WAN. These are the key concepts of a successful IWAN routing design:

- Branch sites should not re-advertise routes learned from one hub to another hub. This prevents transit routing at the branches and keeps traffic flows deterministic.
- Hub routers should advertise summary routes to branch routers to reduce the size of the routing table. This includes a default route for Internet connectivity, enterprise prefixes (includes all branch site locations and LAN networks in the enterprise), DC-specific prefixes, and the optional local PfR MC loopback (to simplify troubleshooting).
- Hub routers always prefer routes learned from the branch router's tunnel interface that is attached to the same transport as the hub router.
- Network traffic should be steered to use the preferred transport through manipulation of the routing protocol's best-path calculation. This provides optimal flow while PfR is in an uncontrolled state.
- The protocol configuration should keep variables to a minimum so that the configuration can be deployed via network management tools like Cisco Prime Infrastructure.

Further Reading

To keep the size of the book small, this chapter does not go into explicit detail about routing protocol behaviors, advanced filtering techniques, or multicast routing. Deploying and maintaining an IWAN environment requires an understanding of these concepts depending on your environment's needs. The book *IP Routing on Cisco IOS, IOS XE, and IOS XR* that is listed here provides a thorough reference to the concepts covered in this chapter.

Bates, T., and R. Chandra. RFC 1966, "BGP Route Reflection: An Alternative to Full Mesh IBGP." IETF, June 1996. http://tools.ietf.org/html/rfc1966.

Cisco. "Cisco IOS Software Configuration Guides." www.cisco.com.

Cisco. "Understanding the Basics of RPF Checking." www.cisco.com.

Edgeworth, Brad, Aaron Foss, and Ramiro Garza Rios. *IP Routing on Cisco IOS, IOS XE, and IOS XR*. Indianapolis: Cisco Press, 2014.

Rekhter, Y., T. Li, and S. Hares. RFC 4271. "A Border Gateway Protocol 4 (BGP-4)." IETF, January 2006. http://tools.ietf.org/html/rfc4271.

This page intentionally left blank

Chapter 5

Securing DMVPN Tunnels and Routers

This chapter covers the following topics:

- Elements of secure transport
- IPsec fundamentals
- IPsec tunnel protection
- Zone-Based Firewalls (ZBFWs)
- Control Plane Policing (CoPP)
- Device hardening

This chapter focuses on securing the IWAN network and encompasses device hardening of the IWAN routers and protecting the data transmitted between them. Device hardening is the process of securing a router as much as possible by reducing security threats.

When employees think about the data that is transmitted on their network, they associate a certain level of sensitivity with it. For example, bank statements, credit card numbers, and product designs are considered highly sensitive. If this information is made available to the wrong party, there could be repercussions for the company or a specific user. Employees assume that their data is secure because their company owns all the infrastructure, but this is not necessarily the case when a WAN is involved. A properly designed network provides data confidentiality, integrity, and availability. Without these components, a business might lose potential customers if they do not think that their information is secure.

The following list provides the terms and functions of data confidentiality, data integrity, and data availability:

- **Data confidentiality:** Ensuring that data is viewable only by authorized users. Data confidentiality is maintained through encryption.
- **Data integrity:** Ensuring that data is modified only by authorized users. Information is valuable only if it is accurate. Inaccurate data can result in an unanticipated cost,

for example, if a product design is modified and the product therefore does not work. When the product breaks, the time it takes to identify and correct the issue has a cost associated with it. Data integrity is maintained via an encrypted digital signature, which is typically a checksum.

- **Data availability:** Ensuring that the network is always available allows for the secure transport of the data. Redundancy and proper design ensure data availability.

Elements of Secure Transport

WAN network designs are based on the concept of trusted SP connections. Original network circuits were point-to-point connections and placed trust in the SP's ability to control access to the infrastructure and assurances of privacy. Even though SPs use peer-to-peer networks, technologies such as MPLS VPNs have provided a layer of logical segmentation.

A certain level of data confidentiality on a WAN is based upon the type of transport and the limited access to the network by the SP's employees. Information security and network engineers assume that the SP network is secure and does not require encryption on the SP WAN circuits.

Figure 5-1 displays the traditional approach to securing data on a network. The entire controlled infrastructure (enterprise and SP) is assumed to be safe. Traffic is encrypted only when exposed to the public Internet.

Also, the Internet edge is the only identified intrusion point for a network. The Internet edge is protected by a firewall such as the Cisco Adaptive Security Appliance which prevents outside users from accessing the network and servers in the DC that hosts e-commerce applications.

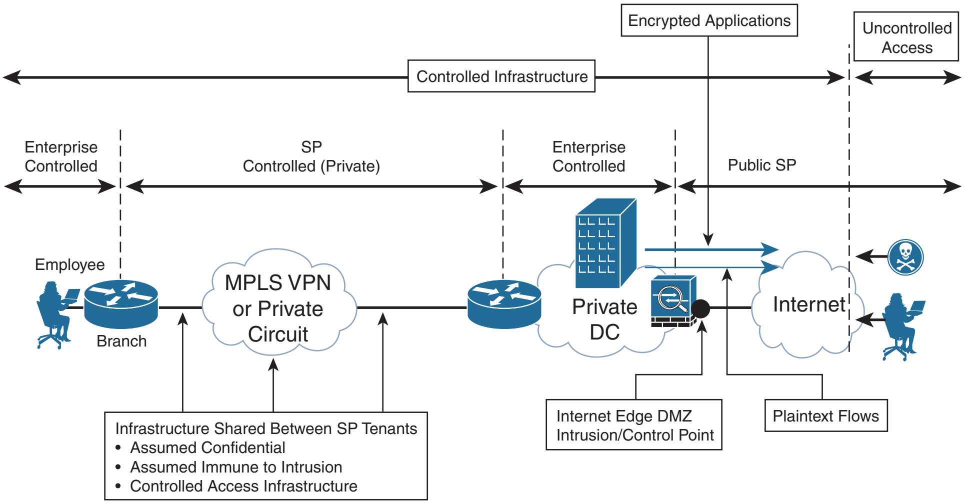

Encrypted Applications
Uncontrolled Access
Controlled Infrastructure
Enterprise Controlled
SP Controlled (Private)
Enterprise Controlled
Public SP
Employee
Branch
MPLS VPN or Private Circuit
Private DC
Internet
Infrastructure Shared Between SP Tenants
• Assumed Confidential
• Assumed Immune to Intrusion
• Controlled Access Infrastructure
Internet Edge DMZ Intrusion/Control Point
Plaintext Flows

Figure 5-1 *Typical WAN Network*

In Figure 5-2, the Internet is used as the transport for the WAN. The Internet does not provide controlled access and cannot guarantee data integrity or data confidentiality. Hackers, eavesdroppers, and man-in-the-middle intrusions are common threats on public transports like the Internet. In addition, branch WAN, corporate WAN, and Internet edge become intrusion points for the network.

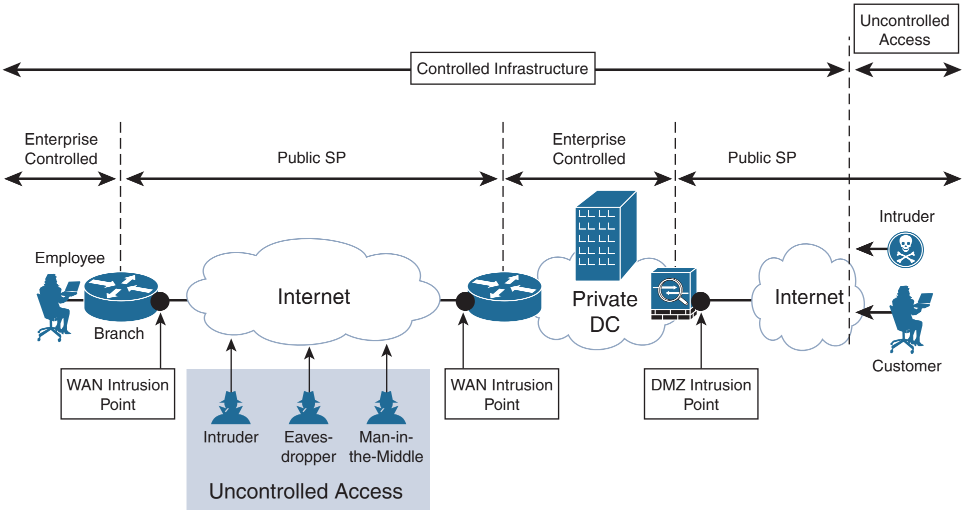


Figure 5-2 *Internet as a WAN Transport*

Data confidentiality and integrity are maintained by adding IPsec encryption to the DMVPN tunnel that uses the Internet as a transport. IPsec is a set of industry standards defined in RFC 2401 to secure IP-based network traffic. The branch and corporate routers are hardened by limiting only the IPsec ports to pass through the Cisco ZBFW. Combining both of these technologies reduces threat vectors on any WAN while providing data confidentiality and integrity. These technologies are shown in Figure 5-3.

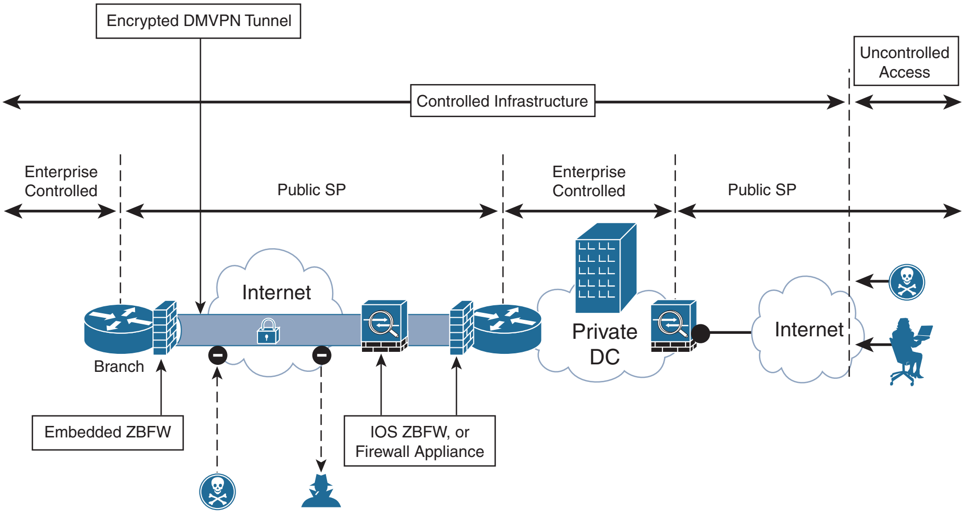


Figure 5-3 *Secured Internet as a WAN Transport*

In the traditional private WAN model, trust is placed in the SP and its ability to control access to a company's network. Occasionally provisioning systems fail and data is leaked into another customer's networks. Other times more blatant attempts to compromise data confidentiality or integrity may occur in other geographic regions. Some organizations that are subject to stringent privacy regulations (healthcare, government, and financial) often require all traffic to be encrypted across all WAN links regardless of the transport used. Adding IPsec tunnel protection is a straightforward process and removes the headaches associated with IPsec point-to-point tunnels.

IPsec Fundamentals

DMVPN tunnels are not encrypted by default, but they can be encrypted by IPsec. IPsec provides encryption through cryptographically based security and was designed with interoperability in mind. When IPsec is integrated with DMVPN tunnels, the encrypted DMVPN tunnels provide a secure overlay network over any transport with the following functions:

- **Origin authentication:** Authentication of origin is accomplished by pre-shared key (static) or through certificate-based authentication (dynamic).
- **Data confidentiality:** A variety of encryption algorithms are used to preserve confidentiality.
- **Data integrity:** Hashing algorithms ensure that packets are not modified in transit.
- **Replay detection:** This provides protection against hackers trying to capture and insert network traffic.

- **Periodic rekey:** New security keys are created between endpoints every specified time interval or within a specific volume of traffic.
- **Perfect forward secrecy:** Each session key is derived independently of the previous key. A compromise of one key does not compromise future keys.

The IPsec security architecture is composed of the following independent components:

- Security protocols
- Security associations
- Key management

Security Protocols

IPsec uses two protocols to provide data integrity and confidentiality. The protocols can be applied individually or combined based upon need. Both protocols are explained further in the following sections.

Authentication Header

The *IP authentication header* provides data integrity, authentication, and protection from hackers replaying packets. The authentication header protocol ensures that the original data packet (before encapsulation/encryption) has not been modified during transport on the public network. It creates a digital signature similar to a checksum to ensure that the packet has not been modified, using protocol number 51 located in the IP header.

Encapsulating Security Payload (ESP)

The *Encapsulating Security Payload (ESP)* provides data confidentiality, authentication, and protection from hackers replaying packets. Typically, *payload* refers to the actual data minus any headers, but in the context of ESP, the payload is the portion of the original packet that is encapsulated within the IPsec headers. The ESP protocol ensures that the original payload (before encapsulation) maintains data confidentiality by encrypting the payload and adding a new set of headers during transport across a public network. ESP uses the protocol number 50 that is located in the IP header.

Key Management

A critical component of secure encryption is the communication of the keys used to encrypt and decrypt the traffic being transported over the insecure network. The process of generating, distributing, and storing these keys is called *key management*. IPsec uses the *Internet Key Exchange (IKE)* protocol by default.

RFC 4306 defines the second iteration of IKE called IKEv2, which provides mutual authentication of each party. IKEv2 introduced the support of Extensible Authentication

Protocol (EAP) (certificate-based authentication), reduction of bandwidth consumption, NAT traversal, and the ability to detect whether a tunnel is still alive.

Security Associations

Security associations (SAs) are a vital component of IPsec architecture and contain the security parameters that were agreed upon between the two endpoint devices. There are two types of SAs:

- **IPsec SA:** Used for data plane functions to secure data transmitted between two different sites
- **IKE SA:** Used for control plane functions like IPsec key management and management of IPsec SAs.

There is only one IKE SA between endpoint devices, but multiple IPsec SAs can be established between the same two endpoint devices.

Note IPsec SAs are unidirectional and require at least two IPsec SAs (one for inbound, one for outbound) to exchange network traffic between two sites.

ESP Modes

Traditional IPsec provides two ESP modes of packet protection: tunnel and transport.

- Tunnel mode encrypts the entire original packet and adds a new set of IPsec headers. These new headers are used to route the packet and also provide overlay functions.
- Transport mode encrypts and authenticates only the packet payload. This mode does not provide overlay functions and routes based upon the original IP headers.

Figure 5-4 displays an original packet, an IPsec packet in transport mode, and an IPsec packet in tunnel mode. The following section expands upon these concepts by explaining the structure of various DMVPN packets. The DMVPN packet structure can be compared to a regular packet in Figure 5-4.

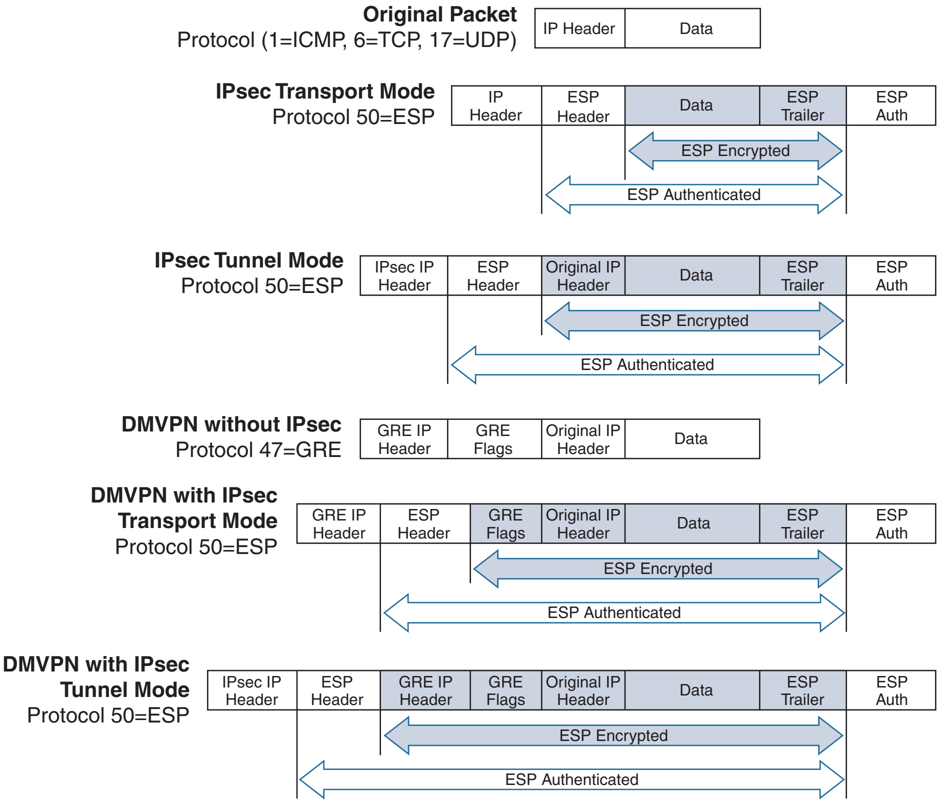


Figure 5-4 *DMVPN Packet Headers*

DMVPN without IPsec

In unencrypted DMVPN packets, the original packets have GRE flags added to them, and then the new GRE IP header is added for routing the packets on the transport (underlay) network. The GRE IP header adds an extra 20 bytes of overhead, and the GRE flags add an extra 4 bytes of overhead. These packets use the protocol field of GRE (47).

Note If a tunnel key is specified, an additional 4 bytes are added to every packet regardless of whether the encryption type (if any) is selected.

DMVPN with IPsec in Transport Mode

For encrypted DMVPN packets that use ESP transport mode, the original packets have the GRE flags added to them, and then that portion of the packets is encrypted. A signature for the encrypted payload is added, and then the GRE IP header is added for routing the packets on the transport (underlay) network.

The GRE IP header adds an extra 20 bytes of overhead, the GRE flags add an extra 4 bytes of overhead, and depending on the encryption mechanism, a varying amount of additional bytes(s) is added for the encrypted signature. These packets use the protocol field of ESP (50).

DMVPN with IPsec in Tunnel Mode

For encrypted DMVPN packets that use ESP tunnel mode, the original packets have the GRE flags added to them, and then the new GRE IP header is added for routing the packets on the transport (underlay) network. That portion of the packets is encrypted, a signature for the encrypted payload is added, and then a new IPsec IP header is added for routing the packets on the transport (underlay) network.

The GRE IP header adds an extra 20 bytes of overhead, the GRE flags add an extra 4 bytes of overhead, the IPsec IP header adds an extra 20 bytes of overhead, and depending on the encryption mechanism, a varying amount of additional bytes(s) is added for the encrypted signature. These packets use the IP protocol field of ESP (50).

It is important to note that the use of IPsec tunnel mode for DMVPN networks does not add any perceived value and adds 20 bytes of overhead. Transport mode should be used for encrypted DMVPN tunnels.

IPsec Tunnel Protection

Enabling IPsec protection on a DMVPN network requires that all devices enable IPsec protection. If some routers have IPsec enabled and others do not, devices with mismatched settings will not be able to establish a connection on the tunnel interfaces.

Pre-shared Key Authentication

The first scenario for deploying IPsec tunnel protection is with the use of a static pre-shared key, which involves the creation of an

- IKEv2 keyring
- IKEv2 profile
- IPsec transform set
- IPsec profile

In this portion of the book, emphasis is placed on the DMVPN routers that are attached to the Internet as shown in Figure 5-5. The following sections explain how to configure IPsec tunnel protection on the DMVPN tunnel 200.

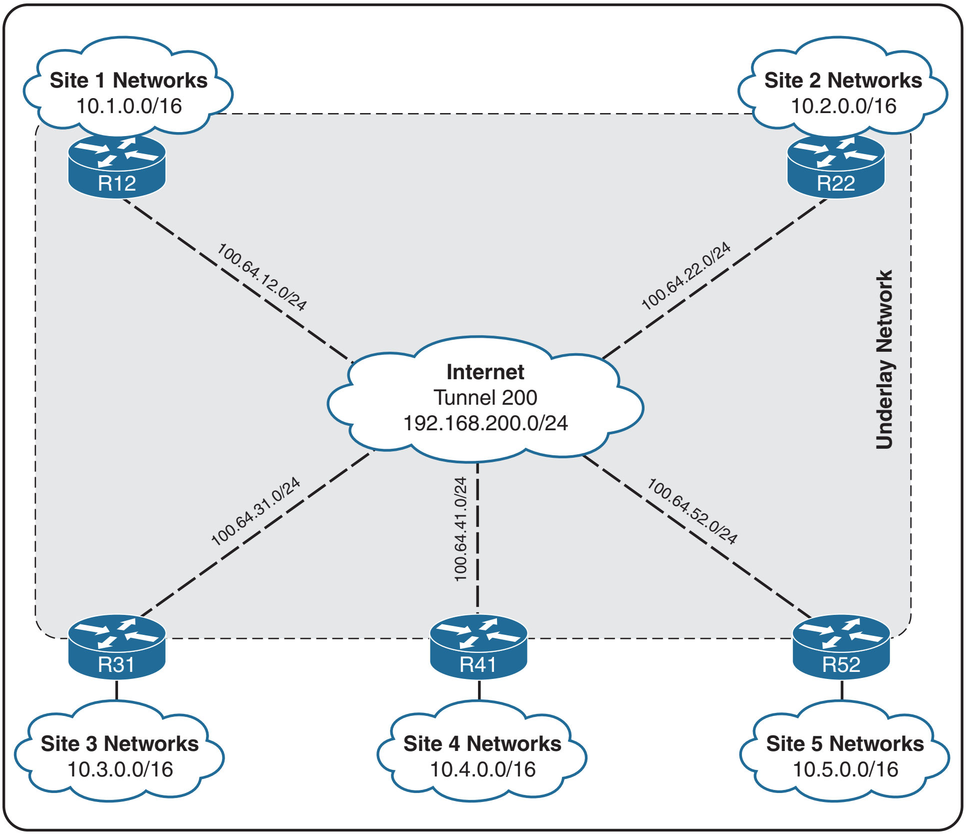


Figure 5-5 *Typical WAN Network*

IKEv2 Keyring

The IKEv2 keyring is a repository of the pre-shared keys. In a keyring it is possible to define which keys apply to which hosts. Identification of the password is based on the IP address of the remote router. The IKEv2 keyring is created with the following steps:

Step 1. **Define the keyring instance.**

The IKEv2 keyring is created with the command **crypto ikev2 keyring** *keyring-name*.

Step 2. **Create a friendly peer name.**

Multiple peers can exist in a keyring. Each peer has a matching qualifier and can use a different password. The peer is created with the command **peer** *peer-name*. For simplicity, only one peer is created, called *ANY*.

Step 3. **Identify the IP address or address range for a peer.**

Multiple peers can reside in a keyring. The IP address is identified so that the appropriate peer configuration is used based upon the remote device's IP address. The command **address** *network subnet-mask* defines the IP address range. For simplicity, the value of 0.0.0.0 0.0.0.0 is used to allow a match against any peer.

IPv6 transport can use the value of ::/0 for any IPv6 peer.

Step 4. **Define a pre-shared key.**

The last step is to define the pre-shared key with the command **pre-shared-key** *secure-key*. Generally a long and alphanumeric password is used for increased security.

Example 5-1 demonstrates a simple keyring that is used to secure the DMVPN routers on the Internet.

Example 5-1 *IKEv2 Keyring*

```
crypto ikev2 keyring DMVPN-KEYRING-INET
 peer ANY
  address 0.0.0.0 0.0.0.0
  pre-shared-key CISCO456
```

IKEv2 Profile

The IKEv2 profile is a collection of nonnegotiable security parameters used during the IKE security association. The IKEv2 profile is later associated with the IPsec profile. Within the IKEv2 profile, local and remote authentication methods must be defined, as well as a match statement (identity, certificate, and so on).

The basic steps for creating an IKEv2 profile are as follows:

Step 1. **Define the IKEv2 profile.**

The IKEv2 profile is defined with the command **crypto ikev2 profile** *ike-profile-name*.

Step 2. **Identify the IP address for the remote router.**

The IP address must be identified for the initial IKEv2 session to establish. The peer IP address is defined with the command **match identity remote address** *ip-address*. For simplicity, the value of 0.0.0.0 is used to allow a match against any peer.

IPv6 transport can use the value of ::/0 for any IPv6 peer.

Step 3. **Configure the local router's identity (optional).**

The local router's identity can be set based on an IP address with the command **identity local address** *ip-address*.

This command is not needed for pre-shared key authentication but is very helpful with the deployment of PKI authentication. The IP address specified should match the IP address used when registering the certificate (the recommended Loopback0 IP address).

Step 4. **Identify the FVRF for the tunnel end.**

The FVRF must be associated to the IKEv2 profile with the command **match fvrf** {*vrf-name* | **any**}. Using the **any** keyword allows either FVRF to be selected.

Step 5. **Define the local authentication method.**

The authentication method must be defined for connection requests that are received by remote peers. The command **authentication local** {**pre-share** | **rsa-sig**} defines the local authentication. Only one local authentication can be selected. The **pre-share** keyword is for pre-shared static keys, and **rsa-sig** is used for certificate-based authentication.

Step 6. **Define the remote authentication method.**

The authentication method must be defined for connection requests that are sent to remote peers. The command **authentication remote** {**pre-share** | **rsa-sig**} defines the remote authentication. Multiple remote authentication methods can be defined by repeating the command. The **pre-share** keyword is for pre-shared static keys, and **rsa-sig** is used for certificate-based authentication.

Step 7. **Define the IKEv2 keyring (required for pre-shared authentication).**

Pre-shared authentication requires that the IKEv2 keyring be associated to the IKEv2 profile. The command **keyring local** *keyring-name* associates the IKEv2 keyring.

Example 5-2 provides a sample IKEv2 profile that uses pre-shared key authentication.

Example 5-2 *Sample IKEv2 Profile*

```
crypto ikev2 profile DMVPN-IKE-PROFILE-INET
 match fvrf INET01
 match identity remote address 0.0.0.0
 authentication remote pre-share
 authentication local pre-share
 keyring local DMVPN-KEYRING-INET
```

The IKEv2 profile settings are displayed with the command **show crypto ikev2 profile** as shown in Example 5-3. Notice that the authentication, FVRF, IKE keyring, and identity IP address are displayed along with the IKE lifetime.

Example 5-3 *Display of IKEv2 Profile Settings*

```
R12-DC1-Hub2# show crypto ikev2 profile
IKEv2 profile: DMVPN-IKE-PROFILE-INET
 Ref Count: 1
 Match criteria:
  Fvrf: INET01
  Local address/interface: none
  Identities:
   address 0.0.0.0
  Certificate maps: none
 Local identity: none
 Remote identity: none
 Local authentication method: pre-share
 Remote authentication method(s): pre-share
 EAP options: none
 Keyring: DMVPN-KEYRING-INET
 Trustpoint(s): none
 Lifetime: 86400 seconds
 DPD: disabled
 NAT-keepalive: disabled
 Ivrf: none
 Virtual-template: none
 mode auto: none
 AAA AnyConnect EAP authentication mlist: none
 AAA EAP authentication mlist: none
 AAA Accounting: none
 AAA group authorization: none
 AAA user authorization: none
```

IPsec Transform Set

The transform set identifies the security protocols (ESP) for encrypting traffic. It specifies the protocol ESP or authentication header that is used to authenticate the data.

Table 5-1 provides a matrix of common IPsec transforms that can be inserted into a transform set. Following are some guidelines:

- Select an ESP encryption transform for data confidentiality.
- Select an authentication header or ESP authentication transform for data confidentiality.

Table 5-1 *IPsec Transform Matrix*

Transform Type	Transform	Description
ESP encryption	esp-aes 128	ESP with the 128-bit Advanced Encryption Standard (AES) encryption algorithm
	esp-aes 192	ESP with the 192-bit AES encryption algorithm
	esp-aes 256	ESP with the 256-bit AES encryption algorithm
	esp-gcm 128	ESP transform using the Galois Counter Mode (GCM) 128-bit cipher
	esp-gcm 192	ESP transform using the GCM 192-bit cipher
	esp-gcm 256	ESP transform using the GCM 256-bit cipher (next-generation encryption)
ESP authentication	esp-sha-hmac	ESP with the Secure Hash Algorithm (SHA) (HMAC variant) authentication algorithm
	esp-sha256-hmac	ESP with the 256-bit SHA2 (HMAC variant) authentication algorithm
	esp-sha384-hmac	ESP with the 384-bit SHA2 (HMAC variant) authentication algorithm
	esp-sha512-hmac	ESP with the 512-bit SHA2 (HMAC variant) authentication algorithm
Authentication header authentication	ah-md5-hmac	Authentication header with the MD5 (Message Digest Algorithm 5) authentication algorithm
	ah-sha-hmac	Authentication header with the SHA authentication algorithm

The transform set is created with the following steps:

Step 1. Create the transform set and identify the transforms.

The transform set and identification of transforms are accomplished with one command. Only one transform set can be selected for ESP encryption, ESP authentication, and authentication header authentication. The command is **crypto ipsec transform-set** *transform-set-name* [*esp-encryption-name*] [*esp-authentication-name*] [*ah-authentication-name*].

Suggested transform set combinations are

```
esp-aes 256 and esp-sha-hmac
esp-aes and esp-sha-hmac
```

Step 2. **Specify the ESP mode.**

The ESP mode is configured with the command **mode** {**transport** | **tunnel**}. The ESP tunnel mode is the default mode and does not provide any benefits while adding 20 bytes of overhead per packet. Use the ESP mode of transport.

Example 5-4 provides a sample IPsec transform set.

Example 5-4 *Sample IPsec Transform Set*

```
crypto ipsec transform-set AES256/SHA/TRANSPORT esp-aes 256 esp-sha-hmac
 mode transport
```

The transform set can be verified with the command **show crypto ipsec transform-set** as shown in Example 5-5.

Example 5-5 *Verification of the IPsec Transform Set*

```
R12-DC1-Hub2# show crypto ipsec transform-set
! Output omitted for brevity
Transform set AES256/SHA/TRANSPORT: { esp-256-aes esp-sha-hmac  }
   will negotiate = { Transport,  },
```

IPsec Profile

The IPsec profile combines the IPsec transform set and the IKEv2 profile. The IPsec profile is created with the following steps:

Step 1. **Create the IPsec profile.**

The IPsec profile is created with the command **crypto ipsec profile** *profile-name*. The configuration context is then placed in IPsec profile configuration submode.

Step 2. **Specify the transform set.**

The transform set is specified with the command **set transform-set** *transform-set-name*.

Step 3. **Specify the IKEv2 profile.**

The IKEv2 profile is specified with the command **set ikev2-profile** *ike-profile-name*.

Example 5-6 provides a sample IPsec profile configuration.

Example 5-6 *Sample IPsec Profile*

```
crypto ipsec profile DMVPN-IPSEC-PROFILE-INET
 set transform-set AES256/SHA/TRANSPORT
 set ikev2-profile DMVPN-IKE-PROFILE-INET
```

The command **show crypto ipsec profile** displays the components of the IPsec profile as shown in Example 5-7.

Example 5-7 *Verification of the IPsec Profile*

```
R12-DC1-Hub2# show crypto ipsec profile
! Output omitted for brevity
IPSEC profile DMVPN-IPSEC-PROFILE-INET
        IKEv2 Profile: DMVPN-IKE-PROFILE-INET
        Security association lifetime: 4608000 kilobytes/3600 seconds
        Responder-Only (Y/N): N
        PFS (Y/N): N
        Mixed-mode : Disabled
        Transform sets={
                AES256/SHA/TRANSPORT:  { esp-256-aes esp-sha-hmac  } ,
```

Encrypting the Tunnel Interface

Now that all the required IPsec components have been configured, the IPsec profile is associated to the DMVPN tunnel interface with the command **tunnel protection ipsec profile** *profile-name* [**shared**].

The **shared** keyword is required for routers that terminate multiple encrypted DMVPN tunnels on the same transport interface. The command shares the IPsec *security association database (SADB)* among multiple DMVPN tunnels. Because the SADB is shared, a unique tunnel key must be defined on each DMVPN tunnel interface to ensure that the encrypted/decrypted traffic aligns to the proper DMVPN tunnel.

Note The topology in this book does not terminate multiple DMVPN tunnels on the same transport interface. The **shared** keyword is not required, nor is the tunnel key.

Example 5-8 provides a sample configuration for encrypting a DMVPN tunnel interface. After the configuration in this section is applied to R12, R22, R31, R41, and R52, the DMVPN tunnels are protected with IPsec.

Example 5-8 *Enabling IPsec Tunnel Protection*

```
interface Tunnel200
 tunnel protection ipsec profile DMVPN-IPSEC-PROFILE-INET
```

IPsec Packet Replay Protection

The Cisco IPsec implementation includes an anti-replay mechanism that prevents intruders from duplicating encrypted packets by assigning a unique sequence number to each encrypted packet. When a router decrypts the IPsec packets, it keeps track of the packets it has received. The IPsec anti-replay service rejects (discards) duplicate packets or old packets.

The router identifies acceptable packet age according to the following logic. The router maintains a sequence number window size (default of 64 packets). The minimum sequence number is defined as the highest sequence number for a packet minus the window size. A packet is considered of age when the sequence number is between the minimum sequence number and the highest sequence number.

At times, the default 64-packet window size is not adequate. Encryption is where the sequence number is set, and this happens before any QoS policies are processed. Packets can be delayed because of QoS priorities, resulting in out-of-order packets (low-priority packets are queued, whereas high-priority packets are immediately forwarded). The sequence number increases on the receiving router because the high-priority packets shift the window ahead, and when the lower-priority packets arrive later, they are discarded.

Increasing the anti-replay window size has no impact on throughput or security. An additional 128 bytes per incoming IPsec SA are needed to store the sequence number on the decryptor. The window size is increased globally with the command **crypto ipsec security-association replay window-size** *window-size*. Cisco recommends using the largest window size possible for the platform, which is 1024.

Dead Peer Detection

When two routers establish an IPsec VPN tunnel between them, it is possible that connectivity between the two routers can be lost for some reason. In most scenarios, IKE and IPsec do not natively detect a loss of peer connectivity, which results in network traffic being blackholed until the SA lifetime expires.

The use of *dead peer detection (DPD)* helps detect the loss of connectivity to a remote IPsec peer. When DPD is enabled in on-demand mode, the two routers check for connectivity only when traffic needs to be sent to the IPsec peer and the peer's liveliness is questionable. In such scenarios, the router sends a DPD R-U-THERE request to query the status of the remote peer. If the remote router does not respond to the R-U-THERE request, the requesting router starts to transmit additional R-U-THERE messages every retry interval for a maximum of five retries. After that the peer is declared dead.

DPD is configured with the command **crypto ikev2 dpd** [*interval-time*] [*retry-time*] **on-demand** in the IKEv2 profile. As a general rule, the interval time is set to twice that of the routing protocol timer (2*20), and the retry interval is set to 5 seconds. In essence, the total time is (2*20(*routing-protocol*)) + (5*5(*retry-count*)) = 65 seconds. This exceeds the holdtime of the routing protocol and engages only when the routing protocol is not operating properly.

DPD is configured on the spoke routers and not on the hubs because of the CPU processing that is required to maintain state for all the branch routers.

NAT Keepalives

Network Address Translation (NAT) keepalives are enabled to keep the dynamic NAT mapping alive during a connection between two peers. NAT keepalives are UDP (User Datagram Protocol) packets that contain an unencrypted payload of 1 byte. When DPD is used to detect peer status, NAT keepalives are sent if the IPsec entity has not transmitted or received a packet within a specified time period. NAT keepalives are enabled with the command **crypto isakmp nat keepalive** *seconds*.

Note This command is placed on the DMVPN spokes because the routing protocol between the spoke and the hub keeps the NAT state, whereas spoke-to-spoke tunnels do not maintain a routing protocol relationship so NAT state is not maintained.

Complete Configuration

Example 5-9 displays the complete configuration to enable IPsec protection on the Internet DMVPN tunnel on R12, R22, R31, R41, and R52 with all the settings from this section.

Example 5-9 *Complete IPsec DMVPN Configuration with Pre-Shared Authentication*

```
R12 and R22
crypto ikev2 keyring DMVPN-KEYRING-INET
 peer ANY
  address 0.0.0.0 0.0.0.0
  pre-shared-key CISCO456
!
crypto ikev2 profile DMVPN-IKE-PROFILE-INET
 match fvrf INET01
 match identity remote address 0.0.0.0
 authentication remote pre-share
 authentication local pre-share
 keyring local DMVPN-KEYRING-INET
!
crypto ipsec transform-set AES256/SHA/TRANSPORT esp-aes 256 esp-sha-hmac
 mode transport
!
crypto ipsec profile DMVPN-IPSEC-PROFILE-INET
 set transform-set AES256/SHA/TRANSPORT
 set ikev2-profile DMVPN-IKE-PROFILE-INET
!
```

```
interface Tunnel200
 tunnel protection ipsec profile DMVPN-IPSEC-PROFILE-INET
!
crypto ipsec security-association replay window-size 1024
```

```
R31, R41, and R51
crypto ikev2 keyring DMVPN-KEYRING-INET
 peer ANY
  address 0.0.0.0 0.0.0.0
  pre-shared-key CISCO456
!
crypto ikev2 profile DMVPN-IKE-PROFILE-INET
 match fvrf INET01
 match identity remote address 0.0.0.0
 authentication remote pre-share
 authentication local pre-share
 keyring local DMVPN-KEYRING-INET
 dpd 40 5 on-demand
!
crypto ipsec transform-set AES256/SHA/TRANSPORT esp-aes 256 esp-sha-hmac
 mode transport
!
crypto ipsec profile DMVPN-IPSEC-PROFILE-INET
 set transform-set AES256/SHA/TRANSPORT
 set ikev2-profile DMVPN-IKE-PROFILE-INET
!
interface Tunnel200
 tunnel protection ipsec profile DMVPN-IPSEC-PROFILE-INET
!
crypto ipsec security-association replay window-size 1024
!
crypto isakmp nat keepalive 20
```

Verification of Encryption on IPsec Tunnels

Now that the DMVPN tunnels have been configured for IPsec protection, the status should be verified. The command **show dmvpn detail** provides the relevant IPsec information.

Example 5-10 demonstrates the command on R31. The output lists the status of the DMVPN tunnel, the underlay IP addresses, and packet counts. Examining the packet counts in this output is one of the steps that can be taken to verify that network traffic is being transmitted out of a DMVPN tunnel or received on a DMVPN tunnel.

Example 5-10 *Verification of IPsec DMVPN Tunnel Protection*

```
R31-Site3-Spoke# show dmvpn detail
! Output omitted for brevity
# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1 100.64.12.1      192.168.200.12    UP 00:03:39     S  192.168.200.12/32
    1 100.64.22.1      192.168.200.22    UP 00:00:57     S  192.168.200.22/32

Crypto Session Details:
--------------------------------------------------------------------------------
Interface: Tunnel200
Session: [0xE7192900]
  Session ID: 1
  IKEv2 SA: local 100.64.31.1/500 remote 100.64.12.1/500 Active
          Capabilities:(none) connid:1 lifetime:23:56:20
  Crypto Session Status: UP-ACTIVE
  fvrf: INET01, Phase1_id: 100.64.12.1
  IPSEC FLOW: permit 47 host 100.64.31.1 host 100.64.12.1
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 22 drop 0 life (KB/Sec) 4280994/3380
        Outbound: #pkts enc'ed 20 drop 0 life (KB/Sec) 4280994/3380
   Outbound SPI : 0x35CF62F4, transform : esp-256-aes esp-sha-hmac
    Socket State: Open

Interface: Tunnel200
Session: [0xE71929F8]
  Session ID: 1
  IKEv2 SA: local 100.64.31.1/500 remote 100.64.22.1/500 Active
          Capabilities:(none) connid:2 lifetime:23:59:03
  Crypto Session Status: UP-ACTIVE
  fvrf: INET01, Phase1_id: 100.64.22.1
  IPSEC FLOW: permit 47 host 100.64.31.1 host 100.64.22.1
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 11 drop 0 life (KB/Sec) 4306711/3542
        Outbound: #pkts enc'ed 9 drop 0 life (KB/Sec) 4306712/3542
   Outbound SPI : 0x366F5BFF, transform : esp-256-aes esp-sha-hmac
    Socket State: Open

Pending DMVPN Sessions:
```

The command **show crypto ipsec sa** includes additional information that was not included in the command **show dmvpn detail**. Example 5-11 displays explicit information about all the security associations. Examine the path MTU, tunnel mode, and replay detection.

Example 5-11 *Verification of IPsec Security Association*

```
R31-Site3-Spoke# show crypto ipsec sa

interface: Tunnel200
    Crypto map tag: Tunnel200-head-0, local addr 100.64.31.1

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (100.64.31.1/255.255.255.255/47/0)
   remote ident (addr/mask/prot/port): (100.64.22.1/255.255.255.255/47/0)
   current_peer 100.64.22.1 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 16, #pkts encrypt: 16, #pkts digest: 16
    #pkts decaps: 18, #pkts decrypt: 18, #pkts verify: 18
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

     local crypto endpt.: 100.64.31.1, remote crypto endpt.: 100.64.22.1
     plaintext mtu 1362, path mtu 1400, ip mtu 1400, ip mtu idb Tunnel200
     current outbound spi: 0x366F5BFF(913267711)
     PFS (Y/N): N, DH group: none

     inbound esp sas:
      spi: 0x66DD2026(1725767718)
        transform: esp-256-aes esp-sha-hmac ,
        in use settings ={Transport, }
        conn id: 4, flow_id: SW:4, sibling_flags 80000000, crypto map:
          Tunnel200-head-0
        sa timing: remaining key lifetime (k/sec): (4306710/3416)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE(ACTIVE)

     inbound ah sas:
     inbound pcp sas:

     outbound esp sas:
      spi: 0x366F5BFF(913267711)
        transform: esp-256-aes esp-sha-hmac ,
        in use settings ={Transport, }
        conn id: 3, flow_id: SW:3, sibling_flags 80000000, crypto map:
          Tunnel200-head-0
        sa timing: remaining key lifetime (k/sec): (4306711/3416)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE(ACTIVE)

     outbound ah sas:
     outbound pcp sas:
```

Note Using encryption over all transports allows for ease of deployment and troubleshooting workflows, because any and all transports are configured exactly the same. No special review or concern for traffic is needed because all paths are configured the same.

Private Key Infrastructure (PKI)

This chapter's first scenario for demonstrating DMVPN tunnel protection used pre-shared keys for simplicity. It is the easiest method to configure, but it can be difficult to manage across a large branch infrastructure. In addition, security can be compromised if a device is stolen or lost, thereby requiring a complete change of pre-shared keys.

Changing the authentication of devices to a *private key infrastructure (PKI)* provides a more secure, robust, and automated method of authentication. Every DMVPN router enrolls and maintains a digital certificate that provides more rigorous confirmation of the identity of both devices. If a device is stolen, the digital certificate for only the stolen device must be revoked in the PKI administration tool. No configuration changes are required on all the routers participating in the DMVPN network.

The basis of digital certificates is that there are two *Rivest-Shamir-Adleman (RSA)* keys. One of the keys is a public key and the other is a private key. Content is encrypted using one of the keys and then decrypted using the other key. In this concept, there is a concern that a device may not be sure of the integrity of the public key—whether the private key actually belongs to the other device.

A *certificate authority (CA)* acts as a trusted third party to establish the integrity of a device's public keys. The CA is responsible for issuing the public RSA key for each device and signs the certificate with its own private key. The device's public key now contains a hash of the CA's public key. The remote device can verify the integrity of the public key by comparing its own copy of the CA's public key hash with the one included in the remote device's public key. This allows both devices to be assured that the public key is unique and that the key they received has not been tampered with.

Both devices must trust the CA in order for the concept to work. In addition, the CA is responsible for canceling or revoking certificates that are no longer valid. The CA maintains a *certificate revocation list (CRL)* of the serial numbers of any certificates that it has revoked. The certificate contains the location of the *CRL distribution point (CDP)* so that any device can verify the validity (whether it was revoked) of any public certificate it received. In the event that a router is lost, stolen, or compromised, only that router's specific certificate needs to be revoked, and it is then listed on the CRL.

On a router, the *trustpoint* is the identification and parameters for a specific CA. The trustpoint also includes a PKI keychain that contains the public key of the CA. Although there are multiple methods of certificate-based authentication, IPsec authentication typically uses the verification of the RSA signature.

At a minimum, all devices check the certificate that they received from the remote device to ensure that it contains the same CA public key defined in the PKI trustpoint. This process does not require communication with the CA because the PKI keychain and trustpoint are statically defined on the router. During this process, the router checks the CRL to see if the certificate is valid and has not been revoked. In order to check the CRL, the device requires connectivity to the CDP.

Note A router can maintain a local cache of the CRL to reduce load on the CA. The CRL provides a CRL validity date that indicates when the CRL expires. A router does not refresh the cached copy of the CRL until the CRL validity date has been reached. The length of time that the CRL remains valid should be reviewed with your security team as a part of the deployment strategy.

In the DMVPN architecture, the hub routers provide connectivity to the corporate headquarters and to branch sites (through NHRP redirects). If a spoke router cannot connect to the hub router, it cannot establish a spoke-to-spoke tunnel. Because the DMVPN hub routers are the gatekeepers to the network, CRL checking should be enabled on them. Spoke routers need to verify that there is not a man-in-the-middle intrusion when connecting to a hub router. All that is required from a spoke's perspective is that the signature be checked to make sure it matches the PKI trustpoint.

This design strategy requires that hub routers always have connectivity to the CA (assuming that the CDP is the CA server), and that the spoke routers have access to the CA only during the initial certificate enrollment process.

Figure 5-6 shows the placement of an IOS CA in our reference architecture. The CA has interfaces that connect into the local LAN (10.1.1.1), the MPLS transport network (172.16.111.1), and the Internet transport (100.64.112.1).

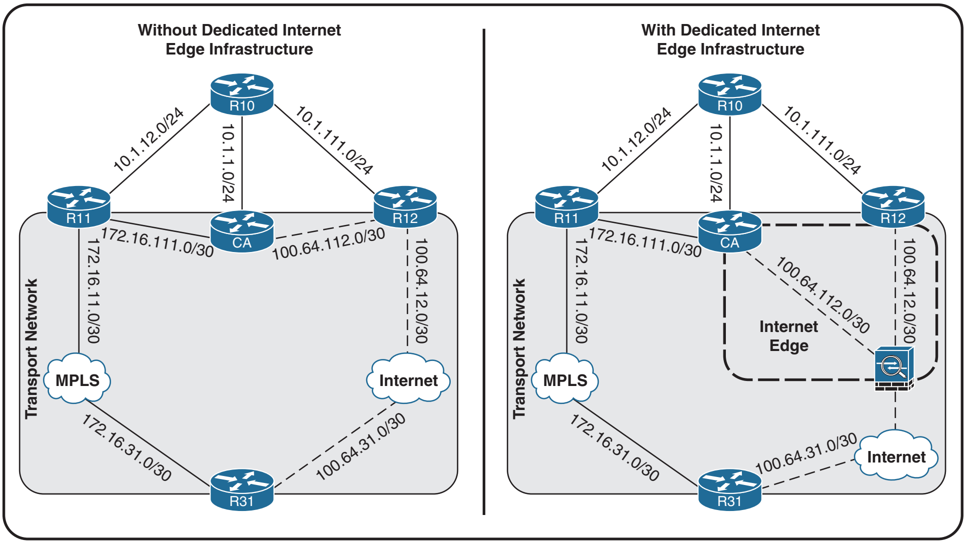


Figure 5-6 *Cisco IOS CA Deployment*

The DMVPN hub routers always connect to the CA's LAN address for any CA-related activities. As stated earlier, a certain level of trust is placed in the SP to ensure privacy, so the MPLS transport is deemed secure enough to expose the CA directly to that network. The Internet transport is deemed insecure and requires either a localized firewall instance (such as the Cisco Zone-Based Firewall) or a dedicated firewall in the organization's Internet edge portion of the WAN infrastructure. An exposed CA server can be a target for hackers or denial-of-service intrusions.

IOS Certificate Authority (CA) Server

As stated earlier, the IOS server has connectivity, as needed, to the LAN, MPLS network, and Internet. The MPLS and Internet interfaces should be associated to their own unique FVRF in a similar fashion to the DMVPN hub router. A simple static default route pointed to the default gateway for each segment is all that is needed. In addition, the MPLS transport network (172.16.111.0/30) needs to be advertised to be reachable via the MPLS provider network, and the Internet transport network (100.64.112.0/30) needs to be advertised to be reachable via the Internet.

The following process is used to configure a dedicated device as an IOS CA server:

Step 1. **Define the domain name.**

The router must have a fully qualified domain name (FQDN) as part of the cryptographic hash generation algorithm. The FQDN is a combination of the host name and the domain name. The domain name is set with the command **ip domain-name** *domain-name*.

Step 2. **Create an RSA key pair.**

The RSA key pair is created with the command **crypto key generate rsa** [**modulus** *modulus-size*]. A modulus size of 2048 increases the strength of the security.

Note As of December 1, 2013, all RSA key pairs must be greater than or equal to 2048 to be considered secure through 2030.

Step 3. **Enable the HTTP service.**

The HTTP service is enabled on the IOS-based CA router with the command **ip http server**. The CA service uses the default port 80 for issuing certificates.

The port can be changed to a different value with the command **ip http port** *port-number*. If the port is changed, the port must be specified in the enrollment URL in the PKI trustpoint configuration for the DMVPN hub and spoke routers.

Step 4. **Define the Cisco IOS CA instance.**

The CA service is defined with the command **crypto pki server** *ca-name*. All the CA configuration resides under the CA instance.

Step 5. **Specify what is kept in the CA database (optional but recommended).**

The CA stores only enough information in the CA database to prevent issuing new certificates without conflict. It is recommended that each certificate that is issued be written to the database with the command **database level complete**.

Step 6. **Specify the location of the database (optional but recommended).**

By default, the CA places database entries and certificates in the flash drive. Unless the router has a local file system that is capable of handling a large number of write operations and sufficient storage, the location may need to be changed. The location can be a remote server (FTP or removable flash drive). The command **database url** [**cnm | crl | crt | p12 | pem | ser**] *url* [**username** *username*] [**password** *password*] specifies the new location.

Step 7. **Disable database archiving (optional).**

The CA database archive function can be disabled with the command **no database archive**.

Step 8. **Define the distinguished name (DN) for the CA.**

Setting a DN on a certificate makes it easier to locate which CA issued the certificate, if multiple CAs exist in the organization. The DN is defined with the command **issuer-name** *dn-string*.

Generally the DN string contains

- Common name (CN) = the FQDN of the CA
- Organization (O) = organization or department name
- Locality name (L) = a city, town, or location
- State or province (ST) = the state or province where the CA resides
- Country (C) = the country where the CA resides

This chapter uses DN = CA-SITE1.IWAN.LAB L = LAB C = US.

Step 9. **Configure the CA to automatically grant CA requests (optional).**

Devices request a certificate from the CA. Approval of these requests is a manual process, which is more secure. The CA can be configured to automatically grant certificates upon enrollment with the command **grant auto**, but this is not as secure as manual administration.

The security team should review the use of automatic approval. Some organizations enable this feature strictly for simplicity during deployment and later change to a manual approval process.

Step 10. **Define the lifetime for certificates (optional but recommended).**

The lifetime of any certificates that are issued to routers should be set with the command **lifetime certificate** *days*. The recommended value is two years (730 days).

The lifetime of certificates that are issued for this CA or sub-CAs should be set with the command **lifetime ca-certificates** *days*. The recommended value is three years (1095 days).

Step 11. **Define the CRL lifetime (optional).**

By default, CRLs are issued every week (168 hours), which may leave a large window for access after a certificate is revoked. The CRL can be issued more frequently by changing the lifetime with the command **lifetime crl** *hours* [*minutes*]. Hours can be set anywhere from 0 to 336. The recommended CRL lifetime is 24 hours.

Step 12. **Define the CRL distribution point (CDP).**

The CDP is the location of the CRL that is made available for devices to check the validity of a certificate and is located in every certificate that this CA issues. The CDP is defined with the command **cdp-url** *url*. IOS-based CAs use the Simple Certificate Enrollment Protocol (SCEP)-based GetCRL function by setting the URL to http://*ca-ip-address*/cgi-bin/pkiclient.exe?operation=GetCRL.

Note IOS requires an escape sequence for placing the question mark (?) in the CLI. Otherwise the CLI provides context-sensitive help. Use Ctrl+Shift+V before trying to type the question mark.

Step 13. **Initiate the CA process.**

Now that the CA has been properly configured, it needs to be initiated with the command **no shutdown**. After the CA is initialized and running, the configuration is locked, which prevents making any changes to the configuration. To unlock the CA, the CA process must be shut down with the **shutdown** command.

Example 5-12 provides output of Figure 5-6's CA-Site 1 router configuration. When the CA instance starts up, it asks for a password that is used for encrypting the private keys. Write this password down and store it in a safe place.

Example 5-12 *Configuration of the CA-Site 1 CA Instance*

```
CA-Site1(cs-server)# ip domain-name IWAN.LAB
CA-Site1(config)# crypto key generate rsa modulus 2048
% The key modulus size is 2048 bits
% Generating 2048 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 1 seconds)
CA-Site1(config)# ip http server
*Dec 13 05:25:21.473: %SSH-5-ENABLED: SSH 1.99 has been enabled
CA-Site1(config)# crypto pki server CA-SITE1
CA-Site1(cs-server)# database level complete
CA-Site1(cs-server)# no database archive
CA-Site1(cs-server)# issuer-name CN=CA-SITE1.IWAN.LAB L=LAB C=US
CA-Site1(cs-server)# grant auto
*Dec 13 05:25:45.088: %PKI-6-CS_GRANT_AUTO: All enrollment requests will be
     automatically granted.
CA-Site1(cs-server)# lifetime certificate 730
CA-Site1(cs-server)# lifetime ca-certificate 1095
CA-Site1(cs-server)# lifetime crl 24
CA-Site1(cs-server)# cdp-url http://10.1.1.1/cgi-bin/pkiclient.exe?operation=GetCRL
CA-Site1(cs-server)# no shut
%Some server settings cannot be changed after CA certificate generation.
% Please enter a passphrase to protect the private key
% or type Return to exit
Password:
Re-enter password:
% Generating 1024 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 0 seconds)

% Certificate Server enabled.
*Dec 13 05:26:25.088: %PKI-6-CS_ENABLED: Certificate server now enabled.
```

As part of the initialization, CA-Site 1's public key is stored in the configuration as part of a keychain. This key is needed to verify the integrity of any certificates it creates. Example 5-13 displays the key so that the structure can be viewed.

Example 5-13 *CA-Site 1's Public Key*

```
CA-Site1# show running-config | section crypto pki certificate chain
crypto pki certificate chain CA-SITE1
 certificate ca 01
  30820227 30820190 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  27312530 23060355 0403131C 43412D53 49544531 2E495741 4E2E4C41 42204C3D
  4C414220 433D5553 301E170D 31353132 31343032 34303434 5A170D31 38313231
  33303234 3034345A 30273125 30230603 55040313 1C43412D 53495445 312E4957
  414E2E4C 4142204C 3D4C4142 20433D55 5330819F 300D0609 2A864886 F70D0101
  01050003 818D0030 81890281 810095BA E5FF31D9 8B2D37C7 15AAA897 E09BF4C5
  D3BC96DF CD65510A 52380267 E1A0F70C B3FFF599 47107357 7AA4BD4A 839C7F19
  23EA8059 D6BA7AA0 9477C8AC 61CBFAD5 810CCDF5 7FC6A364 EF72BAD6 1D1F5C74
  2C46BA81 E81AACE3 4524E417 C4A20DEE 5D4A977B 82730C3D D1CC6095 B4B568F8
  EA8700A5 FA0B1C29 BB97C851 78190203 010001A3 63306130 0F060355 1D130101
  FF040530 030101FF 300E0603 551D0F01 01FF0404 03020186 301F0603 551D2304
  18301680 145E7ADF CC5316AF 8CD7F3A4 FC6E405E E7D1F013 D4301D06 03551D0E
  04160414 5E7ADFCC 5316AF8C D7F3A4FC 6E405EE7 D1F013D4 300D0609 2A864886
  F70D0101 04050003 81810064 5F5E5A36 5CB21FB3 AC107472 3F8B2CA6 5240B3FE
  23398567 6A9517B0 BD7DF914 80E19D11 FEF572A2 58EA6F23 6BCE43F6 115C28F8
  526B2F30 94F4DF30 8613CF18 75A79FA1 DF7F7D37 51C159D8 568666A6 F3DA7E5A
  05566D67 CBDC3B55 0920B642 7C6D673A F2EC1DA2 D2E99A2A 164953D8 DE56174C
  13FF4926 4AAB651C 13AF1B
        Quit
```

During the initialization of the CA service, CA-Site 1 also installs the configuration from Example 5-14.

Example 5-14 *CA-Site 1's PKI Trustpoint*

```
CA-Site1# show running-config | section crypto pki trustpoint
crypto pki trustpoint CA-SITE1
 revocation-check crl
 rsakeypair CA-SITE1
```

The command **show crypto pki server** displays the relevant configuration settings for the CA server. Example 5-15 shows the output of the command. The CA cert fingerprint is of importance and will be used later on the devices as part of the PKI trustpoint configuration.

Example 5-15 *CA-Site 1's CA Settings*

```
CA-Site1# show crypto pki server
Certificate Server CA-SITE1:
    Status: enabled
    State: enabled
    Server's configuration is locked  (enter "shut" to unlock it)
    Issuer name: CN=CA-SITE1.IWAN.LAB L=LAB C=US
    CA cert fingerprint: D8C27666 B6974708 BA381547 B41CAFC6
    Granting mode is: auto
    Last certificate issued serial number (hex): 1
    CA certificate expiration timer: 05:31:13 EST Dec 13 2018
    CRL NextUpdate timer: 08:31:13 EST Dec 14 2015
    Current primary storage dir: nvram:
    Database Level: Complete - all issued certs written as <serialnum>.cer
```

Note Before performing the CA configuration, determine the values that should be used for certificate lifetime, FQDN, and CDP. After any certificates have been generated with these settings, new certificates need to be generated if those settings change.

DMVPN Hub PKI Trustpoints

Now that the CA server has been established, the DMVPN hub routers need to configure their PKI trustpoints to the CA's LAN address. These devices also perform CRL checking of the spoke certificates to ensure that only valid (nonrevoked) certificates are allowed to connect to the DMVPN network. The process for configuring the PKI trustpoints is as follows:

Step 1. **Define the domain name.**

The router must have an FQDN as part of the cryptographic hash generation algorithm. The FQDN is a combination of the host name and the domain name. The domain name is set with the command **ip domain-name** *domain-name*.

Step 2. **Define a PKI trustpoint for the CA.**

The PKI trustpoint defines the settings for a specific CA, along with the methods of verifying the validity of a certificate from that CA. The PKI trustpoint is defined with the command **crypto pki trustpoint** *trustpoint-name*. Although it is not a requirement, the trustpoint name should be standardized among devices that share the same function/characteristics to simplify operations.

Step 3. **Define the enrollment URL.**

The enrollment URL is the FQDN or IP address of the CA from which this device will request a certificate. The enrollment URL is defined with the command **enrollment url http**://*ca-ip-address*[:*port-number*]. Specifying the port number is optional.

The *ca-ip-address* is the address that is reachable via the LAN and is the 10.1.1.1 IP address from Figure 5-6.

Step 4. **Disable the placement of the router serial number in the certificate (optional).**

The certificate requests the router serial number as part of the certificate request. This feature can be disabled with the command **serial-number none**.

Step 5. **Define the FQDN of the router.**

The certificate includes the FQDN of the router. The FQDN provides a friendly, recognizable name to the certificate. The FQDN is a combination of the host name and the domain name and is defined with the command **fqdn** *fqdn*.

Step 6. **Define the IP address associated to the certificate (optional).**

The IP address of the requesting devices is added to the certificate and becomes a part of the IKEv2 identity. The IP address is defined with the command **ip-address** *ip-address*.

The Loopback0 IP address should be used for the IP address.

If this information is not configured in the PKI trustpoint, the router asks for it during certificate creation. Making it a part of the PKI trustpoint provides a quick reference to what is included in the request.

Step 7. **Define the certificate revocation password (optional).**

The certificate requires a revocation password upon its creation. The revocation password is defined with the command **password** *password*.

If this information is not configured in the PKI trustpoint, the router asks for it during certificate creation. Making it a part of the PKI trustpoint ensures a standardized revocation password.

Step 8. **Define the CA fingerprint.**

The CA fingerprint provides a form of authentication to prevent a potential man-in-the-middle intrusion. The fingerprint is available from the CA as shown earlier in Example 5-15. The fingerprint is defined with the command **fingerprint** *fingerprint*.

Step 9. Define the certificate check mechanism.

Within the trustpoint, the mechanism for checking the validity of a certificate is defined with the command **revocation-check** {**none** | **crl**}.

- The **none** option only ensures that the certificate was issued from the CA. The router never checks to see if the certificate has been revoked.
- The **crl** keyword checks to make sure the certificate was issued from the CA and that the certificate has not been revoked. If the certificate has been revoked and is found on the CRL, access is rejected. The router should always have access to the CRL in order to verify the validity of the certificate. If the router is set for **crl** and does not have access to reach the CRL list (CDP), the remote party will never successfully authenticate.

The DMVPN hubs have LAN access to the CA and use the **crl** option to verify that the certificates from a spoke have not been rejected.

Step 10. Define the RSA key pair.

The RSA key pair is created during the certificate request. The key pair size is defined with the command **rsakeypair** *rsa-keypair-label* [*general-purpose-length*] [*encryption-key-length*]. The recommended value for the key length is 2048.

Example 5-16 provides the PKI trustpoint configuration for R11. The FQDN and IP address need to be changed for deployment on the other DMVPN hub routers.

Example 5-16 *DMVPN Hub PKI Trustpoint Configuration*

```
R11
ip domain-name IWAN.LAB
!
crypto pki trustpoint CA-SITE1-HUB
 enrollment url http://10.1.1.1:80
 serial-number none
! The FQDN needs to change per router
 fqdn R11-Hub.IWAN.LAB
 password CISCO123
! The identifying IP address needs to change per router, and should match the
! Loopback 0 IP address.
 ip-address 10.1.0.11
! The Fingerprint was taken from Example 5-15
 fingerprint D8C27666 B6974708 BA381547 B41CAFC6
 revocation-check crl
 rsakeypair CA-SITE1 2048 2048
```

After the PKI trustpoint is defined, it needs to be authenticated with the command **crypto pki authenticate** *pki-trustpoint-name*. The router then connects to the CA's enrollment URL and verifies the fingerprint hash. Example 5-17 demonstrates the PKI trustpoint authentication on R11.

Example 5-17 *Authentication of DMVPN Hub PKI Trustpoint*

```
R11-DC1-Hub1(config)# crypto pki authenticate CA-SITE1-HUB
Certificate has the following attributes:
       Fingerprint MD5: D8C27666 B6974708 BA381547 B41CAFC6
      Fingerprint SHA1: FF2947B8 2062E013 11AD4292 3FA21089 AB79D6C4
Trustpoint Fingerprint: D8C27666 B6974708 BA381547 B41CAFC6
Certificate validated - fingerprints matched.
Trustpoint CA certificate accepted.
```

Now that R11 has authenticated the PKI trustpoint for the CA, the CA public keychain is downloaded and placed into the running configuration as shown in Example 5-18. Notice that the actual certificate information is the same as in Example 5-13.

Example 5-18 *Verification of CA Public Key Signature*

```
R11-Hub# show running-config | section crypto pki certificate chain
crypto pki certificate chain CA-SITE1-HUB
 certificate ca 01
  30820227 30820190 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  27312530 23060355 0403131C 43412D53 49544531 2E495741 4E2E4C41 42204C3D
  4C414220 433D5553 301E170D 31353132 31343032 34303434 5A170D31 38313231
  33303234 3034345A 30273125 30230603 55040313 1C43412D 53495445 312E4957
  414E2E4C 4142204C 3D4C4142 20433D55 5330819F 300D0609 2A864886 F70D0101
  01050003 818D0030 81890281 810095BA E5FF31D9 8B2D37C7 15AAA897 E09BF4C5
  D3BC96DF CD65510A 52380267 E1A0F70C B3FFF599 47107357 7AA4BD4A 839C7F19
  23EA8059 D6BA7AA0 9477C8AC 61CBFAD5 810CCDF5 7FC6A364 EF72BAD6 1D1F5C74
  2C46BA81 E81AACE3 4524E417 C4A20DEE 5D4A977B 82730C3D D1CC6095 B4B568F8
  EA8700A5 FA0B1C29 BB97C851 78190203 010001A3 63306130 0F060355 1D130101
  FF040530 030101FF 300E0603 551D0F01 01FF0404 03020186 301F0603 551D2304
  18301680 145E7ADF CC5316AF 8CD7F3A4 FC6E405E E7D1F013 D4301D06 03551D0E
  04160414 5E7ADFCC 5316AF8C D7F3A4FC 6E405EE7 D1F013D4 300D0609 2A864886
  F70D0101 04050003 81810064 5F5E5A36 5CB21FB3 AC107472 3F8B2CA6 5240B3FE
  23398567 6A9517B0 BD7DF914 80E19D11 FEF572A2 58EA6F23 6BCE43F6 115C28F8
  526B2F30 94F4DF30 8613CF18 75A79FA1 DF7F7D37 51C159D8 568666A6 F3DA7E5A
  05566D67 CBDC3B55 0920B642 7C6D673A F2EC1DA2 D2E99A2A 164953D8 DE56174C
  13FF4926 4AAB651C 13AF1B
        Quit
```

Now that the DMVPN hub router has authenticated the CA, it is time for the hub to submit a certificate request with the command **crypto pki enroll** *pki-trustpoint-name* as shown in Example 5-19. Notice in the output that there are two different MD5 and SHA-1 fingerprints; this is because two certificates were requested. One certificate is the signature certificate and the other one is the encryption certificate.

Example 5-19 *DMVPN Hub Router Certificate Request*

```
R11-Hub(config)# crypto pki enroll CA-SITE1-HUB
*Dec 13 06:22:45.178: %CRYPTO-6-AUTOGEN: Generated new 2048 bit key pair% Start
  certificate enrollment ..

% The subject name in the certificate will include: R11-Hub.IWAN.LAB
% The IP address in the certificate is 10.1.0.11

% Certificate request sent to Certificate Authority
% The 'show crypto pki certificate verbose CA-SITE1-HUB' command will show the
  fingerprint.

*Dec 13 06:22:47.530: %CRYPTO-6-AUTOGEN: Generated new 2048 bit key pair
*Dec 13 06:22:47.577: CRYPTO_PKI: Signature Certificate Request Fingerprint
     MD5: 4682781D 59CAF556 5AC343DE C876A275
*Dec 13 06:22:47.577: CRYPTO_PKI: Signature Certificate Request Fingerprint
     SHA1: D4F17076 B5234124 2D632EDF 5B3FC8B7 EDEEE1A6
*Dec 13 06:22:47.645: CRYPTO_PKI: Encryption Certificate Request Fingerprint
     MD5: C6DBB8D8 F4E98FAC A2B2DBBD CE979FC4
*Dec 13 06:22:47.645: CRYPTO_PKI: Encryption Certificate Request Fingerprint
     SHA1: 34D1C342 882B5254 2A8F38A6 7EE97E14 2F001FD5
Issuer-name  cn=CA-SITE1.IWAN.LAB L=LAB C=US
                      Subject-name  ipaddress=10.1.0.11+hostname=R11-Hub.IWAN.LAB
                      Serial-number 02
                      Auto-Renewal: Not Enabled
*Dec 15 03:46:22.869: %PKI-6-CERTRET: Certificate received from Certificate
  Authority
```

Note If the CA is not set up to automatically accept certificate requests, a request must be approved on the IOS CA with the command **crypto pki server** *ca-name* **grant** {**all** | *request-number*}. After the request is approved, the certificate automatically installs on the requesting router. By default, the router retries the request every minute. The status can be checked on the requesting router with the command **show crypto pki trustpoints status.**

After the CA has approved the certificate request, the certificate is sent to the router, which produces the syslog message "*%PKI-6-CERTRET: Certificate received from Certificate Authority*." The certificates for the router can be viewed with the command

show crypto pki certificates as shown in Example 5-20. Notice the certificate serial number, CRL, purpose, and validity dates.

Example 5-20 *Verification of Local Certificates*

```
R11-DC1-Hub1# show crypto pki certificates
Certificate
  Status: Available
  Certificate Serial Number (hex): 03
  Certificate Usage: Encryption
  Issuer:
    cn=CA-SITE1.IWAN.LAB L=LAB C=US
  Subject:
    Name: R11.IWAN.LAB
    IP Address: 10.1.0.11
    ipaddress=10.1.0.11+hostname=R11.IWAN.LAB
  CRL Distribution Points:
    http://10.1.1.1/cgi-bin/pkiclient.exe?operation=GetCRL
  Validity Date:
    start date: 12:19:11 EST Dec 12 2015
    end   date: 12:19:11 EST Dec 11 2017
  Associated Trustpoints: CA-SITE1

Certificate
  Status: Available
  Certificate Serial Number (hex): 02
  Certificate Usage: Signature
  Issuer:
    cn=CA-SITE1.IWAN.LAB L=LAB C=US
  Subject:
    Name: R11.IWAN.LAB
    IP Address: 10.1.0.11
    ipaddress=10.1.0.11+hostname=R11.IWAN.LAB
  CRL Distribution Points:
    http://10.1.1.1/cgi-bin/pkiclient.exe?operation=GetCRL
  Validity Date:
    start date: 12:19:11 EST Dec 12 2015
    end   date: 12:19:11 EST Dec 11 2017
  Associated Trustpoints: CA-SITE1

CA Certificate
  Status: Available
  Certificate Serial Number (hex): 01
  Certificate Usage: Signature
  Issuer:
    cn=CA-SITE1.IWAN.LAB L=LAB C=US
```

```
Subject:
  cn=CA-SITE1.IWAN.LAB L=LAB C=US
Validity Date:
  start date: 11:33:28 EST Dec 12 2015
  end   date: 11:33:28 EST Dec 11 2018
Associated Trustpoints: CA-SITE1
```

DMVPN Branch PKI Trustpoints

The next step is to enroll the branch routers with the CA. The process is the same as for the hub routers with two exceptions:

- The DMVPN spoke routers do not check the CRL for certificate revocation. It is assumed that the hub is safe and secure.
- The DMVPN tunnel is not established, so connectivity to the CA is through the use of the FVRF, which must be defined.

The process for configuring a PKI trustpoint and requesting a certificate on a DMVPN spoke router is as follows:

Step 1. **Define the domain name.**

The router must have an FQDN as part of the cryptographic hash generation algorithm. The FQDN is a combination of the host name and the domain name. The domain name is set with the command **ip domain-name** *domain-name*.

Step 2. **Define a PKI trustpoint for the CA.**

The PKI trustpoint defines the settings for a specific CA, along with the methods of verifying the validity of a certificate from that CA. The PKI trustpoint is defined with the command **crypto pki trustpoint** *trustpoint-name*. Although it is not a requirement, the trustpoint name should be standardized among devices that share the same function/characteristics to simplify operations.

Step 3. **Define the enrollment URL.**

The enrollment URL is the FQDN or IP address of the CA from which this device requests a certificate. The enrollment URL is defined with the command **enrollment url http:**//*ca-ip-address*[:*port-number*]. Specifying the port number is optional.

The *ca-ip-address* is the address that is reachable via the MPLS transport and is the 172.16.111.1 IP address from Figure 5-6.

Step 4. **Define the VRF used to reach the enrollment URL.**

The VRF must be defined if the router will use a VRF to reach the enrollment URL. The VRF is defined with the command **vrf** *vrf-name*.

Step 5. **Disable the placement of the router serial number from the certificate (optional).**

The certificate requests the router serial number as part of the certificate request. This feature can be disabled with the command **serial-number none**.

Step 6. **Define the FQDN.**

The certificate includes the FQDN of the router. The FQDN provides a friendly, recognizable name to the certificate. The FQDN is a combination of the host name and the domain name and is defined with the command **fqdn** *fqdn*.

Step 7. **Define the IP address associated to the certificate (optional).**

The IP address of the requesting devices is added to the certificate and becomes a part of the IKEv2 identity. The IP address is defined with the command **ip-address** *ip-address*.

The Loopback0 IP address should be used for the IP address.

If this information is not configured in the PKI trustpoint, the router asks for it during certificate creation. Making it a part of the PKI trustpoint provides a quick reference to what is included in the request.

Step 8. **Define the certificate revocation password (optional).**

The certificate requires a revocation password upon its creation. The revocation password is defined with the command **password** *password*.

If this information is not configured in the PKI trustpoint, the router asks for it during certificate creation. Making it a part of the PKI trustpoint ensures a standardized revocation password.

Step 9. **Define the CA fingerprint.**

The CA fingerprint provides a form of authentication to prevent a potential man-in-the-middle intrusion. The fingerprint is available from the CA as shown earlier in Example 5-15. The fingerprint is defined with the command **fingerprint** *fingerprint*.

Step 10. **Define the certificate check mechanism.**

Within the trustpoint, the mechanism for checking the validity of the certificate must be defined with the command **revocation-check** {**none** | **crl**}.

- The **none** option only ensures that the certificate was issued from the CA. The router never checks to see if the certificate has been revoked.

- The **crl** keyword checks to make sure the certificate was issued from the CA and that the certificate has not been revoked. If the certificate has been revoked and is found on the CRL, access is rejected. The router should always have access to the CRL in order to verify access. If the router is set for **crl** and does not have access to reach the CRL list (CDP), the remote party will never successfully authenticate.

The DMVPN spokes may not always have access to the CA and use the **none** option.

Note This recommendation is based on the fact that IOS CA has a basic configuration. A more refined security policy would have the CRL uploaded to a secured web server that provides SSL connectivity. If the hardened CRL is available to the spoke, the option of **crl** may be viable based on the needs of your organization. Hardening the IOS CA is outside the scope of this book. A dedicated out-of-band tunnel to the CA, which is explained later, can eliminate the need to harden the CA so that CRLs can be checked.

Step 11. **Define the RSA key pair.**

The RSA key pair is created during the certificate request. The key pair size is defined with the command **rsakeypair** *rsa-keypair-label* [*general-purpose-length*] [*encryption-key-length*]. The recommended value for the key length is 2048.

Example 5-21 demonstrates the configuration for R31 to configure the PKI trustpoint and request a certificate from the CA. The MPLS-based transport is assumed to be safe enough for certificate requests to be transferred, so R31 uses the MPLS transport for the request.

Example 5-21 *DMVPN Spoke PKI Trustpoint Configuration for MPLS VRF*

```
R31
ip domain-name IWAN.LAB
!
crypto pki trustpoint CA-SITE1-SPOKE-MPLS
 enrollment url http://172.16.111.1:80
 serial-number none
! The FQDN needs to change per router
 fqdn R31-Spoke.IWAN.LAB
 password CISCO123
! The identifying IP address needs to change per router, and should match the
! Loopback 0 IP address.
 ip-address 10.3.0.31
! The Fingerprint was taken from Example 5-15
 fingerprint D8C27666 B6974708 BA381547 B41CAFC6
```

```
! The VRF was selected because it will be used to reach the CA's interface
! exposed to the MPLS transport.
 vrf MPLS01
 revocation-check none
 rsakeypair CA-SITE1 2048 2048
```

Note A router needs to install only one certificate to authenticate multiple DMVPN tunnel interfaces that use a different transport.

As shown earlier in Figure 5-6, the CA may have connectivity to the Internet. Some organizations may not want the certificate to be requested on a public network in plaintext. In scenarios like those, a spoke router can use an existing secured connection to reach the CA.

For dual-router branch sites that connect via an MPLS network on one router, and then to the Internet via a second router, the Internet router can use the DMVPN tunnel (once established) from the MPLS network. The Internet branch router would specify the CA's LAN IP address (10.1.1.1) as part of the PKI trustpoint. Because the router uses the global routing table, a VRF does not need to be defined. Example 5-22 demonstrates the configuration that R52 would use after R51 receives its certificate and establishes DMVPN tunnel 100 to the headquarters network.

Example 5-22 *DMVPN Spoke PKI Trustpoint Configuration for Global*

```
R52
crypto pki trustpoint CA-SITE1-SPOKE-GLOBAL
 enrollment url http://10.1.1.1:80
 serial-number none
! The FQDN needs to change per router
 fqdn R52-Spoke.IWAN.LAB
 password CISCO123
! The identifying IP address needs to change per router, and should match the
! Loopback 0 IP address.
 ip-address 10.5.0.52
! The Fingerprint was taken from Example 5-15
 fingerprint D8C27666 B6974708 BA381547 B41CAFC6
! The revocation-check is set to none because this is a spoke router
 revocation-check none
 rsakeypair CA-SITE1 2048 2048
exit
```

After the PKI trustpoint is defined on R52, the trustpoint is authenticated, and then R52 submits a certificate request to the CA.

PKI IPsec Protection Configurations

Now that signed certificates from the CA have been installed on all the routers, the next phase is to configure the IPsec tunnel protection to use PKI for authentication. The configuration for PKI authentication is almost identical to the pre-shared key IPsec protection configuration with the following exceptions:

- An IKEv2 keyring is not needed.
- In the IKEv2 profile the local and remote authentications use the **rsa-sig** option in lieu of the **pre-share** option.
- The IKEv2 profile defines the PKI trustpoint that was used by the router with the command **pki trustpoint** *pki-trustpoint-name*.
- The IKEv2 profile contains the recommended command **identity local address** *ip-address*. The IP address specified should match the IP address defined in the PKI trustpoint (the Loopback0 interface).

Example 5-23 provides R11's, R31's, and R52's configuration for IPsec tunnel protection with PKI authentication. Because of the change in PKI trustpoint logic, the IKEv2 profile names have changed slightly to reflect the differences, which should help from a support perspective. The differentiating commands between pre-shared and PKI authentication have been highlighted.

Example 5-23 *PKI IPsec Tunnel Protection Configurations*

```
R11
! This template can be used for other DMVPN hub routers. Replace MPLS with
! INET for Internet transports. Primarily this is important for the FVRF.
crypto ikev2 profile DMVPN-IKE-PKI-PROFILE-MPLS-HUB
 match fvrf MPLS01
 match identity remote address 0.0.0.0
 identity local address 10.1.0.11
 authentication remote rsa-sig
 authentication local rsa-sig
 pki trustpoint CA-SITE1-HUB
!
crypto ipsec transform-set AES256/SHA/TRANSPORT esp-aes 256 esp-sha-hmac
 mode transport
!
crypto ipsec profile DMVPN-IPSEC-PROFILE-MPLS-HUB
 set transform-set AES256/SHA/TRANSPORT
 set ikev2-profile DMVPN-IKE-PKI-PROFILE-MPLS-HUB
!
interface Tunnel100
 tunnel protection ipsec profile DMVPN-IPSEC-PROFILE-MPLS-HUB
!
crypto ipsec security-association replay window-size 1024
```

```
R31
! This template can be used for other DMVPN spoke routers that connect to the
! MPLS transport.
crypto ikev2 profile DMVPN-IKE-PKI-PROFILE-MPLS-SPOKE
 match fvrf MPLS01
 match identity remote address 0.0.0.0
 identity local address 10.3.0.31
 authentication remote rsa-sig
 authentication local rsa-sig
 pki trustpoint CA-SITE1-SPOKE-MPLS
 crypto ikev2 dpd 40 5 on-demand
!
crypto ipsec transform-set AES256/SHA/TRANSPORT esp-aes 256 esp-sha-hmac
 mode transport
!
crypto ipsec profile DMVPN-IPSEC-PROFILE-MPLS-SPOKE
 set transform-set AES256/SHA/TRANSPORT
 set ikev2-profile DMVPN-IKE-PKI-PROFILE-MPLS-SPOKE
!
interface Tunnel100
 tunnel protection ipsec profile DMVPN-IPSEC-PROFILE-MPLS-SPOKE
!
crypto ipsec security-association replay window-size 1024
!
crypto isakmp nat keepalive 20
```

```
R52
! This template can be used for DMVPN spoke routers at dual-router branch sites
! that connect to the Internet transport and a DMVPN spoke router attached
! to the MPLS transport. The FVRF is not identified.
crypto ikev2 profile DMVPN-IKE-PKI-PROFILE-SPOKE-GLOBAL
 match identity remote address 0.0.0.0
 identity local address 10.5.0.52
 authentication remote rsa-sig
 authentication local rsa-sig
 pki trustpoint CA-SITE1-SPOKE-GLOBAL
 crypto ikev2 dpd 40 5 on-demand
!
crypto ipsec transform-set AES256/SHA/TRANSPORT esp-aes 256 esp-sha-hmac
 mode transport
!
crypto ipsec profile DMVPN-IPSEC-PROFILE-SPOKE-GLOBAL
 set transform-set AES256/SHA/TRANSPORT
 set ikev2-profile DMVPN-IKE-PKI-PROFILE-SPOKE-GLOBAL
!
```

```
interface Tunnel200
 tunnel protection ipsec profile DMVPN-IPSEC-PROFILE-SPOKE-GLOBAL
!
crypto ipsec security-association replay window-size 1024
!
crypto isakmp nat keepalive 20
```

Note Failing to set the **identity local address** in the IKEv2 still allows DMVPN tunnels to establish. However, the error message "*%CRYPTO-6-IKMP_NO_ID_CERT_ADDR_MATCH: (NOT ERROR BUT WARNING ONLY) ID of 172.16.21.1 (type 1) and certificate addr with 10.2.0.21*" will occur every time an IPsec tunnel is initialized between devices. This creates noise in the router's syslog and may cause unnecessary concern among support staff.

Certificate Registration with Out-of-Band Management Tunnel

If an organization does not want to request the certificate via the Internet but uses the Internet only as a transport, there are two remaining solutions:

- Manually create the request, generate a certificate, and install the certificate through manual file extraction and import techniques. This requires copying and transferring the files between the router and a laptop on which to perform the certificate request functions.
- Create an out-of-band DMVPN tunnel (using pre-shared authentication) that attaches the spoke to the CA. The tunnel is used only for authenticating the PKI trustpoint and creating certificate requests. There should be no routing protocols running across it. The DMVPN hub should have an inbound access list that permits only ICMP and port 80 to the CA. The ACL ensures that the spoke router cannot communicate with another spoke router.

Figure 5-7 depicts the topology where the branch routers establish DMVPN tunnel 10 directly with the CA using pre-shared keys. This tunnel is established for requesting and verifying certificates. An ACL is placed on the CA to prevent spoke-to-spoke network traffic from establishing. Tunnel 200 is established using PKI authentication. Tunnel 10 can be used for CRL checking from a branch router perspective.

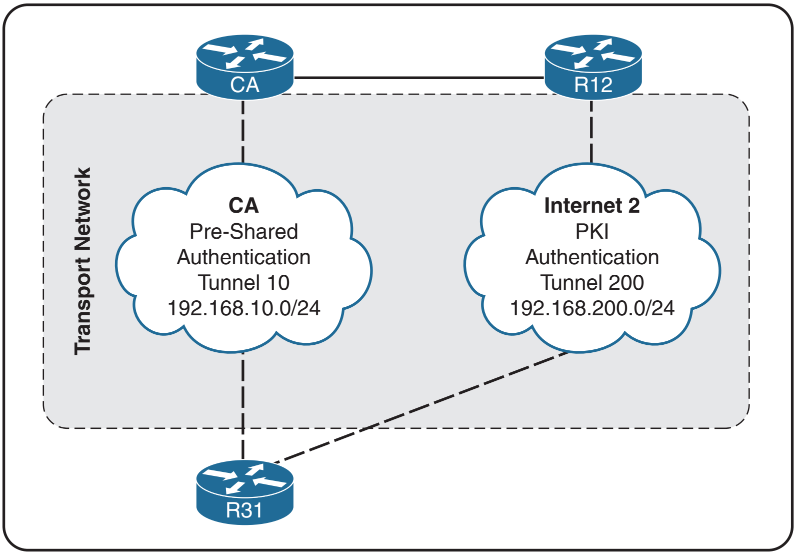


Figure 5-7 *Dedicated Out-of-Band DMVPN Tunnel for CA Management*

Example 5-24 provides the configuration for the out-of-band certificate management DMVPN tunnel 10. The significant change in the configuration is that NHRP redirect is not enabled on the CA server, and an inbound ACL is applied to the tunnel interface.

Example 5-24 *CA DMVPN Hub Tunnel Configuration*

```
CA Server
interface Tunnel10
 description DMVPN-CA-Point-to-Point
 bandwidth 1000
 ip address 192.168.10.1 255.255.255.0
ip access-group ACL-CA-ACCESS-ONLY in
ip access-group ACL-CA-ACCESS-REPLY out
 no ip redirects
 ip mtu 1400
 ip nhrp authentication CISCO-CA
 ip nhrp network-id 10
 ip nhrp holdtime 600
 ip tcp adjust-mss 1360
 load-interval 30
 tunnel source GigabitEthernet0/2
 tunnel mode gre multipoint
 tunnel key 10
 tunnel vrf INET01
 tunnel protection ipsec profile DMVPN-IPSEC-PROFILE-CA
!
```

```
ip access-list extended ACL-CA-ACCESS-ONLY
 permit icmp any host 192.168.10.1
 permit tcp any host 192.168.10.1 eq www
!
ip access-list extended ACL-CA-ACCESS-REPLY
 permit icmp host 192.168.10.1 any
 permit tcp host 192.168.10.1 eq www any
!
crypto ikev2 keyring DMVPN-KEYRING-CA
 peer ANY
  address 0.0.0.0 0.0.0.0
  pre-shared-key CISCO456
!
crypto ikev2 profile DMVPN-IKE-PROFILE-CA
 match fvrf INET01
 match identity remote address 0.0.0.0
 authentication remote pre-share
 authentication local pre-share
 keyring local DMVPN-KEYRING-CA
!
crypto ipsec transform-set AES256/SHA/TRANSPORT esp-aes 256 esp-sha-hmac
 mode transport
!
crypto ipsec profile DMVPN-IPSEC-PROFILE-CA
 set transform-set AES256/SHA/TRANSPORT
 set ikev2-profile DMVPN-IKE-PROFILE-CA
!
crypto ipsec security-association replay window-size 1024
!
crypto isakmp nat keepalive 20
```

Example 5-25 demonstrates the configuration for the CA DMVPN spoke tunnel on a router, with the assumption that R31 connects to only two different Internet transports. The DMVPN tunnel configuration does not have NHRP shortcuts enabled. Notice that the destination address is more refined, because the CA address is explicitly defined.

Example 5-25 *CA DMVPN Spoke Tunnel Configuration*

```
R31
interface Tunnel10
 description DMVPN-CA-Point-to-Point
 bandwidth 1000
 ip address 192.168.10.31 255.255.255.0
 no ip redirects
 ip mtu 1400
 ip nhrp authentication CISCO-CA
 ip nhrp network-id 10
 ip nhrp holdtime 600
 ip nhrp nhs 192.168.10.1 nbma 100.64.112.1 multicast
 ip tcp adjust-mss 1360
 tunnel source GigabitEthernet0/2
 tunnel mode gre multipoint
 tunnel key 10
 tunnel vrf INET01
 tunnel protection ipsec profile DMVPN-IPSEC-PROFILE-CA
!
crypto ikev2 keyring DMVPN-KEYRING-CA
 peer ANY
  address 100.64.112.1 255.255.255.255
  pre-shared-key CISCO456
!
crypto ikev2 profile DMVPN-IKE-PROFILE-CA
 match fvrf INET01
 match identity remote address 0.0.0.0
 authentication remote pre-share
 authentication local pre-share
 keyring local DMVPN-KEYRING-CA
 crypto ikev2 dpd 40 5 on-demand
!
crypto ipsec transform-set AES256/SHA/TRANSPORT esp-aes 256 esp-sha-hmac
 mode transport
!
crypto ipsec profile DMVPN-IPSEC-PROFILE-CA
 set transform-set AES256/SHA/TRANSPORT
 set ikev2-profile DMVPN-IKE-PROFILE-CA
!
crypto ipsec security-association replay window-size 1024
!
crypto isakmp nat keepalive 20
```

Example 5-26 demonstrates the PKI trustpoint configuration that R31 uses to request the certificate from the CA. Notice that the enrollment URL uses the CA's DMVPN tunnel IP address of 192.168.10.1.

Example 5-26 *Spoke PKI Trustpoint Configuration for the CA DMVPN Tunnel*

```
R31
crypto pki trustpoint CA-SITE1-SPOKE-CA
 enrollment url http://192.168.10.1:80
 serial-number none
 password CISCO123
 fqdn R31-Spoke.IWAN.LAB
 ip-address 10.3.0.31
 fingerprint D8C27666 B6974708 BA381547 B41CAFC6
 revocation-check none
 rsakeypair CA-SITE1 2048 2048
exit
```

After the PKI trustpoint has been authenticated, the certificate can be requested. This tunnel can be left up for CRL checking from a spoke router site or shut down if CRL checking is not being performed. This topic should be reviewed with the security team for their input.

IKEv2 Protection

Protecting the router from various IKE intrusion methods was the key reason for the development of IKEv2 based on prior known IKEv1 limitations. The first key concept is limiting the number of packets required to process IKE establishment. CPU utilization increases for every SA state it maintains along with the negotiation of a session. During high CPU utilization, a session that has started may not complete because other sessions are consuming limited CPU resources. Problems can occur when the number of expected sessions is different from the number of sessions that can be established. Limiting the number of sessions that can be in negotiation keeps the CPU resources down so that the expected number of established sessions can be obtained.

The command **crypto ikev2 limit** {**max-in-negotation-sa** *limit* | **max-sa** *limit*} [**outgoing**] limits the number of sessions being established or that are allowed to establish.

- The **max-sa** keyword limits the total count of SAs that a router can establish under normal conditions. The value should be set to double the number of ongoing sessions in order to achieve renegotiation.
- To limit the number of SAs being negotiated at one time, the **max-in-negotiation-sa** keyword should be used.
- To protect IKE from half-open sessions, a cookie can be used to validate that sessions are valid IKEv2 sessions and not a denial-of-service intrusion. The command

crypto ikev2 cookie-challenge *challenge-number* defines the threshold of half-open SAs before issuing an IKEv2 cookie challenge.

In Example 5-27, R41 limits the number of SAs to 10, the number in negotiation to six, and an IKEv2 cookie challenge for sessions above four. R41 has four static sessions to the hub routers (R11, R12, R21, and R22) and is limited to six additional sessions that all use the IKEv2 cookie challenge.

The command **show crypto ikev2 stats** displays the SA restrictions and that the four sessions are currently established to the four DMVPN hub routers.

Example 5-27 *Crypto IKEv2 Limit Configuration*

```
R41-Spoke(config)# crypto ikev2 limit max-sa 10
R41-Spoke(config)# crypto ikev2 limit max-in-negotation-sa 6 outgoing
R41-Spoke(config)# crypto ikev2 limit max-in-negotation-sa 6
R41-Spoke(config)# crypto ikev2 cookie-challenge 4
R41-Spoke(config)# end
```

```
R41-Spoke# show crypto ikev2 stats
-------------------------------------------------------------------------------
                          Crypto IKEv2 SA Statistics
-------------------------------------------------------------------------------
System Resource Limit:   0      Max IKEv2 SAs: 10    Max in nego(in/out): 6/6
Total incoming IKEv2 SA Count:    0        active:        0     negotiating: 0
Total outgoing IKEv2 SA Count:    4        active:        4     negotiating: 0
Incoming IKEv2 Requests: 3        accepted:      3       rejected:     0
Outgoing IKEv2 Requests: 16       accepted:      16      rejected:     0
Rejected IKEv2 Requests: 0        rsrc low:      0       SA limit:     0
IKEv2 packets dropped at dispatch: 0
Incoming IKEV2 Cookie Challenged Requests: 0
    accepted: 0        rejected: 0        rejected no cookie: 0
Total Deleted sessions of Cert Revoked Peers: 0
```

Basic IOS CA Management

The day-to-day maintenance and guidelines for managing an IOS CA are outside the scope of this book, but a list of some basic commands for the management of the IOS CA is provided here:

- **show crypto pki server** *ca-name* **requests:** Displays a list of requests, fingerprint, and subject name (FQDN + IP address)
- **show crypto pki server** *ca-name* **certificates** [*certificate-number* | **expired**]: Displays the certificates that have been assigned by the CA, including the issuance date, the expiration date, and the subject name

- **show crypto pki server** *ca-name* **crl**: Displays the time for the next CRL update, CRL size, revoked certificates, and their serial numbers
- **crypto pki server** *ca-name* **grant {all |** *request-number***}**: Approves all or the specified certificate request
- **crypto pki server** *ca-name* **reject {all |** *request-number***}**: Denies all or the specified certificate request
- **crypto pki server** *ca-name* **revoke** *certificate-serial-number*: Revokes an issued certificate and marks the certificate so that it is added the next time the CRL is updated
- **crypto pki server** *ca-name* **unrevoke** *certificate-serial-number*: Makes a revoked certificate valid again and removes the certificate from the CRL
- **clear crypt pki crl**: Purges the cached CRL on a DMVPN router, typically run on the DMVPN hub router and requiring the router to retrieve the CRL from the CA before the cached copy expires

Resources for additional commands and guidelines for the management of the IOS CA can be found in the "Further Reading" section at the end of this chapter.

Securing Routers That Connect to the Internet

The primary function of a router is to route packets. When a router connects to the Internet, it routes traffic from the Internet into your internal network. The router does not examine the packets to see if users on the Internet are malicious or not.

Network engineers need to plan and design an architecture that protects their network from multiple threats. Following the principle of least privileges, a user should be able to access only resources that he or she requires for legitimate purposes. This requires restricting access to the network to only the IP addresses or protocol ports that are necessary to establish connectivity.

There are two techniques for restricting access on a router:

- Access control lists (ACLs)
- Cisco Zone-Based Firewall (ZBFW)

Access Control Lists (ACLs)

Access-control lists (ACLs) provide the capability to filter network packets that flow into or out of a network interface. ACLs can also provide a method of packet classification for a variety of features, such as QoS or identifying networks within routing protocols.

ACLs are composed of access control entries (ACEs), which identify the action to be taken (permit or deny) and the relevant packet classification for that action. Packet classification starts at the top (lowest sequence) and proceeds down (higher sequence) until a matching pattern is identified. After a match is found, the appropriate action (permit or deny) is taken and processing stops. At the end of every ACL is an implicit "deny all ACEs," which denies all packets that did not match earlier in the ACL.

ACLs are classified as standard (based purely on the source network) or extended (which identifies packets on source, destination, protocol, port, or any combination of attributes).

Standard ACLs use the numbered entry 1–99, 1300–1999 or can be named. Extended ACLs use the numbered entry 100–199, 2000–2699 or can be named too. Because ACLs are used for things besides filtering, a named ACL can indicate its purpose or function, which can simplify troubleshooting. Extended ACLs provide the most granular access and are created with the following steps:

Step 1. **Define the ACL.**

Define the ACL with the command **ip access-list extended** {*acl-number* | *acl-name*} and place the CLI in ACL configuration mode.

Step 2. **Configure ACE entries.**

Configure the specific ACE entry with the command [*sequence*] {**permit** | **deny** } *protocol source source-wildcard destination destination-wildcard*[**log**].

The optional **log** keyword logs the packet information. Some organizations place an implicit **deny ip any any log** so that any traffic that was not permitted in the ACL is logged to the syslog.

From a connectivity standpoint, the ACL needs to permit the IKEv2, ESP, and DHCP, if configured, on the transport interface. From an operational perspective, the router should be able to ping with Internet Control Message Protocol (ICMP) and perform traceroutes. Traceroutes use UDP ports 33434 to 33463.

Example 5-28 displays a sample ACL that permits the minimal number of ports to provide connectivity and operational support.

Example 5-28 *ACL for Interfaces Connected to the Internet*

```
ip access-list extended ACL-INTERNET-TO-ROUTER
 ! IPsec via NAT-Traversal
 permit udp any any eq non500-isakmp
 ! ISAKMP (UDP Port 500) and IKEv2 key exchange
 permit udp any any eq isakmp
 ! IPSEC
 permit esp any any
 ! Allow DHCP
 permit udp any any eq bootpc
 ! Allow remote pings
```

```
permit icmp any any echo
! Allow ping replies (from our requests)
permit icmp any any echo-reply
! Following 2 entries allow traceroute replies (from our requests)
permit icmp any any ttl-exceeded
permit icmp any any port-unreachable
! Allow remote traceroute
permit udp any any range 33434 33463 ttl eq 1
```

The command **show ip access-list** lists all the access lists on a router. In addition, as packets match an entry, the router keeps count of the matches as shown in Example 5-29.

Example 5-29 *Verification of ACL ACEs*

```
R41-Spoke# show ip access-list
Extended IP access list ACL-INTERNET-TO-ROUTER
    10 permit udp any any eq non500-isakmp
    20 permit udp any any eq isakmp (15 matches)
    30 permit esp any any (73 matches)
    40 permit udp any any eq bootpc (2 matches)
    50 permit icmp any any echo (2 matches)
    60 permit icmp any any echo-reply (5 matches)
    70 permit icmp any any ttl-exceeded
    80 permit icmp any any port-unreachable
    90 permit udp any any range 33434 33463 ttl eq 1
```

Zone-Based Firewalls (ZBFWs)

ACLs control access based on protocol, source IP address, destination IP address, and ports used. Unfortunately they are stateless and do not inspect the packet's payload to detect if malicious hackers are using a port that they have found open. Stateful firewalls are capable of looking into Layers 4 through 7 of a network packet and verifying the state of the transmission. A stateful firewall can detect if a port is being piggybacked and can mitigate DDoS intrusions.

The Cisco *Zone-Based Firewall (ZBFW)* is the latest integrated stateful firewall included in the Cisco IOS-based operating systems. ZBFW reduces the need for a second device at a branch site to provide stateful network security.

ZBFW uses a flexible and straightforward approach to providing security by establishing security zones. Router interfaces are assigned to a specific zone, which can maintain a one-to-one or many-to-one relationship. A zone establishes a security border on the network and defines acceptable traffic that is allowed to pass between zones. By default, interfaces in the same security zone can communicate freely with each other, but interfaces in different zones cannot communicate with each other.

Figure 5-8 illustrates the concept of the ZBFW and the association of interfaces to a security zone.

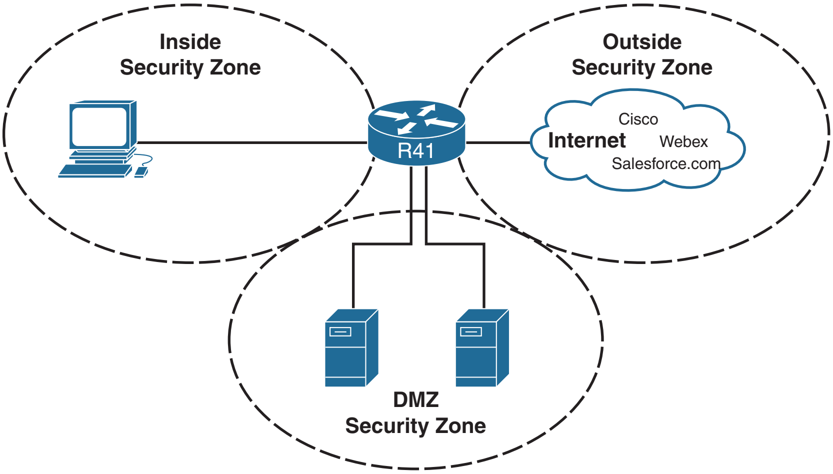


Figure 5-8 *Zone-Based Firewall and Security Zones*

Within the ZBFW architecture, there are two system-built zones: self and default.

Self

This is a system-level zone and includes all the router's IP addresses. By default, traffic to and from this zone is permitted to support management (Secure Shell [SSH] Protocol, Simple Network Management Protocol [SNMP]) and control plane (EIGRP, BGP, and so on) functions.

After a policy is applied to the self zone and another security zone, interzone communication must be explicitly defined.

Default

This is a system-level zone, and any interface that is not a member of another security zone is placed in this zone automatically.

When an interface that is not in a security zone sends traffic to an interface that is in a security zone, the traffic is dropped. Most network engineers assume that a policy cannot be configured to permit these traffic flows, but the solution involves the enablement of the *default zone*. Upon initialization of this zone, any interface not associated to a security zone is placed in this zone. When the unassigned interfaces are in the default zone, a policy map can be created between the two security zones.

The current iteration of PfRv3 uses dynamic autotunnels that cannot be configured or assigned to a security zone. When the default security zone is used for all the inside interfaces, security policies can be built as needed and PfRv3 can work properly.

ZBFW Configuration

This section explains the process for configuring a ZBFW on R41. In the book's topology, the Internet interface (Gi0/2) is assigned to the Outside zone. Initial configuration of the ZBFW requires communication with the Outside and Self zones and requires the creation of the following two inspection policies:

- **Outside-to-Self policy:**
  - Allows receipt of DHCP server traffic like DHCPOffer packets.
  - Allows receipt of IPsec packets from all hosts. IPsec packets can be received from other branch routers that initiate spoke-to-spoke DMVPN tunnels.
  - Allows receipt of and replies to basic ICMP responses for connectivity testing.
- **Self-to-Outside policy:**
  - Allows the transmission of DHCP client traffic such as DHCPDiscover and DHCPDiscover packets
  - Allows the initiation of IPsec packets to all hosts
  - Allows the transmission of all ICMP packets
  - Denies and logs all other traffic out of the Outside security zone.

Note If the router is used for direct Internet access, the MPLS interface (Gi0/1), DMVPN tunnels (100 and 200), and LAN interface (Gi0/3) are associated to the default zone. This configuration is not required initially.

The ZBFW is configured in five steps. The following section explains each of the steps as the Outside-to-Self policy is created:

Step 1. **Define the security zones.**

Security zones are configured using the command **zone security** *zone-name*. A zone needs to be created for the Outside (Internet). The Self zone is defined automatically. Example 5-30 demonstrates the configuration of a security zone.

Example 5-30 *Configuration to Define the Outside Security Zone*

```
zone security OUTSIDE
 description OUTSIDE Zone used for Internet Interface
```

Step 2. **Define the inspection class map.**

The class map for inspection defines a method for classification of traffic. The class map is configured using the command **class-map type inspect [match-all | match-any]** *class-name*. The **match-all** keyword requires that network traffic match all the conditions listed in the class map to qualify (Boolean AND), whereas the **match-any** requires that network traffic match only one of the conditions in the class map to qualify (Boolean OR). If neither keyword is specified, the **match-all** function is selected. Example 5-31 demonstrates the configuration of inspection class maps and the supporting ACLs.

Example 5-31 *Inspect Class Map Configuration*

```
ip access-list extended ACL-IPSEC
 permit udp any any eq non500-isakmp
 permit udp any any eq isakmp
ip access-list extended ACL-PING-AND-TRACEROUTE
 permit icmp any any echo
 permit icmp any any echo-reply
 permit icmp any any ttl-exceeded
 permit icmp any any port-unreachable
 permit udp any any range 33434 33463 ttl eq 1
ip access-list extended ACL-ESP
 permit esp any any
ip access-list extended ACL-DHCP-IN
 permit udp any eq bootps any eq bootpc
ip access-list extended ACL-GRE
 permit gre any any
!
class-map type inspect match-any CLASS-OUTSIDE-TO-SELF-INSPECT
 match access-group name ACL-IPSEC
 match access-group name ACL-PING-AND-TRACEROUTE
class-map type inspect match-any CLASS-OUTSIDE-TO-SELF-PASS
 match access-group name ACL-ESP
 match access-group name ACL-DHCP-IN
 match access-group name ACL-GRE
```

The configuration of inspect class maps can be verified with the command **show class-map type inspect** [*class-name*] as shown in Example 5-32.

Example 5-32 *Verification of Inspect Class Maps*

```
R41-Spoke# show class-map type inspect
 Class Map type inspect match-any CLASS-OUTSIDE-TO-SELF-PASS (id 2)
   Match access-group name ACL-ESP
   Match access-group name ACL-DHCP-IN
   Match access-group name ACL-GRE

 Class Map type inspect match-any CLASS-OUTSIDE-TO-SELF-INSPECT (id 1)
   Match access-group name ACL-IPSEC
   Match access-group name ACL-PING-AND-TRACEROUTE
```

Step 3. **Define the inspection policy map.**

The inspection policy map applies firewall policy actions to the class maps defined in the policy map. The policy map is then associated to a zone pair.

The inspection policy map is defined with the command **policy-map type inspect** *policy-name*. After the policy map is defined, the various class maps are defined with the command **class type inspect** *class-name*. Under the class map, the firewall action is defined with these commands:

- **drop** [**log**]: This is the default action and silently discards packets that match the class map. The **log** keyword adds syslog information that includes source and destination information (IP address, port, and protocol).
- **pass** [**log**]: This action makes the router forward packets from the source zone to the destination zone. Packets are forwarded in only one direction. A policy must be applied for traffic to be forwarded in the opposite direction. The **pass** action is useful for protocols like IPsec ESP and other inherently secure protocols with predictable behavior. The optional **log** keyword adds syslog information that includes the source and destination information.
- **Inspect**: The inspect action offers state-based traffic control. The router maintains connection/session information and permits return traffic from the destination zone without the need to specify it in a second policy.

The inspect policy map has an implicit class default that uses a default **drop** action. This provides the same implicit "deny all" that is found in an ACL. Adding it to the configuration may simplify troubleshooting for junior network engineers.

Example 5-33 demonstrates the configuration of the inspect policy map. Notice that in the class default class, the **drop** command does not have the **log** keyword because of the potential to fill up the syslog.

Example 5-33 *Inspection Policy Map Configuration*

```
policy-map type inspect POLICY-OUTSIDE-TO-SELF
 class type inspect CLASS-OUTSIDE-TO-SELF-INSPECT
   inspect
 class type inspect CLASS-OUTSIDE-TO-SELF-PASS
   pass
 class class-default
   drop
```

The inspection policy map can be verified with the command **show policy-map type inspect** [*policy-name*] as shown in Example 5-34.

Example 5-34 *Verification of the Inspection Policy Map*

```
R41-Spoke# show policy-map type inspect
  Policy Map type inspect POLICY-OUTSIDE-TO-SELF
    Class CLASS-OUTSIDE-TO-SELF-INSPECT
      Inspect
    Class CLASS-OUTSIDE-TO-SELF-PASS
      Pass
    Class class-default
      Drop
```

Step 4. **Define the zone pairs.**

A policy map is now applied to a traffic flow source to a destination configured as **zone-pair security** *zone-pair-name* **source** *source-zone-name* **destination** *destination-zone-name*. The inspection policy map is then applied to the zone pair with the command **service-policy type inspect** *policy-name*. Traffic is statefully inspected between the source and destination, and return traffic is allowed. Example 5-35 defines the zone pairs and associates the policy map to the zone pair.

Example 5-35 *ZBFW Zone Pair Configuration*

```
zone-pair security OUTSIDE-TO-SELF source OUTSIDE destination self
  service-policy type inspect POLICY-OUTSIDE-TO-SELF
```

Note The order of the zone pair is significant; the first zone indicates the source zone and the second zone indicates the destination zone. A second zone pair needs to be created with bidirectional traffic patterns when the **pass** action is selected.

Step 5. **Apply the security zones to the appropriate interfaces.**

An interface is assigned to the appropriate zone by entering the interface configuration submode with the command **interface** *interface-id* and associating the interface to the correct zone with the command **zone-member security** *zone-name* as defined in Step 1.

Example 5-36 demonstrates the Outside security zone being associated to GigabitEthernet 0/2.

Example 5-36 *Application of the Security Zone to the Interface*

```
interface GigabitEthernet 0/2
  zone-member security OUTSIDE
```

Now that the Outside-to-Self policy has been fully defined, traffic statistics can be viewed with the command **show policy-map type inspect zone-pair** [*zone-pair-name*]. Example 5-37 demonstrates the verification of the configured ZBFW policy.

Example 5-37 *Verification of the Outside-to-Self Policy*

```
R41-Spoke# show policy-map type inspect zone-pair

policy exists on zp OUTSIDE-TO-SELF
  Zone-pair: OUTSIDE-TO-SELF

  Service-policy inspect : POLICY-OUTSIDE-TO-SELF

    Class-map: CLASS-OUTSIDE-TO-SELF-INSPECT (match-any)
      Match: access-group name ACL-IPSEC
        2 packets, 208 bytes
        30 second rate 0 bps
      Match: access-group name ACL-PING-AND-TRACEROUTE
        0 packets, 0 bytes
        30 second rate 0 bps

   Inspect
        Packet inspection statistics [process switch:fast switch]
        udp packets: [4:8]

        Session creations since subsystem startup or last reset 2
        Current session counts (estab/half-open/terminating) [0:0:0]
        Maxever session counts (estab/half-open/terminating) [2:1:0]
        Last session created 00:03:39
        Last statistic reset never
        Last session creation rate 0
        Maxever session creation rate 2
        Last half-open session total 0
        TCP reassembly statistics
        received 0 packets out-of-order; dropped 0
        peak memory usage 0 KB; current usage: 0 KB
        peak queue length 0


    Class-map: CLASS-OUTSIDE-TO-SELF-PASS (match-any)
      Match: access-group name ACL-ESP
        186 packets, 22552 bytes
        30 second rate 0 bps
      Match: access-group name ACL-DHCP-IN
        1 packets, 308 bytes
        30 second rate 0 bps
      Match: access-group name ACL-GRE
        0 packets, 0 bytes
        30 second rate 0 bps
      Pass
        187 packets, 22860 bytes

    Class-map: class-default (match-any)
      Match: any
      Drop
        30 packets, 720 bytes
```

Note Making the class maps more explicit and thereby adding more of the explicit class maps to the policy map provides more visibility to the metrics.

Even though the ACLs are not used for blocking traffic, the counters do increase as packets match the ACL entries for the inspect class maps as demonstrated in Example 5-38.

Example 5-38 *ACL Counters from the Inspect Class Maps*

```
R41-Spoke# show ip access
Extended IP access list ACL-DHCP-IN
    10 permit udp any eq bootps any eq bootpc (1 match)
Extended IP access list ACL-ESP
    10 permit esp any any (170 matches)
Extended IP access list ACL-GRE
    10 permit gre any any
Extended IP access list ACL-IPSEC
    10 permit udp any any eq non500-isakmp
    20 permit udp any any eq isakmp (2 matches)
Extended IP access list ACL-PING-AND-TRACEROUTE
    10 permit icmp any any echo
    20 permit icmp any any echo-reply
    30 permit icmp any any ttl-exceeded
    40 permit icmp any any port-unreachable
    50 permit udp any any range 33434 33463 ttl eq 1
```

Now that the Outside-to-Self policy has been defined, it is time to verify that the DMVPN tunnels have established connectivity for the Internet transport as seen in Example 5-39. However, all the necessary tests for checking connectivity on the transport network are not functioning properly. Notice that a simple ping from R41 to R12 fails, even though the DMVPN tunnels verify end-to-end reachability.

Example 5-39 *Verification of Outside Connectivity*

```
R41-Spoke# show dmvpn detail
! Output omitted for brevity
Interface Tunnel200 is up/up, Addr. is 192.168.200.41, VRF ""
# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- -------------- ----- -------- ----- -----------------
    1 100.64.12.1      192.168.200.12    UP 02:09:07     S  192.168.200.12/32
    1 100.64.22.1      192.168.200.22    UP 02:09:07     S  192.168.200.22/32
R41-Spoke# ping vrf INET01 100.64.12.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 100.64.12.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

The reason for the packet failure is that the router needs to have the Self-to-Outside policy defined. Example 5-40 demonstrates the configuration for the Self-to-Outside policy. The ACL-IPsec and ACL-IPsec are reused from the Outside-to-Self policy.

Example 5-40 *Configuration for the Self-to-Outside Policy*

```
ip access-list extended ACL-DHCP-OUT
permit udp any eq bootpc any eq bootps
!
ip access-list extended ACL-ICMP
permit icmp any any
!
class-map type inspect match-any CLASS-SELF-TO-OUTSIDE-INSPECT
 match access-group name ACL-IPSEC
 match access-group name ACL-ICMP

class-map type inspect match-any CLASS-SELF-TO-OUTSIDE-PASS
 match access-group name ACL-ESP
 match access-group name ACL-DHCP-OUT
!
policy-map type inspect POLICY-SELF-TO-OUTSIDE
  class type inspect CLASS-SELF-TO-OUTSIDE-INSPECT
   inspect
class type inspect CLASS-SELF-TO-OUTSIDE-PASS
   pass
class class-default
  drop log
!
zone-pair security SELF-TO-OUTSIDE source self destination OUTSIDE
 service-policy type inspect POLICY-SELF-TO-OUTSIDE
```

Now that the second policy has been configured, R41 can successfully ping R12 as shown in Example 5-41.

Example 5-41 *Successful Ping Test Between R41 and R12*

```
R31-Spoke# ping vrf INET01 100.64.12.2
Sending 5, 100-byte ICMP Echos to 100.64.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

Control Plane Policing (CoPP)

Control Plane Policing (CoPP) is a QoS policy that is applied to traffic to or sourced by the router's control plane CPU. CoPP policies are used to limit known traffic to a given rate while protecting the CPU from unexpected extreme rates of traffic that could impact the stability of the router.

Typical CoPP implementations use only an input policy that allows traffic to the control plane to be policed to a desired rate. In a properly planned CoPP policy, network traffic is placed into various classes that are based on the type of traffic (management, routing protocols, or known IP addresses). The CoPP policy is then implemented to limit traffic to the control plane CPU to a specific rate for each class.

When defining a rate for a CoPP policy, the rate for a class may not be known without further investigation. The QoS **police** command uses conform, exceed, and violate actions, which can be configured to **transmit** or **drop** traffic. By choosing to transmit traffic that exceeds the policed rate, and monitoring the CoPP, the policy can be adjusted over time to meet day-to-day requirements.

Understanding what is needed to define a traffic class can be achieved from protocol documentation or from experience with running networks. Other new features, such as PfRv3, may not be directly known. The use of the Cisco Embedded Packet Capture (EPC) allows capturing network traffic and exporting it to a PCAP file to identify the necessary traffic classes.

EPC operates differently on the IOS and IOS XE platforms.

IOS Embedded Packet Capture (EPC)

The following steps outline how to use EPC on an IOS platform running 15.5(3)M1:

Step 1. **Define a capture buffer.**

A capture buffer must be defined and can include a limit based on time, packet count, or capture file size. The capture buffer is defined with the executive command **monitor capture buffer** *buffer-name* [**limit** {**duration** *seconds* | **packet-count** *total-packets*} [**size** *buffer-size*].

Step 2. **Define a capture point.**

A capture point is defined with the command **monitor capture point** {**ip** | **ipv6**} **process-switched** *capture-point-name* {**both** | **from-us** | **in** | **out**}.

Step 3. **Associate the capture point to the capture buffer.**

The capture point is associated to the capture buffer with the command **monitor capture point associate** *capture-point-name capture-buffer-name*.

Step 4. **Start capturing data packets.**

Enable the capture point to start capturing data with the command **capture point start** {*capture-point-name* | **all**}.

Step 5. **Export the captured data.**

The captured data is then exported with the command **monitor capture buffer** *buffer-name* **export** *export-location*.

Step 6. **Clear the capture buffer.**

The capture buffer is then cleared with the command **monitor capture buffer** *buffer-name* **clear**.

Example 5-42 demonstrates the configuration of a packet capture and export to a TFTP (Trivial File Transfer Protocol) server on R41.

Example 5-42 *IOS Platform EPC*

```
R41-Spoke# monitor capture buffer CoPP
R41-Spoke# monitor capture buffer CoPP limit duration 1200
R41-Spoke# monitor capture buffer CoPP size 10240
R41-Spoke# monitor capture point ip process-switched CoPP both
R41-Spoke# monitor capture point associate CoPP CoPP
R41-Spoke# monitor capture point start CoPP
R41-Spoke# show monitor capture point CoPP
Status Information for Capture Point CoPP
IPv4 Process
Switch Path: IPv4 Process        , Capture Buffer: CoPP
Status : Active

Configuration:
monitor capture point ip process-switched CoPP both
R41-Spoke# monitor capture buffer CoPP export tftp://192.168.0.1/R41.pcap
R41-Spoke# monitor capture buffer CoPP clear
```

IOS XE Embedded Packet Capture

The EPC commands for the IOS XE platform are different from those for IOS and are simplified. The following steps demonstrate EPC on IOS XE running 3.16.1 / 15.5(3)S1. With IOS XE, the entire configuration of the monitor and start can be accomplished in two command lines.

Step 1. **Define the traffic to be captured.**

An access list is required to specify the network traffic that is captured. After creating the extended ACL, the command **monitor capture** *capture-name* **access-list** *access-list-name* is used.

Step 2. **Define the capture duration.**

The length of the packet capture is defined with the command **monitor capture** *capture-name* **limit duration** *seconds*.

Step 3. **Define the capture buffer size.**

The capture buffer size is defined with the command **monitor capture** *capture-name* **buffer [circular] size** *bytes.*

Step 4. **Define the capture source.**

The capture of control plane traffic can be directly enabled, greatly simplifying the process of learning what traffic is important. The command **monitor capture** *capture-name* **control-plane {in | out | both}** specifies the control plane as the source.

Step 5. **Start capturing data packets.**

Start capturing control plane traffic with the command **monitor capture** *capture-name* **start**.

Step 6. **Export the captured data.**

The captured data is then exported with the command **monitor capture** *capture-name* **export** *export-location*.

Step 7. **Stop capturing data packets.**

After enough packets have been captured, the packet capture can be stopped with the command **monitor capture** *capture-name* **stop**.

Example 5-43 demonstrates the EPC on R41 as an IOS XE router.

Example 5-43 *IOS XE Platform EPC*

```
R41-Spoke# config terminal
R41-Spoke(config)# ip access-list extended CoPP-filter
R41-Spoke(config)# permit ip any any
R41-Spoke(config)# end
R41-Spoke# monitor capture CoPP control-plane in access-list CoPP-filter limit
  duration 600 buffer size 10
R41-Spoke# monitor capture CoPP start
R41-Spoke# show monitor capture CoPP

Status Information for Capture CoPP
  Target Type:
   Interface: Control Plane, Direction : in
   Status : Active
  Filter Details:
   Access-list: CoPP-filter
  Buffer Details:
   Buffer Type: LINEAR (default)
   Buffer Size (in MB): 50
  Limit Details:
   Number of Packets to capture: 0 (no limit)
   Packet Capture duration: 600
   Packet Size to capture: 0 (no limit)
   Maximum number of packets to capture per second: 1000
   Packet sampling rate: 0 (no sampling)

%BUFCAP-6-DISABLE_ASYNC: Capture Point CoPP disabled. Reason : Time Duration Reached
R41-Spoke# monitor capture CoPP export tftp://192.168.0.1/R41-XE.pcap
```

Analyzing and Creating the CoPP Policy

After the file has been exported, the packets can be analyzed to identify the traffic classes that are needed. Table 5-2 provides a list of protocols and ports specifically for PfR.

Table 5-2 *PfR-Derived Control Protocols and Ports*

Protocol/Port	Description
UDP/9997	NetFlow version 9_ exports between the local border router and MC with bandwidth updates
	Traffic between remote border routers and local MC with threshold crossing alert (TCA) on-demand exports (ODEs) to indicate that the remote site is receiving traffic that does not meet the policy definition

Protocol/Port	Description
UDP/9997 UDP/9995	Traffic between the local border router and MC is NetFlow v9 exports of traffic classes, bandwidth, application identification, and site prefixes
UDP/9996	Traffic between the local border routers of TCAs
TCP/17749	Traffic between the local border routers and MC for control traffic
IP Protocol 88 224.0.0.10	EIGRP *Service Advertisement Framework (SAF)* between MCs as unicast traffic for policy and site prefix database control plane

Table 5-3 lists other common ports and protocols for routing protocols and connectivity tests.

Table 5-3 *Other Relevant Traffic Classes*

Protocol/Port	Description
Protocol 88 (destination of 224.0.0.10)	EIGRP
Protocol 89 (destination of 224.0.0.5 and 224.0.0.6)	OSPF
TCP/179	BGP
Protocol 103 (destination of 224.0.0.13)	PIM
UDP/161	SNMP queries
UDP/162	SNMP traps
UDP/123	Network Time Protocol (NTP)
TCP/22	SSH
UDP/500	ISAKMP (with/without cryptography)
UDP/4500	ISAKMP NAT
Protocol 50	IPsec with ESP
TCP/80 or TCP/443	Communication with CA
UDP/67	DHCP bootps
UDP/68	DHCP bootpc

After the list of traffic is reviewed, ACLs can be built for matching in a class map. Example 5-44 demonstrates a typical list of ACLs. Notice that these access lists do not restrict access and are open, allowing anyone to send traffic matching the protocols. For some types of external network traffic (such as BGP), the external network address can change and is better managed from a ZBFW perspective. A majority of these protocols are accessed only via controlled internal prefixes, minimizing the intrusion surface. Management protocols are an area that can easily be controlled by using a few jump boxes for direct access, limiting SNMP and other management protocols to a specific range of addresses residing in the NOC.

Example 5-44 *Access List Configuration for CoPP*

```
ip access-list extended ACL-CoPP-ICMP
 permit icmp any any echo-reply
 permit icmp any any ttl-exceeded
 permit icmp any any unreachable
 permit icmp any any echo
 permit udp any any range 33434 33463 ttl eq 1
!
ip access-list extended ACL-CoPP-IPsec
 permit esp any any
 permit gre any any
 permit udp any eq isakmp any eq isakmp
 permit udp any any eq non500-isakmp
 permit udp any eq non500-isakmp any
!
ip access-list extended ACL-CoPP-Initialize
 permit udp any eq bootps any eq bootpc
!
ip access-list extended ACL-CoPP-Management
 permit udp any eq ntp any
 permit udp any any eq snmp
 permit tcp any any eq 22
 permit tcp any eq 22 any established
!
ip access-list extended ACL-CoPP-PfRv3
 permit udp any any range 9995 9997
 permit tcp any any eq 17749
 permit eigrp any any
!
ip access-list extended ACL-CoPP-Routing
 permit tcp any eq bgp any established
 permit eigrp any host 224.0.0.10
 permit ospf any host 224.0.0.5
 permit ospf any host 224.0.0.6
 permit pim any host 224.0.0.13
 permit igmp any any
```

The class configuration for CoPP uses these access lists to match known protocols being used and is demonstrated in Example 5-45.

Example 5-45 *Class Configuration for CoPP*

```
class-map match-all CLASS-CoPP-IPsec
 match access-group name ACL-CoPP-IPsec
class-map match-all CLASS-CoPP-Routing
 match access-group name ACL-CoPP-Routing
class-map match-all CLASS-CoPP-PfRv3
 match access-group name ACL-CoPP-PfRv3
class-map match-all CLASS-CoPP-Initialize
 match access-group name ACL-CoPP-Initialize
class-map match-all CLASS-CoPP-Management
 match access-group name ACL-CoPP-Management
class-map match-all CLASS-CoPP-ICMP
 match access-group name ACL-CoPP-ICMP
```

The policy map for how these classes operate is to **police** traffic to a given rate in order to minimize any ability to overload the router. However, finding the correct rate without impacting network stability is not a simple task. In order to guarantee that CoPP does not introduce issues, the configuration of the violate action is set to **transmit** for all the vital classes until a baseline for normal traffic flows is established. Over time, the rate can be adjusted. Other traffic like ICMP and DHCP is set to **drop** as it should have low packet rates.

In the policy map the class default exists and contains any unknown traffic. Under normal conditions nothing should exist within the class default, but allowing a minimal amount of traffic within this class and monitoring the policy permits discovery of new or unknown traffic that would have otherwise been denied. Example 5-46 displays the CoPP policy.

Example 5-46 *Policy Configuration for CoPP*

```
policy-map POLICY-CoPP
 class CLASS-CoPP-ICMP
  police 8000 conform-action transmit  exceed-action transmit
         violate-action drop
 class CLASS-CoPP-IPsec
  police 64000 conform-action transmit  exceed-action transmit
         violate-action transmit
 class CLASS-CoPP-Initialize
  police 8000 conform-action transmit  exceed-action transmit
         violate-action drop
 class CLASS-CoPP-Management
  police 32000 conform-action transmit  exceed-action transmit
         violate-action transmit
 class CLASS-CoPP-PfRv3
```

```
  police 64000 conform-action transmit  exceed-action transmit
         violate-action transmit
 class CLASS-CoPP-Routing
  police 64000 conform-action transmit  exceed-action transmit
         violate-action transmit
 class class-default
  police 8000 conform-action transmit  exceed-action transmit
         violate-action drop
```

The policy map is then applied to the control plane as demonstrated in Example 5-47.

Example 5-47 *Application of the Policy for CoPP*

```
control-plane
 service-policy input POLICY-CoPP
```

Now that the policy map has been applied to the control plane, it needs to be validated. In Example 5-48, the PfRv3 policy traffic has exceeded the configured rate even in the small lab network. In addition, the class default class sees traffic. To identify what is happening, EPC is used again. This time, the access lists can be reversed from **permit** to **deny** which is used as the filter to gather unexpected traffic.

Example 5-48 *Validation of the Policy for CoPP*

```
R41-Spoke# show policy-map control-plane input
 Control Plane

  Service-policy input: POLICY-CoPP

    Class-map: CLASS-CoPP-ICMP (match-all)
      154 packets, 8912 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group name ACL-CoPP-ICMP
      police:
          cir 8000 bps, bc 1500 bytes, be 1500 bytes
        conformed 154 packets, 8912 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          transmit
        violated 0 packets, 0 bytes; actions:
          drop
        conformed 0000 bps, exceeded 0000 bps, violated 0000 bps

    Class-map: CLASS-CoPP-IPsec (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group name ACL-CoPP-IPsec
      police:
          cir 64000 bps, bc 2000 bytes, be 2000 bytes
        conformed 0 packets, 0 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          transmit
        violated 0 packets, 0 bytes; actions:
          transmit
        conformed 0000 bps, exceeded 0000 bps, violated 0000 bps

    Class-map: CLASS-CoPP-Initialize (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group name ACL-CoPP-Initialize
      police:
          cir 8000 bps, bc 1500 bytes, be 1500 bytes
        conformed 0 packets, 0 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          transmit
        violated 0 packets, 0 bytes; actions:
          drop
        conformed 0000 bps, exceeded 0000 bps, violated 0000 bps

    Class-map: CLASS-CoPP-Management (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group name ACL-CoPP-Management
      police:
          cir 32000 bps, bc 1500 bytes, be 1500 bytes
        conformed 0 packets, 0 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          transmit
        violated 0 packets, 0 bytes; actions:
          transmit
        conformed 0000 bps, exceeded 0000 bps, violated 0000 bps

    Class-map: CLASS-CoPP-PfRv3 (match-all)
      92 packets, 123557 bytes
      5 minute offered rate 4000 bps, drop rate 0000 bps
      Match: access-group name ACL-CoPP-PfRv3
      police:
          cir 64000 bps, bc 2000 bytes, be 2000 bytes
        conformed 5 packets, 3236 bytes; actions:
          transmit
        exceeded 1 packets, 1383 bytes; actions:
          transmit
        violated 86 packets, 118938 bytes; actions:
          transmit
        conformed 1000 bps, exceeded 1000 bps, violated 4000 bps

    Class-map: CLASS-CoPP-Routing (match-all)
      39 packets, 2277 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: access-group name ACL-CoPP-Routing
      police:
          cir 64000 bps, bc 2000 bytes, be 2000 bytes
        conformed 39 packets, 2277 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          transmit
        violated 0 packets, 0 bytes; actions:
          transmit
        conformed 0000 bps, exceeded 0000 bps, violated 0000 bps

    Class-map: class-default (match-any)
      56 packets, 20464 bytes
      5 minute offered rate 1000 bps, drop rate 0000 bps
      Match: any
      police:
          cir 8000 bps, bc 1500 bytes, be 1500 bytes
        conformed 5 packets, 2061 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          transmit
        violated 0 packets, 0 bytes; actions:
          drop
        conformed 0000 bps, exceeded 0000 bps, violated 0000 bps
```

Device Hardening

In addition to protecting the DMVPN tunnels, deploying ZBFW, and configuring authentication, authorization, and accounting (AAA) on the routers, disabling unused services and features improves the overall security posture by minimizing the amount of information exposed externally. In addition to hardening a router, this reduces the amount of router CPU and memory utilization that would be required to process these unnecessary packets.

This section provides a list of additional commands that can harden a router. All interface-specific commands are applied only to the interface connected to the public network.

- **Disable topology discovery tools:** Tools such as Cisco Discovery Protocol and Link Layer Discovery Protocol (LLDP) can provide unnecessary information to routers outside of your control. The services can be disabled with the interface parameter commands **no cdp enable**, **no lldp transmit**, **no lldp receive**.
- **Disable TCP and UDP small services:** The commands **service tcp-keepalive-in** and **service tcp-keepalive-out** ensure that devices send TCP keepalives for inbound/outbound TCP sessions. This ensures that the device on the remote end of the connection is still accessible and that half-open or orphaned connections are removed from the local device.
- **Disable IP redirect services:** An ICMP redirect is used to inform a device of a better path to the destination network. An IOS device sends an ICMP redirect if it detects network traffic hairpinning on it. This behavior is disabled with the interface parameter command **no ip redirects.**
- **Disable Proxy Address Resolution Protocol (ARP):** Proxy ARP is a technique that a router uses to answer ARP requests intended for a different router. The router fakes its identity and sends out an ARP response for the router that is responsible for that network. A man-in-the-middle intrusion enables a host on the network with a spoofed MAC address of the router and allows traffic to be sent to the hacker. Disabling Proxy ARP on the interface is recommended and accomplished with the command **no ip proxy-arp**.
- **Disable service configuration**: Cisco devices support automatic configuration from remote devices via TFTP and other methods. This service should be disabled with the command **no service config.**
- **Disable the Maintenance Operation Protocol (MOP) service**: The MOP service is not needed and should be disabled globally with the command **no mop enabled** and with the interface parameter command **no mop enabled**.
- **Disable the packet assembler/disassembler (PAD) service:** The PAD service is used for X25 and is not needed. It can be disabled with the command **no service pad.**
- **Enhancing SSH standards:** SSH version 2 has many benefits and closes a potential security hole that is found in SSH version 1. SSH version 2 is certified under FIPS 140-1 and 140-2 NIST/U.S. cryptographic standards and should be used where feasible. The command **ip ssh version 2** disables SSH version 1 on the router.

  By default SSH is assigned a differentiated services code point (DSCP) value of zero, which could be dropped in a QoS policy. The SSH protocol should be changed to a DSCP that provides prioritization in a QoS policy with the command **ip ssh dscp** *dscp-value*.

Summary

This chapter focused on the security components of a WAN network that provides data integrity, data confidentiality, and data availability. A certain level of trust is placed in the SP network to maintain data integrity and confidentiality, but when IPsec protection is enabled on the DMVPN tunnels, the trust boundary is moved from the SP to your own organization's control. DMVPN IPsec tunnel protection can be deployed on any transport using pre-shared keys or PKI.

In addition to securing the transport links, the router itself needs to be hardened from intrusions. Deploying the Cisco Zone-Based Firewall (ZBFW) provides stateful control of traffic that enters or leaves the router. Control plane policing (CoPP) provides a technique to limit router traffic to the router's control plane, ensuring that the router remains stable so that it is available to forward traffic.

Using all the techniques described in this chapter secures the router and its transports so that the router can provide data integrity, data confidentiality, and data availability to the network.

Further Reading

Bolapragada, Vijay, Mohamed Khalid, and Scott Wainner. *IPSec VPN Design*. Indianapolis: Cisco Press, 2005.

Cisco. "Backing Up and Restoring the Cisco IOS CA Server." www.cisco.com.

Cisco. "Cisco Guide to Harden Cisco IOS Devices." www.cisco.com.

Cisco. "Cisco IOS Software Configuration Guides." www.cisco.com.

Cisco. "Next Generation Encryption." www.cisco.com.

Cisco. "Public Key Infrastructure Configuration Guide." www.cisco.com.

Huang, G., S. Beaulieu, and D. Rochefort. RFC 3706, "A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers." IETF, February 2004. http://tools.ietf.org/html/rfc3706.

Karamanian, Andre, Srinivas Tenneti, and Francois Dessart. *PKI Uncovered: Certificate-Based Security Solutions for Next-Generation Networks*. Indianapolis: Cisco Press, 2011.

Kaufman, C., P. Hoffman, Y. Nir, and P. Eronen. RFC 5996, "Internet Key Exchange Protocol Version 2 (IKEv2)." IETF, September 2010. http://tools.ietf.org/html/rfc5996.

Kent, S., and R. Atkinson. RFC 2401, "Security Architecture for the Internet Protocol." IETF, November 1998. http://tools.ietf.org/html/rfc2401.

Kent, S., and K. Seo. RFC 4301, "Security Architecture for the Internet Protocol." IETF, December 2005. http://tools.ietf.org/html/rfc4301.

Chapter 6

Application Recognition

This chapter covers the following topics:

- Application recognition
- Network Based Application Recognition version 2 (NBAR2)
- NBAR2 operations and functions
- NBAR2 auto-learning and customization
- Validation and troubleshooting of NBAR2

"Application" in the context of this book refers to a network application, which is the network traffic between two or more computers that exchange data for a particular service or use. Network applications have various uses, such as client access to a server service (web, mail, database), a client that talks with other clients (media, file transfer, peer-to-peer), network control (BGP, IKE), signaling protocols (SIP, H.323), and operation services (SNMP, syslog).

This chapter explains how network applications can be identified so that the classified applications can be used later for *Quality of Service (QoS)*, *Performance Routing (PfR)*, and *application visibility*.

What Is Application Recognition?

Application recognition is a system that classifies a wide variety of protocols and applications that are sent over a network. In most cases, application recognition associates an application with a particular TCP or UDP connection, which represents the data exchange for a specific service. In some cases, the application is associated with other types of data transactions, such as multiple media sessions multiplexed within a single TCP or UDP connection or non-TCP/UDP traffic.

In addition to identifying the application itself, application recognition can also provide additional metadata about the application. Such metadata can include categorization or attributes of applications, Layer 7 information extracted from the traffic, and other such information.

A particular type of traffic can run as a hierarchy of applications or protocols, for example:

- A web service (such as mail, video, storage) that runs on top of HTTP or HTTPS
- Application traffic tunneled in HTTP as a method for transmission through firewalls

When multiple applications use the same HTTP protocol, it is difficult to differentiate an application that the organization/business needs from a non-business-critical application. Application recognition can identify all protocol and application layers.

Some network services create multiple separate connections associated as one service, for example, FTP control and data, or Session Initiation Protocol (SIP) signaling and Real-Time Transport Protocol (RTP). Application recognition can bundle and associate these connections together and derive attributes from one connection to its associated sessions.

What Are the Benefits of Application Recognition?

Identifying and classifying network traffic is a basic step in implementing network policies such as QoS, application-based routing, monitoring, and security.

Knowledge of the applications and application attributes shifts the policy language from the networking level to a simple and intuitive application-based terminology that can be directly aligned with business goals.

Administrators can implement policies more effectively after identifying the types of applications and protocols that are running on a network. These policies do not change when a new server is brought up or moved to a different location because the application classification remains the same.

Application recognition provides network administrators with the ability to understand the types of applications and protocols operating on the network, the amount of traffic generated by each application, and how each application performs. With this knowledge an administrator can understand how the network is used, identify problems, operate the network better overall, and plan for the future.

NBAR2 Application Recognition

Cisco *Network Based Application Recognition version 2 (NBAR2)* is an application recognition system integrated into various Cisco network devices (such as routers, switches, access points, converged access) and virtual services (such as CSR 1000V and Cisco Virtual Network Analysis Module [vNAM]).

NBAR2 provides the necessary application recognition capability to support services such as QoS, PfR, and visibility. It interoperates with most types of interfaces and services that are applied together on devices. NBAR2 is enabled automatically whenever a service requests application-based policies; there is no need to enable it explicitly. Configuration options for NBAR2 are described later in this chapter.

NBAR2 delivers the best possible application recognition, providing results as early as is achievable and with performance optimized. NBAR2 accomplishes this using a combination of complementary techniques. No single technique can classify every type of network traffic. Each technique addresses different types and granularities of applications and improves the overall application recognition results across a variety of scenarios.

NBAR2 application recognition techniques are based on

- Deep packet inspection (DPI)
- Domain Name System (DNS) queries and snooping
- Statistical and behavioral traffic patterns
- Signaling from authoritative sources and databases
- Auto-learning of local specific applications
- User customization of specific known applications

In addition to application classification, NBAR2 provides a set of categories and attributes, Layer 7 extracted fields, and an application hierarchy to simplify policy configuration and improve the user experience.

NBAR2 Application ID, Attributes, and Extracted Fields

NBAR2 provides an application ID, application attributes, and Layer 7 extracted fields per flow. These are used to simplify application policies (PfR, QoS, and so on) or visibility.

NBAR2 Application ID

The NBAR2 application ID is based on RFC 6759 and uniquely defines an application by the engine ID and selector ID. The application ID remains consistent across most Cisco DPI engines that exist on multiple devices (IOS, IOS XE, NAM, and so on).

The **show ip nbar protocol-id** command displays the applications detected by NBAR2 and associated application IDs, as shown in Example 6-1. In the output, *id* represents the selector ID and *type* represents the engine ID.

Example 6-1 *List of Applications and IDs*

```
R31-Spoke# show ip nbar protocol-id
! Output omitted for brevity
Protocol Name             id            type
----------------------------------------------
3com-amp3                629           L4 IANA
3com-tsmux               106           L4 IANA
3pc                      34            L3 IANA
4chan                    763           L7 STANDARD
58-city                  704           L7 STANDARD
.
sip                      5060          L4 IANA
skinny                   63            L7 STANDARD
skype                    83            L7 STANDARD
smtp                     25            L4 IANA
snmp                     161           L4 IANA
```

NBAR2 Application Attributes

Each application has a set of attributes such as business relevance, category, subcategory, traffic class, application group, peer-to-peer (P2P), encrypted, or tunnel technology. It is recommended to use attributes when applying application-based policies such as QoS or PfR. For example, to apply a policy for business-relevant applications, use the *business-relevance* attribute, or to apply a policy for voice and video traffic, use the *voice-and-video* category. The attributes associated with each application are defined in the NBAR2 Protocol Pack and can be customized according to specific deployment needs.

Note An NBAR2 Protocol Pack contains information on applications officially supported by NBAR2 that are compiled and packed together. Protocol Packs are covered in depth later in this chapter.

Table 6-1 shows the NBAR2 attributes that are available with Protocol Pack version 21 (released in June 2016), subject to change in future releases.

Table 6-1 *NBAR2 Attributes*

Attribute Name	Values
Business relevance	business-irrelevant, business-relevant, default
Categories	anonymizers, backup-and-storage, browsing, business-and-productivity-tools, consumer-file-sharing, consumer-internet, consumer-messaging, consumer-streaming, custom-category, custom1-category, custom2-category, database, email, epayement, file-sharing, gaming, industrial-protocols, instant-messaging, inter-process-rpc, internet-security, layer3-over-ip, location-based-services, net-admin, newsgroup, other, social-networking, software-updates, Trojan, voice-and-video
Subcategories	authentication-services, backup-systems, consumer-audio-streaming, consumer-cloud-storage, consumer-multimedia-messaging, consumer-video-streaming, consumer-web-browsing, control-and-signaling, custom-sub-category, custom1-sub-category, custom2-sub-category, desktop-virtualization, enterprise-cloud-data-storage, enterprise-cloud-services, enterprise-data-center-storage, enterprise-media-conferencing, enterprise-realtime-apps, enterprise-rich-media-content, enterprise-sw-deployment-tools, enterprise-transactional-apps, enterprise-video-broadcast, enterprise-voice-collaboration, file-transfer, naming-services, network-management, os-updates, other, p2p-file-transfer, p2p-networking, remote-access-terminal, routing-protocol, tunneling-protocols
Application group	aol-group, apple-group, apple-talk-group, banyan-group, bittorrent-group, capwap-group, cisco-jabber-group, cisco-phone-group, corba-group, custom-group, custom1-group, custom2-group, dameware-group, edonkey-emule-group, espn-group, fasttrack-group, flash-group, fring-group, ftp-group, gnutella-group, google-group, gtalk-group, icq-group, imap-group, ipsec-group, irc-group, kakao-group, kerberos-group, ldap-group, ms-cloud-group, ms-crm-group, ms-lync-group, msn-messenger-group, netbios-group, nntp-group, npmp-group, other, pop3-group, prm-group, qq-group, skype-group, smtp-group, snmp-group, sqlsvr-group, stun-group, telepresence-group, tftp-group, vmware-group, vnc-group, wap-group, webex-group, xns-xerox-group, xunlei-group, yahoo-group, yahoo-messenger-group
Based on peer-to-peer	p2p-tech-no, p2p-tech-unassigned, p2p-tech-yes
Traffic class	broadcast-video, bulk-data, multimedia-conferencing, multimedia-streaming, network-control, ops-admin-mgmt, real-time-interactive, signaling, transactional-data, voip-telephony
Encrypted	encrypted-no, encrypted-unassigned, encrypted-yes
Tunneled	tunnel-no, tunnel-unassigned, tunnel-yes

The application attributes are displayed with the command **show ip nbar protocol-attribute** *application-name*.

Example 6-2 shows attributes for the cisco-phone application.

Example 6-2 *List of Application Attributes*

```
R31-Spoke# show ip nbar protocol-attribute cisco-phone

           Protocol Name : cisco-phone
               encrypted : encrypted-no
                  tunnel : tunnel-no
                category : voice-and-video
            sub-category : enterprise-voice-collaboration
       application-group : cisco-phone-group
          p2p-technology : p2p-tech-no
           traffic-class : voip-telephony
      business-relevance : business-relevant
```

Note Chapter 14, "Intelligent WAN Quality of Service (QoS)," explains a QoS policy that is based on the *traffic-class* and *business-relevance* attributes to simplify the identification of network traffic.

Business relevance directly refers to applications that are identified as relevant (supporting business objectives), default (applications that may or may not support a business or are just unknown), or irrelevant (not supporting business objectives).

Every application attribute has a set of possible values and a few custom values. Example 6-3 shows the possible values supported for the *category* attribute.

Example 6-3 *List of Options per Attribute*

```
R31-Spoke# show ip nbar attribute category
! Output omitted for brevity
      Name :  category
      Help :  Category attribute
      Type :  group
    Groups :  anonymizers
           :  backup-and-storage
           :  browsing
           :  business-and-productivity-tools
           :  consumer-file-sharing
           :  consumer-internet
           :  consumer-messaging
           :  consumer-streaming
```

```
            :  custom-category
            :  custom1-category
            :  custom2-category
            :  database
            :  email
```

NBAR2 Layer 7 Extracted Fields

NBAR2 supports extraction of Layer 7 information such as HTTP URL, host name, SIP domains, and so on. These can be exported over NetFlow for visibility into the traffic data. Table 6-2 lists the NBAR2 Layer 7 extracted information that is available with Protocol Pack version 21, subject to change in future releases.

Table 6-2 *NBAR2 Layer 7 Extracted Fields*

Extracted Field	Description
sslCommonName	Common name extracted from an SSL certificate
httpUrl	URL extracted from the HTTP transaction
httpHostName	Host name extracted from the HTTP transaction
httpUserAgent	User agent field extracted from the HTTP transaction
httpReferer	Referer extracted from the HTTP transaction
rtspHostName	RTSP host name extracted from the Real Time Streaming Protocol (RTSP) transaction
smtpServer	Server name extracted from a Simple Mail Transfer Protocol (SMTP) transaction
smtpSender	Sender name extracted from an SMTP transaction
pop3Server	Server name extracted from a POP3 transaction
nntpGroupName	Group name extracted from a Network News Transfer Protocol (NNTP) transaction
sipSrcDomain	Source domain extracted from an SIP transaction
sipDstDomain	Destination domain extracted from an SIP transaction

NBAR2 Operation and Functions

This section provides background on how NBAR2 works in order to better understand how to deploy it. NBAR2 uses multiple mechanisms and multiple layers of application recognition, which are selected based on the traffic, the learning of the network, and the level of granularity required.

Figure 6-1 shows a high-level diagram of NBAR2 operation. Application recognition is determined by the NBAR2 software engine on each device. The NBAR2 controller logic can assist with cross-network learning, integration, and propagation of application information across the network but is not a mandatory component of the solution. The NBAR2 engine analyzes network traffic using *application signatures* stored in the Protocol Pack to identify specific application traffic. The signatures for each application include regular expressions and traffic pattern information to identify application traffic. Cisco provides a standard, wide-ranging set of applications in the form of a Protocol Pack, updated periodically.

The NBAR2 engine uses a flow table to track and store states on each TCP or UDP flow. The engine analyzes the traffic to learn patterns (for example, known servers or clients) and stores the pattern information on local cache tables. This information is used by the local device to improve the classification and, optionally, is also fed to the controller logic. The controller logic collects this information from multiple network devices, analyzes it together with additional controller information, and pushes the combined processed information back to the network devices.

The application recognition results are consumed by services running on the device such as PfR, QoS, and visibility.

Note A flow is a tuple that contains the source IPv4/IPv6 address, destination IPv4/IPv6 address, source port, destination port, TCP/UDP protocol, and VRF.

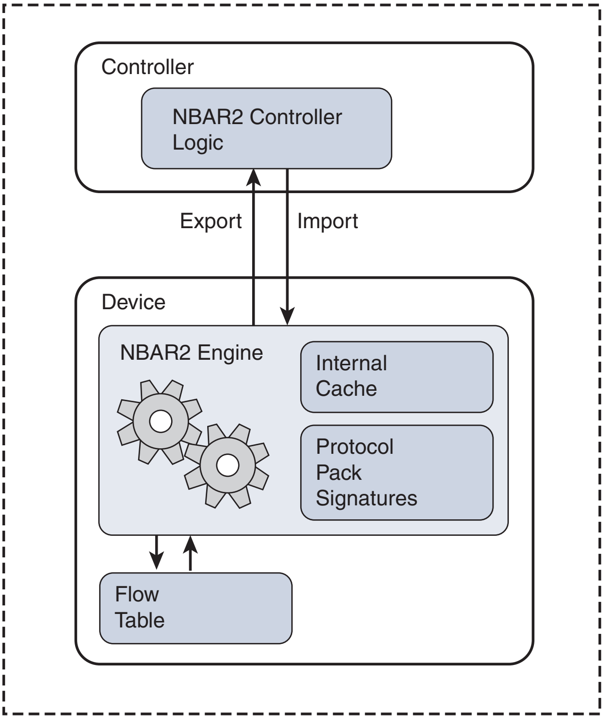


Figure 6-1 *NBAR2 High-Level Diagram*

When using NBAR2 controller logic, the **ip nbar classification cache sync {export | import}** command enables the controller and device integration. The **export** keyword enables exporting application information from the device to the controller. The **import** keyword enables the controller to program application information to the device.

Phases of Application Recognition

NBAR2 attempts to provide the best possible classification as early as possible in the flow. Figure 6-2 shows the phases of application recognition along the flow.

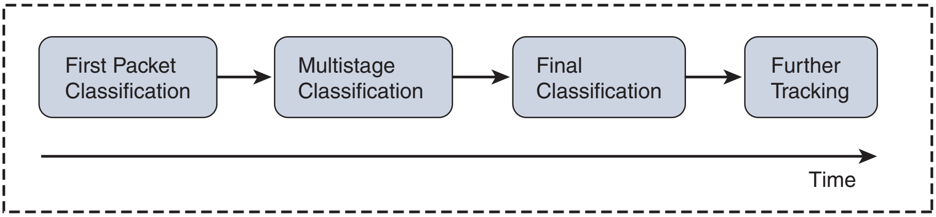


Figure 6-2 *Phases of Application Recognition*

First Packet Classification

In some cases, NBAR2 can provide classification based on the first packet of the flow. First packet classification is mainly based on the flow tuple. NBAR2 stores the flow mapping, socket mapping (server IP and port), or IP mapping to its cache. NBAR2 learns the mapping by traffic aggregation, DNS snooping, or controller logic.

Some services, such as PfR, use application-based policies to select the traffic path. Classifying on the first packet of the flow also causes path selection on the first packet of the flow. If the application is not determined on the first packet, PfR may modify the path later in the flow when the application is determined.

Services such as firewalls, NAT, and WAAS optimization are sensitive to path changes in the middle of the flow. Such services track the state of the flow and, when applied in the path, can close the flow when the path changes, so a good first packet classification is essential in such cases.

Multistage Classification

After attempting to classify a flow based on the first packet, NBAR2 analyzes additional packets of the flow to improve the classification if possible. Additional packets or additional flows contain more information that NBAR2 uses to fine-tune the application. For example, NBAR2 may initially provide a generic application classification such as *Cisco-Collaboration* based on traffic going to a known destination server. After analyzing additional packets, NBAR2 may detect the SIP protocol and refine the classification from *Cisco-Collaboration* to *TelePresence*. When additional packets are observed, NBAR2 may apply statistical classification to further fine-tune the application classification to audio or video streams. At each stage of refining the classification, NBAR2 provides the best available classification result so that application-based policies (such as QoS or PfR) can operate as accurately as possible.

Final Classification

In this phase, NBAR2 stops the classification and reaches the final resolution of the application. The final application classification is stored in the flow table. NBAR2 continues to provide the final application classification without running through the NBAR2 engine. Depending on the traffic and the application mapping, the final classification may occur after a few packets of a flow.

Further Tracking

In some cases, when NBAR2 is required to extract Layer 7 information for subsequent packets, NBAR2 continues to process the packets of the flow. This does not change the classification but can be used to generate additional information on the flow.

Table 6-3 provides examples of the refinement of application classification that can occur in the various phases of a flow and includes an example of a flow whose final classification is available from the first packet analysis.

Table 6-3 *Application Classification Phases Examples*

Phase	Example 1	Example 2	Example 3
First packet classification	Cisco collaboration	Unknown	Office 365
Multistage classification	TelePresence	HTTP	Office 365
Final classification	TelePresence-Audio	YouTube	Office 365
Further tracking	RTP payload type	URL extraction	-

NBAR2 Engine and Best-Practice Configuration

NBAR2 uses multiple techniques to provide the best classification. Each technique fits different traffic types and use cases. This section describes the components and layers of the NBAR2 engine and how to configure them. The configurations listed in this section are best practices. For more configuration options, refer to the documentation “Network Based Application Recognition (NBAR)” on the cisco.com website.

Note For full application recognition capability, NBAR2 requires network traffic to pass symmetrically through the device.

Multipacket Engine

One of the most basic elements of NBAR2 is the capability to inspect multiple packets within a flow and apply cross-packet signatures. NBAR2 tracks and store states for different packets along the flow in a flow table and simultaneously searches many multipacket signatures. Multipacket signatures are applied for many protocols and provide improved classification and accuracy.

DNS Engine

NBAR2 analyzes DNS traffic, checking for known or customized domain names. By inspecting the DNS replies, NBAR2 associates the domain name IP address with the related application. This association is used to map traffic flows sent to this IP address to the related application. To prevent the DNS engine from learning from any response, NBAR2 applies a DNS guard that must see both directions of DNS traffic and checks the validity of the DNS request and reply.

The DNS classification guard is enabled by default. It can be disabled using **no ip nbar classification dns learning guard**.

DNS Authoritative Source (DNS-AS) Engine

DNS-AS provides centralized control of custom application classification information. It leverages the universally available DNS query/response infrastructure to enable local DNS servers within an organization to propagate application classification metadata to devices in an enterprise network.

By centralizing the custom application metadata, a DNS server functioning as a DNS-AS server is an authoritative source for organizational internally hosted application classification information.

The local DNS server may be the organization's main DNS server or a separate dedicated server provisioned with TXT records that contain the application information. The DNS-AS server can be any standard DNS server, and the TXT records should be provisioned to the server using standard DNS configuration files.

Note When using a centralized controller such as Cisco APIC-EM, it is simpler and more efficient to provision local applications using the command **ip nbar custom** *custom-app-name* **composite server-name** *server-name-regex* from the controller or allow auto-learning of local applications.

Figure 6-3 shows a typical DNS-AS operation. For simplicity, the DNS and DNS-AS servers are drawn as the same DNS server, but the solution allows the use of separate servers for DNS and DNS-AS.

Network clients initiate a DNS request (1) normally and receive a response from the DNS server (2). The original DNS request and response are snooped by NBAR2 running on the router. For specific provisioned trusted domains, NBAR2 initiates another DNS A and TXT request (3), requesting metadata provisioned for the domain of the original DNS query. After receiving a reply from DNS-AS (4), NBAR2 associates the server IP address with the application metadata to classify the applications going to that domain and customize the application in the device (5). NBAR2 checks the validity of the DNS-AS queries, rate-limits the number of requests, and maintains the time to live (TTL) of provisioned entries based on the time-to-live value of the DNS response.

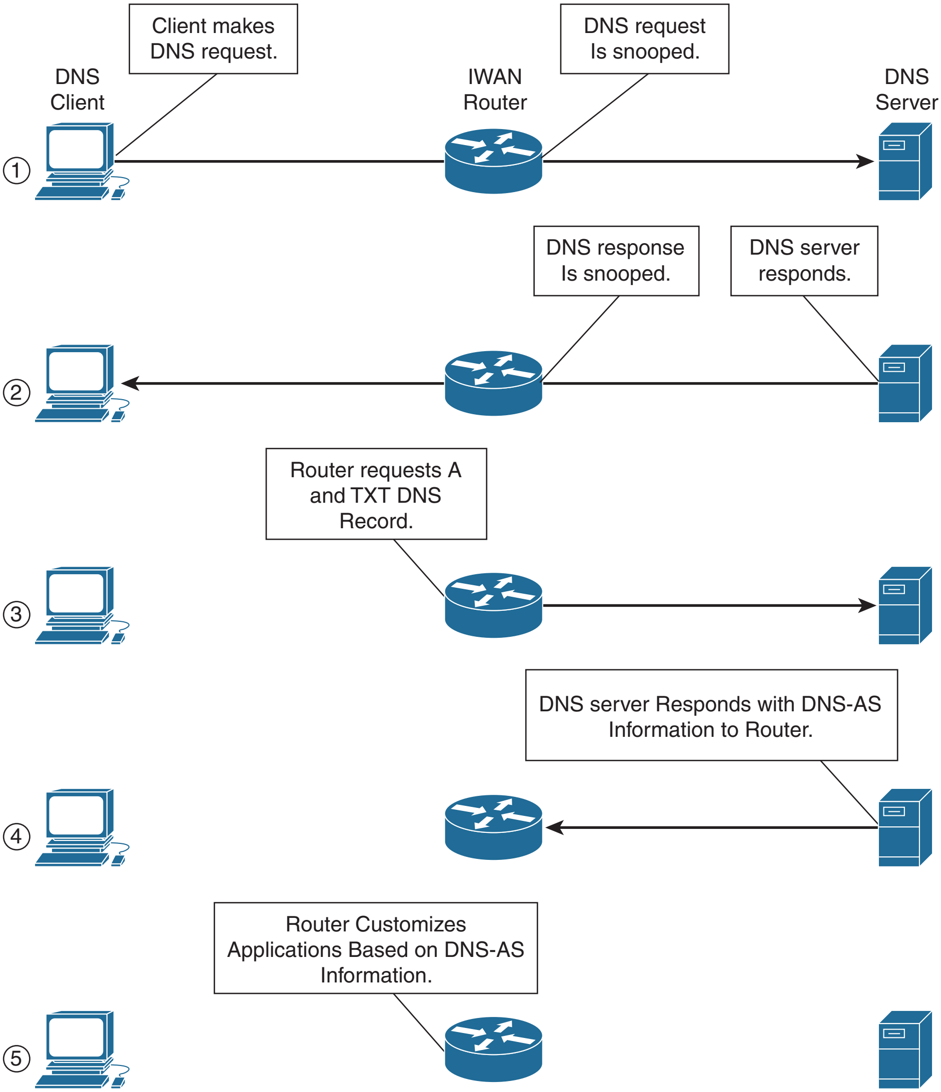

Client makes DNS request.
DNS request Is snooped.
DNS Client
IWAN Router
DNS Server
①
DNS response Is snooped.
DNS server responds.
②
Router requests A and TXT DNS Record.
③
DNS server Responds with DNS-AS Information to Router.
④
Router Customizes Applications Based on DNS-AS Information.
⑤

Figure 6-3 *DNS-AS Operation*

The following steps provision DNS-AS on the devices:

Step 1. **Enable DNS on the router performing the lookups.**

Use the command **ip name-server** [**vrf** *vrf-name*] *dns-server-ip-address*. Multiple DNS server addresses may be specified, separated by spaces.

Step 2. **Configure the trusted domains that DNS-AS will query.**

The command **avc dns-as client trusted-domains** places the router in trusted domains submode. Domains are then added with the command **domain** *regular-expression*. DNS-AS sends a TXT query only for the domains defined in the trusted domains submode.

Step 3. **Enable the DNS-AS client.**

This is accomplished with the command **avc dns-as client enable**.

Step 4. **Enable NBAR2 to snoop DNS traffic on the interface that is facing the source of the network traffic (LAN) and the destination network (WAN—DMVPN tunnels) per device.**

The command **interface** *interface-id* places the configuration into interface parameter configuration submode. Then use **avc dns-as learning** to enable NBAR2 learning.

Example 6-4 shows a DNS-AS configuration example.

Example 6-4 *DNS-AS Configuration*

```
R31-Spoke# configure terminal
R31-Spoke(config)# ip name-server vrf VFR1 10.1.30.40
R31-Spoke(config)# avc dns-as client trusted-domains
R31-Spoke(config-trusted-domains)# domain *mydomain.com
R31-Spoke(config-trusted-domains)# exit
R31-Spoke(config)# avc dns-as client enable
R31-Spoke(config)# interface GigabitEthernet 0/0
R31-Spoke(config-if)# avc dns-as learning
R31-Spoke(config-if)# interface Tunnel 100
R31-Spoke(config-if)# avc dns-as learning
R31-Spoke(config-if)# interface Tunnel 200
R31-Spoke(config-if)# avc dns-as learning
```

The application information TXT resource record entries are provisioned into the DNS-AS server. The DNS-AS TXT records are provisioned in a standard DNS configuration. For information about adding new TXT entries, refer to the documentation for the specific DNS server type. Use the following syntax for adding a TXT record for a given domain: **CISCO-CLS=app-name**:*application-name* | **business**:{YES | NO | DEFAULT} | **app-class**:*traffic-class* | **app-category**:*application-category* | **app-sub-category**:*application-sub-category* | **app-id**:*application-id*.

Table 6-4 provides a description of each attribute in the TXT entry. Refer to Table 6-1 for the possible values for each attribute.

Table 6-4 *NBAR2 DNS-AS TXT Attributes*

Attribute Name	Description
app-name	(Mandatory) The name of the application
business	Application business relevancy
app-class	Application traffic class
app-category	Application category
app-sub-category	Application subcategory
app-id	Application identifier

Example 6-5 shows a DNS-AS TXT entry for the MYDOMAIN application provisioned as an entry for the mydomain.org.com domain. The traffic class is set to multimedia streaming.

Example 6-5 *DNS-AS TXT Entry*

```
"CISCO-CLS=app-name:MYDOMAIN|app-class:MULTIMEDIA-STREAMING"
```

The DNS-AS server must also include the IP address of the domain for the device A query. Multiple IP addresses for a given domain are all associated to the application specified for that domain.

Note The format is provided based on the guidelines at the time of publication of this book. The website www.dns-as.org provides the latest information on structure.

DNS Classification by Domain

DNS traffic can affect application performance. For example, if the DNS query for a site is delayed or routed to a different path, the overall application experience may be delayed. NBAR2 attempts to classify DNS traffic for an application in the same way as the application traffic itself. This ensures that both the DNS traffic and the application traffic are treated with the same traffic policy. This way DNS does not introduce unnecessary delay and affect the overall application experience.

DNS classification by domain is enabled by default but can be disabled using **no ip nbar classification dns classify-by-domain**.

Control and Data Bundling Engine

When NBAR2 detects a control protocol such as SIP or FTP control, it also determines the data flow parameters it initiates. Using these parameters, NBAR2 associates the classification result with the data flow even before the data flow has been sent.

Behavioral and Statistical Engine

NBAR2 uses behavioral and statistical mechanisms to recognize network applications from their traffic patterns. NBAR2 characterizes traffic based on parameters such as packet sizes and inter-packet gaps. When these parameters match predefined patterns, NBAR2 uses the result to improve classification results. For example, NBAR2 uses patterns of different packet sizes to distinguish between audio and video of encrypted traffic.

Layer 3, Layer 4, and Sockets Engine

NBAR2 uses Layer 3 IP addresses and Layer 4 port information to classify traffic. In most cases the IP addresses and ports are learned and stored in the internal NBAR2 cache or by specific Layer 3/Layer 4 application customization. This is the main mechanism used to provide classification based on the first packet of a flow.

A socket is the server IP and server port that NBAR2 learns and associates with a particular application. NBAR2 stores the socket-to-application association and uses it for classification of traffic that matches this association rule. The Layer 3/Layer 4 learning may be populated by

- Customized Layer 3/Layer 4 applications
- NBAR2 controller logic
- DNS/DNS-AS, which associates a given server IP with a specific application
- Socket caching, which associates a given socket with a specific application
- Bundles, which consist of data flow learned from control flows

NBAR2 manages the correctness and lifetime of the associations. When necessary, it removes the entry from the internal cache.

Transport Hierarchy

NBAR2 transport hierarchy (TPH) provides a method for classifying underlying protocols or applications in addition to the final application classification. For example, when applications such as email, video, and so on are running over HTTP, NBAR2 provides HTTP as the transport hierarchy. In this case, the final classification would be email or video, but the transport hierarchy would be HTTP.

Modular Quality of Service Command-Line Interface (MQC) supports matching on the transport hierarchy using the class map command **match protocol** *protocol-name* **in-app-hierarchy**.

Subclassification

For some protocols, NBAR2 allows creating policies and exporting detailed Layer 7 information. To create an MQC policy based on subclassification, use the command **match protocol** *protocol-name sub-classification value*. The *sub-classification* keyword depends on the specified protocol name.

The HTTP protocol supports wildcard matching similar to regex. Matching is case insensitive, so, for example, there is no difference between *cisco* and *CISCO*. Here are some of the available wildcards:

- *** (Asterisk):** Matches any pattern. To match the substring identified in the beginning of a string, end the pattern with the asterisk. To match the substring at the end, use the asterisk at the beginning of the pattern. To match the pattern anywhere in the string, use an asterisk at the beginning and end.

  For example, the pattern ***.cisco.com** matches any site that ends with *.cisco.com*.

- **? (Question mark):** Matches any single character. For example, the pattern **www.?isco.com** matches websites such as *www.cisco.com* or *www.disco.com* but not *www.mycisco.com*.

  The escape sequence (Ctrl+V) must be pressed before the *?* can be entered.

- **[ ] (Brackets):** Matches any character in the pattern using the set of characters provided within the brackets. For example, the pattern **[abc]isco.com** matches websites that contain *aisco.com*, *bisco.com*, or *cisco.com*.

- **( ) (Parentheses):** Matches a group of characters in the identified pattern. For example, the pattern **(ftp).cisco.com** matches *ftp.cisco.com*. Commonly the pipe is used with the parentheses.

- **| (Pipe):** Allows for a Boolean OR when matching in a grouping of characters. For example, the pattern **www.(cisco|webex).com** matches either *www.cisco.com* or *www.webex.com*.

Example 6-6 shows a subclassification based on HTTP host name.

Example 6-6 *HTTP Host Name Subclassification MQC Configuration*

```
R31-Spoke# configure terminal
R31-Spoke(config)# class-map match-any MY-CLASS
R31-Spoke(config-cmap)# match protocol http host www.myhost*
```

Note Advanced customization allows creating a class map that specifies a particular file type. This may be used, for example, to create a policy for specific types of files being downloaded in a network. For example, the pattern ***.mp4** can be used to control HTTP downloads of MP4 streams in a QoS policy.

Custom Applications and Attributes

In every deployment, there are local and specific applications that are not covered by the NBAR2 Protocol Pack provided by Cisco. Local applications are mainly categorized as

- Applications specific to an organization
- Applications specific to one geographic location but not others

NBAR2 provides ways to customize such local applications:

- **Auto-customization:** NBAR2 can customize applications automatically.
    - Single-device customization can be done via the device command-line interface (CLI).
    - Multidevice customization can be done via the NBAR2 controller logic.
- **Manual customization**: Manually customize NBAR2 for local applications.

In both cases, NBAR2 provides a way to learn the applications in the network before setting up any customization.

Auto-learn Traffic Analysis Engine

Local applications that are not identified specifically by a protocol in the NBAR2 Protocol Pack are typically classified by NBAR2 simply as generic applications, such as HTTP, SSL, or unknown applications. NBAR2 learns details from the traffic and attempts to provide more information on the host names, ports, or sockets of the network traffic. Using this data, NBAR2 can automatically create custom protocols for local applications to improve the identification of traffic. NBAR2's auto-learn feature compiles multiple lists (host name, destination port, and other details) by sampling traffic flows for analysis. The lists are sorted by traffic volume.

To see the auto-learned tables, use the command **show ip nbar classification auto-learn** *list-type number-of-entries*.

Example 6-7 shows the top generic hosts collected by the auto-learn feature.

Example 6-7 *Top Generic Hosts Collected by the Auto-learn Feature*

```
R31-Spoke# show ip nbar classification auto-learn top-hosts 5

Total bytes:           2.117 G
Total packets:         2.112 M
Total flows:          30.096 K
Sample rate last:    1
Sample rate average: 1
Sample rate min:     1
Sample rate max:     1
-------------------------------------------------------------------------------
  #|Host                                         |Byte%|Flow%|Pkt% |Type |Field
-------------------------------------------------------------------------------
  1|backup.mydomain.com                           | 77% | <1% | 79% |http |host
  2|10.56.129.50                                  | 11% | 99% | 12% |http |host
  3|mail.mydomain.com                             | 10% | <1% |  7% |http |host
  4|10.56.111.6                                   | <1% | <1% | <1% |http |host
  5|10.210.20.18                                  | <1% | <1% | <1% |http |host
```

Example 6-8 shows the top sockets (server IP plus server port).

Example 6-8 *Top Sockets Collected by the Auto-learn Feature*

```
R31-Spoke# show ip nbar classification auto-learn top-sockets 5

Total bytes:           1.898 G
Total packets:         1.480 M
Total flows:          18.112 K
Sample rate last:    1
Sample rate average: 1
Sample rate min:     1
Sample rate max:     1
------------------------------------------------------------------------------
  #|Port  |IP                                  |Byte%|Flow%|Pkt% |Traffic Type
------------------------------------------------------------------------------
  1|53695 |10.210.20.24                         | 33% | <1% | 29% |UDP
  2|51509 |10.210.20.24                         | 32% | <1% | 28% |UDP
  3|50997 |10.210.20.19                         | 31% | <1% | 27% |UDP
  4|23    |10.210.20.23                         |  1% | <1% |  5% |TCP
  5|23    |10.210.20.17                         |  1% | <1% |  5% |TCP
```

Using the auto-learn results, high-volume local hosts, ports, and sockets can be customized as new local applications.

Traffic Auto-customization

The output in Example 6-7 shows that NBAR2 has learned that backup.mydomain.com consumes 77 percent of the generic traffic bandwidth. This is a potential candidate for a local application customization. To enable auto-customization, use the command **ip nbar auto-custom** *list* [**max-protocols** *number*].

When NBAR2 automatically creates a custom protocol, the new custom protocol inherits the attribute values of the generic application previously classified for that traffic. The attributes can be customized manually as described later in this chapter.

Note It is recommended to auto-customize traffic using the NBAR2 controller logic to make sure all network devices have a consistent application customization.

Manual Application Customization

In some cases, operators may know the local application information, or they may use the auto-learn feature to discover applications but customize the applications manually. This allows further control of the application names and IDs. Manual application customization is done using the command **ip nbar custom** *myname*.

There are various types of application customization:

- **Generic protocol customization:** Customization based on generic protocol identifiers such as
  - HTTP
  - SSL
  - DNS
- **Composite:** Customization based on multiple underlying protocols
- **Layer 3/Layer 4:** Customization based on network or transport-based characteristics such as
  - IPv4/IPv6 address
  - DSCP values
  - TCP/UDP ports
  - Flow source or destination direction
- **Byte offset:** Customization based on specific byte values in the payload

HTTP Customization

HTTP customization may be based on a combination of HTTP fields:

- **cookie:** HTTP cookie
- **host:** Host name of the origin server containing the resource
- **method:** HTTP method
- **referer:** Address from which the resource request was obtained
- **url:** Uniform Resource Locator path
- **user-agent:** Software used by the agent sending the request
- **version:** HTTP version
- **via:** HTTP via field

Example 6-9 shows a customization for an application called MYHTTP using the HTTP host mydomain.com with selector ID 10.

Example 6-9 *HTTP Host Name Customization*

```
R31-Spoke# configure terminal
R31-Spoke(config)# ip nbar custom MYHTTP http host *mydomain.com id 10
```

Note The engine ID and selector ID represent the application ID as described earlier in this chapter. For customized applications, the engine ID used is USER-DEFINED (6) and the selector ID is specified by the custom application CLI. It is recommended to assign the same selector ID for a given application across all routers to maintain consistency.

SSL Customization

Additional customization can be done for SSL-encrypted traffic using information extracted from the SSL server name indication (SNI) or common name (CN). The keyword **unique-name** is used for SSL customization.

Example 6-10 shows a customization for an application named MYSSL, using the SSL unique name mydomain.com with selector ID 11.

Example 6-10 *SSL Unique Name Customization*

```
R31-Spokeconfigure terminal
R31-Spoke(config)# ip nbar custom MYSSL ssl unique-name *mydomain.com id  11
```

DNS Customization

DNS customization has multiple uses:

- Creating a customized application based on information observed in the DNS request/response
- Extending the scope of an existing application by adding local DNS domains to match

NBAR2 examines DNS request/response traffic and can correlate the DNS response to an application. The IP address returned from the DNS response is cached and used for later packet flows associated with that specific application. The command **ip nbar custom** *application-name* **dns domain-name** *domain-name* **id** *application-id* is used for DNS customization.

Example 6-11 shows a customization for an application named MYDNS using the DNS domain name mydomain.com with selector ID 12.

Example 6-11 *DNS Customization*

```
R31-Spoke# configure terminal
R31-Spoke(config)# ip nbar custom MYDNS dns domain-name *mydomain.com id  12
```

Extending an existing application adds local DNS domain names as criteria to the existing application signatures. To extend an existing application, use the command **ip nbar custom** *application-name* **dns domain-name** *domain-name* **extends** *existing-application*. The existing application is used for application-based policies, and the application name is used to specify the name by which the traffic is reported.

Composite Customization

NBAR2 provides a way to customize applications based on domain names appearing in HTTP, SSL, or DNS. This is done using composite customization. Example 6-12 shows a composite application customization for the application named MYDOMAIN using the HTTP, SSL, or DNS domain name mydomain.com with selector ID 13.

Example 6-12 *Composite Customization*

```
R31-Spoke# configure terminal
R31-Spoke(config)# ip nbar custom MYDOMAIN composite server-name *mydomain.com id  13
```

Similar to DNS customization, a composite customization extends an existing application using the command **ip nbar custom** *application-name* **composite server-name** *server-name* **extends** *existing-application*.

Note Composite customization is the preferred method for domain customization.

Layer 3/Layer 4 Customization

Layer 3/Layer 4 customization is based on the packet tuple and is always matched on the first packet of a flow. The classification of IP address is refined with the command **direction** {**source** | **destination** | **any**}.

Example 6-13 shows a TCP application customization for an application called LAYER4CUSTOM using a set of IP addresses, 10.56.1.10 and 10.56.1.11, matched in any direction, and DSCP EF with selector ID 14.

Example 6-13 *TCP Customization*

```
R31-Spoke# configure terminal
R31-Spoke(config)# ip nbar custom LAYER4CUSTOM transport tcp id 14
R31-Spoke(config-custom)# ip address 10.56.1.10 10.56.1.11
R31-Spoke(config-custom)# direction any
R31-Spoke(config-custom)# dscp ef
```

Byte Offset Customization

Byte offset customization enables defining an application based on specific payload content.

Example 6-14 shows a byte offset customization for an application called MYHOME with the ASCII string HOME in the fifth byte of the first packet of the flow with TCP port 4500 and selector ID 15.

Example 6-14 *Byte Offset Customization*

```
R31-Spoke# configure terminal
R31-Spoke(config)# ip nbar custom MYHOME 5 ascii HOME tcp 4500 id 15
```

Manual Application Attributes Customization

As described in earlier sections, NBAR2 protocols provide a set of application attributes for the network application that they identify. The attribute options are predefined in the Protocol Pack. NBAR2 allows setting attributes associated to a given application from the set of available values. To customize the NBAR2 application attributes:

Step 1. Create an attribute map profile.

This is done with the command **ip nbar attribute-map** *profile-name*.

Step 2. **Set the attribute values for each attribute type.**

Use the command **attribute** *attribute-type attribute-value*.

Step 3. **Apply the attribute map to an application.**

The command for this is **ip nbar** *attribute-set application-name attribute-map*.

Not every attribute must be modified when an application attribute is changed. Assume that Skype is used to support business communication and therefore needs to have the business relevance changed from *business-irrelevant* to *business-relevant*.

Example 6-15 shows how the NBAR2 attribute *business-relevance* is identified and modified for the Skype protocol, and then how the change is verified.

Example 6-15 *Modifying an Application's NBAR2 Attributes*

```
R31-Spoke# show ip nbar protocol-attribute skype

           Protocol Name : skype
               encrypted : encrypted-yes
                  tunnel : tunnel-no
                category : consumer-messaging
            sub-category : consumer-multimedia-messaging
       application-group : skype-group
          p2p-technology : p2p-tech-yes
           traffic-class : multimedia-conferencing
      business-relevance : business-irrelevant
```

```
R31-Spoke(config)# ip nbar attribute-map SKYPE-RELEVANT
R31-Spoke(config-attribute-map)# attribute business-relevance business-relevant
R31-Spoke(config-attribute-map)# exit
R31-Spoke(config)# ip nbar attribute-set skype SKYPE-RELEVANT
```

```
R31-Spoke# show ip nbar protocol-attribute skype

           Protocol Name : skype
               encrypted : encrypted-yes
                  tunnel : tunnel-no
                category : consumer-messaging
            sub-category : consumer-multimedia-messaging
       application-group : skype-group
          p2p-technology : p2p-tech-yes
           traffic-class : multimedia-conferencing
      business-relevance : business-relevant
```

NBAR2 State with Regard to Device High Availability

In highly available environments, after a switchover event, NBAR2 running on the device does not maintain the flow classification and restarts the classification process. Application information propagated by the NBAR2 controller logic is updated on the next controller update when the device becomes available.

Encrypted Traffic

In today's networks, a high percentage of network traffic is encrypted. NBAR2 provides multiple mechanisms to classify encrypted traffic, described in this section:

- **Authoritative sources:** NBAR2 integrates with authoritative sources to use a specific, known flow tuple to accurately classify the traffic of a particular application. This is the best encrypted traffic classification mechanism.
- **DNS mechanisms:** NBAR2 uses DNS request/response traffic to classify applications destined for a given domain.
- **SSL certificates:** NBAR2 uses SSL certificate signatures to classify applications destined for a given domain.
- **Statistical and behavioral mechanisms:** NBAR2 uses a set of statistical and behavioral mechanisms to identify certain applications based on the traffic patterns.

NBAR2 Interoperability with Other Services

NBAR2 is required to run together with various routing technologies such as QoS, WAAS, NAT, firewalls, monitoring, PfR, and so on. It provides interoperability with various configuration services used by various use cases. To support this, NBAR2 and other services run in a specific order within the device to provide the best application recognition results:

- NBAR2 processes packets as early as possible after encrypted VPN decapsulation (clear traffic).
- NBAR2 processes traffic in both directions to inspect packets between devices.
- When NBAR2 is applied on multiple interfaces, it attempts to process each packet once and carry the classification results as metadata across the device.
- Special methods are applied when running in combination with application optimization (WAAS) where the application is stored and derived per flow for both unoptimized and optimized traffic, and with NAT where the application is provided in both inside and outside traffic.

NBAR2 Protocol Discovery

NBAR2 protocol discovery provides a simple method for discovering applications passing through an interface.

Protocol discovery can provide the following statistics, per application, for any protocol traffic supported by NBAR2, for each enabled interface:

- Total number of input packets and bytes
- Total number of output packets and bytes
- Input bit rates
- Output bit rates

NBAR2 protocol discovery supports statistics collection for up to 32 interfaces in parallel.

Note NBAR2 protocol discovery is used primarily for discovering applications and for troubleshooting. It is not necessary to enable it for a standard deployment. NBAR2 is enabled automatically whenever a service requests application-based policies.

Enabling NBAR2 Protocol Discovery

NBAR2 protocol discovery is enabled for a specific interface using the following command in interface configuration mode for the desired interface: **ip nbar protocol-discovery [ipv4 | ipv6]**.

Example 6-16 shows how to enable NBAR2 protocol discovery for both IPv4 and IPv6 for interface GigabitEthernet0/0.

Example 6-16 *Enabling NBAR2 Protocol Discovery*

```
R31-Spoke(config)# interface GigabitEthernet 0/0
R31-Spoke(config-if)# ip nbar protocol-discovery
```

Displaying NBAR2 Protocol Discovery Statistics

To display protocol discovery results, use the command **show ip nbar protocol-discovery** [**interface** *type number*] [**stats** {**byte-count** | **bit-rate** | **packet-count** | **max-bit-rate**}] [**protocol** *protocol-name* | **top-n** *number*].

Example 6-17 shows the NBAR2 protocol discovery statistics results for the four most active protocols and for interface GigabitEthernet0/0.

Example 6-17 *NBAR2 Protocol Discovery Statistics*

```
R31-Spoke# show ip nbar protocol-discovery interface GigabitEthernet 0/0 top-n 4

 GigabitEthernet0/0

 Last clearing of "show ip nbar protocol-discovery" counters 1d06h

                            Input                    Output
                            -----                    ------
   Protocol                 Packet Count             Packet Count
                            Byte Count               Byte Count
                            5min Bit Rate (bps)      5min Bit Rate (bps)
                            5min Max Bit Rate (bps)  5min Max Bit Rate (bps)
   ------------------------ ------------------------ ------------------------
   cifs                     19923531                 39002528
                            6473455995               37313113039
                            0                        0
                            21464000                 53795000
   vmware-vsphere           31607442                 16812217
                            32654628841              1706705556
                            0                        0
                            54040000                 1477000
   vnc                      9600398                  4794044
                            9345917474               286538197
                            0                        0
                            16384000                 446000
   webex-meeting            8784096                  5950890
                            8142257441               1578210867
                            0                        0
                            11582000                 1761000
   Total                    116902145                114366234
                            82846046737              58445765327
                            97000                    209000
                            150433000                84222000
```

Clearing NBAR2 Protocol Discovery Statistics

The NBAR2 protocol discovery statistics are cleared using the command **clear ip nbar protocol-discovery** [**interface** *interface-id*].

Example 6-18 shows how to clear NBAR2 protocol discovery statistics for interface GigabitEthernet0/0.

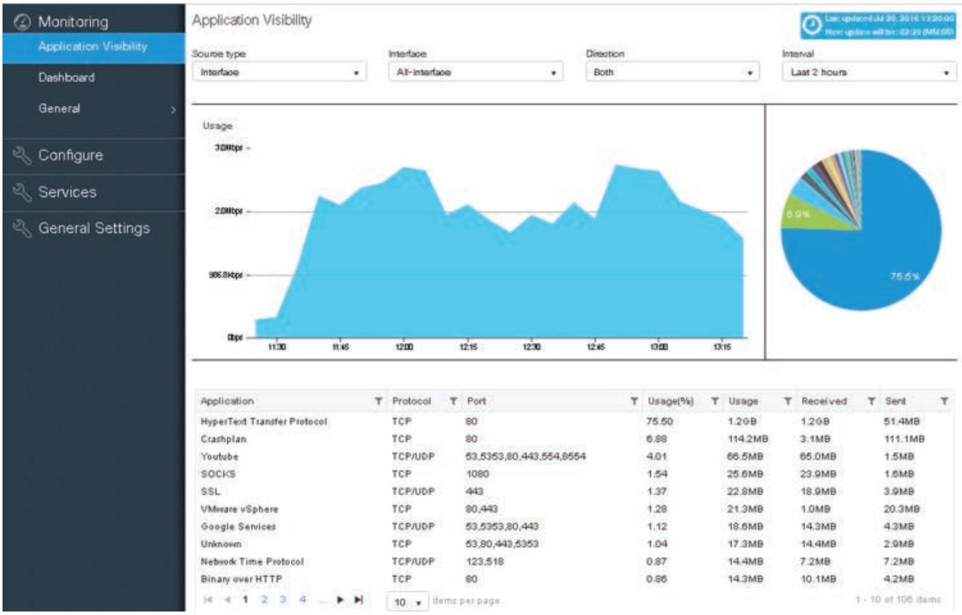
Example 6-18 *Clearing NBAR2 Protocol Discovery Statistics*

```
R31-Spoke# clear ip nbar protocol-discovery interface GigabitEthernet 0/0
```

NBAR2 Visibility Dashboard

NBAR2 provides an on-device traffic visibility dashboard for IOS XE software releases from release 16.3.1. The visibility dashboard includes interactive charts and graphics to provide better traffic visibility, as shown in Figure 6-4.

After the NBAR2 visibility dashboard is enabled, a periodic task is created, which collects the NBAR2 discovery data per minute and stores the data in a local database. The visibility dashboard enables viewing application statistics over time. It is possible to view the statistics according to a specified time window, direction, and interface.



Application Visibility

Source type: Interface
Interface: All-interface
Direction: Both
Interval: Last 2 hours

Application	Protocol	Port	Usage(%)	Usage	Received	Sent
HyperText Transfer Protocol	TCP	80	75.50	1.2GB	1.2GB	51.4MB
Crashplan	TCP	80	6.88	114.2MB	3.1MB	111.1MB
Youtube	TCP/UDP	53,5353,80,443,554,8554	4.01	66.5MB	65.0MB	1.5MB
SOCKS	TCP	1080	1.54	25.6MB	23.9MB	1.6MB
SSL	TCP/UDP	443	1.37	22.8MB	18.9MB	3.9MB
VMware vSphere	TCP	80,443	1.28	21.3MB	1.0MB	20.3MB
Google Services	TCP/UDP	53,5353,80,443	1.12	18.6MB	14.3MB	4.3MB
Unknown	TCP	53,80,443,5353	1.04	17.3MB	14.4MB	2.9MB
Network Time Protocol	TCP/UDP	123,518	0.87	14.4MB	7.2MB	7.2MB
Binary over HTTP	TCP	80	0.86	14.3MB	10.1MB	4.2MB

10 items per page

1 - 10 of 106 items

Figure 6-4 *NBAR2 Visibility Dashboard*

To use the NBAR2 visibility dashboard, perform the following steps:

Step 1. **Enable the HTTP server on the device.**

This is done with the command **ip http server**.

Step 2. **Enable the NBAR2 visibility dashboard.**

Use the command **ip nbar http-services**.

Step 3. **Access the visibility dashboard.**

From a web browser go to the following address: http://*router-ip-or-hostname*/webui/#/applicationVisibility. For example, R31's (loopback IP address of 10.3.0.31) dashboard can be reached at http://10.3.0.31/webui/#/applicationVisibility.

Example 6-19 demonstrates how to enable the NBAR2 visibility dashboard.

Example 6-19 *Enabling the NBAR2 Visibility Dashboard*

```
R31-Spoke(config)# ip http server
R31-Spoke(config)# ip nbar http-services
```

NBAR2 Protocol Packs

An NBAR2 Protocol Pack contains information on applications officially supported by NBAR2 that are compiled and packed together. For each application the Protocol Pack includes information on

- The *application signatures* that enable NBAR2 to identify specific network applications.
- Preconfigured *application attributes* that describe important aspects of the network application. This information is helpful for reporting and for determining traffic policy for the network application and for network visibility.

Release and Download of NBAR2 Protocol Packs

Each software release supports the latest NBAR2 Protocol Pack available at the time of the main software release. NBAR2 provides a hitless way to update the Protocol Pack and supported applications without any traffic or function interruption and without the need to modify the software image on the devices.

The NBAR2 Protocol Pack is released periodically and contains over a thousand applications (and continues to grow) for the most commonly used applications. The list of NBAR2 Protocol Packs is called the NBAR2 Protocol Library. The Protocol Library lists the content of each Protocol Pack, the set of attributes, and release notes. Cisco continues to add applications to Protocol Packs which can be downloaded from the Cisco website at "NBAR Protocol Library." The URL for the Protocol Library web page is provided at the end of this chapter.

NBAR2 Protocol Packs are available for download on the Cisco "Download Software" web page. Specify a device model for which to display the available software, then select "NBAR2 Protocol Packs" in the software type. For each Protocol Pack version, there is a list of software releases for which the Protocol Pack has been released and tested.

Long-lived software releases support multiple Protocol Pack versions, whereas short-lived software releases support only the built-in Protocol Pack within the software image.

NBAR2 Protocol Pack License

To enable the full set of NBAR2 applications, an advanced feature set license is required. The Application Experience software license is used for router devices; check the specific device license requirements on the Cisco website.

Application Customization

The applications included in the NBAR2 Protocol Pack from Cisco provide a foundation for NBAR2 application recognition. To supplement this, NBAR2 includes functions to create custom protocols for localized applications and to learn from local traffic patterns. Localized applications are applications specific to an organization or specific to a geographical location. Learning and customization of protocols are discussed earlier in this chapter.

NBAR2 Protocol Pack Types

NBAR2 provides two ways to load and use Protocol Packs:

- **Built-in:** Each software release has a built-in Protocol Pack bundled with it. The built-in Protocol Pack is the latest one available at the time of the main software release.
- **Loaded:** In addition to the built-in Protocol Pack, multiple additional Protocol Packs can be loaded. NBAR2 uses the most up-to-date loaded Protocol Pack as the active Protocol Pack; the other loaded Protocol Packs are inactive.

NBAR2 Protocol Pack States

An NBAR2 Protocol Pack may be in one of the following states:

- **Active:** The running Protocol Pack that is currently used by the device for application recognition signatures and metadata.
- **Inactive:** A Protocol Pack loaded to the device but not currently running. Inactive Protocol Packs are usually older than the active Protocol Pack or are not compatible with the currently running software.

Identifying the NBAR2 Software Version

Each Protocol Pack release is compatible with specific IOS software versions, described in the Protocol Pack release notes. Compatibility is determined in part by the NBAR2 software engine. The command **show ip nbar version** displays the NBAR2 engine software version.

Example 6-20 shows that the NBAR2 engine software version is 26. The minimum backward-compatible version indicates that Protocol Packs compatible with NBAR2 software engine version 25 are also supported.

Example 6-20 *Displaying the NBAR2 Engine Software Version*

```
R31-Spoke# show ip nbar version

NBAR software version:  26
NBAR minimum backward compatible version:  25

Loaded Protocol Pack(s):

Name:                            Advanced Protocol Pack
Version:                         20.0
Publisher:                       Cisco Systems Inc.
NBAR Engine Version:             26
State:                           Active
```

Verifying the Active NBAR2 Protocol Pack

The command **show ip nbar protocol-pack active** displays the active NBAR2 Protocol Pack.

Example 6-21 shows an active Protocol Pack version 20.0, built for NBAR2 software engine version 26.

Example 6-21 *Verifying the Active NBAR2 Protocol Pack*

```
R31-Spoke# show ip nbar protocol-pack active

Active Protocol Pack:

Name:                            Advanced Protocol Pack
Version:                         20.0
Publisher:                       Cisco Systems Inc.
NBAR Engine Version:             26
State:                           Active
```

Loading an NBAR2 Protocol Pack

At any given time, only one Protocol Pack can be active. Before loading a new Protocol Pack, it is recommended to copy the Protocol Pack to a local disk to avoid any errors after rebooting. To load a Protocol Pack onto a device, use **ip nbar protocol-pack** *protocol-pack* [**force**].

The optional **force** option activates a Protocol Pack of a lower version, even if more up-to-date Protocol Packs are loaded on the system. The **force** option also removes configuration lines used by the active Protocol Pack that are not supported by the loaded Protocol Pack. It is usually used when downgrading a Protocol Pack.

Note An NBAR2 Protocol Pack may be downgraded to a lower version as a part of troubleshooting behavior or to ensure that the Protocol Pack is consistently deployed across all platforms regardless of the OS version on the router.

Example 6-22 shows how to load a new Protocol Pack, newProtocolPack, from the hard disk.

Example 6-22 *Loading a New NBAR2 Protocol Pack*

```
R31-Spoke# configure terminal
R31-Spoke(config)# ip nbar protocol-pack harddisk:newProtocolPack
```

Example 6-23 shows how to revert to the built-in NBAR2 Protocol Pack by using the **default** command.

Example 6-23 *Reverting to the Built-In NBAR2 Protocol Pack*

```
R31-Spoke# configure terminal
R31-Spoke(config)# default ip nbar protocol-pack
```

Example 6-24 shows how to load a lower Protocol Pack version, oldProtocolPack, using the **force** option.

Example 6-24 *Loading an Older Protocol Pack*

```
R31-Spoke# configure terminal
R31-Spoke(config)# ip nbar protocol-pack harddisk:oldProtocolPack force
R31-Spoke(config)# exit
```

Protocol Packs that match a newer software release can be loaded to a device but are not activated. The new Protocol Pack is stored in the running configuration. After the software image is upgraded, NBAR2 chooses the running configuration Protocol Pack at boot time, activating the new Protocol Pack.

NBAR2 Taxonomy File

The NBAR2 taxonomy is an XML-based file that contains information such as the name, description, attributes, global application ID, and other information for every application available in the Protocol Pack. The taxonomy file is mainly used to provide the application details to external servers or to display more details about each application.

The command **show ip nbar protocol-pack {active | inactive | loaded} taxonomy** displays the taxonomy for the Protocol Pack selected.

The NBAR2 taxonomy file generally contains the information for thousands of protocols. It is recommended to redirect the output to a file by using the redirect output modifier, as shown in Example 6-25.

Example 6-25 *Redirecting the NBAR2 XML Taxonomy File to the Hard Disk*

```
R31-Spoke# show ip nbar protocol-pack active taxonomy | redirect
harddisk:taxonomy.xml
```

Protocol Pack Auto Update

The Protocol Pack Auto Update feature assists in updating a large number of devices with the latest compatible Protocol Pack. When a new Protocol Pack becomes available, download the file to a server that is reachable by every device. Platforms with Auto Update enabled check the server periodically. If a newer Protocol Pack is available and compatible, the device downloads the Protocol Pack file and installs it automatically. This procedure centralizes the protocol update, making it unnecessary to update the Protocol Pack on every device manually. It also helps to ensure that all network devices operate consistently with the same Protocol Pack version.

Protocol Pack Configuration Server

Devices with Protocol Pack Auto Update enabled check a configuration file stored on a configuration server. This configuration file provides instructions for the device on the location of the Protocol Pack and the update schedule.

Protocol Pack Source Server

The Protocol Pack source server stores the Protocol Packs themselves. The Protocol Pack configuration server and Protocol Pack source server can be the same server or separate servers.

The following steps are used to enable the Protocol Pack Auto Update function:

Step 1. **Enter Protocol Pack Auto Update configuration mode.**

Use the command **ip nbar protocol-pack-auto-update**.

Step 2. **Configure the source server IP and configuration file location.**

This is done with the command **source-server** *server+location*.

Example 6-26 shows a configuration of source TFTP server address 10.130.1.80 where the configuration file is located in the AutoUpdateConfigDir directory.

Example 6-26 *Protocol Pack Auto Update Configuration*

```
R31-Spoke# configure terminal
R31-Spoke(config)# ip nbar protocol-pack-auto-update
R31-Spoke(config-pp-auto-update)# source-server tftp://10.130.1.80/AutoUpdateConfig-
  Dir
```

The source server configuration provides the instructions for how to load the Protocol Packs. The configuration file is a JSON format file named NBAR_PROTOCOL_PACK_DETAILS.json. Table 6-5 describes the parameters in the JSON file.

Table 6-5 *Routing Protocol Summary*

Parameter	Description
"protocol-pack-server"	(Mandatory) Specifies the server and location of the Protocol Pack server. Example: tftp://10.130.1.200/AutoUpdatePPDir/
"nbar_pp_files"	(Mandatory) Specifies the file locations for Protocol Pack files for various devices and NBAR2 engines.
"schedule": {"daily": | "weekly": | "monthly":} [day] {"hh": hh, "mm": mm}	Specifies the schedule for the NBAR2 Protocol Pack Auto Update interval. Enabled devices check regularly for updates at the scheduled time. ■ **monthly:** Every month on a specific day of the month ■ **weekly:** Every week on a specific day of the week (0 to 6) ■ **hh:hh:** Hour (24-hour time) ■ **mm:mm:** Minute Default value: Daily at 00:00 Specifies the maintenance window (in minutes) for the NBAR2 Protocol Pack update to operate within. NBAR2 will update the Protocol Pack within the maintenance window at the time configured by the schedule parameters. This option is used to spread the upgrade across the network devices. Default value: 60

(Continued)

Table 6-5 *Continued*

Parameter	Description
"clear-previous" {"enable" | "disable"}	Specifies whether to remove unused Protocol Pack files loaded into the device. **enable:** Remove **disable:** Keep Default value: **enable**
"force-upgrade" {"enable" | "disable"}	Specifies whether to use the **force** flag when upgrading a Protocol Pack **enable:** Use the **force** flag. **disable:** Do not use the **force** flag. Default value: **disable**

Example 6-27 shows a typical NBAR_PROTOCOL_PACK_DETAILS.json configuration file. It schedules a daily update at 2:30 a.m. and indicates the directories for Protocol Packs for NBAR2 software engine versions 22 and 23 for ISR devices, and version 23 for ASR and CSR devices.

Example 6-27 *NBAR_PROTOCOL_PACK_DETAILS.json Configuration*

```
{
  "nbar_auto_update_config": {
    "protocol-pack-server": "tftp://10.130.1.200/NbarAutoUpdate/",
    "update-window":30,
    "force-upgrade":true,
    "clear-previous":true,
    "schedule": {
      "weekly": 6,
      "hh": 02,
      "mm": 30
    },
  },
  "nbar_pp_files": {
    "ISR": {
      "22":"isr_protocolpack_dir/pp22",
      "23":"isr_protocolpack_dir/pp23"
    },
    "ASR": {
      "23":"asr_protocolpack_dir/pp23"
    },
    "CSR": {
      "23":["csr_protocolpack_dir/pp23"]
    }
  }
}
```

It is possible to initiate a manual ad hoc Protocol Pack update from a given device using **ip nbar protocol-pack-auto-update now** as shown in Example 6-28.

Example 6-28 *Immediate Protocol Pack Auto Update*

```
R31-Spoke# configure terminal
R31-Spoke(config)# ip nbar protocol-pack-auto-update now
```

```
R31-Spoke# show ip nbar protocol-pack-auto-update

NBAR Auto-Update:
=================

Configuration:
=============
force-upgrade                   : (Default)  Enabled
clear-previous                  : (Default)  Enabled
update-window                   : (Default)  30
source-server                   :  tftp://10.130.1.80/AutoUpdateConfigDir
protocol-pack-directory         : (Default)  harddisk:
schedule                        : (Default)  02:30

Copied files:
==========
File             : harddisk:/NbarAutoUpdate/AsrNbarPP
Copied           : *11:29:11.000 UTC Mon Jan 5 2016



Last run result: SUCCESS
Last auto-update run                         : *11:29:12.000 UTC Mon Jan 5 2016
Last auto-update success                     : *11:29:12.000 UTC Mon Jan 5 2016
Last auto-update successful update           : *11:29:12.000 UTC Mon Jan 5 2016

Last auto-update server-config update        : *16:15:13.000 UTC Mon Jan 5 2016
Success count                                     : 3
Failure count                                     : 0
Success rate                                      : 100 percent

Next AU maintenance estimated to run at      : *17:15:13.000 UTC Mon Jan 5 2016
Next AU update estimated to run at           : *02:41:00.000 UTC Tue Jan 6 2016
```

Validation and Troubleshooting

The health of the NBAR2 system should be checked to ensure that packets are being properly classified. This section describes the most common NBAR2 troubleshooting scenarios and shows how to verify proper operation of the system.

Verify the Software Version

NBAR2 works best with "long-lived" IOS releases, which support Protocol Pack upgrades and bug fixes for the longest possible support period. To check the IOS software version, use **show version**.

Check the Device License

Advanced application recognition by NBAR2 requires the Application Experience license that enables the application visibility and control (AVC) feature. To check the device license, use **show license feature**.

Example 6-29 shows how to verify that AVC is enabled.

Example 6-29   *Verifying That the AVC Feature Is Enabled*

```
R31-Spokeshow license feature
Feature name              Enforcement  Evaluation  Subscription   Enabled  RightToUse
adventerprise             yes          yes         no             yes      yes
advipservices             yes          yes         no             no       yes
ipbase                    no           no          no             no       no
avc                       yes          yes         no             yes      yes
broadband                 no           no          no             no       no
```

Verifying That NBAR2 Is Enabled

NBAR2 is enabled automatically when an application-based service is created. To verify that the application-based policy is attached on the correct interfaces, use the command **show running-configuration**. Then the NBAR statistics should show as *activated* with the command **show platform software nbar statistics** as shown in Example 6-30.

Example 6-30   *Verifying That NBAR2 Is Enabled*

```
R31-Spoke# show platform software nbar statistics | include NBAR
NBAR state is ACTIVATED
NBAR config send mode is ASYNC
NBAR config state is READY
NBAR update ID   73
NBAR batch ID ACK  73
NBAR max protocol ID ever  1933
NBAR Control-Plane Version: 26
```

Verifying the Active NBAR2 Protocol Pack

For best classification results, verify that the latest NBAR2 Protocol Pack is activated on the system. NBAR2 activates Protocol Packs only if they match the NBAR2 engine software version.

Example 6-31 shows how to check the NBAR2 active Protocol Pack and the NBAR2 engine software version. In the example, the NBAR2 engine on the system is version 26, which matches the version required by the Protocol Pack.

Example 6-31 *Verifying the Active NBAR2 Protocol Pack and Software Engine*

```
R31-Spoke# show ip nbar protocol-pack active

Active Protocol Pack:


Name:                            Advanced Protocol Pack
Version:                         20.0
Publisher:                       Cisco Systems Inc.
NBAR Engine Version:             26
State:                           Active
```

```
R31-Spoke# show ip nbar version

NBAR software version:  26
NBAR minimum backward compatible version:  25

Loaded Protocol Pack(s):

Name:                            Advanced Protocol Pack
Version:                         20.0
Publisher:                       Cisco Systems Inc.
NBAR Engine Version:             26
State:                           Active
```

Checking That Policies Are Applied Correctly

To match a given type of traffic, verify that the policy includes all relevant applications. It is recommended to use NBAR2 attributes as much as possible to simplify the configuration and allow dynamically created applications to better fit the policy. For example, to match Microsoft cloud applications, use the application group attribute *ms-cloud-group*, which provisions several related applications as shown in Example 6-32.

For QoS policies, it is recommended to use the NBAR2 *traffic-class* attribute.

Verify that control and data protocols or DNS requests related to a given application are treated properly.

Example 6-32 *Use of Application Attributes*

```
R31-Spoke# show ip nbar attribute application-group ms-cloud-group
  ms-live-accounts    Windows Live Services Authentication
  ms-office-365       Microsoft Office 365
  ms-office-web-apps  Web-based versions of Microsoft Word, Excel,
                      PowerPoint and OneNote
  ms-services         Microsoft Services
  outlook-web-service Microsoft Web-Based email services under Outlook brand,
                      part of Off
  skydrive            SkyDrive Cloud Storage Server From Microsoft
```

Reading Protocol Discovery Statistics

Protocol discovery can be used to view the amount of traffic running through each interface, per application. Use protocol discovery to discover the applications, and to verify that the application-based policies are handling the applications correctly.

Refer to the "NBAR2 Protocol Discovery" section for details on how to enable and show protocol discovery statistics.

Granular Traffic Statistics

A NetFlow per-flow policy can be enabled on specific traffic to view granular per-flow statistics. Refer to Chapter 10, "Application Visibility," for details on how to define NetFlow monitors that contain higher-granularity information. The section "View Raw Data Directly on the Router" explains how to define a monitor that shows raw data on the router without exporting it to an external collector.

Using granular per-flow data can help in verifying that traffic is symmetric and that both directions of the traffic are running through the NBAR2-enabled interfaces.

Discovering Generic and Unknown Traffic

NBAR2 provides the capability to list Layer 7 information of generic HTTP or SSL traffic, using the host name in the HTTP header or the SSL certificate. It can also display the most commonly occurring sockets or ports (top sockets, top ports) for unclassified (unknown) traffic.

Refer to the "Auto-learn Traffic Analysis Engine" section in this chapter for further information on how to view auto-learn results for generic and unknown traffic, and how to customize protocols using the results.

Verifying the Number of Flows

NBAR2 relies on a flow table to track states for each flow. To verify that the number of flows has not reached the maximum allowed, use the command **show ip nbar resources flow**.

Example 6-33 shows how to use the command. Verify that the peak and active sessions and memory have not exceeded the maximum allowed values.

Example 6-33 *Checking NBAR2 Flow Usage*

```
R31-Spoke# show ip nbar resources flow
NBAR flow statistics
        Maximum no of sessions allowed : 5000000
        Maximum memory usage allowed   : 2936012 KBytes
        Active sessions                : 44
        Active memory usage            : 10523 KBytes
        Peak session                   : 1816
        Peak memory usage              : 11169 Kbytes
```

Summary

Network engineers must understand the network traffic that flows across their devices to provide an optimal user experience. More and more applications are sharing network protocols (for example, HTTP) or encrypting the network traffic, making it more difficult to classify the applications that are being used. NBAR2 provides the best available tools and techniques for classifying network applications and provides a wide-ranging set of attributes that are useful for defining and visualizing the network policies.

This chapter provided network engineers with numerous techniques to use when deploying NBAR2 in order to provide application visibility for proper application-based deployment of QoS or Performance Routing policies.

Further Reading

Cisco. *Cisco IOS Software Configuration Guides*. www.cisco.com.

Cisco. “NBAR2 Protocol Library.” www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_nbar/prot_lib/config_library/nbar-prot-pack-library.html.

Claise, B, P. Aitken, and N. Ben-Dvora. RFC 6759, “Cisco Systems Export of Application Information in IP Flow Information Export (IPFIX).” IETF, November 2012. https://tools.ietf.org/html/rfc6759.

This page intentionally left blank

Chapter 7

Introduction to Performance Routing (PfR)

This chapter covers the following topics:

- Performance Routing (PfR)
- Introduction to the IWAN domain
- Intelligent path control principles

Bandwidth cost, WAN latency, and lack of bandwidth availability all contribute to the complexities of running an efficient and cost-effective network that meets the unique, application-heavy workloads of today's enterprise organizations. But as the volume of content and applications traveling across the network grows exponentially, organizations must optimize their WAN investments.

Cisco *Performance Routing (PfR)* is the IWAN intelligent path control component that can help administrators to accomplish the following:

- Augment the WAN with additional bandwidth to including lower-cost connectivity options such as the Internet
- Realize the cost benefits of provider flexibility and the ability to choose different transport technologies (such as MPLS L3VPN, VPLS, or the Internet)
- Offload the corporate WAN with highly secure direct Internet access
- Improve application performance and availability based upon an application's performance requirements
- Protect critical applications from fluctuating WAN performance

Performance Routing (PfR)

Cisco Performance Routing (PfR) improves application delivery and WAN efficiency. PfR dynamically controls data packet forwarding decisions by looking at application type, performance, policies, and path status. PfR protects business applications from fluctuating WAN performance while intelligently load-balancing traffic over the best-performing path based on the application policy.

Simplified Routing over a Transport-Independent Design

One of the critical IWAN components and also a key design decision was to architect the next-generation WAN around a *transport-independent design (TID)*. The choice of DMVPN was extensively explained in Chapter 2, "Transport Independence." This overlay approach allows the use of a single routing protocol over the WAN and greatly simplifies the routing decision process and Performance Routing in multiple ways, two of the main ones being

- Simplified reachability information
- Single routing domain

The first benefit of this overlay approach is simplified reachability information.

The traditional routing protocols were designed to solve the endpoint reachability problem in a hop-by-hop destination-only forwarding environment of unknown topology. The routing protocols choose only the best path based on statically assigned cost. There are a few exceptions where the network path used can be somewhat engineered. Some routing protocols can select a path that is not the shortest one (BGP, *MPLS traffic engineering [TE]*).

Designing deterministic routing behavior is difficult with multiple transport providers but is much simpler thanks to DMVPN. The DMVPN network topology is flat, and it is consistent because it is an overlay network that masks the network complexity underneath. This approach simplifies the logical view of the network and minimizes fundamental topology changes. Logically, only reachability to the next hop across the WAN can change.

An overlay network's routing information is very simple: a set of destination prefixes, and a set of potential transport next hops for each destination. As a result, PfR just needs a *mapping service* that stores and serves all resolved forwarding states for connectivity per overlay network. Each forwarding state contains destination prefix, next hop (overlay IP address), and corresponding transport address.

The second benefit of using overlay networks is the single routing domain design. In traditional hybrid designs, it is common to have two (or more) routing domains:

- One routing domain for the primary path over MPLS—EBGP, static, or default routes
- One routing domain on the secondary path over the Internet—EIGRP, IBGP, or floating static routes

The complexity increases when routes are exchanged between the multiple routing domains, which can lead to suboptimal routing or routing loops. Using DMVPN for all WAN transports allows the use of a single routing protocol for all paths regardless of the of transport choice. Whether the topology is dual hybrid (MPLS plus Internet) or dual Internet (two Internet paths), the routing configuration remains exactly the same, meaning that if there is a change in how your provider chooses to deliver connectivity, or you wish to add or change a provider underneath the DMVPN, the investment in your WAN routing architecture is secure.

EIGRP and IBGP are the best routing protocol options today with DMVPN.

After routing connectivity is established, PfR enters the picture and provides the advanced path control in IWAN. PfR is not a replacement for the routing protocol and never will be. As an adjunct, PfR uses the next-hop information from the routing protocol and overrides it based on real-time performance and link utilization ratio. This next-hop information per destination prefix is critical for PfR to work correctly and is a critical element in the routing design. Having a single routing domain and a very basic mapping service requirement has greatly simplified PfR interaction with the routing protocol.

“Classic” Path Control Used in Routing Protocols

Path control, commonly referred to as “traffic engineering,” is the process of choosing the network path on which traffic is sent. The simplest form is trivial: send all traffic down the primary path unless the path goes down; in that case, send everything through the backup path.

Figure 7-1 illustrates the concept where R31 (branch) sends traffic to R11 (headquarters). When R31’s link to the MPLS provider fails, traffic is sent through the Internet.

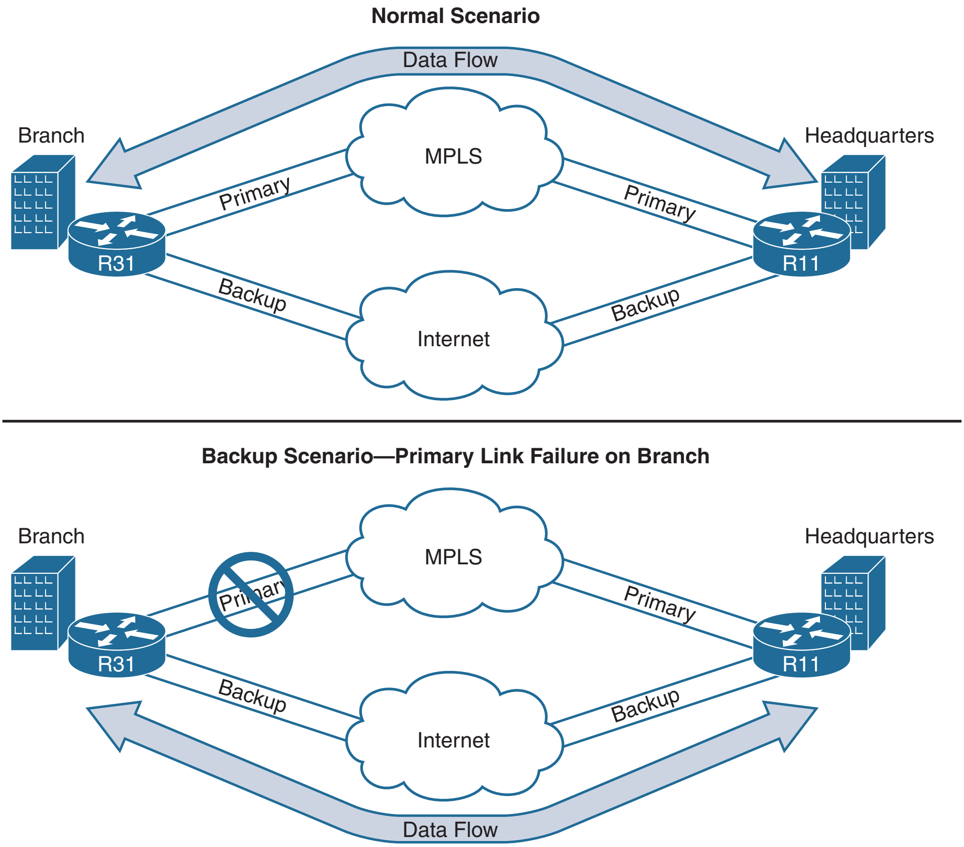


Figure 7-1 *Traffic Flow over Primary and Backup Links*

This approach has two main drawbacks:

- Traffic is forwarded over a single path regardless of the application type, performance, or bandwidth issues.
- The backup path is used only when the primary link goes down and not when there is performance degradation or brownouts over the primary path because the routing protocol peers are usually still up and running and do not detect such performance issues.

Path Control with Policy-Based Routing

The next level of path control lets the administrator specify categories of traffic to send on a specific path as long as that path remains up. One of the most common options is the use of *policy-based routing (PBR)*, routing based on DSCP values:

- DSCP values that are mapped to critical business applications and voice/video types of applications are assigned a next hop that is over the preferred path.
- DSCP values that are mapped to best-effort applications or applications that do not suffer from performance degradation are assigned a next hop over the secondary path.

However, this approach is not intelligent and does not take into account the dynamic behavior of the network. Routing protocols have keepalive timers that can determine if the next hop is available, but they cannot determine when the path selected suffers from degraded performance, and the system cannot compensate.

Figure 7-2 illustrates the situation where R31 (branch) sends traffic to R11 (headquarters). When R31's path across the MPLS provider experiences performance issues, traffic continues to be sent through the MPLS backbone. PBR alone is unaware of any performance problems. An additional mechanism is needed to detect events like these, such as the use of IP SLA probes.

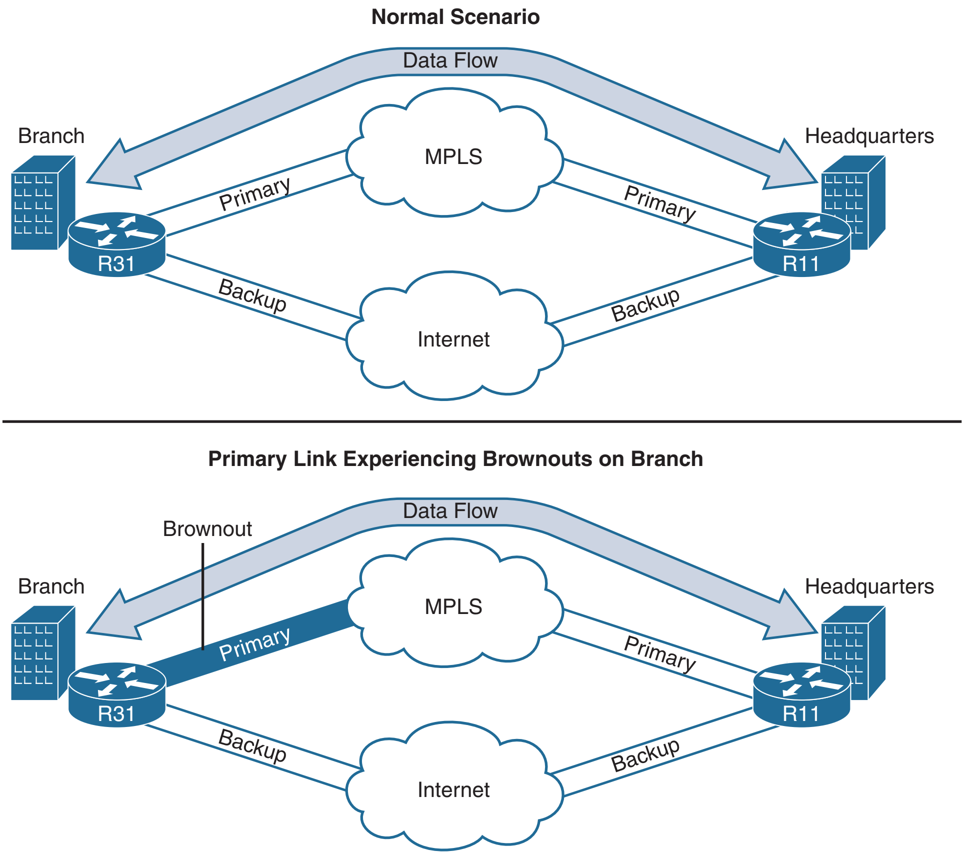


Figure 7-2 *PBR's Inability to Detect Problematic Links*

Intelligent Path Control—Performance Routing

Classic routing protocols or path control with PBR cannot detect performance issues and fall back affected traffic to an alternative path. Intelligent path control solves this problem by monitoring actual application performance on the path that the applications are traversing, and by directing traffic to the appropriate path based on these real-time performance measurements.

When the current path experiences performance degradation, Cisco intelligent path control moves the affected flows according to user-defined policies.

Figure 7-3 illustrates the situation where R31 sends traffic to R11. When R31's path across the MPLS provider experiences performance issues, only affected traffic is sent to the Internet path. The choice of traffic to fall back is based on defined policies. For example, voice or business application flows are forwarded over the secondary path, whereas best-effort traffic remains on the MPLS path.

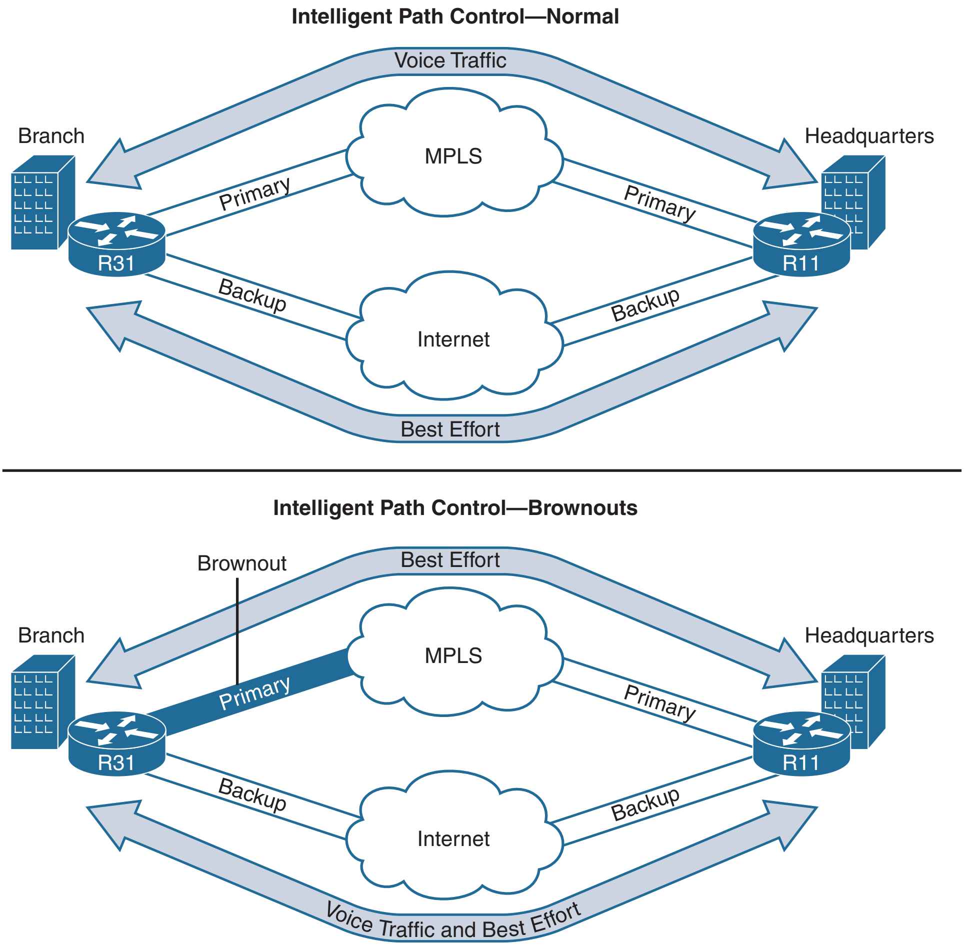


Figure 7-3 *Traffic Flow over Multiple Links with Cisco Intelligent Path Control*

Advanced path control should include the following:

- Detection of issues such as delay, loss, jitter, and defined path preference before the associated application is impacted.
- Passive performance measurement based on real user traffic when available and passively monitored on existing WAN edge routers. This helps support SLAs to protect critical traffic.
- Efficient load distribution across the WAN links for medium-priority and best-effort traffic.
- Effective reaction to any network outages before they can affect users or other aspects of the network. These include *blackouts* that cause a complete loss of connectivity as well as *brownouts* that are network slowdowns caused by path degradation along the route to the destination. Although blackouts can be detected easily, brownouts are much more challenging to track and are usually responsible for bad user experience.
- Application-based policies that are designed to support the specific performance needs of applications (for example, point of sale, enterprise resource planning [ERP], and so on).
- Low WAN overhead to ensure that control traffic is not contributing to overall traffic issues.
- Easy management options, including a single point of administration and the ability to scale without a stacked deployment.

Cisco Performance Routing (PfR), part of Cisco IOS software, provides intelligent path control in IWAN and complements traditional routing technologies by using the intelligence of a Cisco IOS infrastructure to improve application performance and availability.

As explained before, PfR is not a replacement for the routing protocols but instead runs alongside of them to glean the next hop per destination prefix. PfR has APIs with NHRP, BGP, EIGRP, and the routing table to request information. It can monitor and then modify the path selected for each application based on advanced criteria, such as reachability, delay, loss, and jitter. PfR intelligently load-balances the remainder of the traffic among available paths based on the tunnel bandwidth utilization ratio.

Note The routing table, known as the *routing information base (RIB)*, is built from dynamic routing protocols and static and directly connected routes. The routing table is referred to as the RIB throughout the rest of this chapter.

Cisco PfR has evolved and improved over several releases with a focus on simplicity, ease of deployment, and scalability. Table 7-1 provides a list of features that have evolved with each version of PfR.

Table 7-1 *Evolution of PfR Versions and Features*

Version	Features
PfR/Optimized Edge Routing (OER)	Internet edge Basic WAN Provisioning per site per policy Thousands of lines of configuration
PfRv2	Policy simplification App path selection Scale 500 sites Tens of lines of configuration
PfRv3	Centralized provisioning Application Visibility Control (AVC) infrastructure VRF awareness Scale 2000 sites Hub configuration only Multiple data centers Multiple next hops per DMVPN network

Introduction to PfRv3

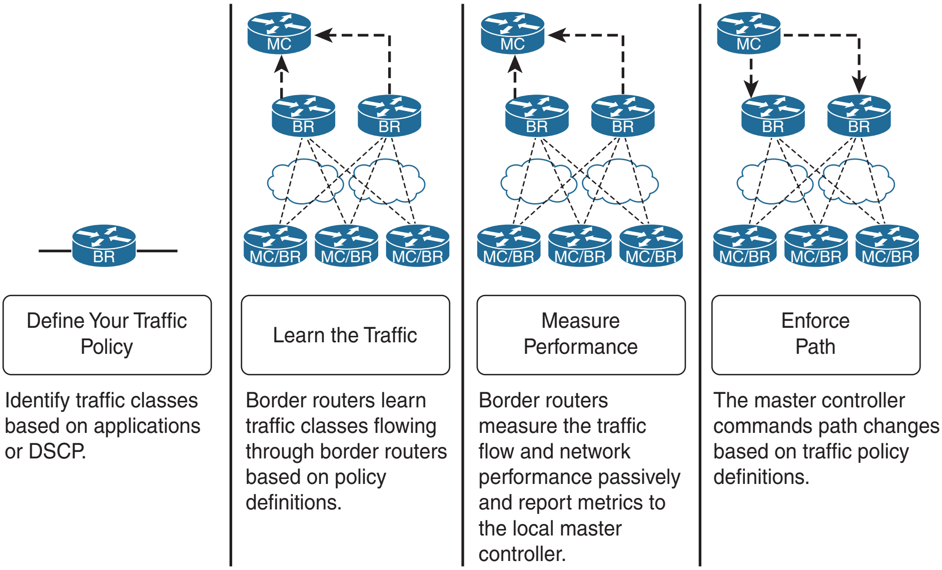
Performance Routing Version 3 (PfRv3) is the latest generation of the original PfR created more than ten years ago. PfRv3 focuses on ease of use and scalability to make it easy to transition to an intelligent network with PfR. It uses one-touch provisioning with multisite coordination to simplify its configuration and deployment from previous versions of PfR. PfRv3 is a DSCP- and application-based policy-driven framework that provides multisite path control optimization and is bandwidth aware for WAN- and cloud-based applications. PfRv3 is tightly integrated with existing AVC components such as Performance Monitor, QoS, and NBAR2.

PfR is composed of devices performing several roles, which are *master controller (MC)* and *border router (BR)*. The MC serves as the control plane of PfR, and the BR is the forwarding plane which selects the path based on MC decisions.

Note The MC and BR are components of the IOS software features on WAN routers.

Figure 7-4 illustrates the mechanics of PfRv3. Traffic policies are defined based on DSCP values or application names. Policies can state requirements and preferences for applications and path selection. A sample policy can state that voice traffic uses preferred path MPLS unless delay is above 200 ms. PfR learns the traffic, then starts measuring the bandwidth and performance characteristics. Then the MC makes a decision by comparing the real-time metrics with the policies and instructs the BRs to use the appropriate path.

Note The BRs automatically build a tunnel (known as an *auto-tunnel*) between other BRs at a site. If the MC instructs a BR to redirect traffic to a different BR, traffic is forwarded across the auto-tunnel to reach the other BR.



Identify traffic classes based on applications or DSCP.

Border routers learn traffic classes flowing through border routers based on policy definitions.

Border routers measure the traffic flow and network performance passively and report metrics to the local master controller.

The master controller commands path changes based on traffic policy definitions.

Figure 7-4 *Mechanics of PfRv3*

Note The first iteration of PfRv3 was introduced in summer 2014 with IOS 15.4(3)M and IOS XE 3.13.

Introduction to the IWAN Domain

An IWAN domain is a collection of sites that share the same set of policies and are managed by the same logical PfR *domain controller*. Each site runs PfR and gets its path control configuration and policies from the logical IWAN domain controller through the IWAN peering service. At each site, an MC is the local decision maker and controls the BRs responsible for performance measurement and path enforcement. The IWAN domain can be an entire enterprise WAN, a particular region, and so forth.

The key point for PfRv3 is that provisioning is fully centralized at a logical domain controller, whereas path control decisions and enforcement are fully distributed within the sites that are in the domain, making the solution very scalable.

Figure 7-5 shows a typical IWAN domain with central and branch sites. R10, R20, R31, R41, and R51 are all MCs for their corresponding sites and retrieve their configuration from the logical domain controller. R11, R12, R21, R22, R31, R41, R51, and R52 are all BRs that report back to their local MC. Notice that R31, R41, and R51 operate as both the MC and the BR for their sites.

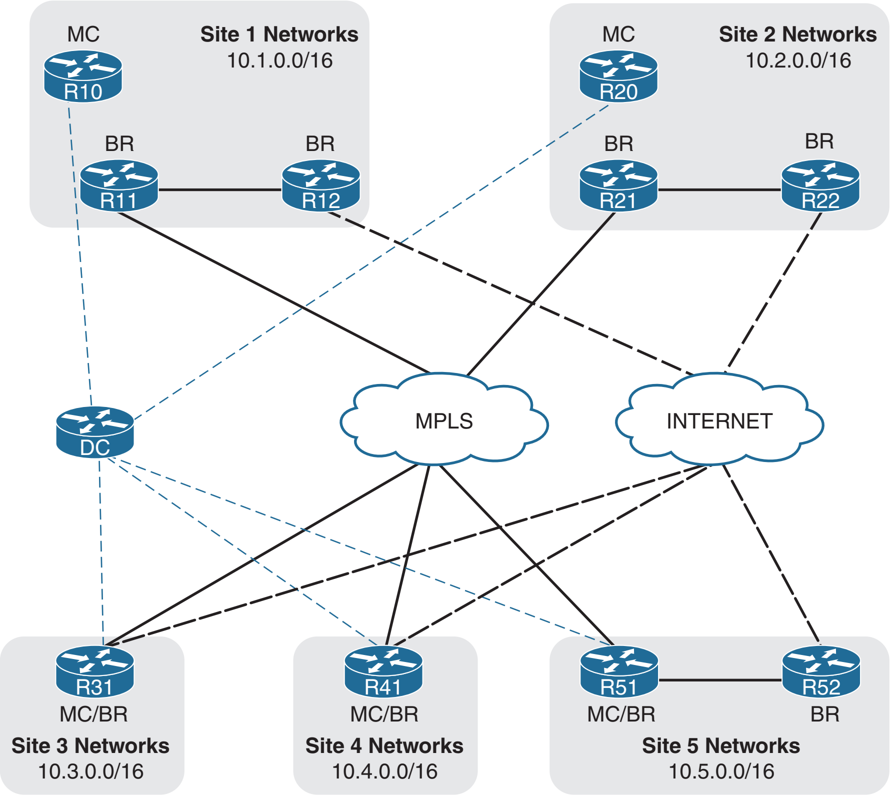


Figure 7-5 *IWAN Domain Concepts*

Note In the remainder of this book, all references to PfR mean PfRv3.

IWAN Sites

An IWAN domain includes a mandatory hub site, optional transit sites, as well as branch sites. Each site has a unique identifier called a *site ID* that is derived from the loopback address of the local MC.

Central and headquarters sites play a significant role in PfR and are called IWAN *Points of Presence (POPs)*. Each site has a unique identifier called a POP ID. These sites house DMVPN hub routers and therefore provide the following traffic flows (streams):

- Traditional DMVPN spoke-to-hub connectivity.
- Spoke-to-hub-to-spoke connectivity until DMVPN spoke-to-spoke tunnels establish.
- Connectivity through NHS chaining until DMVPN spoke-to-spoke tunnels establish.
- Transit connectivity to another site via a data center interconnect (DCI) or shared data center network segment. In essence, these sites act as *transit sites* for the traffic crossing them. Imagine in Figure 7-5 that R31 goes through R21 to reach a network that resides in Site 1. R21 does not terminate the traffic at the local site; it provides transit connectivity to Site 1 via the DCI.
- Data centers may or may not be colocated with the hub site. To elaborate further, some hub sites contain data centers whereas other hub sites do not contain data centers (such as outsourced colocation cages).

Hub site

- The logical domain controller functions reside on this site's MC.
- Only one hub site exists per IWAN domain because of the uniqueness of the logical domain controller's presence. The MC for this site is known as the Hub MC, thereby making this site the hub site.
- MCs from all other sites (transit or branch) connect to the Hub MC for PfR configuration and policies.
- A POP ID of 0 is automatically assigned to a hub site.
- A hub site may contain all other properties of a transit site as defined below.

Transit sites

- Transit sites are located in an enterprise central site, headquarters, or carrier-neutral facilities.
- They provide transit connectivity to access servers in the data centers or for spoke-to-spoke traffic.
- A data center may or may not be colocated with the transit site. A data center can be reached via a transit site.

- A POP ID is configured for each transit site. This POP ID has to be unique in the domain.
- The local MC (known as a *Transit MC*) peers with the Hub MC (domain controller) to get its policies and to monitor configuration and timers.

Branch sites

- These are always DMVPN spokes and are stub sites where traffic transit is not allowed.
- The local Branch MC peers with the logical domain controller (Hub MC) to get its policies and monitoring guidelines.

Figure 7-6 shows the IWAN sites in a domain with two central sites (one is defined as the hub site and the other as a transit site). R10, R11, and R12 belong to the hub site, and R20, R21, and R22 belong to a transit site. R31, R41, R51, and R52 belong to a branch site. The dotted lines represent the site's local MC peering with the Hub MC.

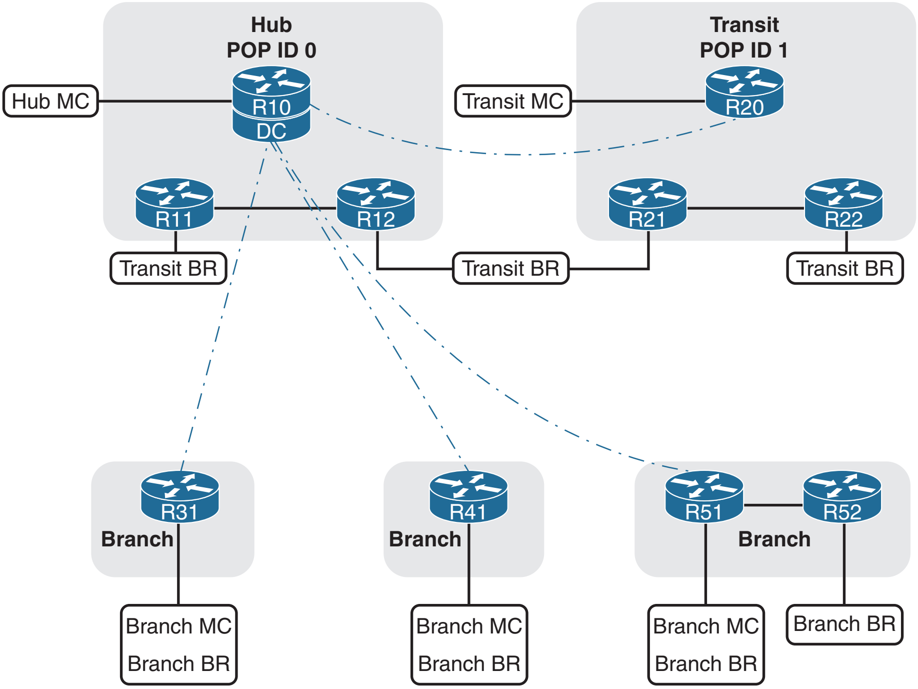


Figure 7-6 *IWAN Domain Hub and Transit Sites*

Device Components and Roles

The PfR architecture consists of two major Cisco IOS components, a master controller (MC) and a border router (BR). The MC is a policy decision point where policies are defined and applied to various traffic classes (TCs) that traverse the BR systems. The MC can be configured to learn and control TCs on the network:

- **Border routers (BRs)** are in the data-forwarding path. BRs collect data from their Performance Monitor cache and smart probe results, provide a degree of aggregation of this information, and influence the packet forwarding path as directed by the site local MC to manage router traffic.
- **The master controller (MC)** is the policy decision maker. At a large site, such as a data center or campus, the MC is a dedicated (physical or logical) router. For smaller branch locations, the MC is typically colocated (configured) on the same platform as the BR. As a general rule, large locations manage more network prefixes and applications than a branch deployment, thus consuming more CPU and memory resources for the MC function. Therefore, it is a good design practice to dedicate a chassis for the MC at large sites.

Each site in the PfR domain must include a local MC and at least one BR.

The branch typically manages fewer active network prefixes and applications. Because of the costs associated with dedicating a chassis at each branch, the network manager can colocate the local MC and BR on the same router platform. CPU and memory utilization should be monitored on platforms operating as MCs, and if utilization is high, the network manager should consider an MC platform with a higher-capacity CPU and memory. The local MC communicates with BRs and the Hub MC over an authenticated TCP socket but has no requirement for populating its own IP routing table with anything more than a route to reach the Hub MC and local BRs.

PfR is an intelligent path selection technology and requires

- At least two external interfaces under the control of PfR
- At least one internal interface under the control of PfR
- At least one configured BR
    - If only one BR is configured, both external interfaces are attached to the single BR.
    - If more than one BR is configured, two or more external interfaces are configured across these BRs.

The BR, therefore, owns external links, or exit points; they may be logical (tunnel interfaces) or physical links (serial, Ethernet, and so on). With the IWAN prescriptive design, external interfaces are always logical DMVPN tunnels.

A device can fill five different roles in an IWAN domain:

- **Hub MC:** This is the MC at the hub site. It acts as MC for the site, makes optimization decisions for that site, and provides the path control policies for all the other MCs. The Hub MC contains the logical PfR domain controller role.
- **Transit MC:** This is the MC at a transit site that makes optimization decision for those sites. There is no policy configuration on Transit MCs because they receive their policies from the Hub MC.
- **Branch MC:** The Branch MC is the MC for branch sites that makes optimization decisions for those sites. There is no policy configuration on Branch MCs because they receive their policies from the Hub MC.
- **Transit BR:** The Transit BR is the BR at a hub or transit site. The WAN interface terminates in the BRs. PfR is enabled on these interfaces. At the time of this writing, only one WAN interface is supported on a Transit BR. This limitation is overcome by using multiple BR devices.

Note Some Cisco documentation may refer to a Transit BR as a Hub BR, but the two function identically because transit site capabilities were included in a later release of PfR.

- **Branch BR:** The Branch BR resides at the branch site and forwards traffic based on the decisions of the Branch MC. The only PfR configuration is the identification of the Branch MC and setting its role as a BR. The WAN interface that terminates on the device is detected automatically.

The PfR Hub MC is currently supported only on the IOS and IOS XE operating systems.

IWAN Peering

PfR uses an IWAN peering service between the MCs and BRs which is based on a publish/subscribe architecture. The current IWAN peering service uses Cisco SAF to distribute information between sites, including but not limited to

- Learned site prefix
- PfR policies
- Performance Monitor information

The IWAN peering service provides an environment for service advertisement and discovery in a network. It is made up of two primary elements: client and forwarder.

- An IWAN peering service client is a producer (advertises to the network), a consumer of services (requests a service from the network), or both.
- An IWAN peering service SAF forwarder receives services advertised by clients, distributes the services reliably through the network, and makes services available to clients.

- An IWAN peering service client needs to send a register message to a forwarder before it is able to advertise (publish) or request (subscribe to) services.

The IWAN peering service also adopts a logical unicast topology to implement the peering system. Each instance that joins the IWAN peering service serves as both a client and a forwarder:

- The Hub MC listens for unicast packets for advertisements or publications from Transit MCs, Branch MCs, and local BRs.
- The Transit MC peers with the Hub MC and listens to its local BRs.
- The Branch MC peers with the Hub MC and listens to its local BRs.
- BRs always peer with their local MC.

Figure 7-7 illustrates the IWAN peering service with the policies advertised from the Hub MC, the advertisement of monitors, and the exchange of site prefixes.

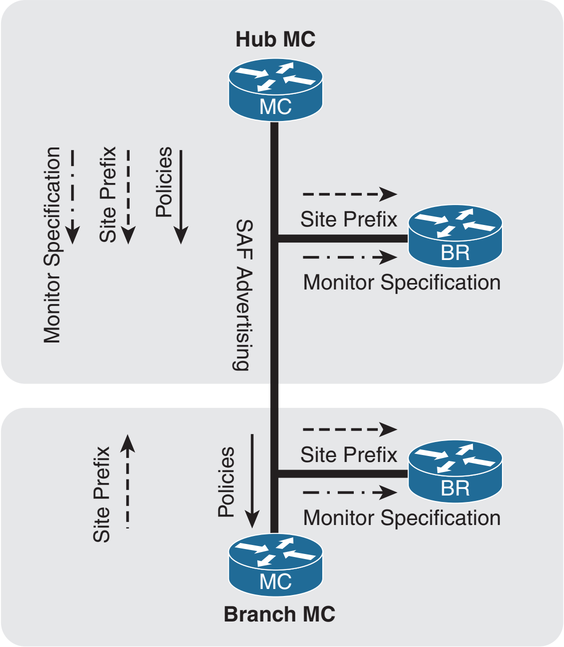


Figure 7-7 *IWAN SAF Peering Service*

SAF is automatically configured when PfR is enabled on a site. SAF dynamically discovers and establishes a peering as defined previously. The Hub MC advertises all policies and monitoring configuration to all the sites. Every site is responsible for advertising its own site prefix information to other sites in the domain.

Each instance must use an interface with an IP address that is reachable (routed) through the network to join in the IWAN peering system. PfRv3 requires that this address be a loopback address. It is critical that all these loopback addresses be reachable across the IWAN domain.

Parent Route Lookups

PfR uses the concept of a *parent route lookup* which refers to locating all the paths that a packet can take to a specific network destination regardless of the best-path calculation. The parent route lookup is performed so that PfR can monitor all paths and thereby prevent network traffic from being blackholed because the BRs have only summary routes in their routing table. PfR has direct API accessibility into EIGRP and BGP and can identify all the paths available for a prefix regardless of whether alternative paths were installed into the RIB.

PfR requires a parent route for every WAN path (primary, secondary, and so on) for PfR to work effectively. PfR searches the following locations in the order listed to locate all the paths for a destination:

1. NHRP cache (when spoke-to-spoke direct tunnels are established)
2. BGP table (where applicable)
3. EIGRP topology table (where applicable)
4. Static routes (where applicable)
5. RIB. Only one path is selected by default. In order for multiple paths to be selected, the same routing protocol must find both paths to be equal. This is known as equal-cost multipathing (ECMP).

Note If a protocol other than EIGRP or BGP is used, all the paths have to be ECMP in the RIB. Without ECMP in the RIB, PfR cannot identify alternative paths, and that hinders PfR's effectiveness.

The following logic is used for parent route lookups:

- The parent route lookup is done during channel creation (see the following section, "Intelligent Path Control Principles," for more information).
- For PfR Internet-bound traffic, the parent route lookup is done every time traffic is controlled.

In a typical IWAN design, BGP or EIGRP is configured to make sure MPLS is the preferred path and the Internet the backup path. Therefore, for any destination prefix, MPLS is the only available path in the RIB. But PfR looks into the BGP or EIGRP table and knows if the Internet is also a possible path and can use it for traffic forwarding in a loop-free manner.

Intelligent Path Control Principles

PfR is able to provide intelligent path control and visibility into applications by integrating with the Cisco Performance Monitoring Agent available on the WAN edge (BR) routers. Performance metrics are passively collected based on user traffic and include bandwidth, one-way delay, jitter, and loss.

PfR Policies

PfR policies are global to the IWAN domain and are configured on the Hub MC, then distributed to all MCs via the IWAN peering system. Policies can be defined per DSCP or per application name.

Branch and Transit MCs also receive the Cisco Performance Monitor instance definition, and they can instruct the local BRs to configure Performance Monitors over the WAN interfaces with the appropriate thresholds.

PfR policies are divided into three main groups:

- **Administrative policies:** These policies define path preference definition, path of last resort, and zero SLA used to minimize control traffic on a metered interface.
- **Performance policies:** These policies define thresholds for delay, loss, and jitter for user-defined DSCP values or application names.
- **Load-balancing policy:** Load balancing can be enabled or disabled globally, or it can be enabled for specific network tunnels. In addition, load balancing can provide specific path preference (for example, the primary path can be INET01 and INET02 with a fallback of MPLS01 and MPLS02).

Site Discovery

PfRv3 was designed to simplify the configuration and deployment of branch sites. The configuration is kept to a minimum and includes the IP address of the Hub MC. All MCs connect to the Hub MC in a hub-and-spoke topology.

When a Branch or Transit MC starts:

- It uses the loopback address of the local MC as its site ID.
- It registers with the Hub MC, providing its site ID, then starts building the IWAN peering with the Hub MC to get all information needed to perform path control. That includes policies and Performance Monitor definitions.
- The Hub MC advertises the site ID information for all sites to all its Branch or Transit MC clients.

At the end of this phase, all MCs have a site prefix database that contains the site ID for every site in the IWAN domain.

Note The site ID is based on the local MC loopback address and is a critical piece of PfR. Routing for MC addresses must be carefully designed to ensure that this address is correctly advertised across all available paths.

Figure 7-8 shows the IWAN peering between all MCs and the Hub MC. R10 is the Hub MC for this topology. R20, R31, R41, and R51 peer with R10. This is the initial phase for site discovery.

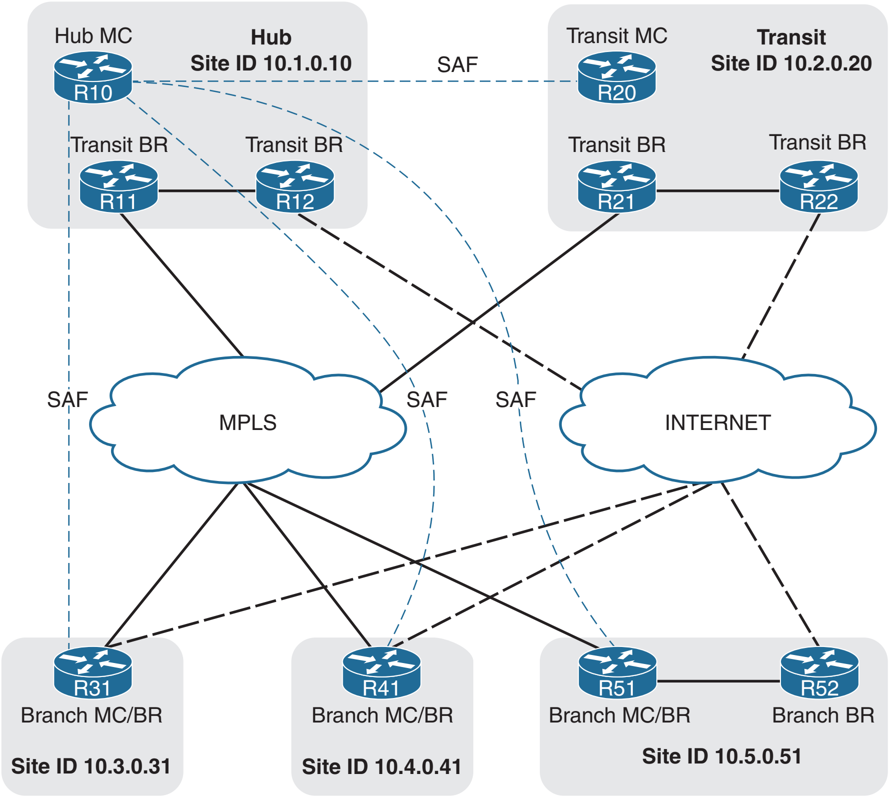


Figure 7-8 *Demonstration of IWAN Peering to the Domain Controller*

Site Prefix Database

PfR maintains a topology that contains all the network prefixes and their associated site IDs. A site prefix is the combination of a network and the site ID for the network prefix attached to that router. The PfR topology table is known as the *site prefix database* and is a vital component of PfR. The site prefix database resides on the MCs and BRs. The site prefix database located on the MC learns and manages the site prefixes and their origins from both local egress flow and advertisements from remote MC peers. The site prefix database located at a BR learns/manages the site prefixes and their origins only from the advertisements from remote peers. The site prefix database is organized as a longest prefix matching *tree* for efficient search.

Table 7-2 provides the site prefix database on all MCs and BRs for the IWAN domain shown in Figure 7-8. It provides a mapping between a destination prefix and a destination site.

Table 7-2 *Site Prefix Database for an IWAN Domain*

Site Name	Site Identifier	Site Prefix
Site 1	10.1.0.10	10.1.0.0/16
Site 1	10.1.0.10	172.16.1.0/24
Site 2	10.2.0.20	10.2.0.0/16
Site 3	10.2.0.31	10.3.3.0/24
Site 4	10.4.0.41	10.4.4.0/24
Site 5	10.5.0.51	10.5.5.0/24

Note The site prefix database can contain multiple network prefixes per site and is not limited to just one. A second entry was added to the table for Site 1 to display the concept.

In order to learn from advertisements via the peering infrastructure from remote peers, every MC and BR subscribes to the site prefix subservice of the PfR peering service. MCs publish and receive site prefixes. BRs only receive site prefixes. An MC publishes the list of site prefixes learned from local egress flows by encoding the site prefixes and their origins into a message. This message can be received by all the other MCs and BRs that subscribe to the peering service. The message is then decoded and added to the site prefix databases at those MCs and BRs. Site prefixes will be explained in more detail in Chapter 8, "PfR Provisioning."

Note Site prefixes are dynamically learned at branch sites but must be statically defined at hub and transit sites. The branch site prefixes can be statically defined too.

PfR Enterprise Prefixes

The enterprise-prefix prefix list defines the boundary for all the internal enterprise prefixes. A prefix that is not from the enterprise-prefix prefix list is considered a PfR Internet prefix. PfR does not monitor performance (delay, jitter, byte loss, or packet loss) for network traffic.

In Figure 7-9, all the network prefixes for remote sites (Sites 3, 4, and 5) have been dynamically learned. The central sites (Site 1 and Site 2) have been statically configured. The enterprise-prefix prefix list has been configured to include all the network prefixes in each of the sites so that PfR can monitor performance.

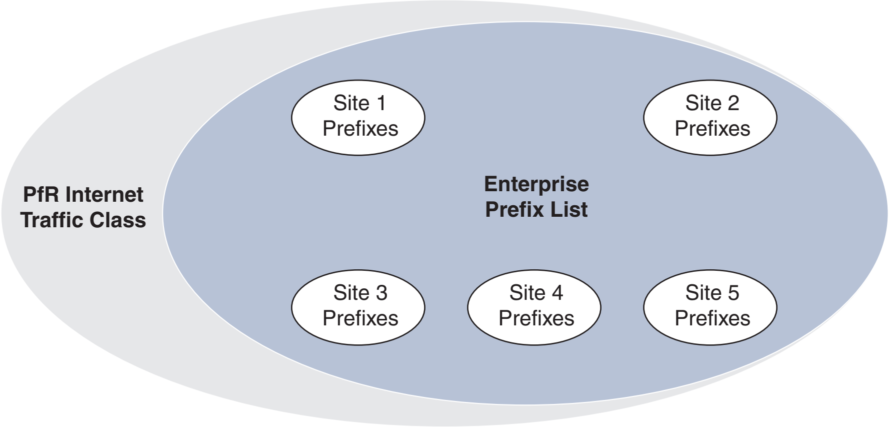


Figure 7-9 *PfR Site and Enterprise Prefixes*

Note In centralized Internet access models, in order for PfR to monitor performance to Internet-based services (email hosting and so forth), the hosting network prefix must be assigned to the enterprise-prefix prefix list. In addition, the hosting network is added to all the site prefix lists for sites that provide Internet connectivity.

WAN Interface Discovery

Border router WAN interfaces are connected to different SPs and have to be defined or discovered by PfR. This definition creates the relationship between the SPs and the administrative policies based on the path name in PfR. A typical example is to define an MPLS-VPN path as the preferred one for all business applications and the Internet-based path as a fallback path when there is a performance issue on the primary.

Hub and Transit Sites

In a PfR domain, a *path name* and a *path identifier* need to be configured for every WAN interface (DMVPN tunnel) on the hub site and all transit sites:

- The *path name* uniquely identifies a transport network. For example, this book uses a primary transport network called MPLS for the MPLS-based transport and a secondary transport network called INET for the Internet-based transport.
- The *path identifier* uniquely identifies a path on a site. This book uses path-id 1 for DMVPN tunnel 100 connected to MPLS and path-id 2 for tunnel 200 connected to INET.

IWAN supports multiple BRs for the same DMVPN network on the hub and transit sites only. The path identifier has been introduced in PfR to be able to track every BR individually.

Every BR on a hub or transit site periodically sends a *discovery packet* with path information to every discovered site. The discovery packets are created with the following default parameters:

- **Source IP address:** Local MC IP address
- **Destination IP address:** Remote site ID (remote MC IP address)
- **Source port:** 18000
- **Destination port:** 19000

Branch Sites

WAN interfaces are automatically discovered on Branch BRs. There is no need to configure the transport names over the WAN interfaces.

When a BR on a branch site receives a *discovery probe* from a central site (hub or transit site):

- It extracts the path name and path identifier information from the probe payload.
- It stores the mapping between the WAN interface and the path name.
- It sends the interface name, path name, and path identifier information to the local MC.
- The local MC knows that a new WAN interface is available and also knows that a BR is available on that path with the path identifier.

The BR associates the tunnel with the correct path information, enables the Performance Monitors, collects performance metrics, collects site prefix information, and identifies traffic that can be controlled.

This discovery process simplifies the deployment of PfR.

Channel

Channels are logical entities used to measure path performance per DSCP between two sites. A channel is created based on real traffic observed on BRs and is based upon a unique combination of factors such as interface, site, next hop, and path. Channels are based on real user traffic or synthetic traffic generated by the BRs called smart probes. A channel is added every time a new DSCP, interface, or site is added to the prefix database or when a new smart probe is received. A channel is a logical construct in PfR and is used to keep track of next-hop reachability and collect the performance metrics per DSCP.

Note In the IWAN 2.1 architecture, multiple next-hop capability was added so that PfR could monitor a path taken through a transit site. A channel is actually created per next hop. In topologies that include a transit site, a channel is created for every next hop to the destination prefix to monitor performance.

Figure 7-10 illustrates the channel creation over the MPLS path for DSCP EF. Every channel is used to track the next-hop availability and collect the performance metrics for the associated DSCP and destination site.

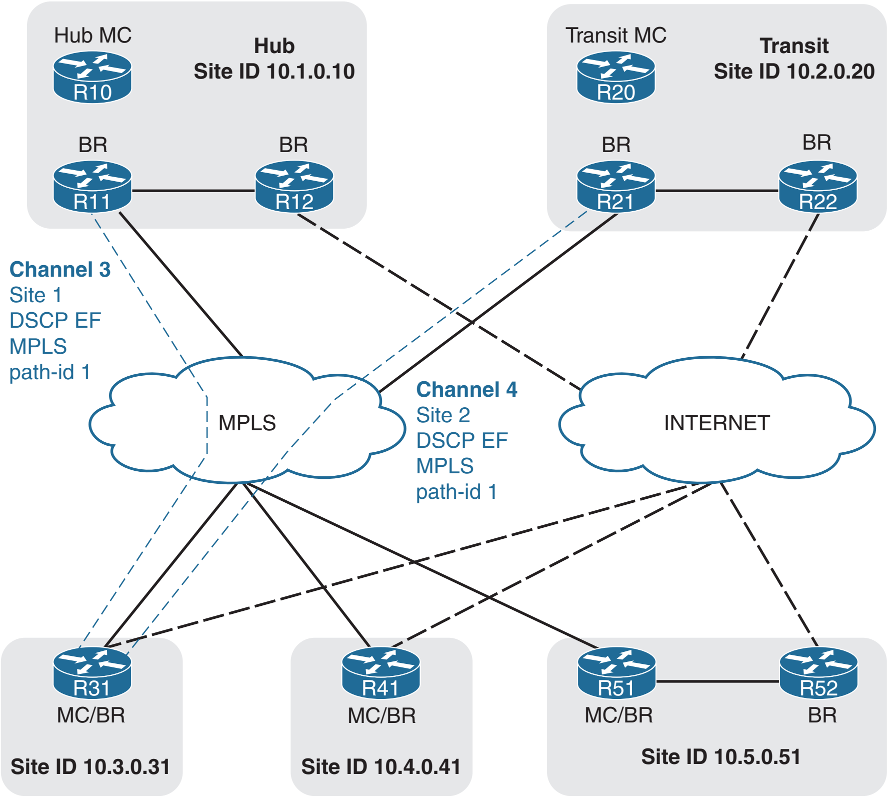


Figure 7-10 *Channel Creation for Monitoring Performance Metrics*

When a channel needs to be created on a path, PfR creates corresponding channels for any alternative paths to the same destination. This allows PfR to keep track of the performance for the destination prefix and DSCP for every DMVPN network. Channels are deemed active or standby based on the routing decisions and PfR policies.

Multiple BRs can sit on a hub or transit site connected to the same DMVPN network. DMVPN hub routers function as NHRP NHSs for DMVPN and are the BRs for PfR. PfR supports multiple next-hop addresses for hub and transit sites only but limits each of the BRs to hosting only one DMVPN tunnel. This limitation is overcome by placing multiple BRs into a hub or transit site.

The combination of multiple next hops and transit sites creates a high level of availability. A destination prefix can be available across multiple central sites and multiple BRs. For example, if a next hop connected on the preferred path DMVPN tunnel 100 (MPLS) experiences delays, PfR is able to fail over to the other next hop available for DMVPN tunnel 100 that is connected to a different router. This avoids failing over to a less preferred path using DMVPN tunnel 200, which uses the Internet as a transport.

Figure 7-11 illustrates a branch with DSCP EF packets flowing to a hub or transit site that has two BRs connected to the MPLS DMVPN tunnel. Each path has the same path name (MPLS) and a unique path identifier (path-id 1 and path-id 2). If BR1 experiences performance issues, PfR fails over the affected traffic to BR2 over the same preferred path MPLS.

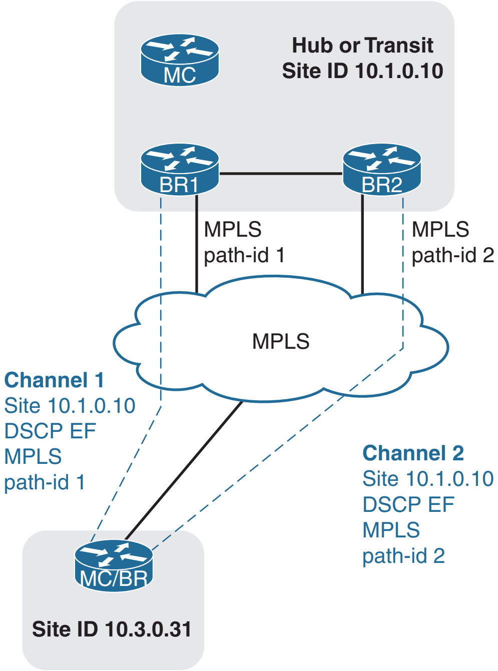


Figure 7-11 *Channels per Next Hop*

A parent route lookup is done during channel creation. PfR first checks to see if there is an NHRP shortcut route available; if not, it then checks for parent routes in the order of BGP, EIGRP, static, and RIB. If at any point an NHRP shortcut route appears, PfR selects that and relinquishes using the parent route from one of the routing protocols. This behavior allows PfR to dynamically measure and utilize DMVPN shortcut paths to protect site-to-site traffic according to the defined polices as well.

A channel is deemed *reachable* if the following happens:

- Traffic is received from the remote site.
- An unreachable event is not received for two monitor intervals.

A channel is declared *unreachable* in both directions in the following circumstances:

- No packets are received since the last unreachable time from the peer, as detected by the BR. This unreachable timer is defined as one second by default and can be tuned if needed.
- The MC receives an unreachable event from a remote BR. The MC notifies the local BR to make the channel unreachable.

When a channel becomes unreachable, it is processed through the threshold crossing alert (TCA) messages, which will be described later in the chapter.

Smart Probes

Smart probes are synthetic packets that are generated from a BR and are primarily used for WAN interface discovery, delay calculation, and performance metric collection for standby channels. This synthetic traffic is generated only when real traffic is not present, except for periodic packets for one-way-delay measurement. The probes (RTP packets) are sent over the channels to the sites that have been discovered.

Controlled traffic is sent at periodic intervals:

- **Periodic probes:** Periodic packets are sent to compute one-way delay. These probes are sent at regular intervals whether actual traffic is present or not. By default this is one-third of the monitoring interval (the default is 30 seconds), so by default periodic probes are sent every 10 seconds.
- **On-demand probes:** These packets are sent only when there is no traffic on a channel. Twenty packets per second are generated per channel. As soon as user traffic is detected on a channel, the BR stops sending on-demand probes.

Traffic Class

PfR manages aggregations of flows called *traffic classes (TCs)*. A traffic class is an aggregation of flows going to the same destination prefix, with the same DSCP or application name (if application-based policies are used).

Traffic classes are learned on the BR by monitoring a WAN interface's egress traffic. This is based on a Performance Monitor instance applied on the external interface.

Traffic classes are divided into two groups:

- **Performance TCs:** These are any TCs with performance metrics defined (delay, loss, jitter).
- **Non-performance TCs:** The default group, these are the TCs that do not have any defined performance metrics (delay, loss, jitter), that is, TCs that do have any match statements in the policy definition on the Hub MC.

For every TC, the PfR route control maintains a list of active channels (current exits) and standby channels.

Note Real-time load balancing affects only non-performance TCs. PfR moves default TCs between paths to keep bandwidth utilization within the boundaries of a predefined ratio. For performance TCs, new TCs use the least loaded path. After a traffic class is established, it stays on the path defined, unless that path becomes out of policy.

Path Selection

Path and next-hop selection in PfR depends on the routing design in conjunction with the PfR policies. From a central site (hub and transit) to a branch site, there is only one possible next hop per path. From a branch site to a central site, multiple next hops can be available and may span multiple sites. PfR has to make a choice among all next hops available to reach the destination prefix of the traffic to control.

Direction from Central Sites (Hub and Transit) to Spokes

Each central site is a distinct site by itself and controls only traffic toward the spoke on the WAN paths to that site. PfR does not redirect traffic between central sites across the DCI or WAN core to reach a remote site. If the WAN design requires that all the links be considered from POP to spoke, use a single MC to control all BRs from both central sites.

Direction from Spoke to Central Sites (Hub and Transit)

The path selection from BR to a central site router can vary based on the overall network design. The following sections provide more information on PfR's path selection process.

Active/Standby Next Hop

The spoke considers all the paths (multiple next hops) toward the central sites and maintains a list of active/standby candidate next hops per prefix and interface.
The concept of *active* and *standby* next hops is based on the routing best metric to gather information about the preferred POP for a given prefix. If the best metric for a given prefix is on a specific central site, all the next hops on that site for all the paths are

tagged as *active* (only for that prefix). A next hop in a given list is considered to have a best metric based on the following metrics/criteria:

- Advertised mask length
- BGP weight and local preference
- EIGRP feasible distance (FD) and successor FD

Transit Site Affinity

Transit Site Affinity (also called POP Preference) is used in the context of a multiple-transit-site deployment with the same set of prefixes advertised from multiple central sites. Some branches prefer a specific transit site over the other sites. The affinity of a branch to a transit site is configured by altering the routing metrics for prefix advertisements to the branch from the transit site. If one of the central sites advertising a specific prefix has the best next hop, the entire site is preferred over the other sites for all TCs to this destination prefix. Transit site preference is a higher-priority filter and takes precedence over path preference. The Transit Site Affinity feature was introduced in Cisco IWAN 2.1.

Path Preference

During Policy Decision Point (PDP), the exits are first sorted on the available bandwidth, Transit Site Affinity, and then a third sort algorithm that places all primary path preferences in the front of the list followed by fallback preferences. A common deployment use case is to define a primary path (MPLS) and a fallback path (INET). During PDP, MPLS is selected as the primary channel, and if INET is within policy it is selected as the fallback.

- With path preference configured, PfR first considers all the links belonging to the preferred path (that is, it includes the active and the standby links belonging to thepreferred path) and then uses the fallback provider links.
- Without path preference configured, PfR gives preference to the active channels and then the standby channels (active/standby is per prefix) with respect to the performance and policy decisions.

Note Active/standby tagging happens whether Transit Site Affinity is enabled or disabled. The active and standby channels (per prefix) may span central sites if they advertise the same prefix. Spoke routers use a hash to choose the active channel.

Transit Site Affinity and Path Preference Usage

Transit Site Affinity and path preference are used in combination to influence the next-hop selection per TC. For example, this book uses a topology with two central sites (Site 1 and Site 2) and two paths (MPLS and INET). Both central sites advertise the same prefix (10.10.0.0/16 as an example), and Site 1 has the best next hop for that prefix

(R11 advertises 10.10.0.0/16 with the highest BGP local preference). Enabling Transit Site Affinity and defining a path preference with MPLS as the primary and INET as the fallback path, the BR identifies the following routers (in order) for the next hop:

1. R11 is the primary next hop for TCs with 10.10.0.0/16 as the destination prefix
2. Then R12 (same site, because of Transit Site Affinity)
3. Then R21 (Site 2, because of path preference)
4. Then R22

Performance Monitoring

The PfR monitoring system interacts with the IOS component called *Performance Monitor* to achieve the following tasks:

- Learning site prefixes and applications
- Collecting and analyzing performance metrics per DSCP
- Generating threshold crossing alerts
- Generating out-of-policy report

Performance Monitor is a common infrastructure within Cisco IOS that passively collects performance metrics, number of packets, number of bytes, statistics, and more within the router. In addition, Performance Monitor organizes the metrics, formats them, and makes the information accessible and presentable based upon user needs. Performance Monitor provides a central repository for other components to access these metrics.

PfR is a client of Performance Monitor, and through the performance monitoring metrics, PfR builds a database from that information and uses it to make an appropriate path decision. When a BR component is enabled on a device, PfR configures and activates three *Performance Monitor instances (PMIs)* over all discovered WAN interfaces of branch sites, or over all configured WAN interfaces of hub or transit sites. Enablement of PMI on these interfaces is dynamic and completely automated by PfR. This configuration does not appear in the startup or running configuration file.

The PMIs are

- **Monitor 1:** Site prefix learning (egress direction)
- **Monitor 2:** Egress aggregate bandwidth per traffic class
- **Monitor 3:** Performance measurements (ingress direction)

Monitor 3 contains two monitors: one dedicated to the business and media applications where failover time is critical (called *quick monitor*), and one allocated to the default traffic.

PfR policies are applied either to an application definition or to DSCP. Performance is measured only per DSCP because SPs can differentiate traffic only based on DSCP and not based on application.

Performance is measured between two sites where there is user traffic. This could be between hub and a spoke, or between two spokes; the mechanism remains the same.

- The egress aggregate monitor instance captures the number of bytes and packets per TC on egress on the source site. This provides the bandwidth utilization per TC.
- The ingress per DSCP monitor instance collects the performance metrics per DSCP (channel) on ingress on the destination site. Policies are applied to either application or DSCP. However, performance is measured per DSCP because SPs differentiate traffic only based on DSCP and not based on discovered application definitions. All TCs that have the same DSCP value get the same QoS treatment from the provider, and therefore there is no real need to collect performance metrics per application-based TC.

PfR passively collects metrics based on real user traffic and collects metrics on alternative paths too. The source MC then instructs the BR connected to the secondary paths to generate smart probes to the destination site. The PMI on the remote site collects statistics in the same way it would for actual user traffic. Thus, the health of a secondary path is known prior to being used for application traffic, and PfR can choose the best of the available paths on an application or DSCP basis.

Figure 7-12 illustrates PfR performance measurement with network traffic flowing from left to right. On the ingress BRs (BRs on the right), PfR monitors performance per channel. On the egress BRs (BRs on the left), PfR collects the bandwidth per TC. Metrics are collected from the user traffic on the active path and based on smart probes on the standby paths.

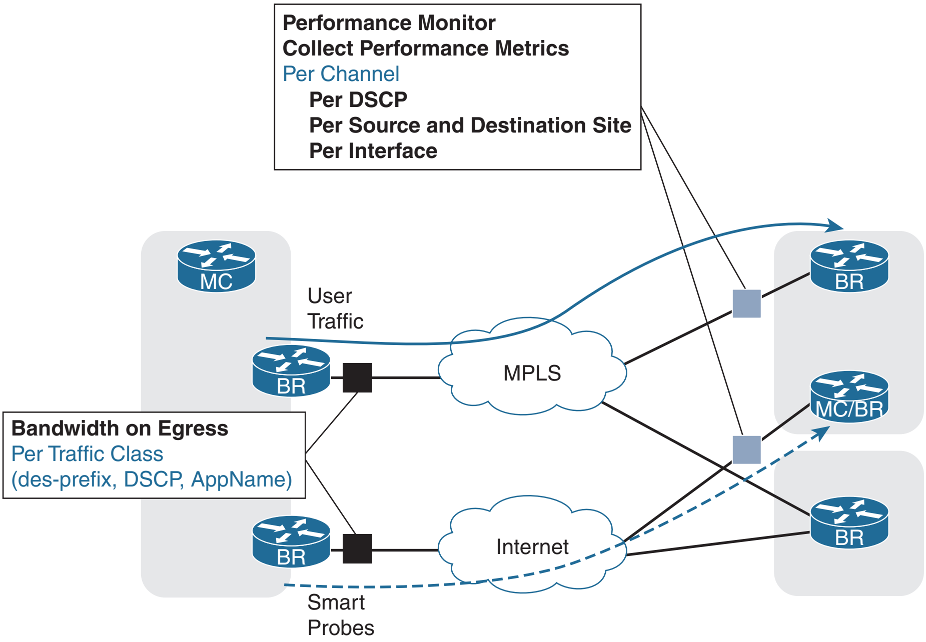


Figure 7-12 *PfR Performance Measurement via Performance Monitor*

Note Smart probes are not IP SLA probes. Smart probes are directly forged in the data plane from the source BR and discarded on the destination BR after performance metrics are collected.

Threshold Crossing Alert (TCA)

Threshold crossing alert (TCA) notifications are alerts for when network traffic exceeds a set threshold for a specific PfR policy. TCAs are generated from the PMI attached to the BR's ingress WAN interfaces and smart probes. Figure 7-13 displays a TCA being raised on the destination BR.

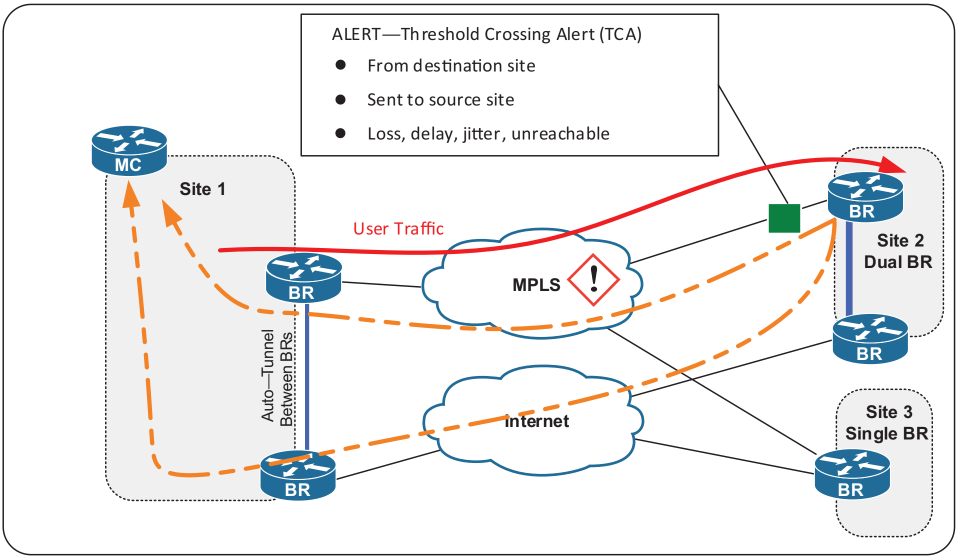


Figure 7-13 *Threshold Crossing Alert (TCA)*

Threshold crossing alerts are managed on both the destination BR and source MC for the following scenarios:

- The destination BR receives performance TCA notifications from the PMI, which monitors the ingress traffic statistics and reports TCA alerts when threshold crossing events occur.
- The BR forwards the performance TCA notifications to the MC on the source site that actually generates the traffic. This source MC is selected from the site prefix database based on the source prefix of the traffic. TCA notifications are transmitted via multiple paths for reliable delivery.

- The source MC receives the TCA notifications from the destination BR and translates the TCA notifications (that contain performance statistics) to an out-of-policy (OOP) event for the corresponding channel.
- The source MC waits for the TCA processing delay time for all the notifications to arrive, then starts processing the TCA. The processing involves selecting TCs that are affected by the TCA and moving them to an alternative path.

Path Enforcement

PfR uses the Route Control Enforcement module for optimal traffic redirection and path enforcement. This module performs lookups and reroutes traffic similarly to policy-based routing but without using an ACL. The MC makes path decisions for every unique TC. The MC picks the next hop for a TC's path and instructs the local BR how to forward packets within that TC.

Because of how path enforcement is implemented, the next hop has to be directly connected to each BR. When there are multiple BRs on a site, PfR sets up an mGRE tunnel between all of them to accommodate path enforcement. Every time a WAN exit point is discovered or an *up/down* interface notification is sent to the MC, the MC sends this notification to all other BRs in the site. An endpoint is added to the mGRE tunnel pointing toward this BR as a result.

When packets are received on the LAN side of a BR, the route control functionality determines if it must exit via a local WAN interface or via another BR. If the next hop is via another BR, the packet is sent out on the tunnel toward that BR. Thus the packet arrives at the destination BR within the same site. Route control gets the packet, looks at the channel identifier, and selects the outgoing interface. The packet is then sent out of this interface across the WAN.

Summary

This chapter provided a thorough overview of Cisco intelligent path control, which is a core pillar of the Cisco IWAN architecture and is based upon Performance Routing (PfR). The following chapters will expand upon these theories while explaining the configuration of PfR.

PfR provides the following benefits for a WAN architecture:

- Maximizes WAN bandwidth utilization
- Protects applications from performance degradation
- Uses passive monitoring to track application performance across the WAN
- Enables the Internet as a viable WAN transport
- Provides multisite coordination to simplify network-wide provisioning

- Provides an application-based policy-driven framework that is tightly integrated with existing Performance Monitor components
- Provides a smart and scalable multisite solution to enforce application SLAs while optimizing network resource utilization

PfRv3 is the third-generation multisite-aware bandwidth and path control/optimization solution for WAN- and cloud-based applications and is available now on Cisco *Integrated Services Router (ISR)* Generation 2 series, ISR-4000 Series, and CSR 1000V and ASR 1000 Series routers.

Further Reading

Cisco. “Performance Routing Version 3.” www.cisco.com.

Cisco. “PfRv3 Transit Site Support.” www.cisco.com.

This page intentionally left blank

Chapter 8

PfR Provisioning

This chapter covers the following topics:

- Dual-data-center enterprise topology
- Overlay routing
- Hub site configuration
- Transit site configuration
- Single CPE branch configuration
- Dual CPE branch configuration
- Path selection and the use of transit site and path preference

Configuring PfR for the IWAN domain includes the following steps:

- **Configuring the hub site:** This is the central site that contains the Hub MC, which is responsible for distributing the PfR domain policies to all other sites. Configuration tasks at the hub site include the definition of the Hub MC and multiple BRs. The BRs are DMVPN hub routers. Every BR terminates only one overlay network. There is only one hub site in an IWAN domain.
- **Configuring one or more transit sites:** This is another type of central site that houses DMVPN hub routers. Network traffic from branch sites may terminate at servers located in this central site itself or at other locations. Transit sites can provide branch-to-branch communications, where branches do not connect to the same WAN transport. Configuration tasks at the transit site include the definition of one MC and one or more Transit BRs per transit site. Every BR terminates only one overlay network. An IWAN domain can have multiple transit sites.

- **Configuring branch sites:** Branch sites are remote sites that house DMVPN spoke routers. Configuration tasks include the definition of the MC and one or more BRs. A BR on a branch can support multiple DMVPN tunnels.

IWAN Domain

The IWAN domain includes multiple central sites and several branch sites. At each site, the MC is the local decision maker and controls the BRs responsible for performance measurement and path enforcement.

One of the central sites is defined as the hub site. The MC defined in this hub site is the Hub MC and acts as

- **A local MC for the site:** It makes decisions and instructs the local BRs how to forward TCs.
- **A global domain controller for the IWAN domain:** It is responsible for defining and distributing the PfR policies for that IWAN domain.

Branch sites typically include a single *customer premises equipment (CPE)* router or two CPEs. Each site is connected to multiple paths that have various SLAs. Every branch site has its own local MC and one or multiple BRs.

Topology

Figure 8-1 displays the dual-hub and dual-cloud topology used in this book to explain the configuration of PfR. The transport-independent design is based on DMVPN with two providers, one being considered the primary (MPLS) and one the secondary (Internet). Branch sites are connected to both DMVPN clouds, and both tunnels are up.

DMVPN tunnel 100 uses 192.168.100.0/24 and is associated with the MPLS transport, and DMVPN tunnel 200 uses 192.168.200.0/24 for the Internet transport. PfR does not have knowledge about the transport (underlay) network and only controls traffic destined for sites that are connected to the DMVPN network.

The dual-hub and dual-cloud topology provides active-active WAN paths that enable connectivity to each DMVPN hub for a transport, and it provides redundancy in the event of a hub failure. In the topology, a DMVPN hub router connects to only one transport on the Transit BRs (Site 1 and Site 2). R11 and R21 are the MPLS DMVPN hub routers for tunnel 100 (192.168.100.0/24). R12 and R22 are the Internet DMVPN hub routers for tunnel 200 (192.168.200.0/24).

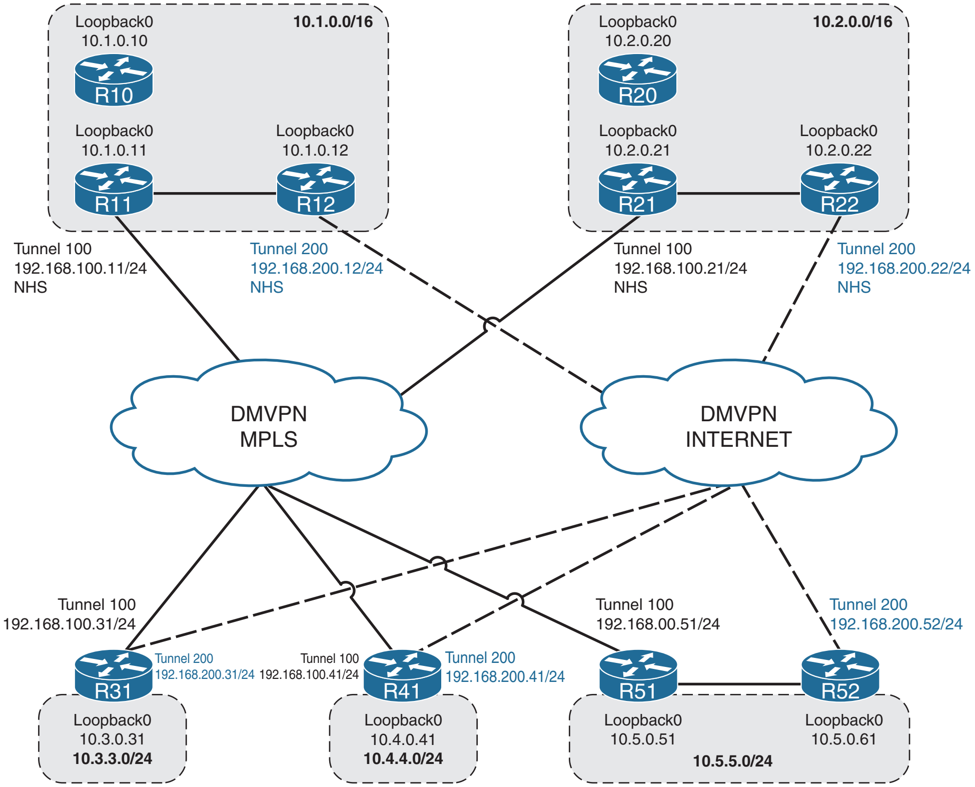


Figure 8-1 *Overlay Network*

Note At the time of this writing, it is mandatory to use only one transport per DMVPN hub to guarantee that spoke-to-spoke tunnels will be established. Normal traffic flow follows the Cisco Express Forwarding path, so in a dual-tunnel hub scenario it is possible for one spoke (R31) to send a packet to a different spoke (R41) which will traverse both tunnel interfaces instead of a single interface, preventing a spoke-to-spoke tunnel from forming. This topology demonstrates one transport per DMVPN hub.

Site 3 and Site 4 do not have redundant routers, so R31 and R41 connect to both transports via DMVPN tunnels 100 and 200. However, at Site 5, redundant routers have been deployed. R51 connects to the MPLS transport with DMVPN tunnel 100, and R52 connects to the Internet transport with DMVPN tunnel 200. R51 and R52 establish connectivity with each other via a cross-link.

> **Note** At a branch site, the BRs must be directly connected to the Branch MC. This can be a direct link, a transit VLAN through a switch, or a GRE tunnel.

Table 8-1 provides the network site, site subnet, Loopback0 IP address, and transport connectivity for the routers.

Table 8-1 *Topology Table*

Router	Site Location	Primary Subnet	Loopback0 IP Address	MPLS Transport	Internet Transport
R10	Site 1, DC 1	10.1.0.0/16	10.1.0.10	–	–
R11	Site 1, DC 1	10.1.0.0/16	10.1.0.11	✓	–
R12	Site 1, DC 1	10.1.0.0/16	10.1.0.12	–	✓
R20	Site 2, DC 2	10.2.0.0/16	10.2.0.20	–	–
R21	Site 2, DC 2	10.2.0.0/16	10.2.0.21	✓	–
R22	Site 2, DC 2	10.2.0.0/16	10.2.0.22	–	✓
R31	Site 3, branch	10.3.0.0/16	10.3.0.31	✓	✓
R41	Site 4, branch	10.4.0.0/16	10.4.0.41	✓	✓
R51	Site 5, branch	10.5.0.0/16	10.5.0.51	✓	–
R61	Site 5, branch	10.5.0.0/16	10.5.0.52	–	✓

As part of the dual-hub and dual-cloud topology, every DMVPN spoke router has two NHRP mappings for every DMVPN tunnel interface. The NHRP mapping correlates to the DMVPN hub router for the transport intended for that DMVPN tunnel. The NHRP mappings are as follows:

- Tunnel 100 (MPLS) uses R11 and R21.
- Tunnel 200 (Internet) uses R12 and R22.

Example 8-1 demonstrates R31's NHRP mappings for DMVPN tunnel 100.

Example 8-1 *R31's Tunnel 100 NHRP Mapping Configuration*

```
interface Tunnel100
 description DMVPN-MPLS
 ip nhrp nhs 192.168.100.11 nbma 172.16.11.1 multicast
 ip nhrp nhs 192.168.100.21 nbma 172.16.21.1 multicast
```

Note The topology uses the DMVPN configuration displayed in Examples 3-48 and 3-49 and includes the use of the front-door VRF explained in Chapter 3, "Dynamic Multipoint VPN."

Overlay Routing

As explained in Chapter 4, "Intelligent WAN (IWAN) Routing," transit routing at branch locations is restricted because it introduces suboptimal routing or unpredictable traffic patterns. NHRP redirect messages establish spoke-to-spoke tunnels when multiple paths exist for a branch site connecting to a network behind a different branch site.

Transit routing at the centralized sites (hub or transit sites) is acceptable and commonly occurs during the following two scenarios:

- Every centralized site advertises a unique set of network prefixes. Each site advertises its local networks with a network summary. In addition to the unique local network prefixes, broader network prefixes encompass the entire enterprise (all data centers, campuses, and remote locations) from all sites. The more specific prefix is used for optimal direct routing, but in failure scenarios, transit connectivity can be established through the alternative site's broader network advertisements as shown in Figure 8-2.
- Both sites advertise the same set of network prefixes. These sites (Site 1 and Site 2) provide transit connectivity to the real data centers connected over a WAN core that connects to both Site 1 and Site 2. This scenario is shown in Figure 8-3.

Advertising Site Local Subnets

In this scenario, each site advertises its local subnets and a network summary for all the routes in the WAN topology. Figure 8-2 illustrates the scenario where Site 1 and Site 2 host their own data centers with two different sets of prefixes. In this scenario, Site 1 advertises the 10.1.0.0/16 network, and Site 2 advertises the 10.2.0.0/16 network.

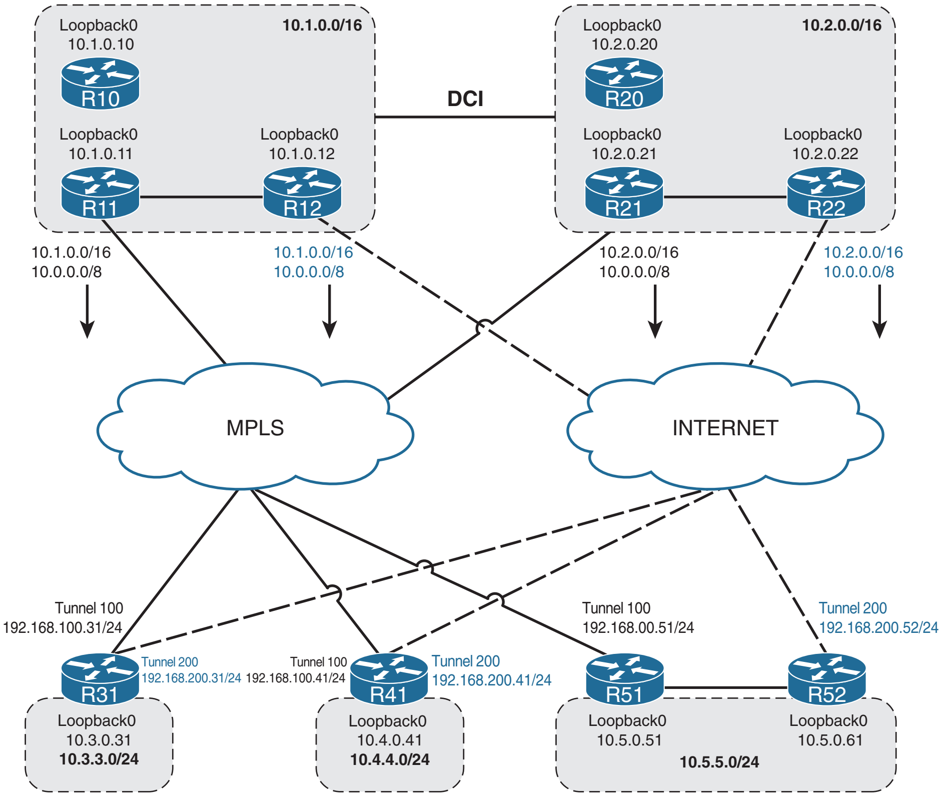


Figure 8-2 *Site 1 and Site 2 Advertising Two Differents Sets of Prefixes*

Table 8-2 displays the potential PfR next-hop addresses for a prefix for a specific tunnel interface. Notice that there is only one potential next-hop address for PfR to use.

Table 8-2 *Routing Table—Different Prefix*

Prefix	Interface	Possible Next Hop
10.1.0.0/16	Tunnel 100	R11
10.1.0.0/16	Tunnel 200	R12
10.2.0.0/16	Tunnel 100	R21
10.2.0.0/16	Tunnel 200	R22

Advertising the Same Subnets

In this scenario, each site advertises its local subnets and a network summary for all the routes in the WAN topology. These sites (Site 1 and Site 2) provide transit connectivity to the real data centers connected over a WAN core that connects to both Site 1 and Site 2.

Figure 8-3 displays a dual-data-center topology where Site 1 and Site 2 advertise the same set of prefixes from all the hub routers (R11, R12, R21, and R22).

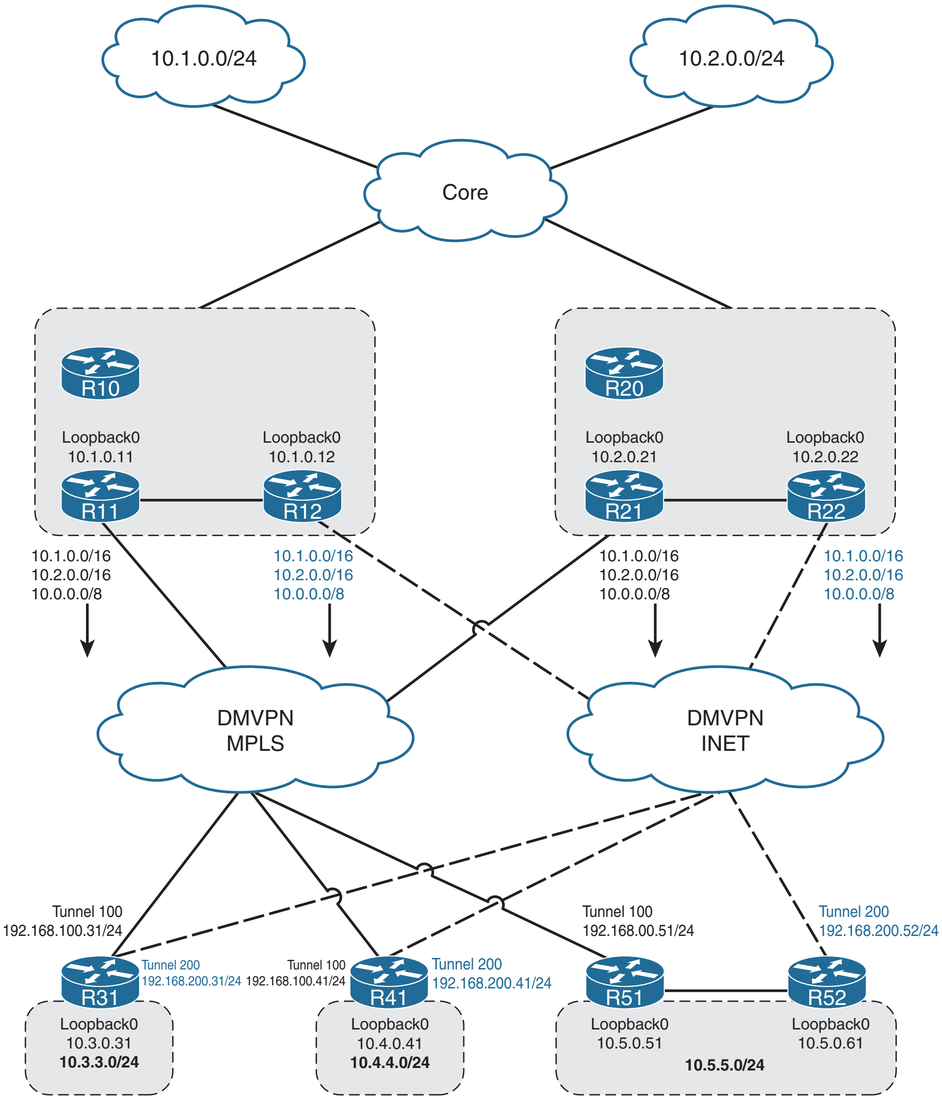


Figure 8-3 *Site 1 and Site 2 Advertising the Same Prefixes*

Table 8-3 displays the potential PfR next-hop addresses for a prefix for a specific tunnel interface. Notice that there are two potential next-hop addresses for PfR to use. The order is not significant.

Table 8-3 *Routing Table—Same Prefix*

Prefix	Interface	Possible Next Hop
10.1.0.0/16	Tunnel 100	R11 or R21
10.1.0.0/16	Tunnel 200	R12 or R22
10.2.0.0/16	Tunnel 100	R11 or R21
10.2.0.0/16	Tunnel 200	R12 or R22

In this scenario, each branch router has multiple next hops available for each tunnel interface. Multiple next hop per DMVPN interface and same prefix from multiple PfR sites are new features introduced with IOS XE 3.15 and IOS 15.5(2)T under the name *Transit Site* support.

Note Multiple next hop per DMVPN tunnel is supported only for traffic flowing from branch sites to central sites (hub or transit site).

Traffic Engineering for PfR

As shown in Chapter 5, it is a good practice to manipulate the routing protocols so that traffic flows across the preferred transport. Influencing the routing table ensures that when PfR is disabled, traffic will follow the Cisco Express Forwarding table derived from the RIB and forward traffic to the DMVPN over the preferred tunnel (transport).

The following logic applies for the transit routing scenarios provided earlier:

- Advertising the site local subnets (Figure 8-2)—R11 is preferred over R12 for 10.1.0.0/16 and R21 is preferred over R22 for prefix 10.2.0.0/16.
- Advertising the same subnets (Figure 8-3)—R11 and R21 are preferred over R12 and R22 for all prefixes.

PfRv3 always checks for a parent route of any destination prefix before creating a channel or controlling a TC. PfR selects next hops in the following order of lookup:

- Check to see if there is an NHRP shortcut route (branch only).
- If not, check in the order of BGP, EIGRP, static, and RIB.
- If at any point an NHRP shortcut route appears, PfRv3 picks that up and relinquishes using the parent route from one of the routing protocols.

Note It is essential to make sure that all destination prefixes are reachable over all available paths so that PfR can create the corresponding channels and control the TCs. Remember that PfR checks within the BGP or EIGRP topology table.

Specific NHRP shortcuts are installed at the spokes by NHRP as and when required. The validity of each NHRP shortcut is determined by the less specific, longest match IGP route present in the RIB. When a preferred path is defined in the routing configuration, it is key to disable the NHRP Route Watch feature on the secondary tunnel because the *covering prefix* is not available in the RIB for that tunnel.

The command **no nhrp route-watch** disables the NHRP Route Watch feature and allows the creation of a spoke-to-spoke tunnel over the secondary DMVPN network.

Example 8-2 illustrates how to disable NHRP Route Watch.

Example 8-2 *Branch Routers Internet Tunnel Configuration with NHRP Route Watch Disabled*

```
R31, R41, and R51
interface Tunnel 200
 no nhrp route-watch
```

PfR Components

Figure 8-4 illustrates the PfR components used in an IWAN domain. You need to configure an MC per site controlling one or more BRs. One of the MCs also hosts the domain controller component.

Note It is mandatory to use a dedicated Hub MC and dedicated Transit MCs.

Site 1 is defined as the hub site for the domain. R10 resides in Site 1 and is defined as the Hub MC (domain controller). PfR policies are defined on the Hub MC, and all other MCs in the domain peer to R10. In Site 1, R11 is a Transit BR for the DMVPN MPLS transport, and R12 is a Transit BR for the DMVPN Internet transport.

Site 2 is defined as a transit site with a POP ID of 1 (POP ID 1). R20 is defined as a Transit MC. R21 is a Transit BR for the DMVPN MPLS network, and R22 is a Transit BR for the DMVPN INET network.

Site 3, Site 4, and Site 5 are defined as branch sites. R31 and R41 are MC and BR for their sites. R51 is a Branch MC/BR (dual function) for Site 5, and R52 is a BR for Site 5.

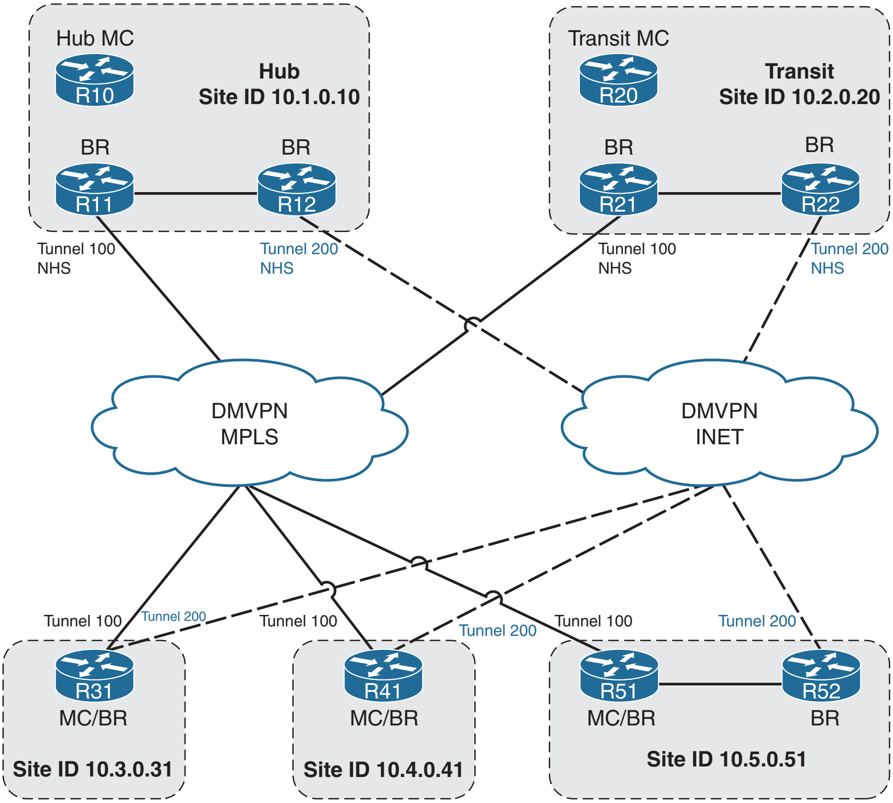


Figure 8-4 *PfR Components*

All sites are assigned a site ID based on the IP address of the loopback interface of the local MC as shown in Table 8-4.

Table 8-4 *PfR Component Table*

Site Name	Site ID	Router	PfR Component	MPLS Transport	Internet Transport
Site 1, DC 1	10.1.0.10	R10	Hub MC	–	–
		R11	Transit BR	✓	–
		R12	Transit BR	–	✓
Site 2, DC 2	10.2.0.10	R20	Transit MC	–	–
		R21	Transit BR	✓	–
		R22	Transit BR	–	✓
Site 3, branch	10.3.0.31	R31	Branch MC/BR	✓	✓

Site Name	Site ID	Router	PfR Component	MPLS Transport	Internet Transport
Site 4, branch	10.4.0.41	R41	Branch MC/BR	✓	✓
Site 5, branch	10.5.0.51	R51	Branch MC	✓	–
		R52	Branch BR	–	✓

Note This book uses the label INET for the Internet-based transport interfaces and related configuration so that it maintains the same length as MPLS.

PfR Configuration

All the PfR components need to communicate with each other in a hierarchical fashion. BRs communicate with their local MC, and the local MCs communicate with the Hub MC. Proper PfR design allocates a loopback interface for communication for the following reasons:

- Loopback interfaces are virtual and are always in an *up* state.
- They ensure that the address is consistent and easy to identify.

Typically, most networks create a dedicated loopback interface for management (SSH, TACACS, or SNMP) and for routing protocols. The same loopback interface can be used for PfR. The IP address assigned to the loopback interface should have a 255.255.255.255 (/32) subnet mask.

Master Controller Configuration

The MC serves as the control plane and decision maker for PfR. It collects performance measurements, bandwidth, and link utilization from the local BRs. Based upon the configured policy, the MC influences the forwarding behavior on BRs to ensure that network traffic uses the best interface based upon the defined policy.

This section focuses on the configuration of the MCs in an IWAN domain. An MC exists in all of the IWAN sites.

Hub Site MC Configuration

The Hub MC is located at the hub site in the IWAN topology. R10 is the Hub MC for the book’s sample topology. R10 controls two BRs in Site 1: R11 and R12.

It is also essential to remember that the Hub MC actually supports two very different roles:

- It is a *local MC* for the site and as such peers with the local BRs and controls them. In that role, it is similar to a Transit MC.
- It is a *global domain controller* with the PfR policies for the entire IWAN domain and as such peers with all MCs in the domain.

The Hub MC should run as a standalone platform, on a physical device, or as a virtual machine (CSR 1000V).

The process for configuring a Hub MC is given in the following steps:

Step 1. **Define the IWAN domain name.**

All PfR-related configuration is defined within the domain section. The command **domain** {**default** | *domain-name*} defines the PfR domain name. The domain name must be consistent for all devices participating in the same PfRv3 configuration.

Step 2. **Define the VRF.**

PfR is configured per VRF. The command **vrf** {**default** | *vrf-name*} defines the VRF for PfR. You can use the default VRF with the command **vrf** *default* or use any VRF defined on the router with the command **vrf** {*vrf-name*}.

Step 3. **Identify the router as the Hub MC.**

The command **master hub** enters MC configuration mode and configures the router as the Hub MC. When the Hub MC is configured, EIGRP SAF autoconfiguration is enabled by default, and requests from remote sites are sent to the Hub MC.

Step 4. **Define the source interface for PfR communication.**

A source interface is defined on the router for communication with other routers in the IWAN domain. The source interface is the loopback interface identified in Step 1. The source interface is configured with the command **source-interface** *interface-id*.

The source interface loopback also serves as a site ID for this particular site.

Step 5. **Define a password (optional).**

The command **password** *password* is used to secure the peering between the MC and BRs.

Step 6. **Define the Hub MC site prefix.**

The site-prefix prefix list defines static site prefixes for the local Hub MC site. The static site prefix list is required for Hub and Transit MCs. A site-prefix prefix list is optional on Branch MCs.

The site prefix is defined under the MC configuration with the command **site-prefixes prefix-list** *prefix-list-name*. Example 8-20 provides R10 and R20's site-prefix prefix list configuration.

Example 8-3 provides the Hub MC configuration that is deployed to R10.

Example 8-3 *R10 Hub MC Configuration*

```
R10 (Hub MC)
ip prefix-list SITE_PREFIX seq 10 permit 10.1.0.0/16
!
domain IWAN
 vrf default
  master hub
   site-prefixes prefix-list SITE_PREFIX
   source-interface Loopback0
   password cisco123
```

Note The site prefix must be defined on the Hub MC or the Hub MC status will be in a *down* state. Failure to define the site prefix might result in messages that look like "*%TCP-6-BADAUTH: No MD5 digest from 10.1.0.10(17749) to 10.1.0.11(52576) (RST) tableid—0*."

Transit Site MC Configuration

The Transit MC is located at all transit sites in an IWAN topology. It should run as a standalone platform, on a physical device, or as a virtual machine (CSR 1000V).

R20 is the Transit MC for the book's sample topology. R20 controls two BRs in Site 2: R21 and R22. A Transit MC needs to peer with the domain controller (Hub MC) to get the policies, monitor configurations, and global parameters.

The process for configuring a Transit MC is given in the following steps:

Step 1. **Define the IWAN domain name.**

All PfR-related configuration is defined within the domain section. The command **domain** {**default** | *domain-name*} defines the PfR domain name. The domain name must be consistent for all devices participating in the same PfRv3 configuration.

Step 2. **Define the VRF.**

PfR is configured per VRF. The command **vrf** {**default** | *vrf-name*} defines the VRF for PfR.

Step 3. **Identify the router as the Transit MC.**

The command **master transit** *pop-id* enters MC configuration mode and configures the MC instance as a transit. EIGRP SAF auto-configuration is enabled by default, and requests are sent to the Hub MC. The POP ID should be unique among other Transit MCs in the same IWAN domain.

Step 4. **Define the source interface for PfR communication.**

A source interface is defined on the router for communication with other routers in the IWAN domain. The source interface is the loopback interface defined at the beginning of this section. The source interface is configured with the command **source-interface** *interface-id*.

The source interface loopback also serves as a site ID for this particular transit site.

Step 5. **Configure the Hub MC IP address.**

This is the IP address of the source interface defined on the Hub MC. The book's example topology uses R10's loopback, 10.1.0.10. Every MC peers with the Hub MC to get the domain policies, monitor configuration, and global parameters.

Step 6. **Define a password (optional).**

The command **password** *password* is used to secure the peering between the MC and BRs.

Example 8-4 displays R20's transit site MC configuration. Notice that R20 assigns the POP ID of 1 for Site 2. Remember that the hub site is assigned a POP ID of 0 by default.

Example 8-4 *R20 Transit MC Configuration*

```
R20 (Transit MC)
domain IWAN
 vrf default
  master transit 1
   source-interface Loopback0
   password cisco123
   hub 10.1.0.10
```

Branch Site MC Configuration

A branch site is a site where no transit traffic is allowed. The PfR configuration is minimized for the Branch BR as the policy is defined at the local MC, which receives the routing policy from the Hub MC.

The MC at a branch site typically does not require a dedicated router to act as an MC. This implies that at least one branch site router contains the MC and the BR roles.

If a branch site has two routers for resiliency, the MC is typically connected to the primary transport circuit (MPLS in the book's examples) and is the HSRP master when the branch routers provide the Layer 2/Layer 3 delineation (all interfaces facing the LAN are Layer 2 only).

The process for configuring a Branch MC is given in the following steps:

Step 1. **Define the IWAN domain name.**

All PfR-related configuration is defined within the domain section. The command **domain** {**default** | *domain-name*} defines the PfR domain name. The domain name must be consistent for all devices participating in the same PfRv3 configuration.

Step 2. **Define the VRF.**

PfR is configured per VRF. The command **vrf** {**default** | *vrf-name*} defines the VRF for PfR. You can use the default VRF or use any VRF defined on the router.

Step 3. **Identify the router as the Branch MC.**

The command **master branch** enters MC configuration mode and configures the MC instance as a branch. EIGRP SAF auto-configuration is enabled by default, and requests are sent to the Hub MC.

Step 4. **Define the source interface for PfR communication.**

A source interface is defined on the router for communication with other routers in the IWAN domain. The source interface is the loopback interface defined at the beginning of this section. The source interface is configured with the command **source-interface** *interface-id*.

The source interface loopback also serves as a site ID for this particular transit site.

Step 5. **Configure the Hub MC IP address.**

This is the IP address of the source interface defined on the Hub MC. The book's example topology uses R10's loopback, 10.1.0.10. Every MC peers with the Hub MC to get the domain policies, monitor configuration, and global parameters.

Step 6. **Define a password (optional).**

The command **password** *password* is used to secure the peering between MC and BRs.

Example 8-5 displays a branch site MC configuration.

Example 8-5 *Branch Site MC Configuration*

```
R31, R41, and R51 (Branch MCs)
domain IWAN
 vrf default
  master branch
   source-interface Loopback0
   password cisco123
   hub 10.1.0.10
```

MC Status Verification

The status of an MC is displayed with the command **show domain** *name* **vrf** *name* **master status.**

Note If the default VRF (global routing table) is used, the specific VRF name can be omitted.

Example 8-6 verifies the status of the Site 3 MC (R31) for the global routing table. The output provides the configured and operational status and the list of the BRs that are controlled. External interfaces are listed with their corresponding path names. Notice that the BR is actually also the MC, but the command gives the IP address of the BR(s). The MC and BR are two completely independent components even if they run on the same platform.

Example 8-6 *R31 Single CPE Branch MC Status*

```
R31-Spoke# show domain IWAN master status
  *** Domain MC Status ***

 Master VRF: Global
  Instance Type:    Branch
  Instance id:      0
  Operational status:  Up
  Configured status:  Up
  Loopback IP Address: 10.3.0.31
  Load Balancing:
   Operational Status: Up
   Max Calculated Utilization Variance: 27%
   Last load balance attempt: 00:00:13 ago
   Last Reason:  No Controlled Traffic Classes Yet for load balancing
   Total unbalanced bandwidth:
         External links: 55 Kbps  Internet links: 0 Kbps
  External Collector: 10.1.200.1 port: 9995
  Route Control: Enabled
  Transit Site Affinity: Enabled
  Load Sharing: Enabled
  Mitigation mode Aggressive: Disabled
  Policy threshold variance: 20
  Minimum Mask Length: 28
  Syslog TCA suppress timer: 180 seconds
  Traffic-Class Ageout Timer: 5 minutes
  Minimum Packet Loss Calculation Threshold: 15 packets
  Minimum Bytes Loss Calculation Threshold: 1 bytes
  Minimum Requirement: Met

  Borders:
    IP address: 10.3.0.31
    Version: 2
    Connection status: CONNECTED (Last Updated 4d01h ago )
    Interfaces configured:
      Name: Tunnel100 | type: external | Service Provider: MPLS | Status: UP |
              Zero-SLA: NO | Path of Last Resort: Disabled
          Number of default Channels: 2

          Path-id list: 0:1 1:1

      Name: Tunnel200 | type: external | Service Provider: INET | Status: UP |
              Zero-SLA: NO | Path of Last Resort: Disabled
          Number of default Channels: 2

          Path-id list: 0:2 1:2

    Tunnel if: Tunnel0
```

Note The BRs are already configured in Example 8-6.

Example 8-7 verifies the status of the Site 2 MC (R20). The output provides the configured and operational status of the MC and lists the local BRs that are controlled (R21 and R22). External interfaces are listed with their corresponding path names but also include the path identifiers because R12 and R22 are Transit BRs.

Example 8-7 *R20 Standalone MC Status*

```
R20-MC# show domain IWAN master status
  *** Domain MC Status ***

 Master VRF: Global
  Instance Type:    Transit
  POP ID:      1
  Instance id:      0
  Operational status:  Up
  Configured status:  Up
  Loopback IP Address: 10.2.0.20
  Load Balancing:
   Operational Status: Up
   Max Calculated Utilization Variance: 0%
   Last load balance attempt: never
   Last Reason:  Variance less than 20%
   Total unbalanced bandwidth:
         External links: 0 Kbps  Internet links: 0 Kbps
  External Collector: 10.1.200.1 port: 9995
  Route Control: Enabled
  Transit Site Affinity: Enabled
  Load Sharing: Enabled
  Mitigation mode Aggressive: Disabled
  Policy threshold variance: 20
  Minimum Mask Length: 28
  Syslog TCA suppress timer: 180 seconds
  Traffic-Class Ageout Timer: 5 minutes
  Minimum Packet Loss Calculation Threshold: 15 packets
  Minimum Bytes Loss Calculation Threshold: 1 bytes
  Minimum Requirement: Met

  Borders:
    IP address: 10.2.0.21
    Version: 2
    Connection status: CONNECTED (Last Updated 00:54:59 ago )
    Interfaces configured:
      Name: Tunnel100 | type: external | Service Provider: MPLS path-id:1 |
              Status: UP | Zero-SLA: NO | Path of Last Resort: Disabled
          Number of default Channels: 0

    Tunnel if: Tunnel0

   IP address: 10.2.0.22
    Version: 2
    Connection status: CONNECTED (Last Updated 00:54:26 ago )
```

```
Interfaces configured:
  Name: Tunnel200 | type: external | Service Provider: INET path-id:2 |
           Status: UP | Zero-SLA: NO | Path of Last Resort: Disabled
      Number of default Channels: 0

Tunnel if: Tunnel0
```

BR Configuration

The BR is the forwarding plane and selects the path based upon the MC's decisions. The BRs are responsible for creation of performance probes and reporting channel performance metrics, TC bandwidth, and interface bandwidth to the MC so that it can make appropriate decisions based upon the PfR policy.

Transit BR Configuration

A Transit BR is the BR at a hub or transit site. DMVPN interfaces terminate at the BRs. The PfR path name and path identifier are defined on the tunnel interfaces. At the time of this writing, only one DMVPN tunnel is supported on a Transit BR. This limitation is overcome by using multiple BR devices.

The configuration for BRs at a hub or transit site is the same and is given in the following steps:

Step 1. **Define the IWAN domain name.**

All PfR-related configuration is defined within the domain section. The command **domain** {**default** | *domain-name*} defines the PfR domain name. The domain name must be consistent for all devices participating in the same PfRv3 configuration.

Step 2. **Define the VRF.**

PfR is configured per VRF. The command **vrf** {**default** | *vrf-name*} defines the VRF for PfR. You can use the default VRF or use any VRF defined on the router.

Step 3. **Identify the router as a BR.**

The command **border** enters border configuration mode. Upon defining the router as a BR, the EIGRP SAF auto-configuration is enabled automatically by default, and requests are sent to the local MC.

Step 4. **Define the source interface.**

The source interface is one of the loopback interfaces defined on the router. The command **source-interface** *interface-id* configures the loopback used as a source for peering with the local MC.

Step 5. **Define the site MC IP address.**

The BR needs to communicate with the site MC. This is the IP address of the loopback interface defined in the site MC.

The MC is identified with the command **master** *ip-address*.

Step 6. **Define a password (optional).**

The command **password** *password* is used to secure the peering between MC and BRs.

Note It is required that only a single transport exist per Transit BR to guarantee that spoke-to-spoke tunnels will be established. Normal traffic flow follows the Cisco Express Forwarding path, so in a dual-tunnel hub scenario it is possible for spoke 1 to send a packet to spoke 2, which traverses both tunnel interfaces instead of a single interface, keeping the traffic spoke-hub-spoke.

Step 7. **Configure the path name and path index.**

The DVMPN tunnel must define the path name and a path identifier. The *path name* uniquely identifies a transport network. For example, this book uses a primary transport network called MPLS and a secondary transport network called INET. The *path identifier* uniquely identifies a path on a site. This books uses path-id 1 for DMVPN tunnel 100 connected to MPLS and path-id 2 for tunnel 200 connected to INET.

The path name and path identifier are stored inside discovery probes. The BRs extract the information from the probe payload and store the mapping between WAN interfaces and the path name. That information is then transmitted to the site's local MC so that the local MC knows about the WAN interface availability for all the site's BRs.

This is the name of the DMVPN network and is limited to eight characters. The path name should be meaningful, such as MPLS or INET. The path index is a unique index for every tunnel per site. The path name and path index are defined under the DMVPN tunnel interface on the hub router with the command **domain** {**default** | *domain-name*} **path** *path-name* **path-id** *path-id*.

Step 8. **Configure the path of last resort (optional).**

PfR provides a *path of last resort* feature that provides a backup mechanism when all the other defined transports are unavailable. PfR does not check the path of last resort's path characteristics (delay, jitter, packet loss) as a part of path selection. When other transports are available, PfR mutes (does not send smart probes) all channels on the path of last resort. When the defined transports fail, PfR unmutes the default channel, which is sent at a slower rate.

The path of last resort is typically used on metered transports that charge based on data consumption (megabytes per month) and not bandwidth consumption (megabits per second). The DMVPN tunnel and routing protocols must be established for PfR's path of last resort to work properly so that it provides a faster failover technique than the one provided in Chapter 3 when deploying on cellular modem technologies.

The path of last resort is configured under the tunnel interface on the Transit BR that connects to the path of last resort with the command **domain** {**default** | *domain-name*} **path** *path-name* **path-id** *path-id* **path-last-resort**.

Step 9. **Configure zero SLA (optional).**

The zero SLA (0-SLA) feature enables users to reduce probing frequency in their network infrastructure. Reduction in the probing process helps to reduce costs, especially when ISPs charge based on traffic, and helps to optimize network performance when ISPs provide limited bandwidth. When this feature is configured, a probe is sent only on the DSCP-0 channel. For all other DSCPs, channels are created if there is traffic, but no probing is performed. The reachability of other channels is learned from the DSCP-0 channel that is available at the same branch site.

The zero SLA feature is configured under the tunnel interface on the Transit BR that connects to the path with the command **domain** {**default** | *domain-name*} **path** *path-name* **path-id** *path-id* **zero-sla**.

Note It is essential that the path name for a specific transport be *exactly* the same as the path name for other transit or hub sites. The path identifier needs to be unique per site. To simplify troubleshooting, maintain a consistent path ID per transport across sites.

Example 8-8 provides the BR configuration for the hub site (Site 1) and the transit site (Site 2) of this book's sample topology. The path names and path identifiers were taken from Table 8-5.

Table 8-5 *Site 2 PfR BRs, Path Names, and Path Identifiers*

Border Router	Tunnel	Path Name	Path Identifier
R11	Tunnel 100	MPLS	1
R12	Tunnel 200	INET	2
R21	Tunnel 100	MPLS	1
R22	Tunnel 200	INET	2

Example 8-8 *Hub Site and Transit Site BR Configuration*

```
R11
domain IWAN
 vrf default
  border
   source-interface Loopback0
   password cisco123
   master 10.1.0.10
!
interface Tunnel100
 domain IWAN path MPLS path-id 1
```

```
R12
domain IWAN
 vrf default
  border
   source-interface Loopback0
   password cisco123
   master 10.1.0.10
!
interface Tunnel200
 domain IWAN path INET path-id 2
```

```
R21
domain IWAN
 vrf default
  border
   source-interface Loopback0
   password cisco123
   master 10.2.0.20
!
interface Tunnel100
 domain IWAN path MPLS path-id 1
```

```
R22
domain IWAN
 vrf default
  border
   source-interface Loopback0
   password cisco123
   master 10.2.0.20
!
interface Tunnel200
 domain IWAN path INET path-id 2
```

Branch Site BR Configuration

A branch site BR is the BR for a branch site that does not allow transit routing. A BR can have one or multiple DMVPN tunnels connected to it.

The configuration for BRs at a branch site is essentially the same as for transit or hub site BRs with one exception: the path names and identifiers are not configured but automatically discovered through the discovery probes.

The process for configuring a Branch BR is given in the following steps:

Step 1. **Define the IWAN domain name.**

All PfR-related configuration is defined within the domain section. The command **domain** {**default** | *domain-name*} defines the PfR domain name. The domain name must be consistent for all devices participating in the same PfRv3 configuration.

Step 2. **Define the VRF.**

PfR is configured per VRF. The command **vrf** {**default** | *vrf-name*} defines the VRF for PfR. You can use the default VRF or use any VRF defined on the router.

Step 3. **Identify the router as a BR.**

The command **border** enters border configuration mode. When the BR is configured, EIGRP SAF auto-configuration is enabled by default, and requests are sent to the local MC.

Step 4. **Define the source interface.**

The source interface is one of the loopback interfaces defined on the router. The command **source-interface** *interface-id* configures the loopback used as a source for peering with the local MC.

Step 5. **Define the site MC IP address.**

The BR needs to communicate with the local MC. This is the IP address for the source interface specified at the branch's MC configuration. The MC is identified with the command **master** {**local** | *ip-address*}. If the router is both an MC and a BR, the keyword **local** should be used.

Step 6. **Define a password (optional).**

The command **password** *password* is used to secure the peering between the MC and the BRs.

Example 8-9 demonstrates the branch site BR configuration for the book's sample topology.

Example 8-9 *Branch Site BR Configuration*

```
R31, R41, and R51
domain IWAN
 vrf default
  border
   source-interface Loopback0
   password cisco123
   master local
```

```
R52
domain IWAN
 vrf default
  border
   source-interface Loopback0
   password cisco123
   master 10.5.0.51
```

BR Status Verification

The status of a BR is displayed with the command **show domain** *name* **vrf** *name* **border status.** Example 8-10 provides the status of a BR in Site 1 (R11) for the global routing table. The output displays the status of the connection with the site MC, the local path.

Example 8-10 *Hub BR Status*

```
R11-Hub# show domain IWAN border status
  **** Border Status ****

Instance Status: UP
Present status last updated: 4d02h ago
Loopback: Configured Loopback0 UP (10.1.0.11)
Master: 10.1.0.10
Master version: 2
Connection Status with Master: UP
MC connection info: CONNECTION SUCCESSFUL
Connected for: 4d02h
External Collector: 10.1.200.1  port: 9995
Route-Control: Enabled
Asymmetric Routing: Disabled
Minimum Mask length: 28
Sampling: off
Channel Unreachable Threshold Timer: 1 seconds
Minimum Packet Loss Calculation Threshold: 15 packets
Minimum Byte Loss Calculation Threshold: 1 bytes
```

```
Monitor cache usage: 2000 (20%) Auto allocated
Minimum Requirement: Met
External Wan interfaces:
     Name: Tunnel100 Interface Index: 15 SNMP Index: 12 SP: MPLS path-id: 1
              Status: UP Zero-SLA: NO Path of Last Resort: Disabled

Auto Tunnel information:


   Name:Tunnel0 if_index: 16
   Virtual Template: Not Configured
   Borders reachable via this tunnel:  10.1.0.12
```

Example 8-11 provides the status of a BR in Site 3 (R31). It gives the status of the connection with the site MC and the two local paths. Notice that even if the BR is colocated with the MC, there is still a connection with the MC. BR and MC are two completely independent components that can run on different platforms or be colocated on the same router as shown in this example.

Example 8-11 *R31 Branch BR Status*

```
R31-Spoke# show domain IWAN border status
  **** Border Status ****

Instance Status: UP
Present status last updated: 4d02h ago
Loopback: Configured Loopback0 UP (10.3.0.31)
Master: 10.3.0.31
Master version: 2
Connection Status with Master: UP
MC connection info: CONNECTION SUCCESSFUL
Connected for: 4d02h
Route-Control: Enabled
Asymmetric Routing: Disabled
Minimum Mask length: 28
Sampling: off
Channel Unreachable Threshold Timer: 1 seconds
Minimum Packet Loss Calculation Threshold: 15 packets
Minimum Byte Loss Calculation Threshold: 1 bytes
Monitor cache usage: 2000 (20%) Auto allocated
Minimum Requirement: Met
External Wan interfaces:
     Name: Tunnel100 Interface Index: 15 SNMP Index: 12 SP: MPLS Status: UP
              Zero-SLA: NO Path of Last Resort: Disabled Path-id List: 0:1, 1:1
     Name: Tunnel200 Interface Index: 16 SNMP Index: 13 SP: INET Status: UP
              Zero-SLA: NO Path of Last Resort: Disabled Path-id List: 0:2, 1:2

Auto Tunnel information:
   Name:Tunnel0 if_index: 18
   Virtual Template: Not Configured
   Borders reachable via this tunnel:
```

NetFlow Exports

MCs and BRs export useful information using the NetFlow v9 export protocol to a NetFlow collector. A network management application can build reports based on information received. Exports include TCA event, route change, TC bandwidth, and performance metrics. To globally enable NetFlow export, a single command line has to be added on the Hub MC in the domain configuration.

Note NetFlow export can be configured globally on the Hub MC. The NetFlow collector IP address and port are pushed to all MCs in the domain. A manual configuration on a specific branch is also available.

Table 8-6 gives the NetFlow records that are exported from the MC and BRs.

Table 8-6 *PfR NetFlow Records Table*

NetFlow Record	PfR Component	Description
TCA Template	MC	Exports every time a TCA is received (on demand, but a TCA is triggered every 30 seconds if the issue persists). TCAs should contain only the metric that was violated.
Route Change Template	MC	Exported to signal a route change due to a TCA received. BR IP address and interface index are from the new path found.
Immitigable Event Summary Template	MC	Exported to summarize all events where no good route was found after a TCA. This shows a major event that PfR was unable to solve.
Bandwidth	MC	Captures the bandwidth of each external interface. Given that an external interface is mapped to a path name, this gives the amount of bandwidth per path name.

NetFlow Record	PfR Component	Description
Egress Measurement Template	BR	Exports the metrics from the egress Performance Monitor that captures TC aggregate bandwidth.
Ingress Measurement Template	BR	Exports the metrics from the ingress Performance Monitor that captures the performance metrics per channel.

The process for configuring NetFlow export on the Hub MC is given in the following steps:

Step 1. **Define the IWAN domain name.**

All PfR-related configuration is defined within the domain section. The command **domain** {**default** | *domain-name*} defines the PfR domain name. The domain name must be consistent for all devices participating in the same PfRv3 configuration.

Step 2. **Define the VRF.**

PfR is configured per VRF. The command **vrf** {**default** | *vrf-name*} defines the VRF for PfR. You can use the default VRF or use any VRF defined on the router.

Step 3. **Enter the Hub MC configuration.**

The command **master hub** enters MC configuration mode.

Step 4. **Define the collector IP address and port.**

The MC and BR need to send NetFlow records to the collector IP address with a specific port. The IP address and port should match what is defined on the collector. The command **collector** {*collector-ip*} **port** {*collector-port*} defines the collector IP address and port used.

Example 8-12 provides the configuration of a NetFlow collector on the Hub MC. This information is distributed to all MCs in the domain.

Example 8-12 *NetFlow Collector Configuration*

```
R10 (Hub MC)
domain IWAN
 vrf default
  master hub
   source-interface Loopback0
   collector 10.1.200.1 port 9999
```

Domain Policies

PfR policies are rules for how PfR should monitor and control traffic. PfR policies are global to the IWAN domain and are defined in the Hub MC and distributed to Branch and Transit MCs using the EIGRP SAF infrastructure. The following sections explain the configuration of the PfR domain policies.

PfR policies include the following:

- **Administrative policies:** These policies specify constraints such as path preference or transit site preference. Critical applications or media applications are forwarded over the preferred path that provides the best expected QoS (typically MPLS) and fail over to the fallback path INET only if performance is out of policy.

Note Transit site preference was added to the IWAN 2.1 release. The PfR policy selects the preferred central site first, and the path second. For example, Site 1 is preferred over Site 2, and MPLS is preferred over the INET path. The hubs are selected in the following order: R11 (MPLS), R12 (INET), R21 (MPLS), and then R22 (INET).

- **Performance policies***:* These policies specify constraints such as delay, jitter, and loss threshold. They define the boundaries within which an application is correctly supported and user experience is good.
- **Load-balancing policy:** When load balancing is enabled, all the traffic that falls in PfR's default class is load balanced.
- **Monitor interval:** This configures the interval time on ingress monitors that track the status of the probes. Lowering the value from the default setting provides a faster failover to alternative paths.

Performance Policies

PfR policies can be defined on a per-application basis or by DSCP QoS marking. Policies are grouped into *classes*. Class groups are primarily used to define a classification order priority but also to group all traffic that has the same administrative policies. Each class group is assigned a sequence number, and PfR looks at the policies by following the sequence numbers.

There are three core principles that should be considered:

- PfR does not support the mixing and matching of DSCP- and application-based policies in the same class group.
- PfR supports the use of predefined policies from the template or creates custom policies.
- Traffic that does not match any of the class group match statements falls into a default bucket called the default class.

The load-balancing configuration is used to define the behavior of all the traffic that falls into PfR's default class. When load balancing is enabled, TCs that fall into PfR's default class are load balanced. When load balancing is disabled, PfRv3 deletes this default class and those TCs are forwarded based on the routing table information.

Note There is a difference between QoS's default class and PfR's default class. PfR's default class is any traffic that is not defined in an administrative or performance policy regardless of the QoS DSCP markings.

Configuring PfR policies for an IWAN domain includes the following:

- A definition of class names with their respective sequence numbers
- A defined policy based on DSCP or application name
- The use of custom or predefined policies
- Defined path preference

The process for configuring Hub MC policies is given in the following steps:

Step 1. **Enter PfR domain configuration mode.**

The command **domain** {**default** | *domain-name*} enters the PfR domain name. Enter the domain you previously configured to enable the Hub MC.

Step 2. **Define the VRF.**

PfR is configured per VRF. The command **vrf** {**default** | *vrf-name*} defines the VRF for PfR.

Step 3. **Enter Hub MC configuration mode.**

The configuration related to performance policies is defined within the master hub section. The command **master hub** enters MC configuration mode.

Step 4. **Enter policy class configuration mode.**

Each class is defined with a name and a sequence number that gives the order of operations for PfR. You cannot mix and match DSCP- and application-based policies in the same class group. All traffic in a class follows the same administrative policies. The command **class** *class-name* **sequence** *sequence-number* defines the class group.

Step 5. **Configure policy on per-application or DSCP basis.**

You can select a DSCP value from 0 to 63 and assign a policy, or you can use application names. If you use application-based policies, NBAR2 is automatically enabled. You should make sure that all MCs and BRs use the same NBAR2 Protocol Packs. You can then select one of the predefined policies or use custom policies. The command **match** {**application** | **dscp**} *services-value* **policy** is used to define a policy with the corresponding threshold values.

You can select the following policy types:

- best-effort
- bulk-data
- low-latency-data
- real-time-video
- scavenger
- voice
- custom

The configuration can leverage the use of predefined policies or can be entirely based on custom policies where you can manually define all thresholds for delay, loss, and jitter.

Step 6. **Define custom policies (optional).**

Configure the user-defined threshold values for loss, jitter, and one-way delay for the policy type. Threshold values are defined in milliseconds .The command **priority** *priority-number* [**jitter** | **loss** | **one-way-delay**] **threshold** *threshold-value* is used to define the threshold values.

Class-type priorities can be configured only for a custom policy. Multiple priorities can be configured for custom policies.

A common example includes the definition of a class group for voice, interactive video, and critical data. Table 8-7 provides a list of class groups with their respective policies.

Table 8-7 *Class Groups and Policies*

Class Group	Match	Performance Policies	Path Preference
VOICE	DSCP EF	Delay Loss Jitter	Preferred: MPLS Fallback: INET
INTERACTIVE-VIDEO	DSCP AF41 DSCP AF42 DSCP CS4	Delay Loss Jitter	Preferred: MPLS Fallback: INET
CRITICAL DATA	DSCP AF21	Delay Loss	Preferred: MPLS Fallback: INET

Table 8-8 gives the predefined templates that can be used for policy definition.

Table 8-8 *PfR Predefined Policy Templates*

Predefined Template	Threshold Definition
Voice	priority 1 one-way-delay threshold 150 threshold 150 (msec)
	priority 2 packet-loss-rate threshold 1 (%)
	priority 2 byte-loss-rate threshold 1 (%)
	priority 3 jitter 30 (msec)
real-time-video	priority 1 packet-loss-rate threshold 1 (%)
	priority 1 byte-loss-rate threshold 1 (%)
	priority 2 one-way-delay threshold 150 (msec)
	priority 3 jitter 20 (msec)
low-latency-data	priority 1 one-way-delay threshold 100 (msec)
	priority 2 byte-loss-rate threshold 5 (%)
	priority 2 packet-loss-rate threshold 5 (%)
bulk-data	priority 1 one-way-delay threshold 300 (msec)
	priority 2 byte-loss-rate threshold 5 (%)
	priority 2 packet-loss-rate threshold 5 (%)
best-effort	priority 1 one-way-delay threshold 500 (msec)
	priority 2 byte-loss-rate threshold 10 (%)
	priority 2 packet-loss-rate threshold 10 (%)
scavenger	priority 1 one-way-delay threshold 500 (msec)
	priority 2 byte-loss-rate threshold 50 (%)
	priority 2 packet-loss-rate threshold 50 (%)

Example 8-13 demonstrates the use of DSCP-based custom policies.

Example 8-13 *Hub MC Custom Policy Configuration*

```
R10 (Hub MC)
domain IWAN
 vrf default
  master hub
   class VOICE sequence 10
    match dscp ef policy custom
```

```
     priority 2 loss threshold 5
     priority 1 one-way-delay threshold 150
   class INTERACTIVE-VIDEO sequence 20
    match dscp af41 policy custom
     priority 2 loss threshold 5
     priority 1 one-way-delay threshold 150
    match dscp cs4 policy custom
     priority 2 loss threshold 5
     priority 1 one-way-delay threshold 150
   class CRITICAL-DATA sequence 30
    match dscp af21 policy custom
     priority 2 loss threshold 10
     priority 1 one-way-delay threshold 600
```

Example 8-14 demonstrates the use of DSCP-based predefined policies with a more comprehensive list of groups.

Example 8-14 *Hub MC Policy Configuration with Predefined Templates*

```
R10 (Hub MC)
domain IWAN
 vrf default
  master hub
   load-balance
   class VOICE sequence 10
    match dscp ef policy voice
    path-preference MPLS fallback INET
   class REAL_TIME_VIDEO sequence 20
    match dscp cs4 policy real-time-video
    match dscp af41 policy real-time-video
    match dscp af42 policy real-time-video
    match dscp af43 policy real-time-video
    path-preference MPLS fallback INET
   class LOW_LATENCY_DATA sequence 30
    match dscp cs3 policy low-latency-data
    match dscp cs2 policy low-latency-data
    match dscp af21 policy low-latency-data
    match dscp af22 policy low-latency-data
    match dscp af23 policy low-latency-data
    path-preference MPLS fallback INET
   class BULK_DATA sequence 40
    match dscp af11 policy bulk-data
    match dscp af12 policy bulk-data
    match dscp af13 policy bulk-data
    path-preference MPLS fallback INET
   class SCAVENGER sequence 50
    match dscp cs1 policy scavenger
    path-preference MPLS fallback INET
   class DEFAULT sequence 60
    match dscp default policy best-effort
    path-preference INET fallback MPLS
```

The use of a class name and sequence numbers helps differentiate network traffic when the design needs to define a dedicated policy for a specific application of a defined group.

Example 8-15 displays a scenario where a network architect wants to define a policy for Apple FaceTime and a different policy for DSCP AF41 traffic. The network architect cannot put these two definitions in the same class because the TelePresence application may also use DSCP AF41, and PfR will not be able to select which policy to apply. If the network architect uses two different classes and gives TelePresence a higher priority (a lower sequence number is preferred), PfR will be able to assign the correct policy.

Example 8-15 *Use of Sequence Number to Prioritize Apple FaceTime*

```
R10 (Hub MC)
domain IWAN
 vrf default
  master hub
   class FACETIME sequence 10
    match application facetime policy real-time-video
    path-preference INET fallback MPLS
   class INTERACTIVE-VIDEO sequence 20
    match dscp cs4 policy real-time-video
    match dscp af41 policy real-time-video
    match dscp af42 policy real-time-video
    path-preference MPLS fallback INET
```

A complicated example would use application-based policies and leverage the IOS classification engine NBAR2.

Load-Balancing Policy

Traffic classes that do not have any performance policies assigned fall into the default class. PfR can dynamically load balance these traffic classes over all available paths based on the tunnel bandwidth utilization. The default range utilization is defined as 20%.

The process for configuring load balancing is given in the following steps:

Step 1. **Enter PfR domain configuration mode.**

The command **domain** {**default** | *domain-name*} enters the PfR domain name. Enter the domain you previously configured to enable the Hub MC.

Step 2. **Define the VRF.**

PfR is configured per VRF. The command **vrf** {**default** | *vrf-name*} defines the VRF for PfR.

Step 3. **Enter Hub MC configuration mode.**

The command **master hub** enters MC configuration mode.

Step 4. **Configure load balancing.**

When load balancing is enabled, all the traffic that falls into the PfR default class is load balanced. When load balancing is disabled, PfRv3 deletes this default class and traffic is not load balanced but routed based on the routing table information. The command **load-balance** enables load balancing globally in the domain.

Example 8-16 demonstrates the use of load balancing.

Example 8-16 *Load-Balancing Configuration*

```
R10 (Hub MC)
domain IWAN
 vrf default
  master hub
   load-balance
```

Note PfR load balancing is based on tunnel bandwidth, and PfR load sharing is within a tunnel between multiple next hops using a hash algorithm.

Path Preference Policies

PfR looks at all paths that satisfy the policies defined on the Hub MC. Some of the available transports may have a better SLA than other transports. The PfR policy should set the primary path on the transports with better SLAs for business applications.

The process for configuring PfR path preference policies is given in the following steps:

Step 1. **Enter PfR domain configuration mode.**

The command **domain** {**default** | *domain-name*} enters the PfR domain name. Enter the domain you previously configured to enable the Hub MC.

Step 2. **Define the VRF.**

PfR is configured per VRF. The command **vrf** {**default** | *vrf-name*} defines the VRF for PfR.

Step 3. **Enter Hub MC configuration mode.**

The command **master hub** enters MC configuration mode.

Step 4. **Enter policy class configuration mode.**

Each class is defined with a name and a sequence number that gives the order of operations for PfR. You cannot mix and match DSCP- and application-based policies in the same class group. All traffic in a class follows the same administrative policies.

Step 5. **Configure path preference per class.**

Configure the path preference for applications. Group policies sharing the same purpose can be defined under the same class path preference. You cannot configure different path preferences under the same class. PfR supports three paths per path preference logic, so you can configure **path-preference** {*path1*} {*path2*} {*path3*} **fallback** {*path4*} {*path5*} {*path6*} **next-fallback** {*path7*} {*path8*} {*path9*}.

Step 6. **Configure the policy for path of last resort (optional).**

Configure the policy for path of last resort per class as an option to the path preference configuration defined in Step 5. The path of last resort is defined with the command **path-last-resort** *path*.

In Example 8-17, the MPLS path should be preferred over the INET path for business and media applications. PfR allows the definition of a three-level path preference hierarchy. With path preference configured, PfR first considers all paths belonging to the primary group and then goes to the fallback group and finally to the next fallback group.

Example 8-17 *Hub MC Custom Policy Configuration*

```
R10 (Hub MC)
domain IWAN
 vrf default
  master hub
   load-balance
   class VOICE sequence 10
    match dscp ef policy custom
     priority 2 loss threshold 5
     priority 1 one-way-delay threshold 150
    path-preference MPLS fallback INET
   class INTERACTIVE-VIDEO sequence 20
    match dscp af41 policy custom
     priority 2 loss threshold 5
     priority 1 one-way-delay threshold 150
    match dscp cs4 policy custom
     priority 2 loss threshold 5
     priority 1 one-way-delay threshold 150
    path-preference MPLS fallback INET
   class CRITICAL-DATA sequence 30
    match dscp af21 policy custom
     priority 2 loss threshold 10
     priority 1 one-way-delay threshold 600
    path-preference MPLS fallback INET
```

Quick Monitor

When PfR is enabled on a BR, it instantiates three different monitors. One of them is applied on ingress and is used to measure the performance of the channel. The default monitor interval is 30 seconds.

This monitor interval can be lowered for critical applications to achieve a fast failover to the secondary path. This is known as *quick monitor*. You can define one quick monitor interval for all DSCP values associated to critical applications.

The process for configuring the quick monitor is given in the following steps:

Step 1. **Enter PfR domain configuration mode.**

The command **domain** {**default** | *domain-name*} enters the PfR domain name. Enter the domain you previously configured to enable the Hub MC.

Step 2. **Define the VRF.**

PfR is configured per VRF. The command **vrf** {**default** | *vrf-name*} defines the VRF for PfR.

Step 3. **Enter Hub MC configuration mode.**

The command **master hub** enters MC configuration mode.

Step 4. **Define the monitor interval for specific QoS DSCPs.**

The default monitor interval is 30 seconds. PfR allows lowering of the monitor interval for critical applications to achieve a fast failover to the secondary path. The command **monitor-interval** *seconds* **dscp** *value* is used to define the quick monitor interval.

Example 8-18 demonstrates defining a monitor interval.

Example 8-18 *Configuration for Quick Monitor*

```
R10 (Hub MC)
domain IWAN
 vrf default
  master hub
   source-interface Loopback0
   enterprise-prefix prefix-list ENTERPRISE_PREFIX
   site-prefixes prefix-list SITE_PREFIX
   monitor-interval 4 dscp af21
   monitor-interval 4 dscp cs4
   monitor-interval 4 dscp af41
   monitor-interval 4 dscp ef
```

Hub Site Master Controller Settings

The enterprise-prefix prefix list defines the boundary for all the internal enterprise prefixes. A prefix that is not from the enterprise prefix list is considered a PfR Internet prefix. PfR does not monitor performance (delay, jitter, byte loss, or packet loss) for network traffic. The enterprise-prefix prefix list is defined only on the Hub MC under the MC configuration with the command **enterprise-prefix prefix-list** *prefix-list-name*.

Note When PfR parses the prefix list, it does not exclude any prefixes with the **deny** statement.

Example 8-19 demonstrates the configuration of the enterprise-prefix prefix list.

Example 8-19 *PfR Enterprise-Prefix Prefix List*

```
R10 (Hub MC)
domain IWAN
 vrf default
  master hub
   enterprise-prefix prefix-list ENTERPRISE_PREFIX
!
ip prefix-list ENTERPRISE_PREFIX seq 10 permit 10.0.0.0/8
```

Hub, Transit, or Branch Site Specific MC Settings

The site-prefix prefix list defines the static-site prefix for the local site and disables automatic site-prefix learning on the BR. The static-site prefix list is required only for Hub and Transit MCs. A site-prefix prefix list is optional on Branch MCs. The site prefix is defined under the MC configuration with the command **site-prefixes prefix-list** *prefix-list-name*.

Example 8-20 provides R10 and R20's site-prefix prefix list configuration. Notice that a second entry is added to R10's SITE_PREFIX prefix list to accommodate the DCI between Site 1 and Site 2.

Example 8-20 *PfR Site-Prefix Prefix List*

```
R10 (Hub MC)
domain IWAN
 vrf default
  master hub
   site-prefixes prefix-list SITE_PREFIX
!
ip prefix-list SITE_PREFIX seq 10 permit 10.1.0.0/16
ip prefix-list SITE_PREFIX seq 20 permit 10.2.0.0/16
```

```
R20 (Transit MC)
domain IWAN
 vrf default
  master transit 1
   source-interface Loopback0
   site-prefixes prefix-list SITE_PREFIX
   hub 10.1.0.10
!
ip prefix-list ENTERPRISE_PREFIX seq 10 permit 10.0.0.0/8
ip prefix-list SITE_PREFIX seq 10 permit 10.1.0.0/16
ip prefix-list SITE_PREFIX seq 20 permit 10.2.0.0/16
```

Note When statically configuring a site prefix list, breaking the prefix list into smaller prefixes provides granularity to support load balancing. Creating smaller prefixes creates additional TCs for the same DSCP, which then provides more path choices for PfR.

For example, instead of creating a 10.1.0.0/16, using a 10.1.0.0/17 and a 10.1.128.0/17 provides more TCs.

Complete Configuration

The PfR configuration section explained the configuration in a step-by-step fashion to provide a thorough understanding of the configuration. Example 8-21 provides a complete sample PfR configuration for the hub site routers.

Example 8-21 *Hub Site PfR Configuration*

```
R10 (Hub MC)
domain IWAN
 vrf default
  master hub
   source-interface Loopback0
   enterprise-prefix prefix-list ENTERPRISE_PREFIX
   site-prefixes prefix-list SITE_PREFIX
   password cisco123
   monitor-interval 2 dscp af21
   monitor-interval 2 dscp cs4
   monitor-interval 2 dscp af41
   monitor-interval 2 dscp ef
   collector 10.1.200.200 port 2055
!
ip prefix-list ENTERPRISE_PREFIX seq 10 permit 10.0.0.0/8
ip prefix-list SITE_PREFIX seq 10 permit 10.1.0.0/16
ip prefix-list SITE_PREFIX seq 20 permit 10.2.0.0/16
```

```
R11
domain IWAN
 vrf default
  border
   source-interface Loopback0
   password cisco123
   master 10.1.0.10
!
interface Tunnel100
 description DMVPN-MPLS
 domain IWAN path MPLS path-id 1
```

```
R12
domain IWAN
 vrf default
  border
   source-interface Loopback0
   password cisco123
   master 10.1.0.10
!
interface Tunnel200
 description DMVPN-INET
 domain IWAN path INET path-id 2
```

Example 8-22 provides a complete sample PfR configuration for the transit site routers.

Example 8-22 *Transit Site PfR Configuration*

```
R20 (Transit MC)
domain IWAN
 vrf default
  master transit 1
   source-interface Loopback0
   password cisco123
   site-prefixes prefix-list SITE_PREFIX
   hub 10.1.0.10
!
ip prefix-list SITE_PREFIX seq 10 permit 10.1.0.0/16
ip prefix-list SITE_PREFIX seq 20 permit 10.2.0.0/16
```

```
R21
domain IWAN
 vrf default
  border
   source-interface Loopback0
   password cisco123
   master 10.2.0.20
!
interface Tunnel100
 description DMVPN-MPLS
 domain IWAN path MPLS path-id 1
```

```
R22
domain IWAN
 vrf default
  border
   source-interface Loopback0
   password cisco123
   master 10.2.0.20
!
interface Tunnel200
 description DMVPN-INET
 domain IWAN path INET path-id 2
```

Example 8-23 provides a complete sample PfR configuration for the branch site routers.

Example 8-23 *Branch Site PfR Configuration*

```
R31, R41, and R51
domain IWAN
 vrf default
  border
   source-interface Loopback0
   password cisco123
   master local
  master branch
   source-interface Loopback0
   password cisco123
   hub 10.1.0.10
```

```
R52
domain IWAN
 vrf default
  border
   source-interface Loopback0
   password cisco123
   master 10.5.0.51
```

Advanced Parameters

PfR configuration has been streamlined and simplified, and it includes many parameters defined by default that work for most customers. The PfR advanced mode is used on the Hub MC to change the default values for the entire IWAN domain.

Unreachable Timer

The BR declares a channel *unreachable* in both directions if no packets are received from the peer within the unreachable timer. The unreachable timer is defined as one second by default and can be tuned if needed.

The command **channel-unreachable-timer** *seconds* configures the unreachable timer on the Hub MC.

Example 8-24 provides a sample PfR advanced configuration for the unreachable timer for the Hub MC.

Example 8-24 *Hub MC PfR Advanced Configuration for Unreachable Timer*

```
R10 (Hub MC)
domain IWAN
 vrf default
  master hub
   advanced
    channel-unreachable-timer 4
```

Smart Probes Ports

Smart probes are generated from BRs when there is no user traffic for a specific channel on a specific path. These probes are configured with the following parameters:

- **Source address:** Source site MC
- **Destination address:** Destination site MC
- **Source port:** 18000
- **Destination port:** 19000

Source and destination ports can be changed on the Hub MC with the command **smart-probes source-port** *src-port-number* and the command **smart-probes destination-port** *dst-port-number.*

Example 8-25 provides a sample PfR advanced configuration for the smart probes source and destination ports for the Hub MC.

Example 8-25 *Hub MC PfR Advanced Configuration for Smart Probes Source and Destination Ports*

```
R10 (Hub MC)
domain IWAN
 vrf default
  border
   smart-probes source-port 180001
   smart-probes destination-port 19001
```

Transit Site Affinity

Transit Site Affinity (also known as Transit Site Preference) is used in the context of a multiple-transit-site deployment with the same set of prefixes advertised from all central sites (Figure 8-3). When routing is configured to define one of the central sites to be preferred, PfR prioritizes the use of the next hops available on that site.

The command **no transit-site-affinity** disables the transit site preference and is configured on the Hub MC.

Example 8-26 provides a sample PfR advanced configuration for the Hub MC.

Example 8-26 *Hub MC PfR Advanced Configuration for Transit Site Preference*

```
R10 (Hub MC)
domain IWAN
 vrf default
  master hub
   no transit-site-affinity
```

Path Selection

Path selection is a combination of the routing configuration and decisions made by PfR administrative and performance policies.

Routing—Candidate Next Hops

As explained in Chapter 7, "Introduction to Performance Routing (PfR)," routing configuration and design are key to PfR and determine the use of possible next hops. That is especially critical for the direction from branches to central sites (hub or transit sites). A branch can also have multiple next hops over a single tunnel interface for a specific destination prefix. These next hops can be spread across multiple transit sites as depicted in Figure 8-3 (Site 1 and Site 2 advertise the same set of prefixes).

The spoke router considers all the paths (multiple NHs) toward the POPs and maintains a list of active/standby candidate next hops per prefix and interface; these are derived based on the routing configuration.

- **Active next hop**: A next hop is considered active for a given prefix if it has the best metric.
- **Standby next hop**: A next hop is considered standby for a given prefix if it advertises a route for the prefix but does not have the best metric.
- **Unreachable next hop**: A next hop is considered unreachable for a given prefix if it does not advertise any route for the prefix.

Routing—No Transit Site Preference

The same set of prefixes can be advertised from multiple transit sites, and routing configuration is done in such a way that there is no preference given to a particular transit site. The goal is to be able to load-balance the traffic across all transit sites to reach the data centers.

Figure 8-5 illustrates the hub routers advertising the same BGP local preference (100) from all BRs from Site 1 and Site 2 (default local preference of 100 used here). The same effect can be achieved with EIGRP.

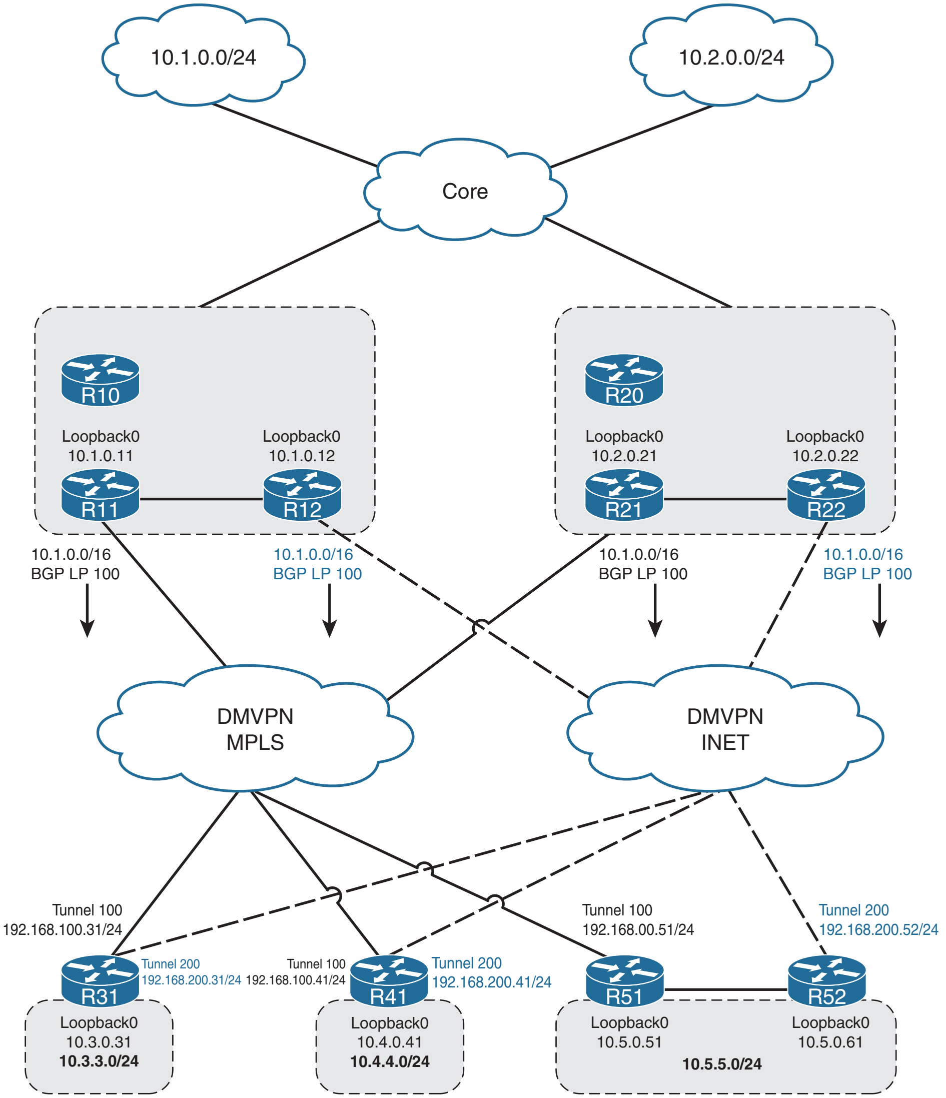


Figure 8-5 *Site 1 and Site 2 Advertising Prefixes with the Same BGP Local Preference*

Table 8-9 displays the PfR table and the status. PfR picks the next-hop information (parent route) from BGP and lists all next hops for prefix 10.1.0.0/16 as *active*.

Table 8-9 *PfR Table—Same Prefix, Same BGP Local Preference*

Prefix	Interface	Possible Next Hop	BGP Local Preference	Status
10.1.0.0/16	Tunnel 100	R11	100	Active
10.1.0.0/16	Tunnel 200	R12	100	Active
10.1.0.0/16	Tunnel 100	R21	100	Active
10.1.0.0/16	Tunnel 200	R22	100	Active

Routing—Site Preference

You may want to define one of the transit sites as a preferred site in the WAN design. In the book's generic topology, Site 1 could be the primary site for all spokes. Another use case is that all branches located in the United States want to use Site 1, which is located in the United States, and branches located in Europe want to use Site 2, which is located in Europe, because geographic proximity reduces latency.

Figure 8-6 illustrates the use of different BGP local preferences from BRs (DMVPN hub routers) located in Site 1 and Site 2. The WAN design sets the local preference of the MPLS transport over the Internet (INET), and Site 1 is preferred over Site 2. The local preference values set on the BRs are provided in Table 8-10.

Table 8-10 *BGP Local Preference Definition*

BR	Site	BGP Local Preference
R11	Site1	100,000
R12	Site1	20,000
R21	Site2	3000
R22	Site2	400

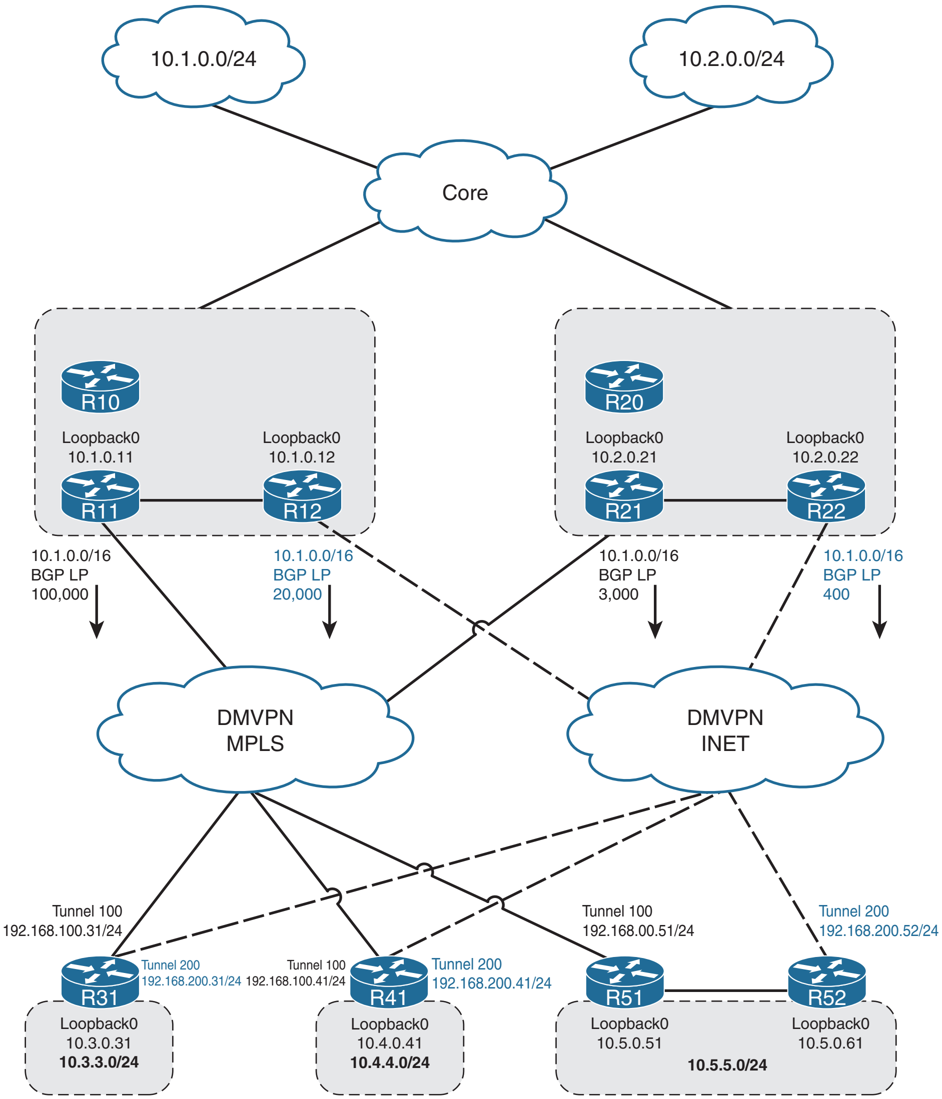


Figure 8-6 *Site 1 and Site 2 Advertising Prefixes with Different BGP Local Preferences*

> **Note** The values used in the BGP configuration should make reading the BGP table easy. A path is more preferred if it has a higher local preference. If the local preference is not displayed or set, a router assumes the default value of 100 during the BGP best-path calculation.

Defining the local preference is a matter of design choice. If all the sites share the same routing policy, it is probably easier to set this preference directly on the Transit BRs. If the routing policy varies at each branch, setting the local preference on an inbound route map based upon the BGP community is the best technique.

Example 8-27 provides the relevant BGP configuration for setting the local preference on R11. The complete BGP configuration was provided in Chapter 4, "Intelligent WAN (IWAN) Routing."

Example 8-27 *R11 BGP Local Preference Configuration*

```
R11-Hub
router bgp 10
 bgp listen range 192.168.100.0/24 peer-group MPLS-SPOKES
 neighbor MPLS-SPOKES peer-group
 neighbor MPLS-SPOKES remote-as 10
!
 address-family ipv4
  aggregate-address 10.1.0.0 255.255.0.0 summary-only
  aggregate-address 10.0.0.0 255.0.0.0 summary-only
  neighbor MPLS-SPOKES activate
  neighbor MPLS-SPOKES send-community
  neighbor MPLS-SPOKES route-reflector-client
  neighbor MPLS-SPOKES weight 50000
  neighbor MPLS-SPOKES route-map BGP-MPLS-SPOKES-OUT out
exit-address-family
!
route-map BGP-MPLS-SPOKES-OUT permit 10
 set local-preference 100000
```

As a result, PfR picks the next-hop information (parent route) from BGP and builds Table 8-11.

Table 8-11 *PfR Table—Same Prefix with Different BGP Local Preference*

Prefix	Interface	Possible Next Hop	BGP Local Preference	Status
10.1.0.0/16	Tunnel 100	R11	100,000	Active
10.1.0.0/16	Tunnel 200	R12	20,000	Active
10.1.0.0/16	Tunnel 100	R21	3000	Backup
10.1.0.0/16	Tunnel 200	R22	400	Backup

Example 8-28 displays the BGP table for the 10.1.0.0/16 prefix. Notice that R12 is not the best path for BGP given the lower local preference compared to R11, but R12 is still considered as a valid next hop for PfR in Table 8-11. Remember that PfR picks at least one next hop for every external tunnel.

Example 8-28 *R31 Next-Hop Information with Local Preference*

```
R31-Spoke# show bgp ipv4 unicast
! Output omitted for brevity
BGP table version is 12, local router ID is 10.3.0.31
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 * i 10.1.0.0/16      192.168.200.22           0    400  50000 i
 * i                  192.168.100.21           0   3000  50000 i
 * i                  192.168.200.12           0  20000  50000 i
 *>i                  192.168.100.11           0 100000  50000 i
```

Note R12 is not the best path for BGP given the lower local preference compared to R11 but is still considered a valid next hop for PfR. Remember that PfR picks at least one next hop for every external tunnel.

PfR Path Preference

Based on available next-hop information, PfR uses its own administrative policies to choose the appropriate next hops per prefix. At the time of this writing, PfR has two main administrative policies:

- Transit site preference (3.16.1 onward)
- Path preference

PfR supports three paths per path preference logic. The command **path-preference** {*path1*} {*path2*} {*path3*} **fallback** {*path4*} {*path5*} {*path6*} **next-fallback** {*path7*} {*path8*} {*path9*} is used in a class group to configure the path preference for applications.

- {*path1*}, {*path2*}, and {*path3*} are the preferred paths for the class group.
- {*path4*}, {*path5*}, and {*path6*} are the secondary paths and are used if all preferred paths are out of policy.

- {*path7*}, {*path8*}, and {*path9*} are then used when all preferred and secondary paths are out of policy.

A generic PfR path preference configuration is shown in Example 8-29.

Example 8-29 *Path Preference Definition*

```
R10 (Hub MC)
domain IWAN
 vrf default
  master hub
   load-balance
   class VOICE sequence 10
    match dscp ef policy custom
     priority 2 loss threshold 5
     priority 1 one-way-delay threshold 150
     path-preference MPLS1 MPLS2 MPLS3 fallback INET1 INET2 INET3 next-fallback
       INET4 INET5 INET6
```

Note The keyword **next-fallback** was introduced with IOS 15.5(3)M and IOS XE 3.16. Initial code supported five paths for path preference and four paths for fallback.

With path preference configured, PfR first considers all the links belonging to the preferred path preference (it includes the active and the standby links belonging to the preferred path) and then uses the fallback provider links. Without path preference configured, PfR gives preference to the active channels and then the standby channels (active/standby is per prefix) with respect to the performance and policy decisions.

PfR Transit Site Preference

Transit site preference is used in the context of a multiple-transit-site deployment with the same set of prefixes advertised from all central sites (Figure 8-3).

A specific transit site is preferred for a specific prefix, as long as there are available *in-policy* channels for that site. Transit site preference is a higher-priority filter and takes precedence over path preference. The concept of active and standby next hops based on routing metrics and advertised mask length in routing is used to gather information about the preferred transit site for a given prefix. For example, if the best metric for a given prefix is on Site 1, all the next hops on that site for all the paths are tagged as active (only for that prefix as shown earlier in Table 8-11).

Figure 8-7 illustrates the transit site preference with Site 3 preferring Site 1 and Site 4 preferring Site 2 to reach the same prefix.

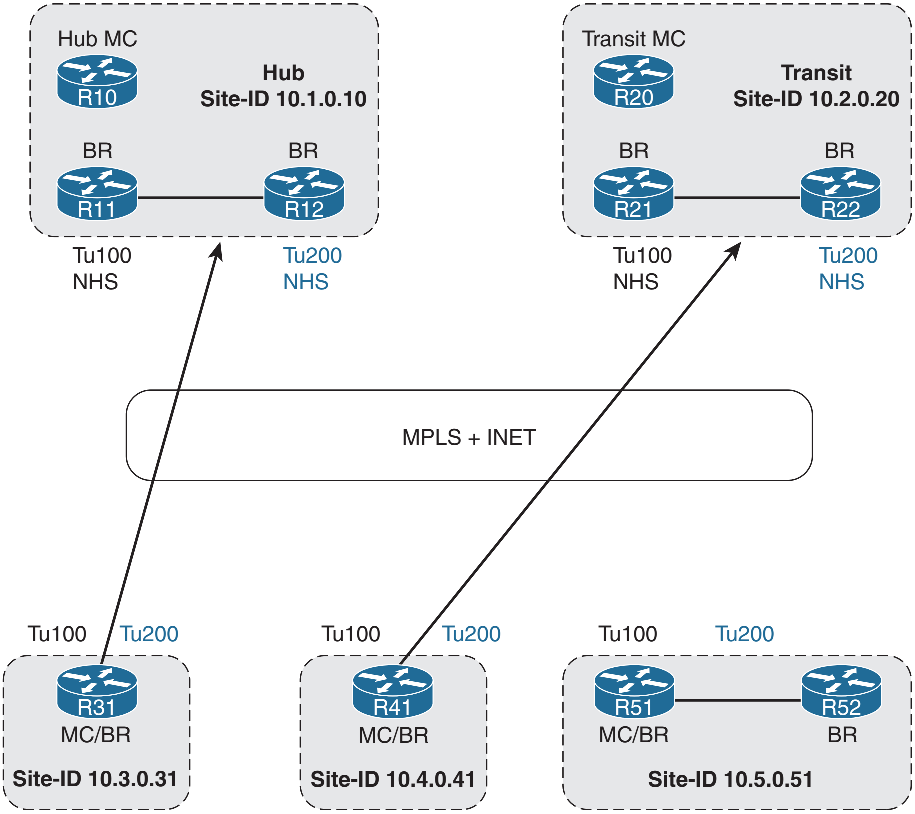


Figure 8-7 *Transit Site Preference*

Note Active and standby channels are per prefix and span the POPs. A spoke will randomly (hash) choose the active channel.

Using Transit Site Preference and Path Preference

The BGP local preference has been configured as in Table 8-11 and is displayed in Figure 8-6 for the 10.1.0.0/16 prefix. The active/standby next-hop tagging happens irrespective of transit site affinity being enabled or disabled. The next-hop status results are listed in Table 8-12.

Table 8-12 *PfR Table—Next-Hop Status*

POP	Path	Next Hop	Prefix	Status
1	MPLS	R11	10.1.0.0/16	Active
1	INET	R12	10.1.0.0/16	Active
2	MPLS	R21	10.1.0.0/16	Standby
2	INET	R22	10.1.0.0/16	Standby

The path selection results are listed in Table 8-13.

Table 8-13 *PfR Table—Path Selection Algorithm*

Prefix	Transit Site Preference	Path Preference	Order
10.1.0.0/16	Site 1	MPLS	R11, R12, R21, R22
10.1.0.0/16	No	MPLS	R11, R21, R12, R22
10.1.0.0/16	Site 1	No	R11/R12, R21/R22
10.1.0.0/16	No	No	R11/R12/R21/R22

Note Transit site preference is enabled by default. Transit site preference was introduced with IOS 15.5(3)M1 and IOS XE 3.16.1.

Summary

This chapter focused on the configuration of Cisco PfR. PfRv3's configuration has been drastically simplified compared to its earlier versions. PfR's provisioning is streamlined now that all of the site's PfR policies are centralized on the Hub MC. Intelligence and automation occur in the background to simplify path discovery, service prefix discovery, and site discovery. The result of this is that the majority of the configuration occurs on the Hub MC, and even then the policies have been simplified.

PfR configuration involves the configuration of the MCs for the IWAN domain. The Transit and Branch MC configuration simply identifies the PfR source interface and the Hub MC IP address. The Hub MC contains the logical domain controller functionality and acts as the local MC for the hub site. The Hub MC should act as a standalone device.

The BR functions consist of identifying the PfR source interface and the local MC IP address. Transit BRs are also required to configure the path ID and path name.

Combining PfR policies with the routing configuration defines the path used for transit sites.

Further Reading

Cisco. "Performance Routing Version 3." www.cisco.com/go/pfr

Cisco. "PfRv3 Configuration." www.cisco.com

Chapter 9

PfR Monitoring

This chapter covers the following topics:

- Checking hub site status
- Checking transit site status
- Checking branch site status
- Monitoring traffic classes
- Monitoring channels
- Transit site preference

The previous chapters described the intelligent path control called Performance Routing (PfR) for the IWAN domain. This chapter covers how to monitor PfR operations.

Monitoring *Performance Routing (PfR)* for the IWAN domain includes the following steps:

1. **Checking the status of the hub site:** This is one of the enterprise central sites that includes the definition of the *Hub master controller (MC)* and multiple *Transit border routers (BRs)*. There is only one hub site in an IWAN domain. The Hub MC contains the PfR policies for the IWAN domain that it distributes to every other MC over a network communications channel referred to as the *Service Advertisement Framework (SAF)*. The Hub MC uses SAF to communicate with the local Transit BRs to learn applications, monitor performance metrics, and enforce path selection. Every BR at the hub site terminates only one overlay network. Checking the status of the hub site ensures that the local BRs are connected to the Hub MC, external tunnels are correctly defined, and the Hub MC is connected to every MC in the domain.

2. **Checking the status of one or more transit sites:** These are the enterprise central sites that are used as transits to reach the data centers or used as the transit for branch-to-branch communication. Every BR on the hub site terminates only one

overlay network. Checking the status of the transit site ensures that the local BRs are connected to the Transit MC, external tunnels are correctly defined, and the Transit MC is connected to the Hub MC to get the PfR policies.

3. **Checking the status of branch sites:** A branch site includes one Branch MC and one or more Branch BRs. A Branch BR can support multiple DMVPN tunnels. Checking the status of a branch site ensures that the Branch MC is communicating with the Hub MC, local BRs are connected, and external tunnels are correctly discovered.

Topology

Figure 9-1 displays the dual-hub and dual-cloud topology used in this book to explain the monitoring of PfR. The DMVPN networks have been configured along with the necessary routing protocol to provide end-to-end connectivity in the topology. Network traffic is generated between branch sites (Site 3, Site 4, and Site 5) and central sites (Site 1 and Site 2) as well as between Site 3 and Site 4 (voice).

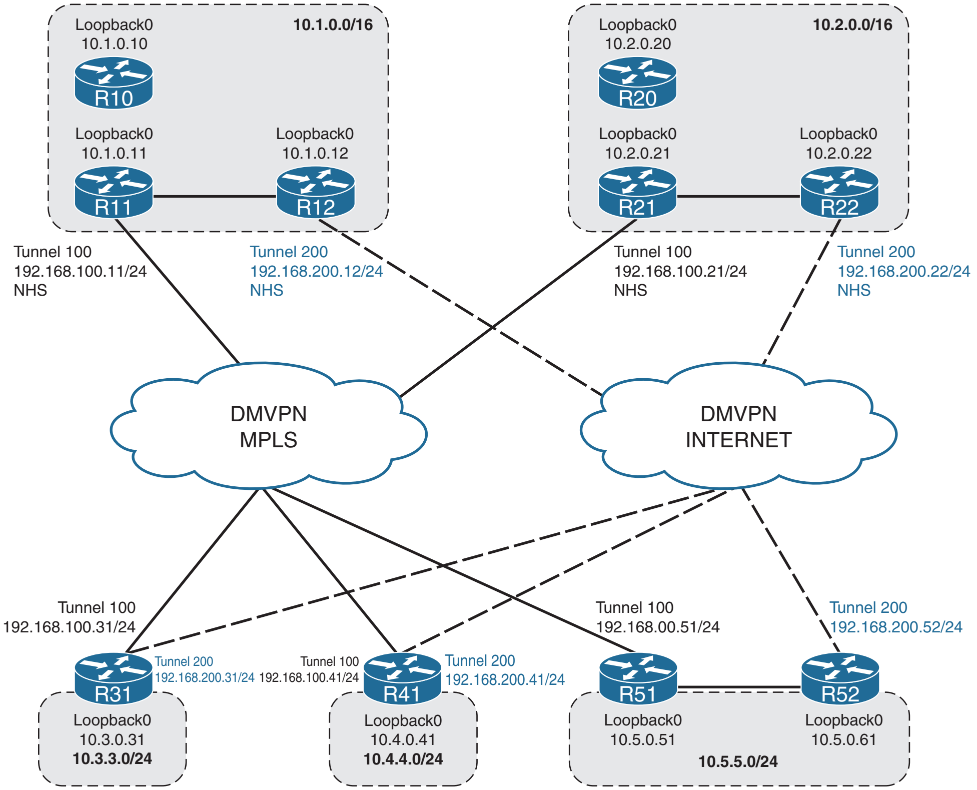


Figure 9-1 *Book's Sample PfR Topology*

Checking the Hub Site

The Hub MC is located at the hub site in the IWAN topology. The Hub MC plays a very important role in the IWAN domain. The very first step is to check its status, locally connected BRs, and associated tunnels. The second step is to verify the list of MCs that are connected. If a branch or transit site MC is missing from the list, the corresponding site will be unable to receive the policies and will not receive the discovery probes. As a result, that site will not be able to control any traffic.

R10 is the Hub MC for this book's sample topology. It controls two BRs in Site 1: R11, which is connected to the MPLS transport, and R12, which is connected to the INET transport.

Check the Routing Table

PfR relies on the routing protocol (EIGRP or BGP) and therefore a proper configuration of these protocols is required. Destination prefixes have to be advertised across all paths via the routing protocol for PfR to work correctly:

- PfR first searches for a parent route to control any traffic class for a specific destination prefix. PfR performs this crucial step to avoid the blackholing of network traffic by a downstream router. The parent route check is done using the NHRP cache, BGP table, EIGRP topology table, static routes, or RIB (when ECMP paths exist) and can be satisfied by an exact match prefix, an aggregate, or the default route.
- A remote MC loopback address is used as a destination address by smart probes (including discovery probes used to help branch sites discover their external interfaces and path names).

Therefore, destination prefixes and remote MC loopback addresses have to be routable over all external interfaces on the BRs.

Table 9-1 provides the tunnel next-hop address information with the site name, MC name, prefix, and loopback address for the BR R11.

Table 9-1 *R11 Site Prefix Information*

Site	MC	Site Prefix	Loopback Address	Tunnel Address
Site 3	R31	10.3.3.0/24	10.3.0.31	192.168.100.31
Site 4	R41	10.4.4.0/24	10.4.0.41	192.168.100.41
Site 5	R51	10.5.5.0/24	10.5.0.51	192.168.100.51

Example 9-1 provides the output of the BGP table on R11 (a similar command would be used to check the EIGRP topology table). All site prefixes and loopbacks are listed and are routable over only the MPLS tunnel during normal conditions. The routing configuration design ensures that the hub router always prefers routes learned from tunnels associated to its transport, which prevents packets from being routed over any other internal interfaces.

Note The BGP (or EIGRP) table is used to validate that a destination prefix is routable over all external paths and not the routing table. The reason for examining the BGP table or EIGRP topology table is that the routing table shows only the best path and not *all* paths. In that case (a Hub BR) only one transport is available, but it is a good practice to view the BGP or EIGRP table. This is typically critical on a single CPE branch with two transports and is critical in the future when multiple transports will be supported on a Hub BR.

Example 9-1 *R11 BGP Topology Table*

```
R11-Hub# show bgp ipv4 unicast
! Output omitted for brevity
     Network          Next Hop            Metric LocPrf Weight Path
 *>  0.0.0.0          192.168.100.11          10         32768 i
 *>  10.0.0.0         0.0.0.0                            32768 i
 *>  10.1.0.0/16      0.0.0.0                            32768 i
 s>  10.1.0.11/32     0.0.0.0                  0         32768 ?
 s>  10.1.12.0/24     0.0.0.0                  0         32768 ?
 s>  10.1.111.0/24    0.0.0.0                  0         32768 ?
 s>i 10.3.0.31/32     192.168.100.31           0    100  50000 ?
 s>i 10.3.3.0/24      192.168.100.31           0    100  50000 ?
 s>i 10.4.0.41/32     192.168.100.41           0    100  50000 ?
 s>i 10.4.4.0/24      192.168.100.41           0    100  50000 ?
 s>i 10.5.0.51/32     192.168.100.51           0    100  50000 ?
 s>i 10.5.0.52/32     192.168.100.51           2    100  50000 ?
 s>i 10.5.5.0/24      192.168.100.51           0    100  50000 ?
```

Table 9-2 provides the tunnel next-hop address information with the site name, MC name, prefix, and loopback address for BR R12.

Table 9-2 *R12 Site Prefix Information*

Site	MC	Site Prefix	Loopback Address	Tunnel Address
Site 3	R31	10.3.3.0/24	10.3.0.31	192.168.200.31
Site 4	R41	10.4.4.0/24	10.4.0.41	192.168.200.41
Site 5	R51	10.5.5.0/24	10.5.0.51	192.168.200.51

Example 9-2 provides the output of the BGP table on R12 (a similar command would be used to check the EIGRP topology table). All site prefixes and loopbacks are listed and are routable over the MPLS tunnel. As with the R11 branch router, these prefixes should not be routed over any internal interfaces.

Example 9-2 *R12 BGP Table*

```
R12-Hub# show bgp ipv4 unicast
! Output omitted for brevity
     Network          Next Hop            Metric LocPrf Weight Path
 *   0.0.0.0          192.168.200.12          10         32768 i
 *>                   192.168.200.12          10         32768 i
 *>  10.0.0.0         0.0.0.0                            32768 i
 *>  10.1.0.0/16      0.0.0.0                            32768 i
 s>  10.1.0.12/32     0.0.0.0                  0         32768 ?
 s>  10.1.12.0/24     0.0.0.0                  0         32768 ?
 s>  10.1.112.0/24    0.0.0.0                  0         32768 ?
 s>i 10.3.0.31/32     192.168.200.31           0    100  50000 ?
 s>i 10.3.3.0/24      192.168.200.31           0    100  50000 ?
 s>i 10.4.0.41/32     192.168.200.41           0    100  50000 ?
 s>i 10.4.4.0/24      192.168.200.41           0    100  50000 ?
 s>i 10.5.0.51/32     192.168.200.52           2    100  50000 ?
 s>i 10.5.0.52/32     192.168.200.52           0    100  50000 ?
 s>i 10.5.5.0/24      192.168.200.52           0    100  50000 ?
```

Checking the Hub MC

The status of a Hub MC is shown with the command **show domain** *domain-name* [**vrf** *vrf-name*] **master status**. If the default VRF (global routing table) is used, the specific VRF name can be omitted.

Example 9-3 verifies the status of the Site 1 Hub MC (R10) for the global routing table. The output provides the role of the MC, the configured and operational status, and the list of the BRs that are controlled (R11 and R12), and it displays whether Transit Site Affinity is enabled. External interfaces are listed with their corresponding path names and path identifiers. The output also provides the status of load balancing. If the load-balancing status is *disabled*, the MC will not control the default class traffic classes. If the load-balancing status is *enabled*, the MC will control and load-share default class traffic classes among all external interfaces.

Example 9-3 *R10 Hub MC Status*

```
R10-HUB-MC# show domain IWAN master status

  *** Domain MC Status ***

 Master VRF: Global

  Instance Type:    Hub
  Instance id:      0
  Operational status:  Up
  Configured status:  Up
  Loopback IP Address: 10.1.0.10
  Global Config Last Publish status: Peering Success
  Load Balancing:
   Admin Status: Enabled
   Operational Status: Up
   Enterprise top level prefixes configured: 1
   Max Calculated Utilization Variance: 2%
   Last load balance attempt: never
   Last Reason:  Variance less than 20%
   Total unbalanced bandwidth:
         External links: 0 Kbps  Internet links: 0 Kbps
  External Collector: 10.1.200.1 port: 9995
  Route Control: Enabled
  Transit Site Affinity: Enabled
  Load Sharing: Enabled
  Mitigation mode Aggressive: Disabled
  Policy threshold variance: 20
  Minimum Mask Length: 28
  Syslog TCA suppress timer: 180 seconds
  Traffic-Class Ageout Timer: 5 minutes
  Channel Unreachable Threshold Timer: 1 seconds
  Minimum Packet Loss Calculation Threshold: 15 packets
  Minimum Bytes Loss Calculation Threshold: 1 bytes

  Borders:
    IP address: 10.1.0.12
    Version: 2
    Connection status: CONNECTED (Last Updated 23:03:11 ago )
    Interfaces configured:
      Name: Tunnel200 | type: external | Service Provider: INET path-id:2 |
        Status: UP | Zero-SLA: NO | Path of Last Resort: Disabled
          Number of default Channels: 1
    Tunnel if: Tunnel0

    IP address: 10.1.0.11
    Version: 2
    Connection status: CONNECTED (Last Updated 23:03:06 ago )
    Interfaces configured:
      Name: Tunnel100 | type: external | Service Provider: MPLS path-id:1 |
        Status: UP | Zero-SLA: NO | Path of Last Resort: Disabled
          Number of default Channels: 1

    Tunnel if: Tunnel0
```

Checking the Hub BRs

The status of a BR is shown with the command **show domain** *domain-name* [**vrf** *vrf-name*] **border status.**

Example 9-4 provides the status of a BR in Site 1 (R11) for the global routing table. The output displays the status of the connection with the local site MC, the local path. Tunnel 0 is the auto-tunnel automatically built between the BRs on the site. In this example, the address listed (10.1.0.12) is the loopback address of the site BR connected to the path INET (R12).

Example 9-4 *Hub BR Status*

```
R11-Hub# show domain IWAN border status
  **** Border Status ****

Instance Status: UP
Present status last updated: 4d02h ago
Loopback: Configured Loopback0 UP (10.1.0.11)
Master: 10.1.0.10
Master version: 2
Connection Status with Master: UP
MC connection info: CONNECTION SUCCESSFUL
Connected for: 4d02h
External Collector: 10.1.200.1  port: 9995
Route-Control: Enabled
Asymmetric Routing: Disabled
Minimum Mask length: 28
Sampling: off
Channel Unreachable Threshold Timer: 1 seconds
Minimum Packet Loss Calculation Threshold: 15 packets
Minimum Byte Loss Calculation Threshold: 1 bytes
```

```
Monitor cache usage: 2000 (20%) Auto allocated
Minimum Requirement: Met
External Wan interfaces:
     Name: Tunnel100 Interface Index: 15 SNMP Index: 12 SP: MPLS path-id: 1
              Status: UP Zero-SLA: NO Path of Last Resort: Disabled

Auto Tunnel information:


   Name:Tunnel0 if_index: 16
   Virtual Template: Not Configured
   Borders reachable via this tunnel:  10.1.0.12
```

Verification of Remote MC SAF Peering with the Hub MC

Every site's local MC establishes an SAF peering with the Hub MC within a PfR domain. These MCs should peer with the Hub MC when the SAF peering is correctly set up between the remote MC and the Hub MC. The command **show eigrp service-family ipv4** [**vrf** *vrf-name*] **neighbors detail** is used to check the list of SAF neighbors.

Note Do not confuse the overlay routing protocol used (BGP or EIGRP) with PfR's SAF which uses EIGRP to announce PfR services between sites.

The IWAN domain for the book's sample topology includes two central sites (Site 1 and Site 2) and three remote sites (Site 3, Site 4, and Site 5) as well as the local Hub BRs (R11 and R12). Table 9-3 provides the loopback IP address, network site, and PfR role for the routers.

Table 9-3 *SAF Neighbor Table*

Loopback	Site	Role
10.1.0.11	Site 1	BR
10.1.0.12	Site 1	BR
10.2.0.20	Site 2	Transit MC
10.3.0.31	Site 3	Branch MC
10.4.0.41	Site 4	Branch MC
10.5.0.51	Site 5	Branch MC

Example 9-5 verifies the EIGRP neighbors for the Hub MC (R10) at Site 1 for the global routing table. The output provides the list of MCs connected to the Hub MC and the BRs at the hub site. The hold timer and uptime fields are helpful for troubleshooting SAF session stability.

Example 9-5 *PfR Hub MC SAF Neighbors*

```
R10-HUB-MC# show eigrp service-family ipv4 neighbors
EIGRP-SFv4 VR(#AUTOCFG#) Service-Family Neighbors for AS(59501)
H   Address               Interface            Hold Uptime   SRTT   RTO  Q  Seq
                                               (sec)         (ms)       Cnt Num
5   10.4.0.41             Lo0                   572 22:59:45    5   100  0  948
4   10.2.0.20             Lo0                   531 23:00:20    1   100  0  960
3   10.5.0.51             Lo0                   505 23:01:05    5   100  0  2845
2   10.3.0.31             Lo0                   506 23:01:10   12   100  0  949
1   10.1.0.11             Lo0                   568 23:01:28    1   100  0  2
0   10.1.0.12             Lo0                   555 23:01:33    1   100  0  1
```

The PfR EIGRP configuration is automatically generated by PfR and can be viewed with the command **show derived-config** as shown in Example 9-6.

Example 9-6 *PfR Hub MC EIGRP SAF Generated Configuration*

```
R10-HUB-MC# show derived-config | section eigrp
router eigrp #AUTOCFG# (API-generated auto-configuration, not user configurable)
 service-family ipv4 autonomous-system 59501
  sf-interface Ethernet0/0
   shutdown
   hello-interval 120
   hold-time 600
  exit-sf-interface
  !
  sf-interface Ethernet0/1
   shutdown
   hello-interval 120
   hold-time 600
  exit-sf-interface
  !
  sf-interface Loopback0
   hello-interval 120
   hold-time 600
  exit-sf-interface
  !
  topology base
  exit-sf-topology
  remote-neighbors source Loopback0 unicast-listen
 exit-service-family
```

The status of the peering services on the Hub MC is shown with the command **show domain** *domain-name* [**vrf** *vrf-name*] **master peering.** All the services are automatically generated by the Hub MC and then published by the EIGRP SAF framework. The Hub MC publishes all four types of services and subscribes to the Site-Prefix and Globals services from remote branch sites.

Example 9-7 verifies the status of the peering announcements of Site 1's Hub MC (R10) for the global routing table. The output displays the SAF peering services to all remote MCs such as Site-Prefix, Cent-Policy, PMI, and Globals service Publish and Subscription status.

Example 9-7 *PfR Hub MC Peering Status*

```
R10-HUB-MC# show domain IWAN master peering
Peering state: Enabled
Origin:         Loopback0(10.1.0.10)
Peering type:      Listener
Subscribed service:
     cent-policy (2) :
     site-prefix (1) :
       Last Notification Info: 01:37:01 ago, Size: 193, Compressed size: 164,
         Status: Peering Success, Count: 48
     Capability (4) :
       Last Notification Info: 01:36:27 ago, Size: 488, Compressed size: 248,
         Status: Peering Success, Count: 48
     globals (5) :
     pmi (3) :

Published service:
     site-prefix (1) :
       Last Publish Info: 01:02:30 ago, Size: 284, Compressed size: 167,
         Status: Peering Success
     cent-policy (2) :
       Last Publish Info: 23:02:37 ago, Size: 2501, Compressed size: 482,
         Status: Peering Success
     pmi (3) :
       Last Publish Info: 23:02:34 ago, Size: 2645, Compressed size: 493,
         Status: Peering Success
     Capability (4) :
       Last Publish Info: 01:02:28 ago, Size: 419, Compressed size: 211,
         Status: Peering Success
     globals (5) :
       Last Publish Info: 23:02:48 ago, Size: 616, Compressed size: 307,
         Status: Peering Success
```

The list of all discovered sites is shown with the command **show domain** *domain-name* [**vrf** *vrf-name*] **master discovered-sites.** Example 9-8 verifies the list of sites on the Hub MC and gives the number of traffic classes for each DSCP. This command is used to check whether a site is correctly registered in the IWAN domain. Site identifier 255.255.255.255 is the Internet.

Example 9-8 *PfR Hub MC Discovered Sites*

```
R10-HUB-MC# show domain IWAN master discovered-sites

  *** Domain MC DISCOVERED sites ***

  Number of sites:  5
 *Traffic classes [Performance based][Load-balance based]

 Site ID: 10.2.0.20
  Site Discovered:5d14h ago
    Off-limits: Disabled
    DSCP :default[0]-Number of traffic classes[0][0]
    DSCP :af21[18]-Number of traffic classes[0][0]
    DSCP :cs4[32]-Number of traffic classes[0][0]
    DSCP :af41[34]-Number of traffic classes[0][0]
    DSCP :ef[46]-Number of traffic classes[0][0]

 Site ID: 10.3.0.31
  Site Discovered:5d14h ago
    Off-limits: Disabled
    DSCP :default[0]-Number of traffic classes[1][1]
    DSCP :af21[18]-Number of traffic classes[1][0]
    DSCP :cs4[32]-Number of traffic classes[0][0]
    DSCP :af41[34]-Number of traffic classes[0][0]
    DSCP :ef[46]-Number of traffic classes[1][0]

 Site ID: 10.4.0.41
  Site Discovered:5d14h ago
    Off-limits: Disabled
    DSCP :default[0]-Number of traffic classes[0][0]
    DSCP :af21[18]-Number of traffic classes[0][0]
    DSCP :cs4[32]-Number of traffic classes[0][0]
    DSCP :af41[34]-Number of traffic classes[0][0]
    DSCP :ef[46]-Number of traffic classes[1][0]

 Site ID: 10.5.0.51
  Site Discovered:5d14h ago
    Off-limits: Disabled
```

```
   DSCP :default[0]-Number of traffic classes[0][0]
   DSCP :af21[18]-Number of traffic classes[0][0]
   DSCP :cs4[32]-Number of traffic classes[0][0]
   DSCP :af41[34]-Number of traffic classes[0][0]
   DSCP :ef[46]-Number of traffic classes[0][0]

Site ID: 255.255.255.255
 Site Discovered:5d14h ago
   Off-limits: Disabled
   DSCP :default[0]-Number of traffic classes[0][0]
   DSCP :af21[18]-Number of traffic classes[0][0]
   DSCP :cs4[32]-Number of traffic classes[0][0]
   DSCP :af41[34]-Number of traffic classes[0][0]
   DSCP :ef[46]-Number of traffic classes[0][0]
```

At this point of verification, the Hub BRs can generate discovery probes (smart probes) to all remote sites to help them discover their external interfaces and their path names.

Checking the Transit Site

The Transit MC is located at all the transit sites in an IWAN topology. It should run as a standalone platform, on a physical device, or as a virtual machine (CSR 1000V). Checking a transit site is similar to checking the hub site. The only difference is that a Transit MC receives the policies, monitor configurations, and global parameters from the Hub MC.

- The first step is to check its status, locally connected BRs, and associated tunnels. That validates the configuration and reachability of all loopback addresses within the transit site as well as the tunnel path name and identifier definitions.
- A second and very important step is to make sure the path names are the same on all hub and transit sites. Automatic discovery of the external interfaces is not used on a transit site, so external interfaces (DMVPN tunnels), path names, and path identifiers are configured. Path names that are not the same for a specific DMVPN network can create an issue on the branch sites.
- A third check is to ensure that the Transit MC is connected to the hub site to receive the policies and monitor definitions. This peering could be over a DMVPN network or over a DCI link.
- A routing table check similar to the one performed on the hub site has to be done on every transit site. Prefixes have to be correctly advertised in the overlay routing protocol for PfR to work correctly. That includes the Branch MC loopback addresses.

Note PfR does not control traffic between hub and transit sites. Therefore, no channel is created between them.

Check the Branch Site

A branch site is a site where no transit traffic is allowed. The PfR configuration is minimized for the Branch BR because the policies and performance monitors are defined on the Hub MC and the external interfaces are automatically discovered. A Branch MC needs to peer with the Hub MC to get the policies, monitor configurations, and global parameters.

For the book's sample topology:

- R31 is the Branch MC in Site 3. The BR is combined with the MC on the same router.
- R41 is the Branch MC in Site 4. The BR is combined with the MC on the same router.
- R51 is the Branch MC in Site 5 and controls two BRs: R51 (combined with the MC) and R52.

Check the Routing Table

A routing table check similar to the one performed on the hub and transit sites has to be done on every branch site. Prefixes have to be correctly advertised in the overlay routing protocol for PfR to work correctly.

Note Special attention is required for a single CPE branch because all external interfaces are located on the same router. A parent route for each destination prefix must be present in the BGP or EIGRP topology table for each external interface. An external interface may not be listed in the routing table but must be present in the BGP or EIGRP topology table (this happens when one path is preferred over another).

Table 9-4 provides the tunnel next-hop address information with the site name, MC name, prefix, and loopback address for BR R31.

Table 9-4 *R31 Site Prefix Information*

Site	BR Name	Site Prefix	Loopback Address	Tunnel Address
Site 1	R11	10.1.0 0/16 10.0.0.0/8	10.1.0.11	192.168.100.11
Site 1	R12	10.1.0.0/16 10.0.0.0/8	10.1.0.12	192.168.200.12
Site 2	R21	10.2.0.0/16 10.0.0.0/8	10.2.0.21	192.168.100.21
Site 2	R22	10.2.0.0/16 10.0.0.0/8	10.2.0.22	192.168.200.22

Example 9-9 provides the output of the BGP table on R31 with the command **show bgp ipv4 unicast**. The command **show ip eigrp topology** is used to check the EIGRP topology table for routers using EIGRP on the WAN. All site prefixes and loopbacks are listed and are routable over the MPLS and INET tunnels.

Example 9-9 *R31 BGP Topology Table*

```
R31-Spoke# show bgp ipv4 unicast
! Output omitted for brevity
     Network          Next Hop            Metric LocPrf Weight Path
 * i 0.0.0.0          192.168.200.12          10  20000  50000 i
 * i                  192.168.200.22          10    400  50000 i
 *>i                  192.168.100.11          10 100000  50000 i
 * i                  192.168.100.21          10   3000  50000 i
 * i 10.0.0.0         192.168.200.12           0  20000  50000 i
 * i                  192.168.200.22           0    400  50000 i
 *>i                  192.168.100.11           0 100000  50000 i
 * i                  192.168.100.21           0   3000  50000 i
 * i 10.1.0.0/16      192.168.200.12           0  20000  50000 i
 *>i                  192.168.100.11           0 100000  50000 i
 * i 10.2.0.0/16      192.168.200.22           0    400  50000 i
 *>i                  192.168.100.21           0   3000  50000 i
 *>  10.3.0.31/32     0.0.0.0                  0         32768 ?
 *>  10.3.3.0/24      0.0.0.0                  0         32768 ?
```

Check Branch MC Status

The Branch MC status is shown with the command **show domain** *domain-name* [**vrf** *vrf-name*] **master status**. If the default VRF (global routing table) is used, the specific VRF name can be omitted.

Example 9-10 verifies the status of Site 3's single CPE Branch MC (R31) for the global routing table. The output provides the role of the MC (branch), the configured and operational status and the list of the BRs that are controlled (R31, combined with MC on the same router), and whether Transit Site Affinity is enabled.

External interfaces are listed with their corresponding path names and path identifiers. The output also provides the status of load balancing. The field *Minimum Requirement: Met* output ensures that this MC is correctly connected to the Hub MC and has received the policies and Performance Monitor definitions. This check is absolutely critical to make sure the branch site is correctly set up.

Example 9-10 *R31 Branch MC Status*

```
R31-Spoke# show domain IWAN master status
  *** Domain MC Status ***

 Master VRF: Global

  Instance Type:    Branch
  Instance id:      0
  Operational status:  Up
  Configured status:  Up
  Loopback IP Address: 10.3.0.31
  Load Balancing:
   Operational Status: Up
   Max Calculated Utilization Variance: 23%
   Last load balance attempt: 00:00:12 ago
   Last Reason: No more default Traffic Class with non zero bandwidth left move
   Total unbalanced bandwidth:
         External links: 45 Kbps  Internet links: 0 Kbps
  External Collector: 10.1.200.1 port: 9995
  Route Control: Enabled
  Transit Site Affinity: Enabled
  Load Sharing: Enabled
  Mitigation mode Aggressive: Disabled
  Policy threshold variance: 20
  Minimum Mask Length: 28
  Syslog TCA suppress timer: 180 seconds
  Traffic-Class Ageout Timer: 5 minutes
  Minimum Packet Loss Calculation Threshold: 15 packets
  Minimum Bytes Loss Calculation Threshold: 1 bytes
  Minimum Requirement: Met

  Borders:
    IP address: 10.3.0.31
    Version: 2
    Connection status: CONNECTED (Last Updated 1w1d ago )
    Interfaces configured:
      Name: Tunnel200 | type: external | Service Provider: INET | Status: UP |
        Zero-SLA: NO | Path of Last Resort: Disabled
          Number of default Channels: 4

          Path-id list: 0:2 1:2

      Name: Tunnel100 | type: external | Service Provider: MPLS | Status: UP |
        Zero-SLA: NO | Path of Last Resort: Disabled
```

```
          Number of default Channels: 4

          Path-id list: 0:1 1:1

    Tunnel if: Tunnel0
```

Step 1. **Check that the minimum requirement is met.**

Verify whether or not the field *Minimum Requirement* from the command **show domain** *domain-name* [**vrf** *vrf-name*] **master status** is *met*. If the minimum requirement is not met, check the SAF peering; it should correctly peer with the Hub MC. The command **show eigrp service-family ipv4 vrf** [**vrf** *vrf-name*] **neighbors detail** is used to check the list of SAF neighbors.

Example 9-11 illustrates the SAF peering session from the local Branch MC (R31) to the Hub MC (R10). The duration of uptime should be used as a gauge of session stability. An uptime of less than 15 minutes may indicate session instability due to a hold timer of 10 minutes.

Example 9-11 *R31 SAF neighbors*

```
R31-Spoke# show eigrp service-family ipv4 neighbors detail
EIGRP-SFv4 VR(#AUTOCFG#) Service-Family Neighbors for AS(59501)
H   Address                 Interface            Hold Uptime   SRTT   RTO  Q  Seq
                                                 (sec)         (ms)       Cnt Num
0   10.1.0.10               Lo0                   525 23:20:43   10   100  0  119
   Remote Static neighbor (static multihop)
   Version 20.0/4.0, Retrans: 0, Retries: 0, Prefixes: 11
   Topology-ids from peer - 0
   Topologies advertised to peer:   base

Max Nbrs: 65535, Current Nbrs: 0
```

The command **show domain** *domain-name* [**vrf** *vrf-name*] **master peering** helps check that the services are correctly exchanged over the peering session with the Hub MC.

Example 9-12 illustrates the peering services exchanged between the Hub MC (R10) and the local Branch MC (R31). This MC has subscribed to the Policy, Site-Prefix, Capability, and Globals services. The Branch MC also publishes site prefix information and capabilities to the Hub MC. The Capability service is used during establishment of the SAF peering session to communicate to the peer the list of services supported.

Example 9-12 *Branch MC R31 Services*

```
R31-Spoke# show domain IWAN master peering
Peering state: Enabled
Origin:         Loopback0(10.3.0.31)
Peering type:      Listener, Peer(With 10.1.0.10)
Subscribed service:
     cent-policy (2) :
       Last Notification Info: 23:22:16 ago, Size: 2501, Compressed size: 502,
         Status: Peering Success, Count: 2
     site-prefix (1) :
       Last Notification Info: 01:22:21 ago, Size: 284, Compressed size: 187,
         Status: Peering Success, Count: 893
     Capability (4) :
       Last Notification Info: 01:22:19 ago, Size: 419, Compressed size: 231,
         Status: Peering Success, Count: 900
     globals (5) :
       Last Notification Info: 23:22:16 ago, Size: 616, Compressed size: 327,
         Status: Peering Success, Count: 4
Published service:
     site-prefix (1) :
       Last Publish Info: 01:56:55 ago, Size: 193, Compressed size: 145,
         Status: Peering Success
     Capability (4) :
       Last Publish Info: 01:56:53 ago, Size: 457, Compressed size: 229,
         Status: Peering Success
```

Step 2. **Check external interface path names.**

Check that external interfaces are listed with their correct path names. It is very important to make sure that external interfaces are correctly discovered with the correct path name. That means that smart probes are correctly received and decoded by local BRs. This step is also used to check the list of Transit BRs that generate the discovery probes by checking the path ID list. A Transit BR generates discovery probes that include a PfR label. This PfR label is formed as follows: <*POP-ID*>:<*PATH-ID*>.

Table 9-5 provides the label information with the site name, BR name, and path name for the Transit BRs.

Table 9-5 *SAF Neighbor Table*

Label	Site	BR	Path Name
0:1	Site 1	R11	MPLS
0:2	Site 1	R12	INET
1:1	Site 2	R21	MPLS
1:2	Site 2	R22	INET

Note The hub site has a POP ID of 0, and the transit site MC defines that site's POP ID.

If external interfaces are not correctly discovered, smart probes are not correctly received and troubleshooting should be done on the BRs.

Step 3. **Check PfR policies.**

Domain policies are defined on the Hub MC and sent over the SAF infrastructure to all MC peers. Each time domain policies are updated on the Hub MC, they are refreshed and sent over to all MC peers. PfR policies are stored forever on a Branch MC. If the Hub MC fails, the local MC will continue to operate based on its local policies.

If a Branch MC does not receive updates, confirm whether MTU settings are consistent across the path. If the MTU is not consistent, EIGRP SAF packets may be dropped unexpectedly.

The policies received from the Hub MC are displayed with the command **show domain** *domain-name* [**vrf** *vrf-name*] **master policy**.

Example 9-13 shows the policies of Site 3 for the global routing table.

Example 9-13 *R31 Policies*

```
R31-Spoke# show domain IWAN master policy
---------------------------------------------------------------------------------


  class VOICE sequence 10
    path-preference MPLS fallback INET
    class type: Dscp Based
      match dscp ef policy custom
        priority 2 packet-loss-rate threshold 5.0 percent
        priority 1 one-way-delay threshold 150 msec
        priority 2 byte-loss-rate threshold 5.0 percent
        Number of Traffic classes using this policy: 2

  class VIDEO sequence 20
    path-preference MPLS fallback INET
    class type: Dscp Based
      match dscp af41 policy custom
        priority 2 packet-loss-rate threshold 5.0 percent
        priority 1 one-way-delay threshold 150 msec
        priority 2 byte-loss-rate threshold 5.0 percent
      match dscp cs4 policy custom
        priority 2 packet-loss-rate threshold 5.0 percent
        priority 1 one-way-delay threshold 150 msec
        priority 2 byte-loss-rate threshold 5.0 percent

  class CRITICAL sequence 30
    path-preference MPLS fallback INET
    class type: Dscp Based
      match dscp af21 policy custom
        priority 2 packet-loss-rate threshold 10.0 percent
        priority 1 one-way-delay threshold 600 msec
        priority 2 byte-loss-rate threshold 10.0 percent

  class default
      match dscp all
        Number of Traffic classes using this policy: 2
```

Check the Branch BR

The status of a Branch BR is shown with the command **show domain** *domain-name* [**vrf** *vrf-name*] **border status**.

Example 9-14 provides the status of a BR in Site 3 (R31) for the global routing table. In this example, the MC and BR are colocated in R31, but each component is completely independent, and a session is established between the BR and the MC. Both external interfaces (DMVPN tunnels) are on the same router.

Example 9-14 *Branch BR R31 Status*

```
R31-Spoke# show domain IWAN border status
--------------------------------------------------------------------
  **** Border Status ****

Instance Status: UP
Present status last updated: 1w1d ago
Loopback: Configured Loopback0 UP (10.3.0.31)
Master: 10.3.0.31
Master version: 2
Connection Status with Master: UP
MC connection info: CONNECTION SUCCESSFUL
```

```
Connected for: 1w1d
External Collector: 10.1.200.1  port: 9995
Route-Control: Enabled
Asymmetric Routing: Disabled
Minimum Mask length: 28
Sampling: off
Channel Unreachable Threshold Timer: 1 seconds
Minimum Packet Loss Calculation Threshold: 15 packets
Minimum Byte Loss Calculation Threshold: 1 bytes
Monitor cache usage: 2000 (20%) Auto allocated
Minimum Requirement: Met
External Wan interfaces:
     Name: Tunnel200 Interface Index: 16 SNMP Index: 13 SP: INET Status: UP
       Zero-SLA: NO Path of Last Resort: Disabled Path-id List: 0:2, 1:2
     Name: Tunnel100 Interface Index: 15 SNMP Index: 12 SP: MPLS Status: UP
       Zero-SLA: NO Path of Last Resort: Disabled Path-id List: 0:1, 1:1

Auto Tunnel information:

   Name:Tunnel0 if_index: 17
   Virtual Template: Not Configured
   Borders reachable via this tunnel:
```

Step 1. **Check that the minimum requirement is met.**

Verify whether or not the field *Minimum Requirement* from the command **show domain** *domain-name* [**vrf** *vrf-name*] **master status** is *met*. If the minimum requirement is not met, check the SAF peering; it should correctly peer with the local Branch MC. The command **show eigrp service-family ipv4 vrf** [**vrf** *vrf-name*] **neighbors detail** is used to check the list of SAF neighbors.

Step 2. **Check external interface path names.**

Check that the WAN tunnel interfaces are identified with their correct path names. That means that smart probes are correctly received and decoded by the local Branch BRs. Check that the PfR path names are correct. If interfaces and/or path names are not listed, check that smart probes are received from the hub. By default, discovery probes (smart probes) are forged and generated from the Transit BRs with the following parameters:

- **Source address:** Loopback address of the MC on the source site
- **Destination address:** Loopback address of the MC on the destination site
- **Source port:** 18000
- **Destination port:** 19000

The discovery probe definition requires that the Branch MC loopback addresses be announced and routable from the Transit BRs (R11, R12, R21, and R22) as explained earlier in the section “Check the Routing Table.”

There are several ways to troubleshoot whether the discovery probes are correctly received over an interface, but these tools are beyond the scope of this chapter. Example 9-15 illustrates the use of an access list to check whether the *smart probes packets (SMP)* are correctly received on every external interface. This access list can be used in a conditional debug.

Example 9-15 *R31 Access List to Match SMP Packets*

```
R31-Spoke
access-list 100 permit udp any eq 18000 any eq 19000
```

Embedded Packet Capture (EPC) can also be used to capture the smart probes packets on the external interface on either the source or destination BR.

Step 3. **Check that monitor specifications are received from the Hub MC.**

PfR leverages the AVC infrastructure with the use of Performance Monitor for passive performance monitoring. Performance Monitor instances (PMIs) are received from the Hub MC to the local MC and then forwarded to the BRs. These include four possible monitors (one is optional):

- **Ingress-per-DSCP:** PMI applied over the DMVPN tunnels on ingress to measure the channel performance.
- **Ingress-per-DSCP-quick:** PMI applied over the DMVPN tunnels on ingress to measure the channel performance. This is used for DSCP configured with a monitor interval different from the default (shorter value) and used for fast failover (see Chapter 8, “PfR Provisioning”).
- **Egress-aggregate:** PMI applied over the DMVPN tunnels on egress to collect the bandwidth per traffic class.
- **Egress-prefix-learn:** PMI applied over the DMVPN tunnels on egress to collect the source prefixes.

The command **show domain** *domain-name* [**vrf** *vrf-name*] **border pmi** displays the Performance Monitor configurations.

Example 9-16 displays the PMIs as well as the tunnels over which they are applied.

Example 9-16 *BR R31 PMIs*

```
R31-Spoke# show domain IWAN border pmi
****CENT PMI INFORMATION****

Ingress policy CENT-Policy-Ingress-0-2:
Ingress policy activated on:
  Tunnel200 Tunnel100
-------------------------------------------------------------------------
PMI[Ingress-per-DSCP]-FLOW MONITOR[MON-Ingress-per-DSCP-0-0-0]
    monitor-interval:30
    key-list:
      pfr site source id ipv4
      pfr site destination id ipv4
      ip dscp
      interface input
      policy performance-monitor classification hierarchy
      pfr label identifier
    Non-key-list:
      transport packets lost rate
      transport bytes lost rate
      pfr one-way-delay
      network delay average
      transport rtp jitter inter arrival mean
      counter bytes long
      counter packets long
      timestamp absolute monitoring-interval start
    DSCP-list:N/A

    Exporter-list:
      10.1.200.1
-------------------------------------------------------------------------
PMI[Ingress-per-DSCP-quick ]-FLOW MONITOR[MON-Ingress-per-DSCP-quick -0-0-1]
    monitor-interval:2
    key-list:
     pfr site source id ipv4
     pfr site destination id ipv4
     ip dscp
     interface input
     policy performance-monitor classification hierarchy
     pfr label identifier
   Non-key-list:
      transport packets lost rate
      transport bytes lost rate
      pfr one-way-delay
      network delay average
      transport rtp jitter inter arrival mean
      counter bytes long
      counter packets long
      timestamp absolute monitoring-interval start
    DSCP-list:
      ef-[class:CENT-Class-Ingress-DSCP-ef-0-2]
        packet-loss-rate:react_id[2]-priority[2]-threshold[5.0 percent]
        one-way-delay:react_id[3]-priority[1]-threshold[150 msec]
        network-delay-avg:react_id[4]-priority[1]-threshold[300 msec]
        byte-loss-rate:react_id[5]-priority[2]-threshold[5.0 percent]
      af41-[class:CENT-Class-Ingress-DSCP-af41-0-3]
        packet-loss-rate:react_id[6]-priority[2]-threshold[5.0 percent]
        one-way-delay:react_id[7]-priority[1]-threshold[150 msec]
        network-delay-avg:react_id[8]-priority[1]-threshold[300 msec]
        byte-loss-rate:react_id[9]-priority[2]-threshold[5.0 percent]
      cs4-[class:CENT-Class-Ingress-DSCP-cs4-0-4]
        packet-loss-rate:react_id[10]-priority[2]-threshold[5.0 percent]
        one-way-delay:react_id[11]-priority[1]-threshold[150 msec]
        network-delay-avg:react_id[12]-priority[1]-threshold[300 msec]
        byte-loss-rate:react_id[13]-priority[2]-threshold[5.0 percent]
      af21-[class:CENT-Class-Ingress-DSCP-af21-0-5]
        packet-loss-rate:react_id[14]-priority[2]-threshold[10.0 percent]
        one-way-delay:react_id[15]-priority[1]-threshold[600 msec]
        network-delay-avg:react_id[16]-priority[1]-threshold[1200 msec]
        byte-loss-rate:react_id[17]-priority[2]-threshold[10.0 percent]

    Exporter-list:None
-------------------------------------------------------------------------


Egress policy CENT-Policy-Egress-0-3:
Egress policy activated on:
  Tunnel200 Tunnel100
-------------------------------------------------------------------------
PMI[Egress-aggregate]-FLOW MONITOR[MON-Egress-aggregate-0-0-2]
    monitor-interval:30
    Trigger Nbar:No
    minimum-mask-length:28
    key-list:
      ipv4 destination prefix
      ipv4 destination mask
      pfr site destination prefix ipv4
      pfr site destination prefix mask ipv4
      ip dscp
      interface output
    Non-key-list:
      timestamp absolute monitoring-interval start
      counter bytes long
      counter packets long
      ip protocol
      pfr site destination id ipv4
      pfr site source id ipv4
      pfr br ipv4 address
      interface output physical snmp
    DSCP-list:N/A
    Class:CENT-Class-Egress-ANY-0-6

    Exporter-list:
      10.3.0.31
      10.1.200.1
-------------------------------------------------------------------------
PMI[Egress-prefix-learn]-FLOW MONITOR[MON-Egress-prefix-learn-0-0-3]
    monitor-interval:30
    minimum-mask-length:28
    key-list:
      ipv4 source prefix
      ipv4 source mask
      routing vrf input
    Non-key-list:
      counter bytes long
      counter packets long
      timestamp absolute monitoring-interval start
      interface input
    DSCP-list:N/A
    Class:CENT-Class-Egress-ANY-0-6

    Exporter-list:
      10.3.0.31
```

Step 4. **Check that performance monitors are correctly applied.**

Check that Performance Monitors are correctly applied on the external interfaces, on ingress and egress. Example 9-16 displays where the PMIs are applied. This is key to determining whether they are correctly applied on all external tunnel interfaces.

Monitoring Operations

The previous sections described how to check that all sites are correctly registered in the IWAN domain, that policies are received on all MCs, and that PMIs are correctly applied to the tunnels.

This section explains the operational aspects of PfR: how it tracks path health, the logic for path selection, and how PfR implements intelligent path control.

Routing Table

PfR overrides the routing table information with a new interface and next hop. The MC makes a decision by comparing the real-time metrics with the policies and instructs the BRs to use the appropriate path. Checking routing information is usually the very first step in a troubleshooting or monitoring workflow. This could be very misleading because the actual next hop may not be the one displayed in the routing table. To address this concern, a new flag (‘*p*’) has been added in the **show ip route** [**vrf** *vrf-name*] to indicate that a specific prefix can actually be controlled by PfR.

Example 9-17 provides the output of the routing table on BR R31 from the book’s sample topology and illustrates the use of the ‘*p*’ flag.

Example 9-17 *BR R31 Routing Table*

```
R31-Spoke# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       a - application route
       + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is 192.168.100.11 to network 0.0.0.0

B*    0.0.0.0/0 [19/10] via 192.168.100.11, 2w5d
    p 10.0.0.0/8 is variably subnetted, 8 subnets, 4 masks
B   p    10.0.0.0/8 [19/0] via 192.168.100.11, 2w5d
B   p    10.1.0.0/16 [19/0] via 192.168.100.11, 2w5d
B   p    10.2.0.0/16 [19/0] via 192.168.100.21, 2w5d
C   p    10.3.0.31/32 is directly connected, Loopback0
C   p    10.3.3.0/24 is directly connected, Ethernet1/0
L   p    10.3.3.31/32 is directly connected, Ethernet1/0
H   p    10.4.0.41/32 [250/255] via 192.168.200.41, 6d10h, Tunnel200
                      [250/255] via 192.168.100.41, 6d12h, Tunnel100
```

```
H   p    10.4.4.0/24 [250/255] via 192.168.100.41, 6d12h, Tunnel100
      192.168.100.0/24 is variably subnetted, 3 subnets, 2 masks
C        192.168.100.0/24 is directly connected, Tunnel100
L        192.168.100.31/32 is directly connected, Tunnel100
H        192.168.100.41/32 is directly connected, 6d12h, Tunnel100
      192.168.200.0/24 is variably subnetted, 3 subnets, 2 masks
C        192.168.200.0/24 is directly connected, Tunnel200
L        192.168.200.31/32 is directly connected, Tunnel200
H        192.168.200.41/32 is directly connected, 6d10h, Tunnel200
```

Monitor the Site Prefix

The PfR site prefix is the database infrastructure for *inside* prefixes for every site. Every local site learns the site prefix itself from the egress performance monitor on the external interface, then publishes across all sites over the EIGRP SAF framework. Each local site subscribes to all remote Site-Prefix services as well, so all sites share one synchronized prefix database.

There are four different prefix types in the site prefix database:

- **Local site prefix with flag '*L*'**: Locally learned prefix that is learned by the egress site prefix monitor
- **Local site prefix with flag '*C*'**: Site prefix configured by the static site prefix list, mostly on the transit site
- **Remote site prefix with flag '*S*'**: Site prefix learned from remote EIGRP SAF neighbors
- **Enterprise prefix with '*T*'**: Summary prefix that defines the enterprise site prefix boundary

The command **show domain** *domain-name* [**vrf** *vrf-name*] **master site-prefix** displays the list of prefixes advertised from all MCs and the associated site ID. The site ID is the IP address of that site's MC interface which is Loopback0 in this book's configuration.

Example 9-18 illustrates all the site prefixes of the book's topology. Prefixes 10.1.0.0/16 and 10.2.0.0/16 are advertised from both Site 1 (10.1.0.10) and Site 2 (10.2.0.20). Notice that these entries were defined statically with a site prefix list on the site's corresponding MC, and that they were learned remotely (from R41's perspective).

Example 9-18 *Branch BR R41 Site Prefix Database*

```
R41-Spoke# show domain IWAN master site-prefix
  Change will be published between 5-60 seconds
  Next Publish 01:42:19 later
  Prefix DB Origin: 10.4.0.41
  Last publish Status : Peering Success
  Total publish errors : 0
  Prefix Flag: S-From SAF; L-Learned; T-Top Level; C-Configured; M-shared

Site-id                Site-prefix            Last Updated          DC Bitmap  Flag
--------------------------------------------------------------------------------
10.1.0.10               10.1.0.10/32           00:28:04 ago          0x1          S
10.1.0.10               10.1.0.0/16            00:22:45 ago          0x3          S,C,M
10.2.0.20               10.1.0.0/16            00:22:45 ago          0x3          S,C,M
10.2.0.20               10.2.0.20/32           00:22:45 ago          0x2          S
10.1.0.10               10.2.0.0/16            00:22:45 ago          0x3          S,C,M
10.2.0.20               10.2.0.0/16            00:22:45 ago          0x3          S,C,M
10.3.0.31               10.3.0.31/32           00:20:15 ago          0x0          S
10.3.0.31               10.3.3.0/24            00:20:15 ago          0x0          S
10.4.0.41               10.4.0.41/32           5d16h    ago          0x0          L
10.4.0.41               10.4.4.0/24            00:00:06 ago          0x0          L
10.5.0.51               10.5.0.51/32           00:22:24 ago          0x0          S
255.255.255.255        *10.0.0.0/8             00:28:04 ago          0x1          S,T
```

The site prefix monitor is automatically enabled on BRs. The command **show domain** *domain-name* **border pmi | section prefix-learn** is issued on a BR to display the PMI used to collect the site prefix information.

Example 9-19 illustrates the PMI *Egress-prefix-learn*. The source prefix and source mask are defined as key fields.

Example 9-19 *Branch BR R41 Site Prefix PMI*

```
R41-Spoke# show domain IWAN border pmi | sec prefix-learn
PMI[Egress-prefix-learn]-FLOW MONITOR[MON-Egress-prefix-learn-0-0-5]
    monitor-interval:30
    minimum-mask-length:28
    key-list:
      ipv4 source prefix
      ipv4 source mask
      routing vrf input
    Non-key-list:
      counter bytes long
      counter packets long
      timestamp absolute monitoring-interval start
      interface input
    DSCP-list:N/A
    Class:CENT-Class-Egress-ANY-0-7
```

Monitor Traffic Classes

PfR manages aggregation of flows called *traffic classes (TCs)*. A traffic class is an aggregation of flows going to the same destination prefix, with the same DSCP or application name (if application-based policies are used). TCs are learned on the BRs by monitoring a WAN interface's egress traffic. This is based on a PMI applied on the external interface of every local BR. Information relative to a TC is sent to the local MC.

The command **show domain** *domain-name* [**vrf** *vrf-name*] **border pmi | section Egress-aggregate** gives the configuration of the PMI used to collect the TC information.

Example 9-20 illustrates the PMI *Egress-aggregate*. Key field definition allows the creation of TCs with the corresponding metrics.

Example 9-20 *Branch BR R31 Site Prefix PMI*

```
R31-Spoke# show domain IWAN border pmi | section Egress-aggregate
PMI[Egress-aggregate]-FLOW MONITOR[MON-Egress-aggregate-0-0-4]
    monitor-interval:30
    Trigger Nbar:No
    minimum-mask-length:28
    key-list:
      ipv4 destination prefix
      ipv4 destination mask
      pfr site destination prefix ipv4
      pfr site destination prefix mask ipv4
      ip dscp
      interface output
    Non-key-list:
      timestamp absolute monitoring-interval start
      counter bytes long
      counter packets long
      ip protocol
      pfr site destination id ipv4
      pfr site source id ipv4
      pfr br ipv4 address
      interface output physical snmp
    DSCP-list:N/A
    Class:CENT-Class-Egress-ANY-0-7
```

The TC summary provides a summary view of all traffic and how it is controlled (as well as its current path) in the PfR domain. The possible states for a TC are

- **UNCONTROLLED:** No parent route is found.
- **CONTROLLED:** A path that meets the criteria has been found.
- **OUT OF POLICY:** No path meets the criteria set in the policy.

The command **show domain** *domain-name* [**vrf** *vrf-name*] **master traffic-classes summary** displays a summary view of all TCs on a specific MC.

Example 9-21 displays the summary list of TCs on Branch MC R31:

- The first column gives the destination prefix.
- The second column is the destination site.
- The third column gives the application name if application policies are used.
- The fourth column gives the DSCP value.
- The fifth column is the TC identifier (TC ID).
- The sixth column gives the state of the TC.
- The last column gives the path used by a TC.

Example 9-21 *Branch MC R31 TC Summary*

```
R31-Spoke# show domain IWAN master traffic-classes summary

APP - APPLICATION, TC-ID - TRAFFIC-CLASS-ID, APP-ID - APPLICATION-ID
SP - SERVICE PROVIDER, PC = PRIMARY CHANNEL ID,
BC - BACKUP CHANNEL ID, BR - BORDER, EXIT - WAN INTERFACE
UC - UNCONTROLLED, PE - PICK-EXIT, CN - CONTROLLED, UK - UNKNOWN

Dst-Site-Pfx      Dst-Site-Id       State DSCP      TC-ID APP-ID    APP
  Current-Exit


10.4.4.0/24       10.4.0.41         CN    ef[46]    1     N/A       N/A
  MPLS(0:0|0:0)/10.3.0.31/Tu100(Ch:6)
10.1.0.0/16       10.1.0.10         CN    default[0 3     N/A       N/A
  INET(0:2|0:0)/10.3.0.31/Tu200(Ch:1)
10.1.0.0/16       10.1.0.10         CN    ef[46]    2     N/A       N/A
  MPLS(0:1|0:0)/10.3.0.31/Tu100(Ch:8)
10.10.0.0/16      10.1.0.10         CN    default[0 5     N/A       N/A
  INET(0:2|0:0)/10.3.0.31/Tu200(Ch:1)
10.10.0.0/16      10.1.0.10         CN    af21[18]  4     N/A       N/A
  MPLS(0:1|0:0)/10.3.0.31/Tu100(Ch:10)
 Total Traffic Classes: 5 Site: 5  Internet: 0
```

From there, traffic can be viewed by examining individual TCs on the MC. The list of TCs is shown with the command **show domain** *domain-name* [**vrf** *vrf-name*] **master traffic-classes**. This command allows filtering of the output based on DSCP, destination site, destination prefix, path used, and a few more criteria.

Example 9-22 displays the detailed list of TCs on Branch MC R31 for DSCP value EF (voice traffic). It shows a first TC for destination prefix 10.4.4.0/24 which is Site 4 (spoke-to-spoke voice traffic) and another TC for destination prefix 10.1.0.0/16 which is Site 1 (spoke-to-hub traffic). It gives the state (controlled) of every TC and the current performance status (in-policy). Any TC performance issues are listed in the TC route change history (last five route change reasons).

Example 9-22 *Branch MC R31 TC Details*

```
R31-Spoke# show domain IWAN master traffic-classes dscp ef

 Dst-Site-Prefix: 10.4.4.0/24         DSCP: ef [46] Traffic class id:1
  Clock Time:                 22:00:26 (CET) 07/25/2016
  TC Learned:                 00:09:55 ago
  Present State:              CONTROLLED
  Current Performance Status: in-policy
  Current Service Provider:   MPLS since 00:09:24
  Previous Service Provider:  Unknown
  BW Used:                    23 Kbps
  Present WAN interface:      Tunnel100 in Border 10.3.0.31
  Present Channel (primary):  6 MPLS pfr-label:0:0 | 0:0 [0x0]
  Backup Channel:             5 INET pfr-label:0:0 | 0:0 [0x0]
  Destination Site ID bitmap: 0
  Destination Site ID:        10.4.0.41
  Class-Sequence in use:      10
  Class Name:                 VOICE using policy User-defined
    priority 2 packet-loss-rate threshold 5.0 percent
    priority 1 one-way-delay threshold 150 msec
    priority 2 byte-loss-rate threshold 5.0 percent
  BW Updated:                 00:00:25 ago
  Reason for Latest Route Change:    Uncontrolled to Controlled Transition
  Route Change History:
             Date and Time                   Previous Exit
  Current Exit                                Reason

    1: 21:51:02 (CET) 07/25/16   None(0:0|0:0)/0.0.0.0/None (Ch:0)
  MPLS(0:0|0:0)/10.3.0.31/Tu100 (Ch:6)              Uncontrolled to Controlled
  Transition
--------------------------------------------------------------------------------

 Dst-Site-Prefix: 10.1.0.0/16         DSCP: ef [46] Traffic class id:2
  Clock Time:                 22:00:26 (CET) 07/25/2016
  TC Learned:                 00:09:55 ago
  Present State:              CONTROLLED
  Current Performance Status: in-policy
  Current Service Provider:   MPLS since 00:04:49
  Previous Service Provider:  INET pfr-label: 0:2 | 0:0 [0x20000] for 181 sec
  BW Used:                    24 Kbps
  Present WAN interface:      Tunnel100 in Border 10.3.0.31
  Present Channel (primary):  8 MPLS pfr-label:0:1 | 0:0 [0x10000]
  Backup Channel:             7 INET pfr-label:0:2 | 0:0 [0x20000]
  Destination Site ID bitmap: 1
  Destination Site ID:        10.1.0.10
  Class-Sequence in use:      10
  Class Name:                 VOICE using policy User-defined
    priority 2 packet-loss-rate threshold 5.0 percent
    priority 1 one-way-delay threshold 150 msec
    priority 2 byte-loss-rate threshold 5.0 percent
  BW Updated:                 00:00:25 ago
  Reason for Latest Route Change:    Backup to Primary path preference transition
  Route Change History:
             Date and Time                    Previous Exit
  Current Exit                                 Reason


    1:  21:55:37 (CET) 07/25/16   INET(0:2|0:0)/10.3.0.31/Tu200 (Ch:7)
          MPLS(0:1|0:0)/10.3.0.31/Tu100 (Ch:8)               Backup to Primary path
          preference transition
    2:  21:52:35 (CET) 07/25/16   MPLS(0:1|0:0)/10.3.0.31/Tu100 (Ch:8)
          INET(0:2|0:0)/10.3.0.31/Tu200 (Ch:7)               Out-of-Policy (One Way
          Delay : 605 msec)
    3:  21:51:02 (CET) 07/25/16   None(0:0|0:0)/0.0.0.0/None (Ch:0)
          MPLS(0:1|0:0)/10.3.0.31/Tu100 (Ch:8)               Uncontrolled to
          Controlled Transition
--------------------------------------------------------------------------------
Total Traffic Classes: 2 Site: 2  Internet: 0
```

The TC to Site 1 has a primary channel which is *MPLS* with a PfR label "*0:1 | 0:0*," and a backup channel which is *INET* with a PfR label "*0:2 | 0:0*." A PfR label uniquely identifies a path between sites across DMVPN networks (embedded in GRE encapsulation).

Figure 9-2 displays the PfR label construct. A channel to a transit site has a PfR label in the form <*POP-ID:PATH-ID | 0:0*>. A channel from spoke to spoke has a PfR label in the form <*0:0 | 0:0*>.

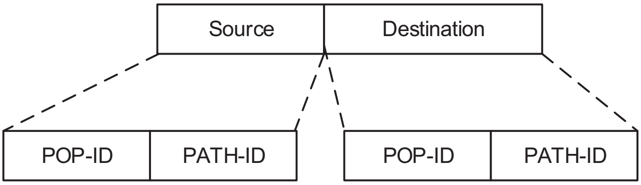


Figure 9-2 *PfR Label Format*

A reexamination of Example 9-22 shows a TC for 10.1.0.0/16 with

- A primary channel over MPLS with a PfR label "*0:1 | 0:0*," which indicates Site 1 (*POP-ID 0*) and R11 (*PATH-ID 1*)
- A backup channel over INET with a PfR label "*0:2 | 0:0*," which indicates Site 1 (*POP-ID 0*) and R12 (*PATH-ID 2*)

This output shows that this TC is forwarded over the MPLS network to R11 on Site 1. A backup path is INET to R12 on Site 1.

All TCs are synchronized to the BR that enforces the path based on the TC database. The list of TCs on a BR is shown with the command **show** *domain-name* [**vrf** *vrf-name*] **border traffic-classes.**

Example 9-23 displays the list of TCs on Branch BR R31. This output displays the primary channel (local or remote) and the next-hop information (with the routing protocol used to get this information).

Example 9-23 *Branch BR R31 TC*

```
R31-Spoke# show domain IWAN border traffic-classes

 Src-Site-Prefix: ANY Dst-Site-Prefix: 10.10.0.0/16
  DSCP: af21 [18] Traffic class id: 4
  TC Learned: 00:14:58 ago
  Present State: CONTROLLED
  Destination Site ID: 10.1.0.10
  If_index: 11
  Primary chan id: 10
  Primary chan Presence: LOCAL CHANNEL
  Primary interface: Tunnel100
  Primary Nexthop: 192.168.100.11 (BGP)
  Backup chan id: 9
  Backup chan Presence: LOCAL CHANNEL
  Backup interface: Tunnel200
--------------------------------------------------------------------
```

```
 Src-Site-Prefix: ANY Dst-Site-Prefix: 10.1.0.0/16
  DSCP: default [0] Traffic class id: 3
  TC Learned: 00:14:58 ago
  Present State: CONTROLLED
  Destination Site ID: 10.1.0.10
  If_index: 12
  Primary chan id: 1
  Primary chan Presence: LOCAL CHANNEL
  Primary interface: Tunnel200
  Primary Nexthop: 192.168.200.12 (BGP)
  Backup chan id: 4
  Backup chan Presence: LOCAL CHANNEL
  Backup interface: Tunnel100
--------------------------------------------------------------------


 Src-Site-Prefix: ANY Dst-Site-Prefix: 10.1.0.0/16
  DSCP: ef [46] Traffic class id: 2
  TC Learned: 00:14:58 ago
  Present State: CONTROLLED
  Destination Site ID: 10.1.0.10
  If_index: 11
  Primary chan id: 8
  Primary chan Presence: LOCAL CHANNEL
  Primary interface: Tunnel100
  Primary Nexthop: 192.168.100.11 (BGP)
  Backup chan id: 7
  Backup chan Presence: LOCAL CHANNEL
  Backup interface: Tunnel200
--------------------------------------------------------------------


 Src-Site-Prefix: ANY Dst-Site-Prefix: 10.4.4.0/24
  DSCP: ef [46] Traffic class id: 1
  TC Learned: 00:14:58 ago
  Present State: CONTROLLED
  Destination Site ID: 10.4.0.41
  If_index: 11
  Primary chan id: 6
  Primary chan Presence: LOCAL CHANNEL
  Primary interface: Tunnel100
  Primary Nexthop: 192.168.100.41 (NHRP)
  Backup chan id: 5
  Backup chan Presence: LOCAL CHANNEL
  Backup interface: Tunnel200
--------------------------------------------------------------------
```

Monitor Channels

A *channel* is a logical construct used in PfR to keep track of the next-hop reachability and collect the performance metrics per DSCP. A channel is added every time a new DSCP, interface, or site is added to the prefix database, or when a new smart probe is received. With the transit site support, a channel is created per DSCP and per next hop.

The command **show domain** *domain-name* [**vrf** *vrf-name*] **master channel summary** is used on an MC to display a summary state of all channels on a specific site. This command is executed on the local MC and is used to quickly check that a channel is available for a specific DSCP to a specific destination site.

Example 9-24 displays a summary of all channels available on Site 3 (R31 is the MC). For example, channel 53 gives the following information:

- The channel is for traffic with DSCP EF.
- The channel is for traffic to destination Site 1 (10.1.0.10).
- The channel is to R11 because the PfR label is "*0:1 | 0:0*," which stands for *POP-ID 0* (Site1) and *PATH-ID 1* (R11).
- The channel is available.

Example 9-24 *Branch MC Router R31*

```
R31-Spoke# show domain IWAN master channels summary
Ch-ID - Channel ID, SP - Service Provider
TCA - counts for Received/Processed/Unreachable
A - Available, UA - Un-Available

Ch-ID Dst-Site-ID     DSCP    SP    pfr-Label             Status TCA

62    10.1.0.10       af21    INET  0:2 | 0:0 [0x20000]   A      52/31/0
54    10.4.0.41       ef      INET  0:0 | 0:0 [0x0]       A      820/421/2
59    10.2.0.20       default INET  1:2 | 0:0 [0x1020000] A      1/1/1
58    10.2.0.20       ef      MPLS  1:1 | 0:0 [0x1010000] A      985/514/0
60    10.2.0.20       ef      INET  1:2 | 0:0 [0x1020000] A      1733/927/0
55    Internet        af21    INET  0:2 | 0:0 [0x20000]   A      0/0/0
43    Internet        af21    MPLS  1:1 | 0:0 [0x1010000] A      0/0/0
53    10.1.0.10       ef      MPLS  0:1 | 0:0 [0x10000]   A      60/31/0
35    10.4.0.41       ef      MPLS  0:0 | 0:0 [0x0]       A      17/17/17
61    Internet        af21    INET  1:2 | 0:0 [0x1020000] A      0/0/0
7     10.2.0.20       default MPLS  1:1 | 0:0 [0x1010000] A      0/0/0
56    10.1.0.10       default INET  0:2 | 0:0 [0x20000]   A      61/58/61
15    10.1.0.10       default MPLS  0:1 | 0:0 [0x10000]   A      113/106/113
42    Internet        af21    MPLS  0:1 | 0:0 [0x10000]   A      0/0/0
52    10.1.0.10       af21    MPLS  0:1 | 0:0 [0x10000]   A      4/2/0
57    10.1.0.10       ef      INET  0:2 | 0:0 [0x20000]   A      3390/1805/0
```

The command **show domain** *domain-name* [**vrf** *vrf-name*] **master channel** is used on an MC to list all channels with more details. Filters based on DSCP, destination site, or path name can also be used.

Example 9-25 displays the channels for DSCP EF on Branch MC R31 when voice traffic with DSCP EF is forwarded from Site 3 to 10.1.0.0/16 which is advertised from Site 1.

- Actual voice traffic is initially forwarded on MPLS based on the routing table (see Example 9-17).
- PfR takes control of this traffic, and a channel for DSCP EF is therefore created for R11 (MPLS transport).
- A channel is also created on the secondary path for R12 (INET transport).
- The site prefix list shown here gives the status of the prefix. 10.1.0.0/16 is seen as *active* on both MPLS and INET paths, which means both can be used for traffic forwarding. State *routable* means that there is a route to that prefix but no traffic has been seen. A state *standby* would mean that the channel is a secondary choice when active channels are out of policy.
- Path preference for EF traffic is MPLS, and therefore the voice traffic will be forwarded over MPLS as a primary path and then fall back to INET if the preferred path is out of policy.
- On-demand export (ODE) gives the last two buckets of performance metrics. ODE statistics are acquired when an MC receives a TC from a destination site and then asks the destination MC to send the performance statistics for the affected channel.
- TCA statistics give the accumulated number of TCAs because of loss, delay, and jitter as well as unreachable status. It also lists the latest TCA received.

Example 9-25 *Branch MC Router R31*

```
R31-Spoke# show domain IWAN master channels dscp ef
   Legend: * (Value obtained from Network delay:)

Channel Id: 53  Dst Site-Id: 10.1.0.10  Link Name: MPLS  DSCP: ef [46] pfr-label:
  0:1 | 0:0 [0x10000] TCs: 1
  Channel Created: 6d12h ago
  Provisional State: Initiated and open
  Operational state: Available
  Channel to hub: TRUE
  Interface Id: 15
  Supports Zero-SLA: Yes
  Muted by Zero-SLA: No
  Estimated Channel Egress Bandwidth: 1 Kbps
  Immitigable Events Summary:
   Total Performance Count: 0, Total BW Count: 0
  Site Prefix List
    10.1.0.10/32 (Routable)
    10.1.0.0/16 (Active)
    10.2.0.0/16 (Routable)
   ODE Statistics:
      Received: 2600
  ODE Stats Bucket Number: 1
   Last Updated  : 00:13:36 ago
    Packet Count  : 80
    Byte Count    : 4872
    One Way Delay : 6 msec*
    Loss Rate Pkts: 0.0 %
    Loss Rate Byte: 0.0 %
    Jitter Mean   : 7000 usec
    Unreachable   : FALSE
  ODE Stats Bucket Number: 2
   Last Updated  : 00:15:40 ago
    Packet Count  : 81
    Byte Count    : 4952
    One Way Delay : 3 msec*
    Loss Rate Pkts: 0.0 %
    Loss Rate Byte: 0.0 %
    Jitter Mean   : 666 usec
    Unreachable   : FALSE
   TCA Statistics:
      Received: 58 ; Processed: 30 ; Unreach_rcvd: 0 ; Local Unreach_rcvd: 0
      TCA lost byte rate: 0
      TCA lost packet rate: 58
      TCA one-way-delay: 0
      TCA network-delay: 0
      TCA jitter mean: 0
  Latest TCA Bucket
   Last Updated  : 14:29:34 ago
    One Way Delay : NA
    Loss Rate Pkts: 5.12 %
    Loss Rate Byte: NA
    Jitter Mean   : NA
    Unreachability: FALSE

Channel Id: 57  Dst Site-Id: 10.1.0.10  Link Name: INET  DSCP: ef [46] pfr-label:
  0:2 | 0:0 [0x20000] TCs: 0
  Channel Created: 6d12h ago
  Provisional State: Initiated and open
  Operational state: Available
  Channel to hub: TRUE
```

```
Interface Id: 16
Supports Zero-SLA: Yes
Muted by Zero-SLA: No
Estimated Channel Egress Bandwidth: 0 Kbps
Immitigable Events Summary:
 Total Performance Count: 0, Total BW Count: 0
Site Prefix List
  10.1.0.10/32 (Routable)
  10.1.0.0/16 (Active)
  10.2.0.0/16 (Routable)
 ODE Statistics:
    Received: 2834
ODE Stats Bucket Number: 1
 Last Updated  : 00:13:38 ago
  Packet Count  : 34
  Byte Count    : 2856
  One Way Delay : 4 msec*
  Loss Rate Pkts: 8.10 %
  Loss Rate Byte: 0.0 %
  Jitter Mean   : 3529 usec
  Unreachable   : FALSE
ODE Stats Bucket Number: 2
 Last Updated  : 00:15:42 ago
  Packet Count  : 36
  Byte Count    : 3024
  One Way Delay : 4 msec*
  Loss Rate Pkts: 0.0 %
  Loss Rate Byte: 0.0 %
  Jitter Mean   : 3888 usec
  Unreachable   : FALSE
 TCA Statistics:
    Received: 3327 ; Processed: 1772 ; Unreach_rcvd: 0 ; Local Unreach_rcvd: 0
    TCA lost byte rate: 0
    TCA lost packet rate: 3327
    TCA one-way-delay: 0
    TCA network-delay: 0
    TCA jitter mean: 0
Latest TCA Bucket
 Last Updated  : 00:13:38 ago
  One Way Delay : NA
  Loss Rate Pkts: 8.10 %
  Loss Rate Byte: NA
  Jitter Mean   : NA
  Unreachability: FALSE
```

The command **show domain** *domain-name* [**vrf** *vrf-name*] **border channel summary** is used to display a summary state of all channels on a specific site. This command is executed on a BR and is used to quickly check that a channel is available for a specific DSCP to a specific destination site and is also used to check the next-hop information. Viewing the routing table is not enough to identify the next hop when PfR controls traffic to a specific destination prefix.

Example 9-26 displays a summary of all channels available on Site 3 (R31 is the MC). Each line gives the main parameters for a channel. Channel 53 as an example based on the book's sample topology gives the following information:

- Channel 53 is the channel for DSCP EF.
- The next hop is R11 based on the PfR label, which is "*0:1 | 00*," which stands for *POP-ID 0* (Site 1), *PATH-ID 1* (R11).
- The next hop is 192.168.100.11, which is R11 on Site 1.
- The channel is reachable on transmit and receive.

Example 9-26 *Branch MC Router R31 Channel Summary*

```
R31-Spoke# show domain IWAN border channels summary
Ch-ID - Channel ID, SP - Service Provider
RX/TX - RX/TX Reachability
R - Reachable, UR - Un-Reachable
IN - Initial State, NR - No Route, UK - Unknown

Ch-ID Dst-Site-ID     DSCP    Next Hop       SP    pfr-Label            RX/TX
7     10.2.0.20       default 192.168.100.21 MPLS  1:1 | 0:0 [0x1010000] R/R
15    10.1.0.10       default 192.168.100.11 MPLS  0:1 | 0:0 [0x10000]   R/R
35    10.4.0.41       ef      192.168.100.41 MPLS  0:0 | 0:0 [0x0]       R/R
52    10.1.0.10       af21    192.168.100.11 MPLS  0:1 | 0:0 [0x10000]   R/R
53    10.1.0.10       ef      192.168.100.11 MPLS  0:1 | 0:0 [0x10000]   R/R
54    10.4.0.41       ef      192.168.200.41 INET  0:0 | 0:0 [0x0]       R/R
56    10.1.0.10       default 192.168.200.12 INET  0:2 | 0:0 [0x20000]   R/R
57    10.1.0.10       ef      192.168.200.12 INET  0:2 | 0:0 [0x20000]   R/R
58    10.2.0.20       ef      192.168.100.21 MPLS  1:1 | 0:0 [0x1010000] R/R
59    10.2.0.20       default 192.168.200.22 INET  1:2 | 0:0 [0x1020000] R/R
60    10.2.0.20       ef      192.168.200.22 INET  1:2 | 0:0 [0x1020000] R/R
62    10.1.0.10       af21    192.168.200.12 INET  0:2 | 0:0 [0x20000]   R/R
```

The command **show domain** *domain-name* [**vrf** *vrf-name*] **border channel** is used to display a detailed state of all channels on a specific site. This command is executed on a BR and is used to check the status of the channels and validate bidirectional setup.

Example 9-27 displays an extract of all channels for DSCP EF available on Site 3 (R31 is a BR). This output illustrates the channel to Site 1 over MPLS for DSCP EF. Transmit (TX) and Receive (RX) reachability is highlighted and shows that the channel is up and bidirectional.

Example 9-27 *Branch BR Router R31 Channel Details*

```
R31-Spoke# show domain IWAN border  channels dscp ef
--------------------------------------------------------------------
Border Smart Probe Stats:

 Smart probe parameters:
   Source address used in the Probe: 10.3.0.31
   Unreach time: 1000 ms
   Probe source port: 18000
   Probe destination port: 19000
   Interface Discovery: ON
   Probe freq for channels with traffic :0 secs
   Discovery Probes: OFF
   Number of transit probes consumed :0
   Number of transit probes re-routed: 0
   DSCP's using this: [18] [32] [34] [46] [64]
   All the other DSCPs use the default interval: 10 secs

 Channel id: 53
  Channel create time: 1w6d ago
  Site id : 10.1.0.10
  DSCP : ef[46]
  Service provider : MPLS
  Pfr-Label : 0:1 | 0:0 [0x10000]
  Exit path-id sent on wire: 0
  Exit dia bit: FALSE
  Chan recv dia bit:FALSE
  Number of Data Packets sent : 46422195
  Number of Data Packets received : 46511454
  Last Data Packet sent : 00:00:00 ago
  Last Data Packet Received : 00:00:00 ago
  Number of Probes sent : 1917758
  Number of Probes received : 1876372
  Last Probe sent : 00:00:00 ago
  Last Probe received : 00:00:00 ago
  Channel state : Initiated and open
  Channel next_hop : 192.168.100.11
  RX Reachability : Reachable
  TX Reachability : Reachable
```

```
Channel is sampling 0 flows
Channel remote end point: 192.168.100.11
Channel to hub: TRUE
Version: 3
Supports Zero-SLA: Yes
Muted by Zero-SLA: No
Plr rx state: No
Plr tx state: No
Plr establish state: No
Probe freq with traffic : 1 in 666 ms
Probe status desc : Real Traffic Present
```

Transit Site Preference

Transit site preference is used in the context of a multiple-transit-site deployment with the same set of prefixes advertised from all central sites:

- A specific transit site is preferred for a specific prefix, as long as there are available in-policy channels for that site.
- Transit site preference is a higher-priority filter and takes precedence over path preference.
- The concept of *active* and *standby* next hops based on routing metrics and advertised mask length in routing is used to gather information about the preferred transit site for a given prefix. For example, if the best metric for a given prefix is on Site 1, all the next hops on that site for all the paths are tagged as *active*.

Figure 9-3 illustrates the transit site preference with branch sites preferring Site 1 to reach the prefix 10.1.0.0/16. Prefixes 10.1.0.0/16 and 10.2.0.0/16 are now advertised from both central sites.

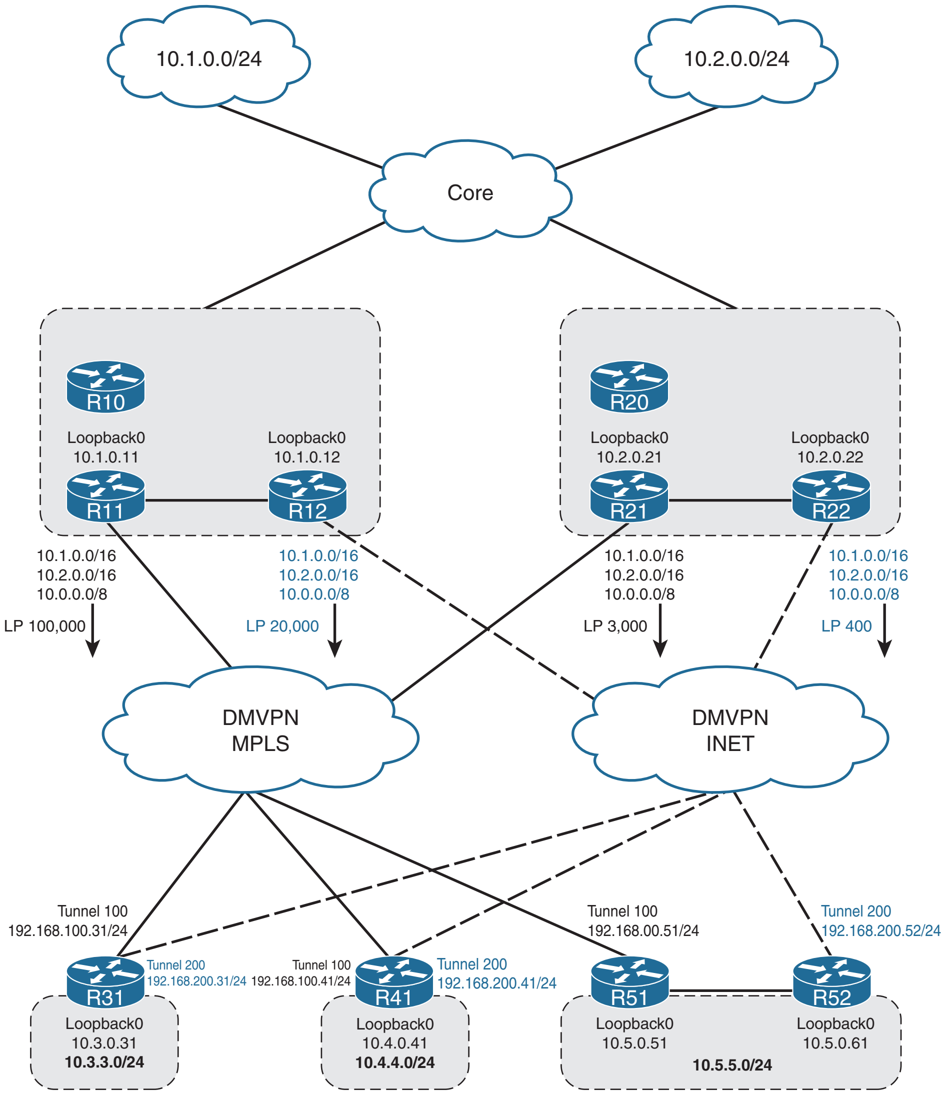


Figure 9-3 *Transit Site Preference*

Example 9-28 provides the output of the BGP table on R31. The destination prefixes 10.1.0.0.16 and 10.2.0.0/16 are now advertised from all Transit BRs. Site 1 is preferred because of the configured local preference of 100000 on DMVPN-MPLS (R11) and 20000 for DMVPN-INET (R12).

Example 9-28 *R31 BGP Topology Table*

```
R31-Spoke# show bgp ipv4 unicast
! Output omitted for brevity
     Network          Next Hop            Metric LocPrf Weight Path
* i 10.1.0.0/16      192.168.100.21          0   3000  50000 i
 * i                  192.168.200.22          0    400  50000 i
 * i                  192.168.200.12          0  20000  50000 i
 *>i                  192.168.100.11          0 100000  50000 i
 * i 10.2.0.0/16      192.168.200.12          0  20000  50000 i
 *>i                  192.168.100.11          0 100000  50000 i
 * i                  192.168.200.22          0    400  50000 i
 * i                  192.168.100.21          0   3000  50000 I
```

Channels to Site 1 for DSCP EF are in the *active* state for prefix 10.1.0.0/16 as shown in Example 9-29. The best metrics for prefix 10.1.0.0/16 are on Site 1, and all the next hops on that site are tagged as *active*.

Example 9-29 *Branch MC Router R31 Channels for Site 1 and DSCP EF (Extract)*

```
R31-Spoke# show domain IWAN master channels dscp ef
! Output omitted for brevity

   Legend: * (Value obtained from Network delay:)

Channel Id: 53  Dst Site-Id: 10.1.0.10  Link Name: MPLS  DSCP: ef [46] pfr-label:
  0:1 | 0:0 [0x10000] TCs: 1
  Channel Created: 6d12h ago
  Provisional State: Initiated and open
  Operational state: Available
  Channel to hub: TRUE
  Interface Id: 15
  Supports Zero-SLA: Yes
  Muted by Zero-SLA: No
  Estimated Channel Egress Bandwidth: 1 Kbps
  Immitigable Events Summary:
   Total Performance Count: 0, Total BW Count: 0
  Site Prefix List
    10.1.0.10/32 (Routable)
    10.1.0.0/16 (Active)
    10.2.0.0/16 (Routable)

! Output omitted for brevity

Channel Id: 57  Dst Site-Id: 10.1.0.10  Link Name: INET  DSCP: ef [46] pfr-label:
  0:2 | 0:0 [0x20000] TCs: 0
  Channel Created: 6d12h ago
  Provisional State: Initiated and open
  Operational state: Available
  Channel to hub: TRUE
  Interface Id: 16
  Supports Zero-SLA: Yes
  Muted by Zero-SLA: No
  Estimated Channel Egress Bandwidth: 0 Kbps
  Immitigable Events Summary:
   Total Performance Count: 0, Total BW Count: 0
  Site Prefix List
    10.1.0.10/32 (Routable)
    10.1.0.0/16 (Active)
    10.2.0.0/16 (Routable)
   ODE Statistics:
      Received: 2834
```

Channels to Site 2 for DSCP EF are in *standby* state for prefix 10.1.0.0/16 as shown in Example 9-30. This is because Site 1 is preferred because of the highest local preference configured on R11 (Site 1, MPLS) and R12 (Site 1, INET).

Example 9-30 *Branch MC Router R31 Channels for Site 2 and DSCP EF (Extract)*

```
R31-Spoke# show domain IWAN master channels dscp ef
! Output omitted for brevity

   Legend: * (Value obtained from Network delay:)

Channel Id: 58  Dst Site-Id: 10.2.0.20  Link Name: MPLS  DSCP: ef [46] pfr-label:
  1:1 | 0:0 [0x1010000] TCs: 0
  Channel Created: 1w6d ago
  Provisional State: Initiated and open
  Operational state: Available
  Channel to hub: TRUE
  Interface Id: 15
  Supports Zero-SLA: Yes
  Muted by Zero-SLA: No
  Estimated Channel Egress Bandwidth: 1 Kbps
  Immitigable Events Summary:
   Total Performance Count: 0, Total BW Count: 0
  Site Prefix List
    10.1.0.0/16 (Standby)
    10.2.0.20/32 (Routable)
    10.2.0.0/16 (Routable)
```

```
[Output omitted for brevity]

Channel Id: 60  Dst Site-Id: 10.2.0.20  Link Name: INET  DSCP: ef [46] pfr-label:
  1:2 | 0:0 [0x1020000] TCs: 0
  Channel Created: 1w6d ago
  Provisional State: Initiated and open
  Operational state: Available
  Channel to hub: TRUE
  Interface Id: 16
  Supports Zero-SLA: Yes
  Muted by Zero-SLA: No
  Estimated Channel Egress Bandwidth: 0 Kbps
  Immitigable Events Summary:
   Total Performance Count: 0, Total BW Count: 0
  Site Prefix List
    10.1.0.0/16 (Standby)
    10.2.0.20/32 (Routable)
    10.2.0.0/16 (Routable)
```

With Transit Site Affinity Enabled (by Default)

Transit Site Affinity takes precedence over path preference, so in that scenario R11 (Site 1, MPLS) is preferred, and R12 (Site 1, INET) is a secondary path if the path to R11 is out of policy (even if path preference is for MPLS, PfR will prefer Site 1 over Site 2). PfR will then fail over to Site 2.

Example 9-31 displays the TC for 10.1.0.0/16 and DSCP EF. The primary channel is the path over MPLS to R11 (pfr-label is "*0:1 | 0:0*," *POP-ID 0*, and *PATH-ID 1*, hence R11) and a secondary path is over INET to R12 (pfr-label is "*0:2 | 0:0*," *POP-ID 0*, and *PATH-ID 2*, hence R12).

Example 9-31 *Branch MC Router R31 TC for 10.1.0.0.16 and DSCP EF*

```
R31-Spoke# show domain IWAN master traffic-classes dscp ef
! Output omitted for brevity
 Dst-Site-Prefix: 10.1.0.0/16         DSCP: ef [46] Traffic class id:16
  Clock Time:                 19:19:24 (EST) 12/22/2015
  TC Learned:                 1w6d ago
  Present State:              CONTROLLED
  Current Performance Status: in-policy
  Current Service Provider:   MPLS since 01:38:17
  Previous Service Provider:  INET pfr-label: 0:2 | 0:0 [0x20000] for 180 sec
  BW Used:                    1 Kbps
  Present WAN interface:      Tunnel100 in Border 10.3.0.31
  Present Channel (primary):  53 MPLS pfr-label:0:1 | 0:0 [0x10000]
  Backup Channel:             57 INET pfr-label:0:2 | 0:0 [0x20000]
  Destination Site ID bitmap: 3
  Destination Site ID:        10.1.0.10
  Alternate Destination site: 10.2.0.20
  Class-Sequence in use:      10
  Class Name:                 VOICE using policy User-defined
    priority 2 packet-loss-rate threshold 5.0 percent
    priority 1 one-way-delay threshold 150 msec
    priority 2 byte-loss-rate threshold 5.0 percent
```

With Transit Site Affinity Disabled (Configured)

Transit Site Affinity is enabled by default. The command **no transit-site-affinity** has to be configured in advanced mode on the Hub MC to disable transit site preference and allow the use of all channels to Site 1 and Site 2. Example 9-32 displays the configuration on the Hub MC R10 to disable transit site preference.

Example 9-32 *R10 Transit Site Preference Disabled*

```
R10-HUB-MC
domain IWAN
 vrf default
  master hub
   advanced
    no transit-site-affinity
```

Example 9-33 displays the TC for 10.1.0.0/16 and DSCP EF. The primary channel is the path over MPLS to R11 (pfr-label is "*0:1 | 0:0*," *POP-ID 0*, and *PATH-ID 1*, hence R11) and a secondary path is again over MPLS but to R21 (pfr-label is "*1:1 | 0:0*," *POP-ID 1*, and *PATH-ID 1*, hence R21). In that example, path preference is MPLS and therefore PfR uses the path over MPLS to Site 2 as a secondary path. There is no site preference, and Site 1 and Site 2 can be used to forward traffic.

Example 9-33 *Branch MC Router R31 TC for 10.1.0.0.16 and DSCP EF*

```
R31-Spoke# show domain IWAN master traffic-classes dscp ef

 Dst-Site-Prefix: 10.1.0.0/16         DSCP: ef [46] Traffic class id:21
  Clock Time:                 19:24:42 (EST) 12/22/2015
  TC Learned:                 00:02:11 ago
  Present State:              CONTROLLED
  Current Performance Status: in-policy
  Current Service Provider:   MPLS since 00:01:40
  Previous Service Provider:  Unknown
  BW Used:                    1 Kbps
  Present WAN interface:      Tunnel100 in Border 10.3.0.31
  Present Channel (primary):  53 MPLS pfr-label:0:1 | 0:0 [0x10000]
  Backup Channel:             67 MPLS pfr-label:1:1 | 0:0 [0x1010000]
  Destination Site ID bitmap: 3
  Destination Site ID:        10.1.0.10
  Alternate Destination site: 10.2.0.20
  Class-Sequence in use:      10
  Class Name:                 VOICE using policy User-defined
    priority 2 packet-loss-rate threshold 5.0 percent
    priority 1 one-way-delay threshold 150 msec
    priority 2 byte-loss-rate threshold 5.0 percent
  BW Updated:                 00:00:11 ago
  Reason for Latest Route Change:    Uncontrolled to Controlled Transition
  Route Change History:
             Date and Time                    Previous Exit
  Current Exit                                 Reason


    1:  19:23:02 (EST) 12/22/15   None(0:0|0:0)/0.0.0.0/None (Ch:0)
  MPLS(0:1|0:0)/10.3.0.31/Tu100 (Ch:53)              Uncontrolled to Controlled
  Transition
--------------------------------------------------------------------------------
 Total Traffic Classes: 2 Site: 2  Internet: 0
```

Summary

This chapter focused on the monitoring of intelligent path control (Performance Routing, PfR) for the IWAN architecture.

PfR relies on the routing protocol (EIGRP or BGP) and thus a proper configuration of these protocols is required. Destination prefixes have to be reachable over all paths (DMVPN tunnels) used in the IWAN domain and so must have at least a parent route in the BGP or EIGRP topology table for every external path.

Each site in the IWAN domain runs PfR and gets its path control configuration and policies from the Hub MC through the IWAN peering service. External tunnels are discovered through the use of discovery probes. Therefore, it is critical to make sure the IWAN peering is established between the local MC and the Hub MC, and that policies and monitor configurations are received and correctly applied on every tunnel.

Finally, PfR operations involve the use of monitoring channels that must be checked to make sure they are operational. TC next hop, performance metrics, out-of-policy events, route changes, and statistics are all available locally on the MC or BR.

All events and metrics are exported through NetFlow v9 records to a NetFlow collector. This allows the monitoring of an IWAN domain from a central place with the help of reporting tools like Cisco Prime Infrastructure, LiveAction, or LivingObjects.

Further Reading

Cisco. "Embedded Packet Capture." www.cisco.com/c/en/us/td/docs/ios-xml/ios/epc/configuration/xe-3s/asr1000/epc-xe-3s-asr1000-book/nm-packet-capture-xe.html.

Cisco. "Performance Routing Version 3." www.cisco.com/go/pfr.

Cisco. "PfRv3 Configuration." www.cisco.com.

This page intentionally left blank

Chapter 10

Application Visibility

This chapter covers the following topics:

- Application Visibility fundamentals
- Performance metrics
- Flexible NetFlow
- Performance Monitor
- ezPM
- Metrics export—NetFlow v9 and IPFIX

The growth in cloud computing and new voice and video collaboration applications are increasing the bandwidth, resiliency, and performance-monitoring requirements for the WAN. These new enterprise trends make it important for enterprise networks to be application aware to cost-effectively address these stringent requirements. For the staff responsible for planning, operating, and maintaining the network and network services, it is indispensable to have visibility into the current health of the network from end to end. It is also essential to gather short- and long-term data in order to fully understand how the network is performing and what applications are active on it. Capacity planning is one of the most important issues faced by organizations in managing their networks.

Application Visibility is a key component of IWAN to meet the needs of the modern cloud and collaboration applications. With Application Visibility it becomes possible to understand what applications are in use in enterprise networks and examine their performance.

Application Visibility Fundamentals

Application Visibility is an integral part of Cisco IOS software and is a key component of IWAN that collects and measures application performance data.

Overview

Network operators want to understand how their network is being used and by which applications. Traditionally, this knowledge has been obtained by exporting information about the flows traversing the network using *Traditional NetFlow (TNF)* and *Flexible NetFlow (FNF)*, and then analyzing the data using a network management system (NMS) or analytic system. Exported fields that can be used to classify flows include IP addresses, port numbers, DSCP markings, and application names using NBAR, among other techniques. Collected metrics are traditionally bytes and packets that give information about the bandwidth used per application and client/server pair.

But organizations want real application visibility into the network and need to understand application performance. *Performance Monitor* is the next-generation monitoring engine that adds application-level visibility to a variety of network devices, beginning with branch and WAN aggregation routers and wireless LAN controllers. Performance Monitor recognizes and classifies thousands of applications, including voice and video, email, file sharing, gaming, peer-to-peer (P2P), encrypted, and cloud-based applications, and uses this classification to perform per-application monitoring of bandwidth statistics (traditional NetFlow statistics), of transactional application response time (ART) metrics, and of media application metrics such as latency and jitter. The per-application metrics are exported via NetFlow version 9 and Internet Protocol Flow Information Export (IPFIX) for analysis, reporting, and visualization by partner network management systems. All of this is accomplished without the need to deploy and manage separate hardware or software probes in each network location; it is integrated directly into the Cisco devices.

Performance Monitor is also an integral part of the Cisco initiative called *Application Visibility and Control (AVC)*. This is why IWAN Application Visibility is commonly named AVC.

Components

The Cisco IWAN Application Visibility solution leverages multiple technologies to recognize and analyze applications, and it combines several Cisco IOS and IOS XE components, as well as communicates with external tools, to integrate the following functions into a powerful solution:

- **Application recognition and classification**: Operating on Cisco IOS and Cisco IOS XE, NBAR2 uses multiple innovative technologies such as DPI, DNS, DNS-AS, server based, and more to identify a wide variety of applications within the network traffic flow, using L3 to L7 data. NBAR2 can monitor thousands of applications and supports Protocol Pack updates for expanding application recognition, without requiring IOS upgrade or router reload. For more information, refer to Chapter 6, “Application Recognition.”

- **Metrics collection:** Traditionally the monitoring engine used to collect bytes and packets has been Flexible NetFlow. Performance Monitor is the next-generation monitoring engine that can provide performance metrics for TCP-based applications (ART metrics), RTP-based applications (media applications), and HTTP-based applications. Performance Monitor allows the definition of a policy for the traffic to be monitored.

- **Metrics export:** Metrics are aggregated and exported in NetFlow v9 (RFC 3954) or IPFIX (RFC 7011) format to a management and reporting package. Metrics records are sent out directly from the data plane when possible, to maximize system performance. When more complex processing is required, such as when the router maintains a history of exported records, records may be exported by the route processor, which is slower than direct export from the data plane.

- **Management and reporting systems:** Management and reporting systems, such as Cisco Prime Infrastructure or third-party tools, receive the network metrics data in NetFlow v9 or IPFIX format and provide a wide variety of system management and reporting functions. These functions include configuring metrics reporting, creating application and network performance reports, system provisioning, configuring alerts, and assisting in troubleshooting.

Figure 10-1 illustrates the basic elements of Application Visibility with the metering engine available on IWAN platforms, the exporting protocol, and the NMS that collects all performance metrics.

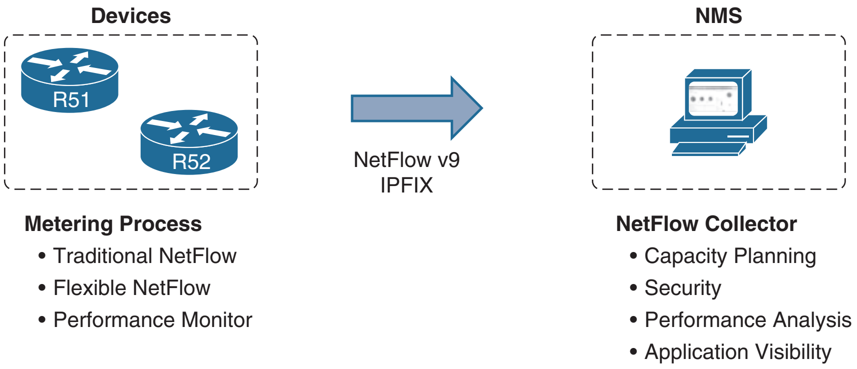


Figure 10-1 *Application Visibility Building Blocks*

The key advantages to using Flexible NetFlow or Performance Monitor can be summarized as follows:

- Flexibility and scalability of flow data
- The ability to monitor a wide range of packet information, producing new information about network behavior
- Enhanced network anomaly and security detection

- User-configurable flow information to perform customized traffic identification and the ability to focus and monitor specific network behavior
- Convergence of multiple accounting technologies into one accounting mechanism

Both Flexible NetFlow and Performance Monitor include two key components that perform the following functions:

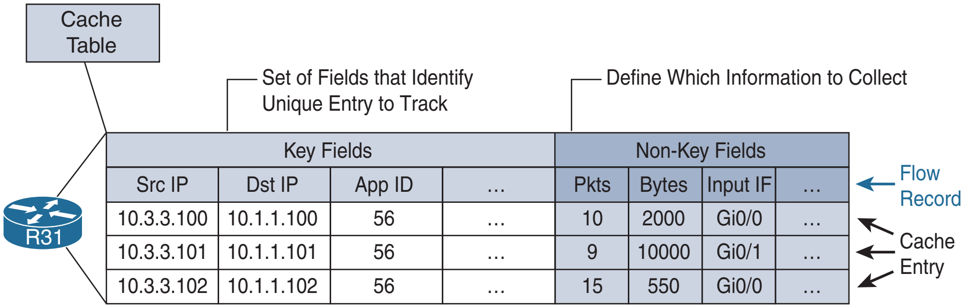
- **Flow caching** analyzes and collects IP data flows within a router or switch and prepares data for export. Flexible NetFlow and Performance Monitor have the ability to implement multiple flow caches or flow monitors for tracking different applications simultaneously. For instance, the user can simultaneously track one application's accessibility for security purposes and execute traffic analysis for performance issues for a different application in separate caches. This gives the ability to pinpoint and monitor specific information about the applications.
- **NetFlow reporting collection** uses exported data from multiple routers and filters, aggregates the data according to customer policies, and stores this summarized or aggregated data. NetFlow collection systems allow users to complete real-time visualization or trending analysis of recorded and aggregated flow data. Users can specify the router and aggregation scheme and time interval desired. Collection systems can be commercial or third-party freeware products and can be optimized for specific NetFlow applications such as traffic or security analysis.

Flows

Flexible NetFlow and Performance Monitor use the concept of flows. A *flow* is defined as a stream of packets between a given source and a given destination. The definition of a flow is based on the definition of key fields. For example, a flow can include all packets between a given source IP and destination IP, or it can be defined as all traffic for a specific application. Performance Collection defines the concept of key fields and non-key fields:

- Flexible NetFlow and Performance Monitor use the values in *key fields* in IP datagrams, such as the IP source or destination address and the source or destination transport protocol port, as the criteria for determining when a new flow must be created in the cache while network traffic is being monitored. When the value of the data in the key field of a datagram is unique with respect to the flows that already exist, a new flow is created.
- Flexible NetFlow and Performance Monitor use *non-key fields* as the criteria for identifying fields from which data is captured from the flows. The flows are populated with data that is captured from the values in the non-key fields.
- The combination of key fields and non-key fields is called a NetFlow record.

Figure 10-2 illustrates a NetFlow cache with key fields and non-key fields. The flow definition could include more fields, which are represented by ellipses, or have fewer fields. The application ID that was identified by NBAR2 is displayed in decimal format. The command **show ip nbar protocol-id** correlates the application ID to the specific application.


Cache Table
Set of Fields that Identify Unique Entry to Track
Define Which Information to Collect
R31
Flow Record
Cache Entry

Key Fields				Non-Key Fields			
Src IP	Dst IP	App ID	…	Pkts	Bytes	Input IF	…
10.3.3.100	10.1.1.100	56	…	10	2000	Gi0/0	…
10.3.3.101	10.1.1.101	56	…	9	10000	Gi0/1	…
10.3.3.102	10.1.1.102	56	…	15	550	Gi0/0	…

Figure 10-2 *NetFlow Record and Cache*

Figure 10-3 shows an example of the process for inspecting packets and creating flow records in the cache. In this example, two unique flows are created in the cache because different values are in the source and destination IP address key fields. Using application name as a key field might create multiple flows unless *account-on-resolution* is used.

Key Fields	Packet 1
Source IP	10.3.3.100
Destination IP	10.1.1.100
Source Port	23
Destination Port	22078
Layer 3 Protocol	TCP-6
TOS Byte	0
Non-Key Fields	Packet 1
Length	1250

Key Fields	Packet 2
Source IP	10.3.3.101
Destination IP	10.4.4.100
Source Port	23
Destination Port	22079
Layer 3 Protocol	TCP-6
TOS Byte	0
Non-Key Fields	Packet 2
Length	519

NetFlow Cache After Packet 1

Source IP	Destination IP	Dst Interface	Protocol	Source Port	Dst Port	TOS	Packets
10.3.3.100	10.1.1.100	Tunnel 100	TCP-6	23	22078	22078	11000

NetFlow Cache After Packet 2

Source IP	Destination IP	Dst Interface	Protocol	Source Port	Dst Port	TOS	Packets
10.3.3.100	10.1.1.100	Tunnel 100	TCP-6	23	22078	22078	11000
10.3.3.100	10.1.1.100	Tunnel 100	TCP-6	23	22078	22079	10359

Figure 10-3 *NetFlow Creating a Cache Entry*

Observation Point

The location where a router monitors traffic is called the *observation point* and is the interface where the flows are metered. The location of the observation point is exported to a network management station or analytic system.

Flow Direction

The flow direction is defined in RFC 5102 and is the direction of the flow observed at the observation point. Flow direction is *ingress* or *egress* and identifies whether the flow comes into an interface or out of an interface when the flow is collected. Figure 10-4 illustrates Performance Collection enabled:

- Observation point A is configured on R31's ingress interface, GigabitEthernet0/3.
- Observation point B is configured on R31's egress interface, tunnel 100.

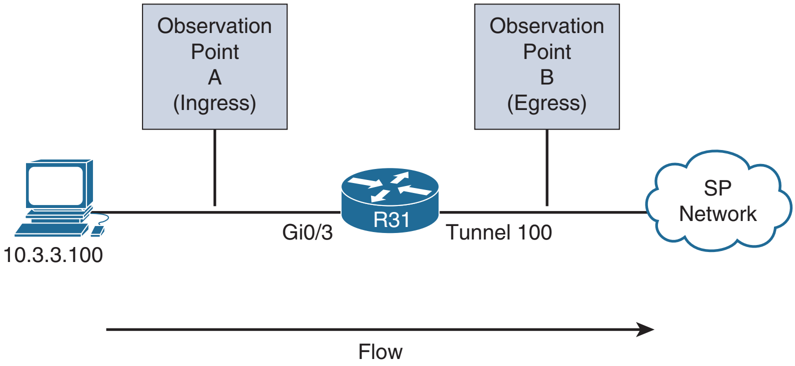


Figure 10-4 *Flow Direction and Observation Point*

Traffic leaving host 10.3.3.100 is *ingress* to R31's GigabitEthernet0/3 and *egress* from R31's tunnel 100 interface. The traffic coming back from the server is *ingress* to R31's tunnel 100 interface and *egress* from R31's GigabitEthernet0/3. As a result, Performance Collection can potentially report both forward and return traffic for observation points A and B. Therefore, IPFIX requires differentiation between *ingress* and *egress* to detect the direction of the network traffic flow for an interface.

Source/Destination IP Versus Connection

The definition of a flow in a NetFlow record usually includes the source and destination IP addresses. This creates two unidirectional flows. The use of connections creates a bidirectional flow and therefore reduces the size of the NetFlow cache, and it is always associated with the client or server regardless of where the bidirectional flow resides in the network.

Figure 10-5 shows the configuration differences that are available when defining the collection of a flow's records.

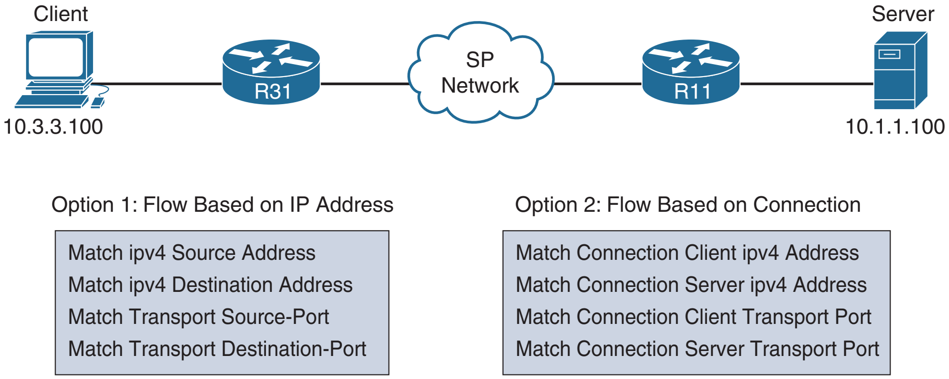

Client
10.3.3.100
R31
SP Network
R11
Server
10.1.1.100
Option 1: Flow Based on IP Address
Match ipv4 Source Address
Match ipv4 Destination Address
Match Transport Source-Port
Match Transport Destination-Port
Option 2: Flow Based on Connection
Match Connection Client ipv4 Address
Match Connection Server ipv4 Address
Match Connection Client Transport Port
Match Connection Server Transport Port

Figure 10-5 *Configuration of Flow Based on IP Address or Connection*

Figure 10-6 illustrates the use of source and destination IP addresses versus connection in a record definition. When connection is used, a single bidirectional record is created.

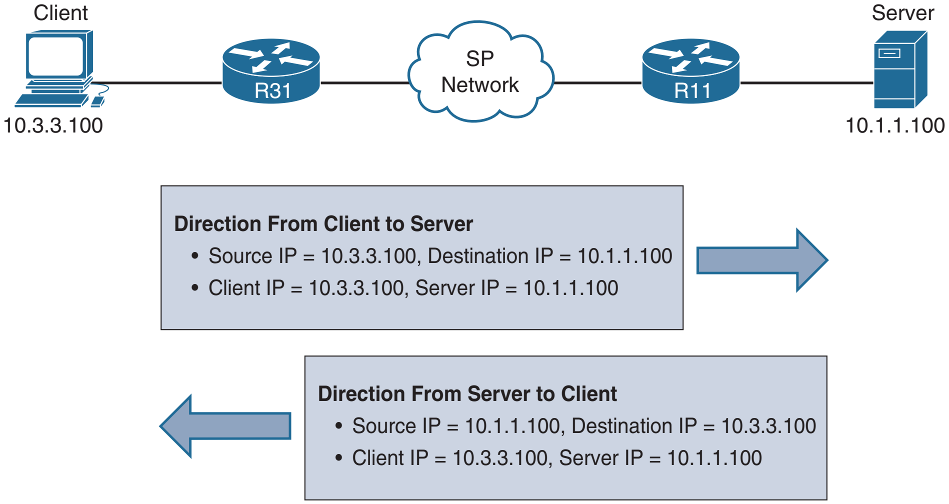

Client
10.3.3.100
R31
SP Network
R11
Server
10.1.1.100
Direction From Client to Server
• Source IP = 10.3.3.100, Destination IP = 10.1.1.100
• Client IP = 10.3.3.100, Server IP = 10.1.1.100
Direction From Server to Client
• Source IP = 10.1.1.100, Destination IP = 10.3.3.100
• Client IP = 10.3.3.100, Server IP = 10.1.1.100

Figure 10-6 *Source/Destination IP Address Versus Connection*

Performance Metrics

In addition to bandwidth statistics, Application Visibility can provide performance metrics for TCP and media applications running over RTP.

Application Response Time Metrics

Application response time (ART) measures the performance of a TCP-based application. It separates the application delivery path into multiple segments and provides insight into application behavior (network versus server bottleneck) to accelerate problem isolation. ART metrics provide essential information on application performance as experienced by clients in branch offices. Enabling performance measurements in different network segments and using ART metrics helps to

- Quickly determine where a problem lies and identify the source of an application's performance problems, which can be related to the network, the application, or the application server
- Quickly resolve problems before users notice them
- Understand application behavior over time to support planning for change: implementing new network resources, applying policies, and so forth
- Enable deployment and verification of WAN optimization services
- Clarify user expectations to support the development of service levels

Figure 10-7 illustrates path segments between the client and the server. *Server network delay (SND)* approximates WAN delay.

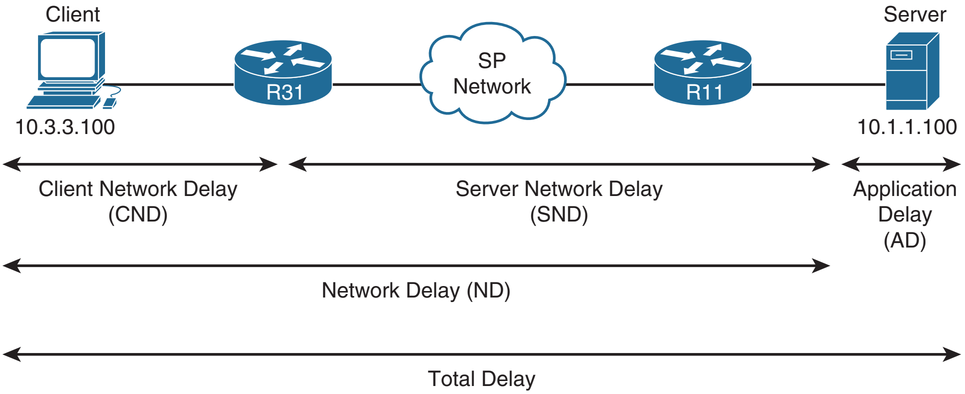


Figure 10-7 *Performance Collection on R31—ART Network Path Segments*

ART metrics are metrics extracted or calculated by the ART engine. These metrics are available only for TCP flows. ART client/server bytes and packets are for Layer 3 and Layer 4 measurements.

Figure 10-8 illustrates the TCP performance in the context of different types of delay and response times.

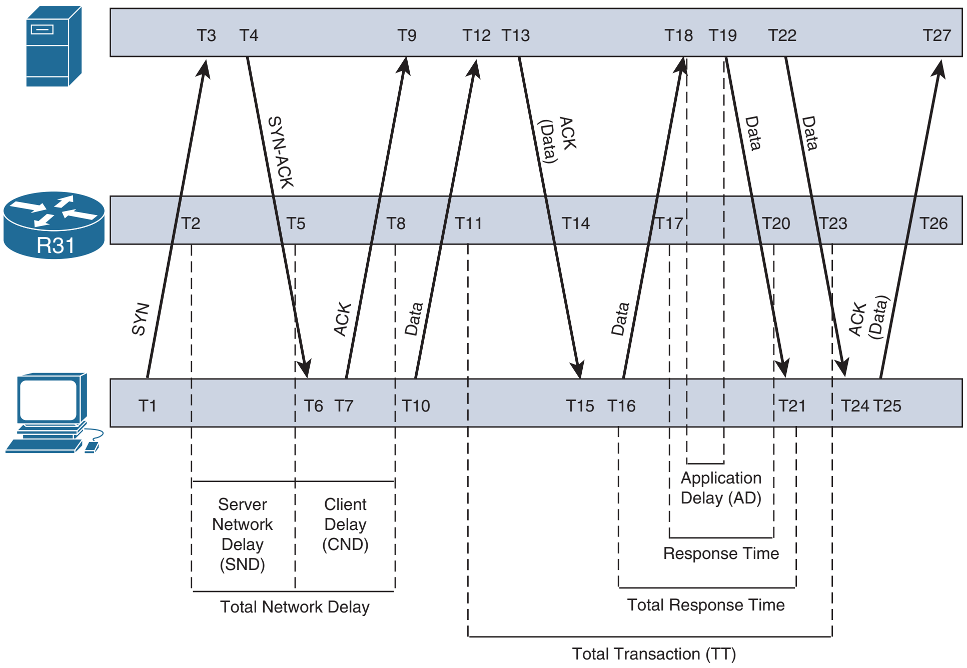


Figure 10-8 *Application Response Time Metrics*

From Figure 10-8, the following major performance metrics are calculated:

- *Server network delay (SND)* identifies the processing time for the operating system running on the server, assuming that the TCP handshake is managed by the kernel.
- *Client network delay (CND)* identifies the processing time for the operating system on the client, assuming that the TCP handshake is managed by the kernel.
- *Network delay (ND)* = CND + SND.
- *Application delay (AD)* = response time (RT) – SND.
- *Transaction time (TT)* quantifies the user experience, and AD helps with troubleshooting server issues.

Media Metrics

Performance Monitor calculates RTP packet drops by keeping track of the sequence numbers that are part of the RTP header. Unlike a TCP connection, a media stream based on RTP and UDP is always unidirectional. Thus, a Performance Monitor policy applied in the input direction on a LAN interface on a branch site's router collects RTP metrics only for media streams leaving the site. To collect RTP metrics for media streams entering a branch site, you need to apply the policy either on the WAN interface (input) or on the LAN interface (output). Media monitoring can be applied at different locations to measure performance before and after the observation point. Figure 10-9 illustrates the RTP performance metrics for a collector instance applied on ingress or egress direction.

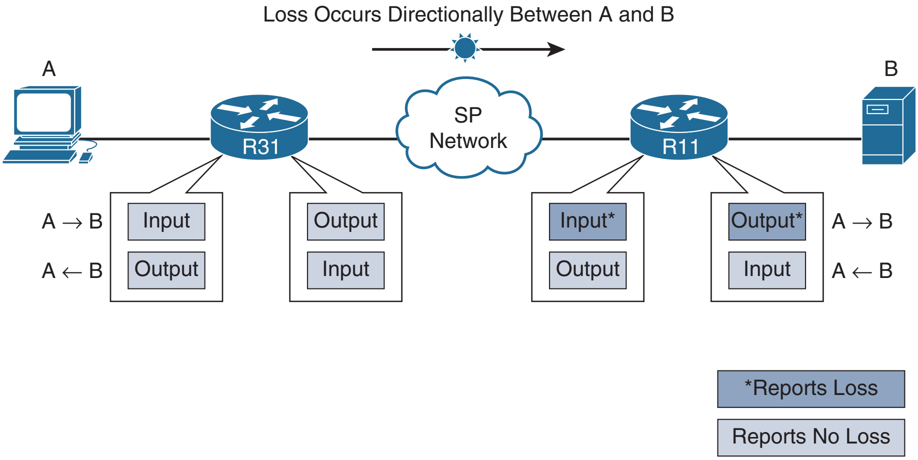


Figure 10-9 *RTP Metrics*

Another field in the RTP header is the synchronization source identifier (SSRC). This identifier is used to distinguish between different audio and video channels if they share the same UDP session. In the case of the Cisco TelePresence System, the multiscreen video channels share the same UDP stream (IP source, IP destination, and Layer 4 ports). For the Cisco TelePresence System, the SSRC is used to differentiate the unique video channels.

RTP jitter values are calculated by analyzing the timestamp field in the RTP header. The timestamp does not actually refer to regular time but to ticks of the encoder's clock. For video, the encoding clock rate is usually 90 kHz, and in traditional voice it is 8 kHz. However, with modern wideband audio codecs, the frequency may be a variety of values. Performance Monitor tries to derive the clock rate from the payload type field in the RTP header, so the RTP payload type gives an idea of the kind of media in an RTP stream. The static RTP payload types can be found on the IANA website (www.iana.org/assignments/rtp-parameters).

Web Statistics

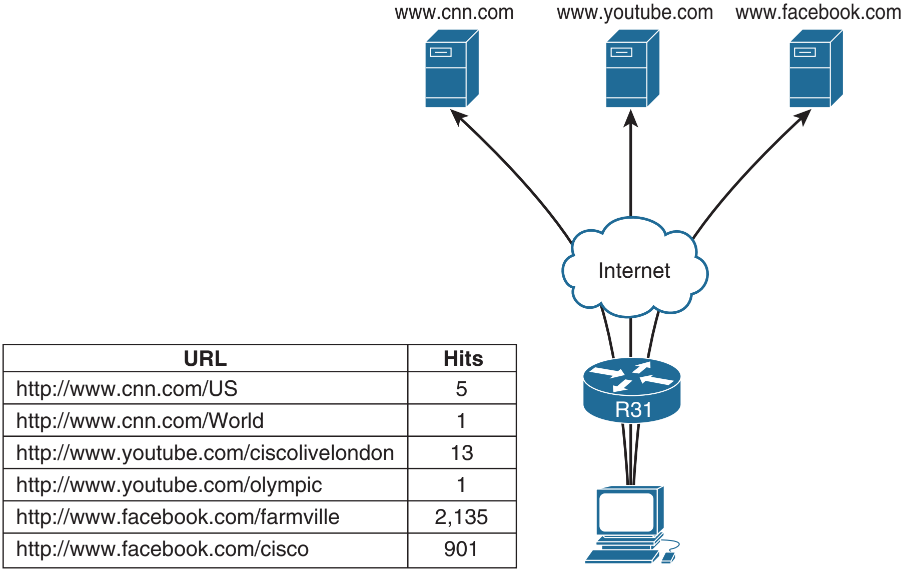
Performance Collection collects and exports web metrics such as host name, URIs (aka URLs), SSL common names, and more. This information can be used to identify which web traffic is business relevant and which traffic is business irrelevant. Performance metrics can be collected for those applications that are business relevant.

HTTP Host

Figure 10-10 illustrates HTTP traffic from a user to various destinations. The host names in this example are www.cnn.com, www.youtube.com, and www.facebook.com. The host is collected only when it is observed.

URI Statistics

A pattern is used to export the list of URIs and the corresponding hit counts. For example, in Figure 10-10 there are the following flows during the five-minute window: host name www.cnn.com, source IP 10.3.100.15, destination IP 100.64.20.2, destination port 80, and protocol TCP.



URL	Hits
http://www.cnn.com/US	5
http://www.cnn.com/World	1
http://www.youtube.com/ciscolivelondon	13
http://www.youtube.com/olympic	1
http://www.facebook.com/farmville	2,135
http://www.facebook.com/cisco	901

Figure 10-10 *URL Statistics*

The result is exported with the source and destination IP addresses, port number, and URL.

> **Note** There are many more extracted fields and metrics, and Performance Monitor has the flexibility to be provisioned as needed. For more information about all metrics available with Performance Collection, refer to the "AVC Metrics Definitions" on the Cisco website, www.cisco.com.

Flexible NetFlow

Flexible NetFlow is commonly used to collect metrics about traffic forwarded by a device and supports both IPv4 and IPv6. It can be a first option to obtain application visibility, mostly based on application names and bandwidth utilization. This section on Flexible NetFlow is followed by the Performance Monitor section. Performance Monitor provides more benefits than FNF by including information on a specific class of traffic, history, or reaction. Performance Monitor runs on top of Flexible NetFlow, so any feature available in FNF is also available in Performance Monitor. It is recommended to use Performance Monitor to get the full application visibility that includes performance metrics.

Note Performance Monitor is now included on the WAN edge router platforms that support IWAN.

Flexible NetFlow Overview

Flexible NetFlow (FNF) is an integral part of Cisco IOS software that collects and measures data, thus allowing every router or switch in the network to become a source of telemetry and a monitoring device. FNF allows extremely granular and accurate traffic measurements and high-level aggregated traffic collection.

Configuration Principles

The following process must be completed to enable NetFlow data collection and optional data export:

- **Create an FNF flow record or select a built-in flow record:** A combination of key and non-key fields is called a *record*. Flexible NetFlow records are assigned to FNF flow monitors to define the cache that is used for storing flow data.

  FNF flow records define the criteria that are used to *match* a flow and additional criteria that routers should *collect* if present. Key fields are the mandatory criteria in a flow that must be matched in order for routers to create a cache and optionally export it. All key fields must be matched. Non-key fields are not mandatory criteria, but data is also collected if it exists. Flexible NetFlow includes several predefined records that can be used instead of creating a custom NetFlow record from scratch.

- **Create a flow exporter for each external NetFlow collector:** Flow exporters define where to send the flow data that has been collected. The flow record exporter destinations are NetFlow analyzers used for off-box analysis. Flow exporters are created as separate entities in the configuration. They are assigned to flow monitors to provide data export capability. Several flow monitors can be defined to provide several export destinations.

- **Create a flow monitor and associate it with either a custom or built-in flow record:** Flow monitors are the FNF components that are applied to interfaces to perform network traffic monitoring. Flow data is collected from the network traffic and added to the flow monitor cache during the monitoring process based on the key and non-key fields in the flow record.
- **Associate the flow monitor to an interface:** The flow monitors must be associated to an interface in an ingress or egress fashion.

The user-defined flow record facilitates the creation of various configurations for traffic analysis and data export on a networking device with a minimum number of configuration commands. Each flow monitor can have a unique combination of flow record, flow exporter, and cache type.

Create a Flexible NetFlow Flow Record

Flexible NetFlow requires the explicit configuration of a flow record that consists of both key fields and non-key fields. The following steps are required to create an FNF record:

Step 1. **Create a user-defined flow record.**

Customized flow records are used to analyze traffic data for a specific purpose. A customized flow record must have at least one match criterion for use as the key field and typically has at least one collect criterion for use as a non-key field.

The command **flow record** *record-name* defines a Flexible NetFlow record.

Step 2. **Define the key fields.**

Key fields are used to define the flow. The command **match** {**application** | **datalink** | **flow** | **interface** | **ipv4** | **ipv6** | **routing** | **timestamp** | **transport**} is used to configure a key field for the flow record.

For information about the key fields available for the **match** command, refer to the *Cisco IOS Flexible NetFlow Command Reference* at www.cisco.com/c/en/us/td/docs/ios/fnetflow/command/reference/fnf_book.html.

Step 3. **Repeat Step 2 as required to configure additional key fields for the record.**

Step 4. **Define the non-key fields.**

This step is to configure the metrics that Flexible NetFlow collects for every flow. The command **collect** defines a non-key field.

Step 5. **Repeat Step 4 as required to configure additional non-key fields for the record.**

Example 10-1 provides an example of a flow record configuration that is deployed to R11 to get a basic understanding of the traffic.

Example 10-1 *R11 Flexible NetFlow Record Example*

```
R11
flow record NMS-FLOWRECORD
 description NMS Record
 match ipv4 tos
 match ipv4 protocol
 match ipv4 source address
 match ipv4 destination address
 match transport source-port
 match transport destination-port
 match interface input
 match flow direction
 collect routing source as
 collect routing destination as
 collect routing next-hop address ipv4
 collect ipv4 dscp
 collect ipv4 id
 collect ipv4 source prefix
 collect ipv4 source mask
 collect ipv4 destination mask
 collect transport tcp flags
 collect interface output
 collect flow sampler
 collect counter bytes
 collect counter packets
 collect timestamp sys-uptime first
 collect timestamp sys-uptime last
 collect application name
```

Create a Flow Exporter

The NetFlow data that is stored in the cache of the network device can be more effectively analyzed when exported to an external collector. A flow exporter is required only when exporting data to an external collector. This procedure may be skipped if data is analyzed only on the network device.

The following steps are performed to create a flow exporter for a remote system for further analysis and storage:

Step 1. Create a flow exporter.

A flow exporter defines the parameters for an NMS running a NetFlow collector. The command **flow exporter** *exporter-name* defines a new exporter.

Step 2. **Define the destination.**

Each flow exporter supports only one destination. If the data needs to be exported to multiple destinations, multiple flow exporters are configured and assigned to the flow monitor. The command **destination** {*hostname* | *ip-address*} [**vrf** *vrf-name*] specifies the destination for the export. The IP address can be either an IPv4 or IPv6 address.

Step 3. **Define the export protocol.**

Data can be exported using NetFlow v9 or IPFIX. The command **export-protocol {netflow-v5 | netflow-v9 | ipfix}** defines the export protocol. NetFlow v5 is available for traditional NetFlow and cannot be used for Application Visibility, which preferably uses IPFIX. NetFlow v9 can be used for Flexible NetFlow with a user-based record that does not require a variable-length field (such as URL monitoring).

Step 4. **Define the UDP port.**

The command **transport udp** *udp-port* defines the port used at the NMS.

Step 5. **Define option templates.**

The command **option** *option-name* is used to add the option templates that a network administrator wants to export to the NMS. Option templates include application table, interface table, and more. For information about the option names that are available to configure option templates, refer to the *Cisco IOS Flexible NetFlow Command Reference*.

Different NetFlow collector applications support different export version formats (NetFlow v9 and/or IPFIX) and expect to receive the exported data on a particular UDP port. Example 10-2 provides an example of a flow exporter configuration that is deployed on R11 to export to an NMS.

Example 10-2 *R11 Flexible NetFlow Exporter*

```
R11
flow exporter NMS-FLOWEXPORTER
 destination 10.151.1.95
 source Loopback0
 transport udp 2055
 export-protocol ipfix
 option interface-table
 option c3pl-class-table timeout 300
 option c3pl-policy-table timeout 300
 option application-table
```

Create a Flow Monitor

The network device must be configured to monitor the flows through the device on a per-interface basis. The flow monitor must include a flow record and optionally one or more flow exporters if data is to be collected and analyzed. After the flow monitor is created, it is applied to device interfaces. The flow monitor stores flow information in a cache, and the timer values for this cache are modified within the flow monitor configuration.

The following steps are required to create a Flexible NetFlow monitor and assign it to an interface:

Step 1. **Create a flow monitor.**

Customized flow records are used to analyze traffic data for a specific purpose. A customized flow record must have at least one match criterion for use as the key field and typically has at least one collect criterion for use as a non-key field.

The command **flow monitor** *monitor-name* defines a Flexible NetFlow monitor.

Step 2. **Assign the flow record.**

Each flow monitor requires a record to define the contents and layout of its cache entries. The record format can be one of the predefined record formats or a user-defined record. The command **record** *record-name* specifies the record for the flow monitor.

Predefined records such as netflow-original or netflow include a list of records that have predefined key and non-key fields. For information about predefined records, refer to the *Cisco IOS Flexible NetFlow Command Reference*.

Step 3. **Assign the flow exporter (optional).**

Each flow monitor may have one or more exporters to export the data in the flow monitor cache to an NMS running a NetFlow collector. The command **exporter** *exporter-name* assigns an exporter to this flow monitor.

Step 4. **Set the cache timers.**

The commands **cache timeout active** *active-timer* and **cache timeout inactive** *inactive-timer* define the cache timers.

Example 10-3 provides an example of a flow monitor configuration that is deployed to R11.

Example 10-3 *R11 Flexible NetFlow Monitor*

```
R11
flow monitor NMS-FLOWMONITOR
 description NMS Monitor
 exporter NMS-FLOWEXPORTER
 cache timeout inactive 10
 cache timeout active 60
 record NMS-FLOWRECORD
```

Apply a Flow Monitor to the WAN

A best practice for NetFlow is to monitor all inbound and outbound traffic on the network device. This method covers all traffic regardless of encryption or application optimization. The command **ip flow monitor** *monitor-name* **{input | output}** assigns a flow monitor to an interface (physical or logical).

Example 10-4 provides an example of a flow monitor assigned to tunnel 100 on R11.

Example 10-4 *Applying a Flexible NetFlow Monitor to the WAN*

```
R11
interface Tunnel100
 ip flow monitor MONITOR-STATS input
 ip flow monitor MONITOR-STATS output
```

Example 10-5 provides the complete configuration for a sample FNF configuration. Figure 10-11 depicts the dependencies for Example 10-5 and FNF when there are multiple flow records, exporters, and monitors.

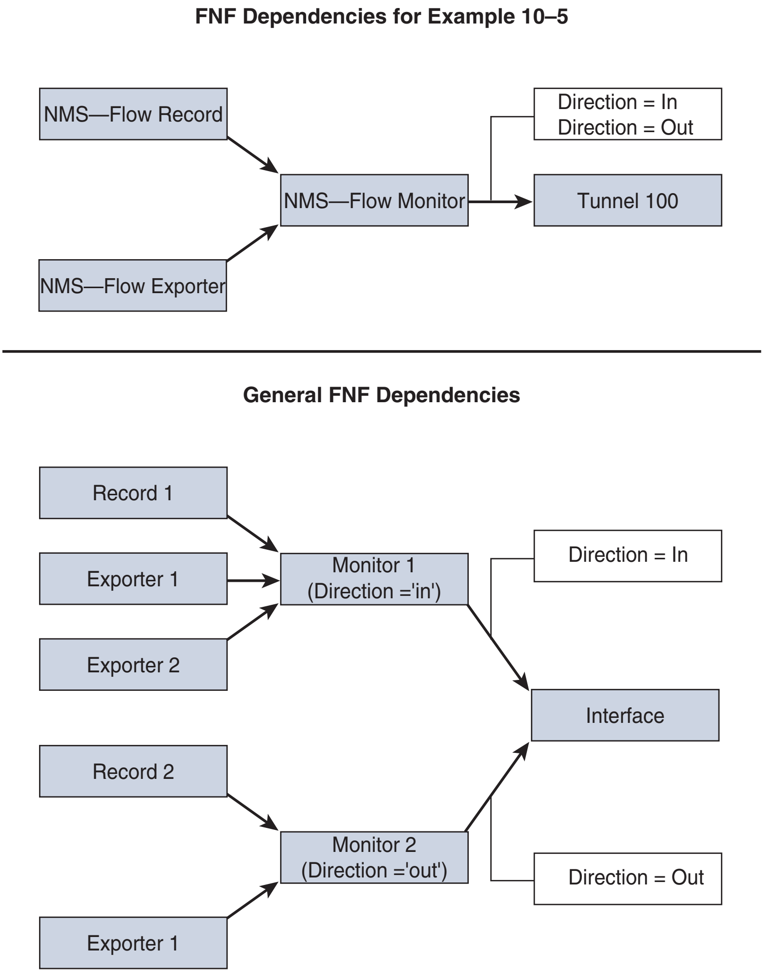


Figure 10-11 *Flexible NetFlow Configuration Principle*

Example 10-5 *Complete Sample Flexible NetFlow Monitor*

```
R11
! Creation of the Flow
flow record NMS-FLOWRECORD
 description NMS Record
 match ipv4 tos
 match ipv4 protocol
 match ipv4 source address
 match ipv4 destination address
 match transport source-port
 match transport destination-port
 match interface input
 match flow direction
 collect routing source as
 collect routing destination as
 collect routing next-hop address ipv4
 collect ipv4 dscp
 collect ipv4 id
 collect ipv4 source prefix
 collect ipv4 source mask
 collect ipv4 destination mask
 collect transport tcp flags
 collect interface output
 collect flow sampler
 collect counter bytes
 collect counter packets
 collect timestamp sys-uptime first
 collect timestamp sys-uptime last
 collect application name
!
! Creation of the Flow Exporter
flow exporter NMS-FLOWEXPORTER
 destination 10.151.1.95
 source Loopback0
 transport udp 2055
 export-protocol ipfix
 option interface-table
 option c3pl-class-table timeout 300
 option c3pl-policy-table timeout 300
 option application-table
!
! Creation of Flow Monitor
flow monitor NMS-FLOWMONITOR
 description NMS Monitor
 exporter NMS-FLOWEXPORTER
 cache timeout inactive 10
 cache timeout active 60
 record NMS-FLOWRECORD

! Association of the Flow Monitor to the interface in both directions
interface Tunnel100
 ip flow monitor MONITOR-STATS input
ip flow monitor MONITOR-STATS output
```

Flexible NetFlow for Application Visibility

Flexible NetFlow can be used for a variety of use cases ranging from simple statistics gathering to full Application Visibility. The following examples are given to illustrate flow record definition flexibility. Based on customer requirements, one or more flow records can be defined to collect what is necessary.

Use Case 1: Flow Statistics

Example 10-6 illustrates a flow record used to collect usage aggregated by source/destination IP addresses and ports.

Example 10-6 *R11 Flexible NetFlow Record Statistics*

```
R11
flow record RECORD-FLOW-STATS
 match ipv4 dscp
 match ipv4 protocol
 match ipv4 source address
 match ipv4 destination address
 match transport source-port
 match transport destination-port
 match interface input
 match flow direction
 collect interface output
 collect counter bytes long
 collect counter packets
 collect routing next-hop address ipv4
```

Note Watch out for direction. In FNF, direction is not exported by default. In a large-scale aggregation, tracking and storing every single flow severely limits the scalability of the solution. Advanced filtering is available with Performance Monitor.

Use Case 2: Application Client/Server Statistics

Example 10-7 illustrates a flow record to collect usage aggregated by application name and source/destination address. The **match application name** command calls on NBAR2, and the optional **account-on-resolution** keyword provides accurate accounting until classification. The record is added to the cache only when the application classification is final, therefore representing an accurate accounting for the entire flow.

Example 10-7 *R11 Flexible NetFlow Record Application Client/Server Statistics*

```
R11
flow record RECORD-APPLICATION-CLIENT-SERVER-STATS
 match ipv4 dscp
 match ipv4 protocol
 match ipv4 source address
 match ipv4 destination address
 match interface input
 match flow direction
 match application name [account-on-resolution]
 collect interface output
 collect counter bytes long
 collect counter packets
 collect routing next-hop address ipv4
```

Use Case 3: Application Usage

Example 10-8 illustrates a flow record defined to collect usage aggregated by application, flow direction, and interface.

Example 10-8 *R11 Flexible NetFlow Record Application Usage*

```
R11
flow record RECORD-APPLICATION-STATS
 match interface input
 match flow direction
 match application name {account-on-resolution}
 collect interface output
 collect counter bytes long
 collect counter packets
```

Monitoring NetFlow Data

The data stored in the cache of the network device can be viewed in several different ways to address common use cases. These methods are covered briefly to provide examples of how to access the flow data.

View Raw Data Directly on the Router

The simplest method of viewing the NetFlow cache is via the command **show flow monitor** *monitor-name* **cache**, which provides a summary of the cache status followed by a series of individual cache entries.

The FNF configuration shown in Example 10-9 is a basic example used to check existing flows on branch routers and get the DSCP used as well as the next hop.

Example 10-9 *R31 Basic Flexible NetFlow Example*

```
R31
flow record RECORD-STATS
 match ipv4 dscp
 match ipv4 protocol
 match ipv4 source address
 match ipv4 destination address
 match transport source-port
 match transport destination-port
 match interface input
 match flow direction
 collect routing next-hop address ipv4
 collect counter bytes
!
!
flow monitor MONITOR-STATS
 cache timeout inactive 60
 cache timeout active 60
 cache timeout update 1
 record RECORD-STATS
!
interface Tunnel 100
 ip flow monitor MONITOR-STATS input
 ip flow monitor MONITOR-STATS output
!
interface Tunnel 200
 ip flow monitor MONITOR-STATS input
 ip flow monitor MONITOR-STATS output
```

Example 10-10 illustrates the use of the command **show flow monitor** *monitor-name* **cache.**

Example 10-10 *R31 Basic Flexible NetFlow Cache*

```
R31-Spoke# show flow monitor MONITOR-STATS cache
  Cache type:                               Normal
  Cache size:                                 4096
  Current entries:                              30
  High Watermark:                               33

  Flows added:                              528193
  Flows aged:                               528163
```

```
    - Active timeout          (60 secs)     528163
    - Inactive timeout        (60 secs)          0
    - Event aged                                 0
    - Watermark aged                             0
    - Emergency aged                             0

IPV4 SOURCE ADDRESS:       10.1.100.10
IPV4 DESTINATION ADDRESS:  10.3.3.100
TRNS SOURCE PORT:          1967
TRNS DESTINATION PORT:     30000
INTERFACE INPUT:           Tu100
FLOW DIRECTION:            Input
IP DSCP:                   0x2E
IP PROTOCOL:               17
ipv4 next hop address:     10.3.3.100
counter bytes:             104

IPV4 SOURCE ADDRESS:       10.3.3.103
IPV4 DESTINATION ADDRESS:  10.4.4.103
TRNS SOURCE PORT:          30000
TRNS DESTINATION PORT:     1967
INTERFACE INPUT:           Et1/0
FLOW DIRECTION:            Output
IP DSCP:                   0x2E
IP PROTOCOL:               17
ipv4 next hop address:     192.168.100.11
counter bytes:             880

IPV4 SOURCE ADDRESS:       10.1.102.10
IPV4 DESTINATION ADDRESS:  10.3.3.102
TRNS SOURCE PORT:          1967
TRNS DESTINATION PORT:     7000
INTERFACE INPUT:           Tu100
FLOW DIRECTION:            Input
IP DSCP:                   0x12
IP PROTOCOL:               17
ipv4 next hop address:     0.0.0.0
counter bytes:             624

[Output omitted for brevity]
```

Note Voice traffic is configured to use DSCP EF (0×2E). Critical applications are configured to use DSCP AF21 (0×12).

There are a couple of options available to format the output; the most useful one is to display a flow table. Example 10-11 illustrates the use of the command **show flow monitor** *monitor-name* **cache format table.**

Example 10-11 *R31 Basic Flexible NetFlow Cache Format Table*

```
R31-Spoke# show flow monitor MONITOR-STATS cache format table
  Cache type:                               Normal
  Cache size:                                 4096
  Current entries:                              35
  High Watermark:                               38

  Flows added:                                2567
  Flows aged:                                 2532
    - Active timeout          (60 secs)       2532
    - Inactive timeout        (60 secs)          0
    - Event aged                                 0
    - Watermark aged                             0
    - Emergency aged                             0

IPV4 SRC ADDR    IPV4 DST ADDR    TRNS SRC PORT  TRNS DST PORT  INTF INPUT
  FLOW DIRN  IP DSCP  IP PROT  ipv4 next hop addr       bytes
===============  ===============  =============  =============
====================  =========   =======  =======  ==================  ==========
10.1.100.10      10.3.3.100               20000          30000  Tu100
  Input      0x2E          17  10.3.3.100              173460
10.3.3.100       10.1.100.10              30000          20000  Gi0/3
  Output     0x2E          17  192.168.100.11          114120
10.3.3.101       10.1.101.10                  0           2048  Gi0/3
  Output     0x00           1  192.168.200.12            4000
10.3.3.103       10.4.4.103               30000          20000  Gi0/3
  Output     0x2E          17  192.168.100.41          111240
10.4.4.103       10.3.3.103               20000          30000  Tu100
  Input      0x2E          17  10.3.3.103              111180
10.4.4.103       10.3.3.103               30000          20000  Tu100
  Input      0x2E          17  10.3.3.103              111060
10.3.3.103       10.4.4.103               20000          30000  Gi0/3
  Output     0x2E          17  192.168.100.41          111060
10.3.3.103       10.4.4.103               30000           1967  Gi0/3
  Output     0x2E          17  192.168.100.41             160
10.4.4.103       10.3.3.103                1967          30000  Tu100
  Input      0x2E          17  10.3.3.103                 104
10.3.3.102       10.1.102.10               7000           1967  Gi0/3
  Output     0x12          17  192.168.100.11            2160
10.1.102.10      10.3.3.102                1967           7000  Tu100
  Input      0x12          17  0.0.0.0                   1404
10.3.3.102       10.1.102.10               7000           7000  Gi0/3
  Output     0x12           6  192.168.100.11            4428
```

```
10.1.102.10      10.3.3.102                7000           7000  Tu100
  Input      0x12           6  10.3.3.102                3348
10.1.101.10      10.3.3.101                   0              0  Tu200
  Input      0x00           1  10.3.3.101                2100
10.4.4.103       10.3.3.103               30000           1967  Tu100
  Input      0x2E          17  10.3.3.103                  80
10.3.3.103       10.4.4.103                1967          30000  Gi0/3
  Output     0x2E          17  192.168.100.41              52
10.3.3.100       10.1.100.10              30000           1967  Gi0/3
  Output     0x2E          17  192.168.100.11              80
10.1.100.10      10.3.3.100                1967          30000  Tu100
  Input      0x2E          17  10.3.3.100                  52
```

If the specific fields are known, such as the source/destination IP address or the TCP/UDP port number, the cache can be searched for exact matches, or regular expressions can be used for broader match criteria.

Example 10-12 illustrates the use of the command **show flow monitor** *monitor-name* **cache filter** available on IOS with a filter on the destination port. Voice traffic is running on port 30000, and the command shows how to verify that RTP streams have the proper QoS DSCP settings.

Example 10-12 *R31 Basic Flexible NetFlow Cache with Destination Port Filter Option*

```
R31-Spoke# show flow monitor MONITOR-STATS cache filter transport destination-port
  30000
  Cache type:                               Normal
  Cache size:                                 4096
  Current entries:                              29
  High Watermark:                               33

  Flows added:                              528550
  Flows aged:                               528521
    - Active timeout          (60 secs)     528521
    - Inactive timeout        (60 secs)          0
    - Event aged                                 0
    - Watermark aged                             0
    - Emergency aged                             0

IPV4 SOURCE ADDRESS:       10.1.100.10
IPV4 DESTINATION ADDRESS:  10.3.3.100
TRNS SOURCE PORT:          20000
TRNS DESTINATION PORT:     30000
INTERFACE INPUT:           Tu100
FLOW DIRECTION:            Input
IP DSCP:                   0x2E
IP PROTOCOL:               17
ipv4 next hop address:     10.3.3.100
counter bytes:             50940

IPV4 SOURCE ADDRESS:       10.1.100.10
IPV4 DESTINATION ADDRESS:  10.3.3.100
TRNS SOURCE PORT:          1967
TRNS DESTINATION PORT:     30000
INTERFACE INPUT:           Tu100
FLOW DIRECTION:            Input
IP DSCP:                   0x2E
IP PROTOCOL:               17
ipv4 next hop address:     10.3.3.100
counter bytes:             52

Matched 2 flows
```

Note The filter option is available only on IOS. Voice traffic is configured to use DSCP EF (0×2E).

View Reports on NetFlow Collectors

Although viewing the NetFlow cache directly on the router can be useful for real-time troubleshooting, it is recommended to use an external NetFlow collector that can build and display reports. One key advantage of using an external collector is the ability to aggregate and correlate flow data collected across multiple network devices. The NetFlow data, cached locally on the network device, is relatively short lived and is typically aged out by new flows within minutes. An external collector is essential to maintain a long-term view of the traffic patterns on a network.

Flexible NetFlow Summary

Flexible NetFlow is a useful network management tool for identifying, isolating, and correcting network problems across multiple devices, such as a misconfigured QoS policy. NetFlow applications can generate multiple reports and filter down to an individual conversation between two endpoints if the FNF record is configured with IP address as a key field. As seen in the previous examples, a proper FNF record definition is critical and is based on customer requirements.

Evolution to Performance Monitor

Performance Monitor is the monitoring engine used in IWAN for Application Visibility and supports both IPv4 and IPv6. As an example, Performance Routing (PfR) instantiates three Performance Monitor instances (PMIs) that provide an aggregated view of performance between sites. To get the full view of the performance metrics per flow, Performance Monitor can be configured and tuned according to the enterprise needs. Performance Monitor provides a wide variety of network metrics data. The monitoring agent collects

- Traffic statistics such as bandwidth usage
- TCP performance metrics such as response time and latency
- RTP performance metrics such as packet loss and jitter

Performance Monitor runs on top of Flexible NetFlow (and therefore supports all FNF metrics) and provides the following advantages:

- Many more metrics
- Ability to apply a monitor on specific traffic with class maps
- History
- Event generation
- Ability to react when a threshold is crossed

Performance Monitor can be configured in two ways:

- **Explicitly configured** with a model that is very similar to the one used in FNF but brings additional flexibility with a class map that allows filtering of traffic. The configuration can be long but provides a very flexible model.
- **Automatically configured** based on predefined templates. This configuration, called *Easy Performance Monitor (ezPM)*, drastically simplifies the configuration and is the recommended option because it includes Cisco validated records, monitors, class maps, and policy maps. The configuration consists of only a couple of lines.

Principles

Performance Monitor is a passive monitoring engine that is an integral part of Cisco IOS software and collects and measures performance data. Performance Monitor also uses the concept of flows, similar to Flexible NetFlow, but provides much more information. A *flow* is defined as a stream of packets. The definition of a flow is based on the definition of key fields. Performance Monitor includes *metric providers* that collect data and a *Metric Mediation Agent (MMA)* that correlates data from various metric providers and exposes them to the user or NMS.

The MMA manages, correlates, and aggregates metrics from different metric providers. It performs the following functions:

- Controls the traffic monitoring and filtering policy
- Correlates data from multiple metric providers into the same record
- Aggregates metrics
- Supports history and alert functions

Metric providers collect and calculate metrics and provide them to the MMA for correlation. There are a variety of metric providers; some collect simple, stateless metrics per packet, and other more complex metric providers track states and collect metrics per flow, transforming the metrics at the time of export and making sophisticated calculations. These transformations may require punting of records to the route processor (RP) before the metrics are exported to the management and reporting system.

The MMA compiles multiple metric providers of different types into the same record.

The exporter is used to export records to the NMS. Metrics are aggregated and exported in NetFlow v9 (RFC 3954) or IPFIX (RFC 7011) format to a management and reporting package.

Figure 10-12 illustrates the architecture of Performance Monitor.

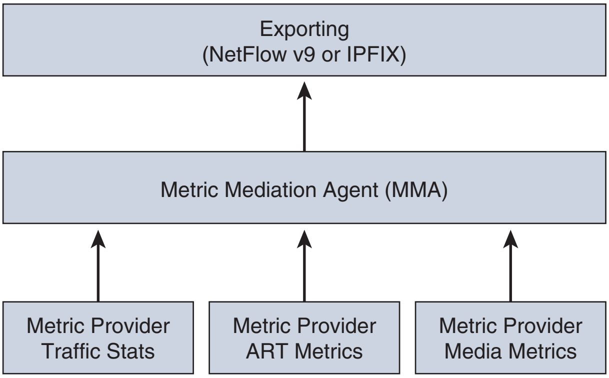


Figure 10-12 *Performance Monitor Architecture*

Performance Monitor includes the following major providers:

- **Traffic statistics:** Collects the traditional metrics collected by Flexible NetFlow
- **ART metrics:** Collects performance metrics for TCP-based applications
- **Media metrics:** Collects performance metrics for RTP-based applications

Performance Monitor collects information with different levels of granularity based on the configuration used or the profile used with ezPM. A higher level of granularity has an effect on memory and CPU. Table 10-1 summarizes some of the key differences between the two main ezPM profiles.

Table 10-1 *ezPM Profile Comparison*

Application Performance Profile	Application Statistics Profile
Use Cases	
■ All use cases ■ Performance metrics per application ■ Field extraction (URL)	■ Network/site/device/link planning ■ Top applications ■ Clients/servers
Reporting	
■ ART and media performance metrics ■ URL and other field extractions ■ Ability to filter a subset of interface traffic and use different reports for different traffic types ■ *account-on-resolution*	■ Bytes, packets, flows reported per application, interface, direction, protocol, and IP version ■ Top clients/servers per application (optional) ■ All interface traffic—no option to filter the monitored traffic

Some use cases may require a combination of high-level and low-level granularity. It may be interesting to configure a low level of granularity to collect interface traffic (with applications and their bandwidth usage only) and a very high level of granularity for a just a small subset of the traffic (for example, to report performance metrics for specific critical applications).

Performance Monitor Configuration Principles

The Performance Monitor configuration includes many of the same basic elements that are in the Flexible NetFlow configuration:

- Flow record—type *performance-monitor*
- Flow monitor—type *performance-monitor*
- Flow exporter
- Class map
- Policy map—type *performance-monitor*

Figure 10-13 illustrates the Performance Monitor configuration. A policy includes one or more classes. Each class has a flow monitor of type *performance-monitor* associated with it, and each flow monitor has a flow record of type *performance-monitor* and an optional flow exporter associated with it. Compared to the Flexible NetFlow configuration, the use of a class map allows a better and finer filtering. For example, Performance Monitor can be configured to track performance only for voice traffic.

The following steps are required to enable Performance Monitor:

Step 1. **Create a flow exporter for each external NetFlow collector.**

Flow exporters export the data in the flow monitor cache to a remote system running a NetFlow collector for analysis and storage. Flow exporters are created as separate entities in the configuration. Flow exporters are assigned to flow monitors to provide data export capability for the flow monitors. Several flow monitors can be defined to provide several export destinations.

Step 2. **Configure a flow record to specify the key and non-key fields that will be monitored.**

A flow record is configured using the **match** and **collect** commands. A flow exporter can optionally be configured to specify the export destination. For Cisco Performance Monitor, a *performance-monitor* type flow record is required.

Step 3. **Configure a flow monitor that includes the flow record and flow exporter.**

For Cisco Performance Monitor, a *performance-monitor* type flow monitor is required.

Step 4. **Configure a class map to specify the filtering criteria using the class-map command.**

Step 5. **Configure a policy map to include one or more classes and one or more *performance-monitor* type flow monitors using the policy-map command.**

For Cisco Performance Monitor, *performance-monitor* type policies are required.

Step 6. **Associate a *performance-monitor* type policy to the appropriate interface using the service-policy type *performance-monitor* command.**

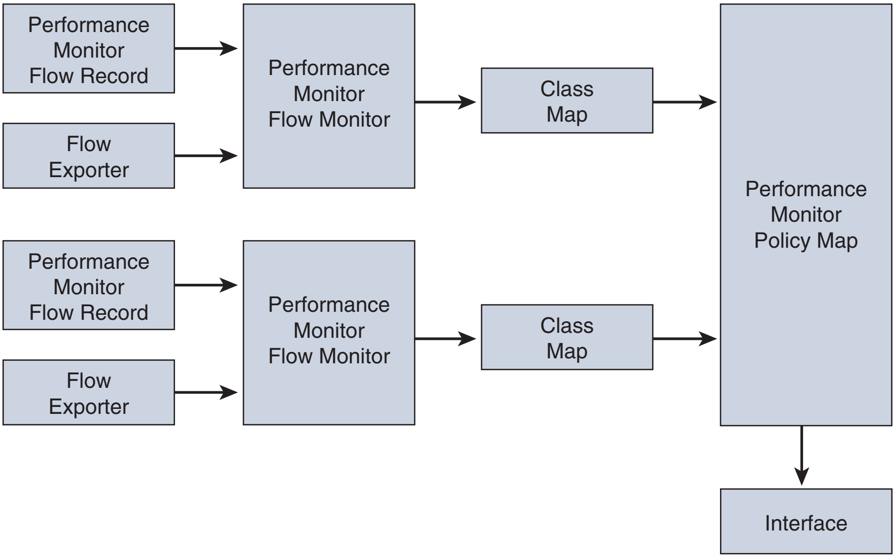

Performance Monitor Flow Record
Flow Exporter
Performance Monitor Flow Monitor
Class Map
Performance Monitor Flow Record
Flow Exporter
Performance Monitor Flow Monitor
Class Map
Performance Monitor Policy Map
Interface

Figure 10-13 *Performance Monitor Configuration Principles*

Example 10-13 illustrates the use of Performance Monitor to collect statistics for IPv6 traffic and ART metrics for IPv6 TCP traffic.

Example 10-13 *Performance Monitor Configuration Example*

```
R11
! Flow Exporter
flow exporter AVC_FLOW_EXPORT
 transport udp 4825
 export-protocol ipfix
 source Loopback0
 dscp 57
 template data timeout 300
 option c3pl-class-table timeout 300
 option c3pl-policy-table timeout 300
!
! Flow Record for IPv6 Traffic
flow record type performance-monitor AVC_RECORD_IPV6
 match connection client ipv6 address
 match connection server ipv6 address
 match connection server transport port
 match ipv6 protocol
 collect interface input
 collect interface output
 collect connection client counter bytes network long
 collect connection server counter bytes network long
 collect connection client counter packets long
 collect connection server counter packets long
 collect ipv6 dscp
 collect policy qos classification hierarchy
 collect connection delay response client-to-server sum
 collect connection new-connections
 collect connection server counter responses
 collect connection initiator
 collect connection transaction counter complete
 collect application name
!
! Flow Record for IPv6 TCP Traffic
flow record type performance-monitor AVC_RECORD_IPV6_TCP
 ! Key fields for Performance-monitor
 match connection client ipv6 address
 match connection server ipv6 address
 match connection server transport port
 match ipv6 protocol
 collect interface input
 collect interface output
 collect connection client counter bytes network long
 collect connection server counter bytes network long
 collect connection client counter packets long
 collect connection server counter packets long
 collect ipv6 dscp
 collect policy qos classification hierarchy
 collect connection delay network to-server sum
 collect connection delay network to-client sum
 collect connection delay application sum
 collect connection delay response to-server histogram bucket1
 collect connection delay response to-server histogram bucket2
 collect connection delay response to-server histogram bucket3
 collect connection delay response to-server histogram bucket4
 collect connection delay response to-server histogram bucket5
 collect connection delay response to-server histogram bucket6
 collect connection delay response to-server histogram bucket7
 collect connection delay response to-server histogram late
 collect connection delay response client-to-server sum
 collect connection new-connections
 collect connection server counter responses
 collect connection initiator
 collect connection transaction counter complete
 collect application name
```

```
!
! Flow Monitor for IPv6 TCP Traffic
flow monitor type performance-monitor AVC_FLOW_MONITOR_IPV6
 record AVC_RECORD_IPV6
 exporter AVC_FLOW_EXPORT
 cache timeout synchronized 300
 cache type synchronized
 history size 0
 cache entries 1000000
!
! Flow Monitor for IPv6 TCP Traffic
flow monitor type performance-monitor AVC_FLOW_MONITOR_IPV6_TCP
 record AVC_RECORD_IPV6_TCP
 exporter AVC_FLOW_EXPORT
 cache timeout synchronized 300
 cache type synchronized
 history size 0
 cache entries 1000000
!
! Define Class-Map to match IPv6 and TCP IPv6 traffic
ipv6 access-list ACL-IPV6
 permit ipv6 any any
ipv6 access-list ACL-IPV6_TCP
 permit tcp any any
!
class-map match-any CLASS-ALLOW_CUSTOMER_TRAFFIC_IPV6
 match access-group name ACL-IPV6
class-map match-any CLASS-ALLOW_CUSTOMER_TRAFFIC_IPV6_TCP
 match access-group name ACL-IPV6_TCP
!
! Define policy-map using class-map to match all traffic
policy-map type performance-monitor POLICY-PERF_MON_POLICY_INPUT
 class CLASS-ALLOW_CUSTOMER_TRAFFIC_IPV6_TCP
  flow monitor AVC_FLOW_MONITOR_IPV6_TCP
 class CLASS-ALLOW_CUSTOMER_TRAFFIC_IPV6
  flow monitor AVC_FLOW_MONITOR_IPV6
!
policy-map type performance-monitor POLICY-PERF_MON_POLICY_OUTPUT
 class CLASS-ALLOW_CUSTOMER_TRAFFIC_IPV6_TCP
  flow monitor AVC_FLOW_MONITOR_IPV6_TCP
 class CLASS-ALLOW_CUSTOMER_TRAFFIC_IPV6
  flow monitor AVC_FLOW_MONITOR_IPV6
!
interface GigabitEthernet0/0/1
 service-policy type performance-monitor input POLICY-PERF_MON_POLICY_INPUT
 service-policy type performance-monitor output POLICY-PERF_MON_POLICY_OUTPUT
```

The Performance Monitor configuration is extremely flexible and provides a large set of options and fields, but it can also quickly become complex. Example 10-13 is an example with just a subset of the traffic. This is the reason why it is recommended to use a profile-based approach with ezPM.

Easy Performance Monitor (ezPM)

As seen in the previous section, the Performance Monitor configuration can quickly become complex with many options and parameters. The ezPM feature provides a simple and effective method of provisioning monitors and is based on templates that are validated as part of the IWAN solution tests. ezPM adds functions without affecting the traditional, full-featured Performance Monitor configuration model for provisioning monitors, but it does not provide the full flexibility of the traditional Performance Monitor configuration model.

ezPM provides *profiles* that represent typical deployment scenarios. ezPM profiles include the following:

- Application Statistics
- Application Performance
- Application Experience (legacy only)

After selecting a profile and specifying a small number of parameters, ezPM provides the remaining provisioning details, which greatly simplify the Performance Monitor configuration. There is also an option to display the entire configuration.

The Application Statistics profile is appropriate for the following use cases:

- Common deployments, capacity planning
- Aggregated application-level statistics (examples: top *N* applications, bandwidth per application, top clients/servers per application)
- Per-interface/application statistics
- Per-client/server/application/interface statistics

The Application Performance profile is appropriate when performance metrics and maximum information are required:

- Common deployments, capacity planning, but with more details than the Application Statistics profile
- Aggregated application-level statistics (examples: top *N* applications, bandwidth per application, top clients/servers per application)
- All application statistics with the main addition of application performance metrics
- Finer granularity

The Application Experience profile remains available only to support legacy configurations, but it is recommended to use the improved Application Performance profile for new configurations.

Application Statistics Profile

Application Statistics is a simple profile used to collect application statistics. It does not report performance statistics in contrast to the Application Performance profile. This profile is a good means to understanding the enterprise traffic profile and discovering applications and their bandwidth usage.

The Application Statistics profile provides two different traffic monitors, *application-stats* and *application-client-server-stats*, described in Table 10-2. The monitors operate on all IPv4 and IPv6 traffic and are based on a *coarse-grain* model.

Table 10-2 *Application Statistics Traffic Monitors*

Monitor Name	Default Traffic Classification
application-stats	All IPv4 and IPv6 traffic; traffic statistics per application
application-client-server-stats	All IPv4 and IPv6 traffic; traffic statistics per application, client, and server (superset of Application Statistics)

The *application-stats* monitor collects data (and mostly bandwidth information) per application. All metrics are aggregated per application. This monitor does not keep individual client/source information. The *application-client-server-stats* monitor on the other end collects metrics of every client/server in addition to application name and bandwidth information.

The Application Statistics profile operates with only one monitor, because the *application-client-server-stats* monitor reports the same information as the *application-stats* monitor, plus additional information.

Application Performance Profile

The Application Performance profile is an improved form of the Application Experience profile, optimized for maximum performance and still exporting the maximum possible amount of available information for monitored traffic.

The Application Performance profile enables the use of five different traffic monitors, described in Table 10-3. The monitors operate on all IPv4 and IPv6 traffic.

Table 10-3 *Application Performance Traffic Monitors*

Monitor Name	Default Traffic Classification
ART	All TCP applications
URL	HTTP applications
Media	RTP applications using transport hierarchy
application-client-server-stats	Remaining TCP/UDP traffic not matching other classifications
application-stats	**Cisco IOS:** Remaining TCP/UDP/ICMP traffic **Cisco IOS XE:** Remaining IP traffic

Application Experience Profile

The Application Experience profile enables the use of five different traffic monitors, described in Table 10-4. The monitors operate on all IPv4 and IPv6 traffic.

Table 10-4 *Application Experience Traffic Monitors*

Monitor Name	Default Traffic Classification
ART	All TCP applications
URL	HTTP applications
Media	RTP applications
conversation-traffic-stats	Remaining traffic not matching other classifications
application-traffic-stats	DNS or DHT

It is recommended to use the new Application Performance profile described in the previous section. The Application Experience profile remains available only for backward compatibility.

ezPM Configuration Steps

ezPM provides a simple and effective method of provisioning monitors based on profiles. The configuration is straightforward but also allows the use of parameterized options such as class-replace, ipv4/ipv6, and so on. The cache size is automatically set based on the platform used, as opposed to Flexible NetFlow which requires explicit configuration. The following steps are required to create a Performance Monitor instance and assign it to an interface using ezPM:

Step 1. **Choose a profile and create a Performance Monitor context.**

The command **performance monitor context** context-name **profile** {application-statistics | **application performance**} creates a Performance Monitor context.

Step 2. **Define the exporter.**

Each context may have one or more exporters to export the data in the Performance Monitor cache to an NMS running a NetFlow collector. The command **exporter destination** {*hostname* | *ip-address*} **source interface** *interface-id* [**port** *port-value* **transport udp vrf** *vrf-name*] defines the exporter (collector running on the NMS).

Step 3. **Define the monitor to use.**

Choose the monitor to enable based on the profile used. The command **traffic monitor** *traffic-monitor-name* is used to assign an exporter to this flow monitor.

Options for traffic monitor type include the following:

- Application Performance profile:
  - *url*
  - *application-response-time*
  - *application-traffic-stats*
  - *conversation-traffic-stats*
  - *media*
- Application Statistics profile:
  - *application-stats*
  - *application-client-server-stats*

This command includes more parameters such as the cache size, cache type, IPv4 and/or IPv6, and others. For information about the options available for the **traffic monitor** *traffic-monitor-name* command, refer to the *Cisco Application Visibility and Control User Guide* at www.cisco.com/c/en/us/td/docs/ios/solutions_docs/avc/guide/avc-user-guide.html.

Step 4. **To configure additional traffic monitor parameters, repeat Step 3.**

Step 5. **Assign the Performance Monitor context to the interface.**

The interface is selected with the command **interface** *interface-id*. Then the performance monitor is assigned with the command **performance monitor context** *context-name.*

Example 10-14 provides an example of a Performance Monitor with the Application Statistics profile where the *application-client-server-stats* monitor is used.

Example 10-14 *ezPM Configuration Example for Application Statistics*

```
R11-Hub
! Easy performance monitor context
!
performance monitor context MYTEST profile application-statistics
 exporter destination 10.1.1.200 source Loopback0
 traffic-monitor application-client-server-stats
!
! Interface attachments
interface GigabitEthernet0/0/2
 performance monitor context MYTEST
```

The command **show performance-monitor context** *context-name* **configuration** gives the configuration generated by ezPM. Example 10-15 shows the output for the configuration used in Example 10-14.

Example 10-15 *ezPM Equivalent Configuration for Application Statistics Profile and Monitor* application-client-server-stats

```
R31-Spoke# show performance monitor context MYTEST configuration
!===============================================================================
!                  Equivalent Configuration of Context MYTEST                  !
!===============================================================================
!Access Lists
!=============
!Class-maps
!===========
class-map match-all MYTEST-app_cs_stats_ipv4
 match protocol ip
!
class-map match-all MYTEST-app_cs_stats_ipv6
 match protocol ipv6
!
!Samplers
!=========
!Records and Monitors
!=====================
!
flow record type performance-monitor MYTEST-app_cs_stats_ipv4
 description ezPM record
 match ipv4 version
 match ipv4 protocol
 match application name account-on-resolution
 match connection client ipv4 address
 match connection server ipv4 address
 match connection server transport port
 match flow observation point
 collect routing vrf input
 collect ipv4 dscp
 collect flow direction
 collect timestamp sys-uptime first
 collect timestamp sys-uptime last
 collect connection initiator
 collect connection new-connections
 collect connection sum-duration
 collect connection server counter packets long
 collect connection client counter packets long
 collect connection server counter bytes network long
 collect connection client counter bytes network long
!
flow monitor type performance-monitor MYTEST-app_cs_stats_ipv4
 record MYTEST-app_cs_stats_ipv4
 cache entries 7500
 cache timeout synchronized 60 export-spread 15
 history size 1
!
flow record type performance-monitor MYTEST-app_cs_stats_ipv6
 description ezPM record
 match ipv6 version
 match ipv6 protocol
 match application name account-on-resolution
 match connection client ipv6 address
 match connection server transport port
 match connection server ipv6 address
 match flow observation point
 collect routing vrf input
 collect ipv6 dscp
 collect flow direction
 collect timestamp sys-uptime first
 collect timestamp sys-uptime last
 collect connection initiator
 collect connection new-connections
 collect connection sum-duration
 collect connection server counter packets long
 collect connection client counter packets long
 collect connection server counter bytes network long
 collect connection client counter bytes network long
!
flow monitor type performance-monitor MYTEST-app_cs_stats_ipv6
 record MYTEST-app_cs_stats_ipv6
 cache entries 7500
 cache timeout synchronized 60 export-spread 15
 history size 1
!
!Policy-maps
!============
policy-map type performance-monitor MYTEST-in
 parameter default account-on-resolution
 class MYTEST-app_cs_stats_ipv4
  flow monitor MYTEST-app_cs_stats_ipv4
 class MYTEST-app_cs_stats_ipv6
  flow monitor MYTEST-app_cs_stats_ipv6
!
policy-map type performance-monitor MYTEST-out
 parameter default account-on-resolution
 class MYTEST-app_cs_stats_ipv4
  flow monitor MYTEST-app_cs_stats_ipv4
 class MYTEST-app_cs_stats_ipv6
  flow monitor MYTEST-app_cs_stats_ipv6
!
!Interface Attachments
!======================
interface GigabitEthernet0/0/2
 service-policy type performance-monitor input MYTEST-in
 service-policy type performance-monitor output MYTEST-out
```

Example 10-16 provides an example of a Performance Monitor with the Application Performance profile with all monitors enabled.

Example 10-16 *ezPM Configuration Example for Application Performance Profile*

```
R11-Hub
! Easy performance monitor context
!
performance monitor context MYTEST profile application-performance
 exporter destination 10.1.1.200 source Loopback0
 traffic-monitor all
!
!
! Interface attachments
! --------------------
!
interface GigabitEthernet0/0/2
 performance monitor context MYTEST
```

Example 10-17 provides an example of a Performance Monitor with the Application Performance profile with only a subset of the monitors enabled and only for IPv6 traffic.

Example 10-17 *ezPM Configuration Example for Application Performance Profile for only IPv6 Traffic*

```
R11-Hub
! Easy performance monitor context
performance monitor context MYTEST profile application-performance
 exporter destination 1.2.3.4 source GigabitEthernet0/0/1 port 4739
 traffic-monitor application-response-time ipv6
 traffic-monitor application-client-server-stats ipv6
 traffic-monitor media ipv6
!
! Interface attachments
interface GigabitEthernet0/0/1
 performance monitor context MYTEST
interface GigabitEthernet0/0/2
 performance monitor context MYTEST
```

Monitoring Performance Monitor

Viewing the Performance Monitor cache directly on the router is not easy, and it is recommended to use an external NetFlow collector that can build and display reports. An external collector has the ability to aggregate the information collected across multiple network devices.

Metrics Export

Performance Monitor stores all traffic metrics in a local cache of every router. This NetFlow data, cached locally on the network device, is relatively short lived and is typically aged out by new flows within minutes. An external collector is essential to maintain a long-term view of the traffic patterns on a network. This NetFlow data is exported to an external NMS that includes a NetFlow collector. The protocol used to export these metrics is an IETF standard called NetFlow v9 and IPFIX.

Flow Record, NetFlow v9, and IPFIX

The basic output of Flexible NetFlow or Performance Monitor is a *flow record*. Several different formats for flow records have evolved as NetFlow has matured. The most recent evolutions of the NetFlow flow record format are known as NetFlow version 9 and IPFIX. The distinguishing feature of the NetFlow v9 and IPFIX formats is that they are *template based*. Templates provide an extensible design to the record format, a feature that should allow future enhancements to NetFlow services without requiring concurrent changes to the basic flow record format.

Figure 10-14 illustrates the principle of the NetFlow v9 and IPFIX export protocols with a template-based exporting process.

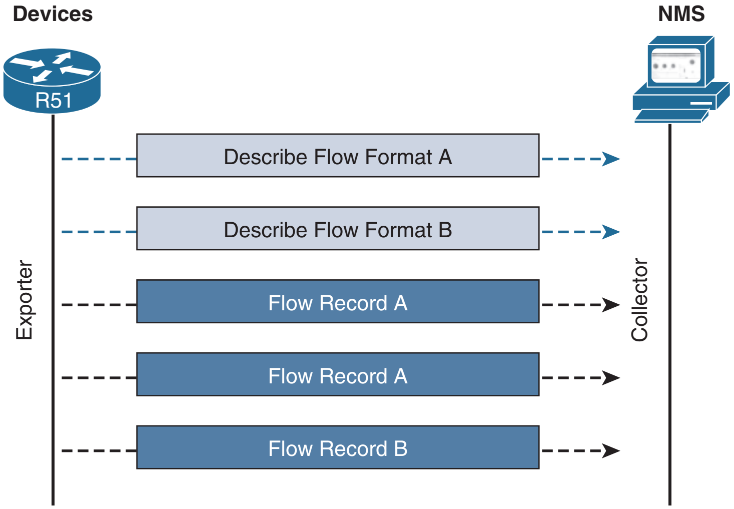


Figure 10-14 *NetFlow v9 and IPFIX Export*

Terminology

One of the difficulties of describing the NetFlow v9 or IPFIX packet format is that many distinctly different but similar-sounding terms are used to describe portions of the NetFlow output. To forestall any confusion, these terms are described here:

- **Export packet:** Built by a device (for example, a router) with monitoring services enabled, this type of packet is addressed to another device (for example, a NetFlow collector). The other device processes the packet (parses, aggregates, and stores information on IP flows).
- **FlowSet:** Following the packet header, an export packet contains information that must be parsed and interpreted by the collector device. FlowSet is a generic term for a collection of records that follow the packet header in an export packet. There are two different types of FlowSets: template and data. An export packet contains one or more FlowSets, and both template and data FlowSets can be mixed within the same export packet.
- **Template ID:** The template ID is a unique number that distinguishes a template record from all other template records produced by the same export device. A collector application that receives export packets from several devices should be aware that uniqueness is not guaranteed across export devices. Thus, the collector should also cache the address of the export device that produced the template ID in order to enforce uniqueness.

- **Template record:** A template record defines the format of subsequent data records that may be received in current or future export packets. Templates are used to describe the type and length of individual fields within a NetFlow data record that match a template ID. A template record within an export packet does not necessarily indicate the format of data records within that same packet. A collector application must cache any template records received, and then parse any data records it encounters by locating the appropriate template record within the cache.

- **Options template:** An options template is a special type of template record used to communicate the format of data related to the NetFlow process.

- **Data record:** A data record provides information about an IP flow that exists on the device that produced an export packet. Each group of data records (that is, each data FlowSet) references a previously transmitted template ID, which can be used to parse the data contained within the records.

- **Options data record:** The options data record is a special type of data record (based on an options template) with a reserved template ID that provides information about the NetFlow process itself. An example is application name to identifier mapping, or interface name to interface identifier mapping.

- **Template FlowSet:** A template FlowSet is a collection of one or more template records that have been grouped together in an export packet.

- **Data FlowSet:** A data FlowSet is a collection of one or more data records that have been grouped together in an export packet.

A NetFlow export format consists of a packet header followed by one or more template or data FlowSets. A template FlowSet provides a description of the fields that will be present in future data FlowSets. These data FlowSets may occur later within the same export packet or in subsequent export packets. Figure 10-15 illustrates the NetFlow v9 and IPFIX packet format and the intermix of template and data FlowSets.

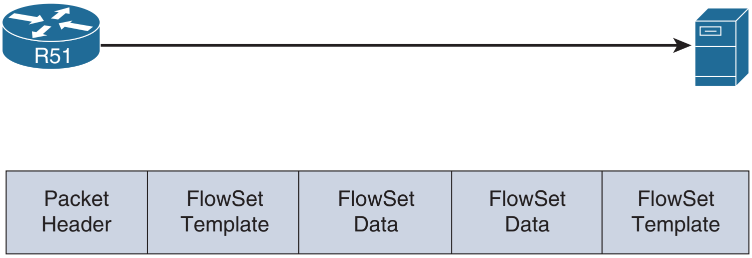


Figure 10-15 *NetFlow v9 and IPFIX Packet Format*

NetFlow Version 9 Packet Header Format (RFC 3954)

The NetFlow v9 exporter sends packets to collectors. These packets include *template FlowSets*, which define the following data formats by a set of field types and their lengths, and *data FlowSets*, which are sets of actual statistics. The NetFlow v9 exporter sends templates separately from (and less frequently than) data sets. There are other sets of template/data FlowSets called *option template/data FlowSets*, by which an exporter can send a collector metadata related to a NetFlow process.

IPFIX Packet Header Format (RFC 7011)

IPFIX is the IETF standard version of the Cisco NetFlow v9 protocol. The packet format is very similar with a few differences and additional features, such as variable-length records. IPFIX is especially required when URL collection is enabled. IPFIX is the default export protocol used in ezPM profiles.

Figure 10-16 illustrates the packet format and the intermix of template and data FlowSets.

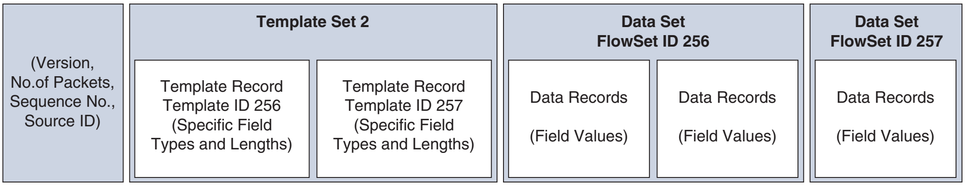

(Version, No.of Packets, Sequence No., Source ID)
Template Set 2
Template Record Template ID 256 (Specific Field Types and Lengths)
Template Record Template ID 257 (Specific Field Types and Lengths)
Data Set FlowSet ID 256
Data Records (Field Values)
Data Records (Field Values)
Data Set FlowSet ID 257
Data Records (Field Values)

Figure 10-16 *Variable-Length Records in NetFlow v9 and IPFIX Packet Format*

Note Set identifiers are used as follows:

- ID = 2 is reserved for template FlowSets.
- ID = 3 is reserved for data FlowSets.

The header follows the same format as prior NetFlow versions so collectors will be backward compatible. Each data record represents one flow.

Monitoring Exports

The command **show flow export templates** displays the template records exported from a router that has Flexible NetFlow or Performance Monitor configured with an exporter.

Example 10-18 provides an example of the templates exported from a router that has the FNF configuration defined in Example 10-6. The options templates are displayed (interface, application names, class map and policy map mapping tables), then the template for the *RECORD-STATS* record that was defined. These templates give the identifier and length of individual fields within a NetFlow data record.

Example 10-18 *Templates Exported*

```
R31-Spoke# show flow export templates
Flow Exporter EXPORTER-LA:
  Client: Option options interface-table
  Exporter Format: IPFIX (Version 10)
  Template ID     : 256
  Source ID       : 0
  Record Size     : 100
  Template layout
  _____________________________________________________________________________
  |                 Field                   |    ID | Ent.ID | Offset |  Size |
  -----------------------------------------------------------------------------
  | INTERFACE INPUT SNMP                    |    10 |        |      0 |     4 |
  | interface name short                    |    82 |        |      4 |    32 |
  | interface name long                     |    83 |        |     36 |    64 |
  -----------------------------------------------------------------------------

  Client: Option options application-name
  Exporter Format: IPFIX (Version 10)
  Template ID     : 257
  Source ID       : 0
  Record Size     : 83
  Template layout
  _____________________________________________________________________________
  |                 Field                   |    ID | Ent.ID | Offset |  Size |
  -----------------------------------------------------------------------------
  | APPLICATION ID                          |    95 |        |      0 |     4 |
  | application name                        |    96 |        |      4 |    24 |
  | application description                 |    94 |        |     28 |    55 |
  -----------------------------------------------------------------------------

  Client: Option classmap option table
  Exporter Format: IPFIX (Version 10)
  Template ID     : 258
  Source ID       : 0
  Record Size     : 300
  Template layout
 _____________________________________________________________________________
  |                 Field                   |    ID | Ent.ID | Offset |  Size |
  -----------------------------------------------------------------------------
  | C3PL CLASS CCE-ID                       |  8233 |      9 |      0 |     4 |
  | c3pl class name                         |  8234 |      9 |      4 |    40 |
  | c3pl class type                         |  8235 |      9 |     44 |   256 |
  -----------------------------------------------------------------------------
```

```
Client: Option policymap option table
Exporter Format: IPFIX (Version 10)
Template ID    : 259
Source ID      : 0
Record Size    : 300
Template layout
_____________________________________________________________________________
|                 Field                   |    ID | Ent.ID | Offset |  Size |
-----------------------------------------------------------------------------
| C3PL POLICY CCE-ID                      |  8236 |      9 |      0 |     4 |
| c3pl policy name                        |  8237 |      9 |      4 |    40 |
| c3pl policy type                        |  8238 |      9 |     44 |   256 |
-----------------------------------------------------------------------------

Client: Flow Monitor MONITOR-STATS
Exporter Format: IPFIX (Version 10)
Template ID    : 260
Source ID      : 0
Record Size    : 27
Template layout
_____________________________________________________________________________
|                 Field                   |    ID | Ent.ID | Offset |  Size |
-----------------------------------------------------------------------------
| ipv4 source address                     |     8 |        |      0 |     4 |
| ipv4 destination address                |    12 |        |      4 |     4 |
| interface input snmp                    |    10 |        |      8 |     4 |
| transport source-port                   |     7 |        |     12 |     2 |
| transport destination-port              |    11 |        |     14 |     2 |
| flow direction                          |    61 |        |     16 |     1 |
| ip dscp                                 |   195 |        |     17 |     1 |
| ip protocol                             |     4 |        |     18 |     1 |
| routing next-hop address ipv4           |    15 |        |     19 |     4 |
| counter bytes                           |     1 |        |     23 |     4 |
-----------------------------------------------------------------------------
```

Monitoring Performance Collection on Network Management Systems

Network management systems used to collect FNF and Performance Monitor data must include a NetFlow collector that supports NetFlow v9 and IPFIX export protocols. These applications offer an effective way to monitor and control application performance leveraging Cisco capabilities.

Deployment Considerations

Application Visibility is commonly used in combination with other components of the Cisco IWAN solution.

Performance Routing

Performance Routing metric collection is based on Performance Monitor. When PfR is enabled on a router, it provides bandwidth statistics per traffic class and performance statistics per channel (refer to Chapter 7, "Introduction to Performance Routing (PfR)," for more details). This gives statistics between the border routers but does not provide granularity per application or connection. To have a deeper view and end-to-end visibility of the performance metrics, Performance Monitor with ezPM can be used to complement the metrics exported with PfR.

Interoperability with WAAS

Cisco WAAS provides WAN optimization and application acceleration. The IWAN architecture takes advantage of both AVC and WAAS WAN optimization. However, it is not desirable to run NBAR on the WAN interface because WAN optimization occurs first, so NBAR sees only compressed traffic and may not be able to properly see application traffic after WAAS is applied.

A general recommendation when NBAR is used is to use QoS without NBAR on the outbound interface, enabling NBAR on the LAN interface only to classify applications and mark the corresponding DSCP value. Figure 10-17 illustrates the use of NBAR with WAAS.

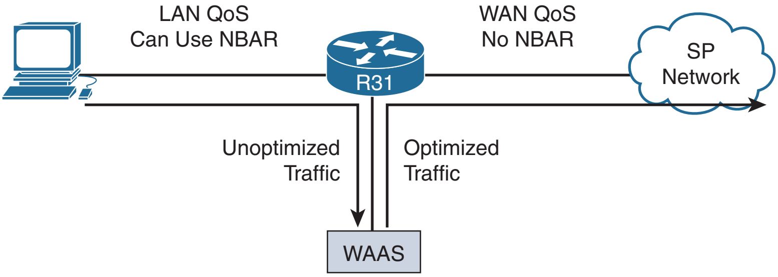


Figure 10-17 *AVC and WAAS WAN Optimization with QoS and NBAR*

The Cisco IWAN Performance Collection solution operates closely with Cisco WAAS, reporting performance on both optimized and unoptimized traffic. Because optimized traffic may be exported twice (pre- and post-WAAS), a new *segment* field is exported within the record in order to describe the type of traffic at the monitoring location. Figure 10-18 illustrates the segment definitions.

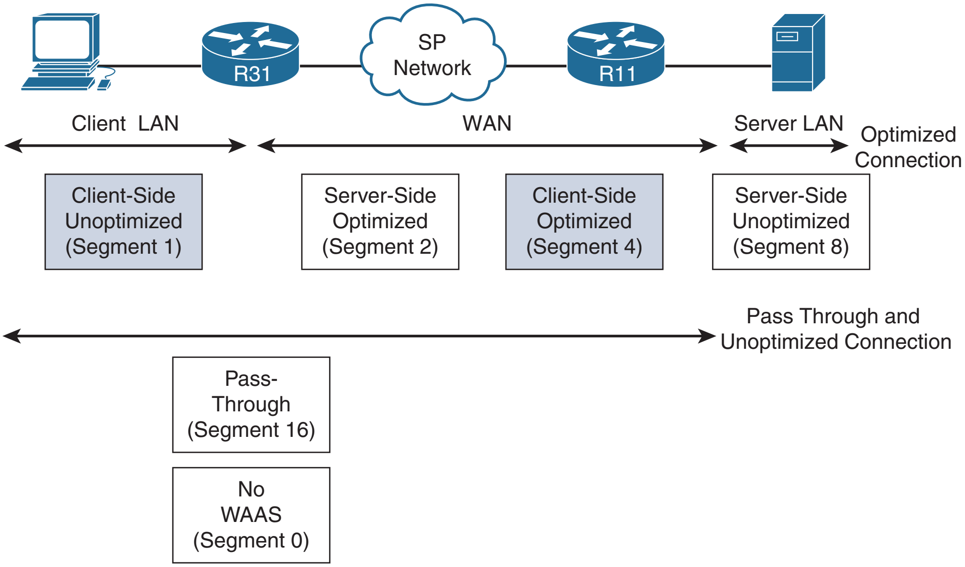
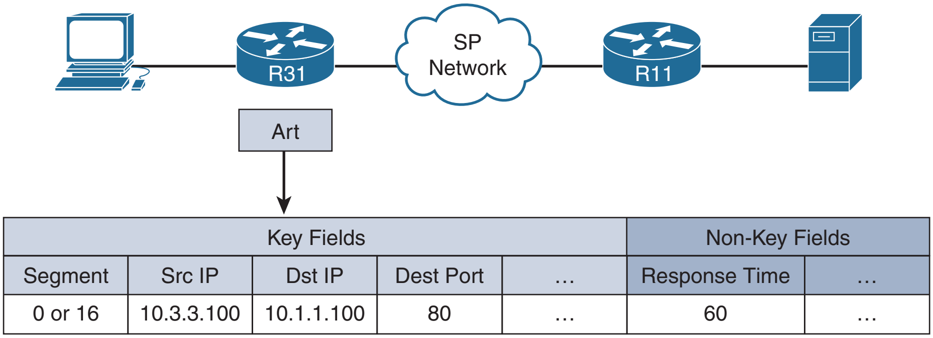

R31
SP Network
R11
Client LAN
WAN
Server LAN
Optimized Connection
Client-Side Unoptimized (Segment 1)
Server-Side Optimized (Segment 2)
Client-Side Optimized (Segment 4)
Server-Side Unoptimized (Segment 8)
Pass Through and Unoptimized Connection
Pass-Through (Segment 16)
No WAAS (Segment 0)

Figure 10-18 *WAAS Segment Definitions*

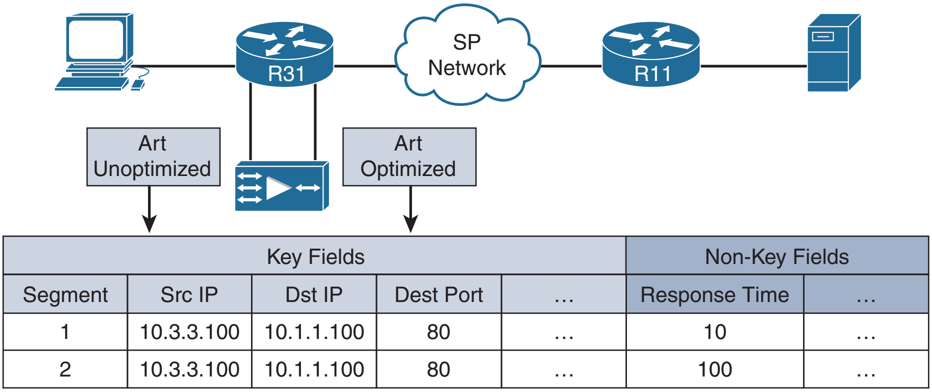
Without WAAS, the router sees only one TCP segment. Figure 10-19 illustrates ART monitoring and export without WAAS.



Key Fields					Non-Key Fields	
Segment	Src IP	Dst IP	Dest Port	...	Response Time	...
0 or 16	10.3.3.100	10.1.1.100	80	...	60	...

Figure 10-19 *Performance Monitoring and Export Without WAAS*

When deploying Cisco WAAS with AppNav as the redirection protocol, AppNav creates two logical interfaces, *AppNav-Uncompress* and *AppNav-Compress*. Performance Monitor monitors traffic on the AppNav logical interfaces and exports two records. Figure 10-20 illustrates Performance Monitor with WAAS and AppNav.



Key Fields					Non-Key Fields	
Segment	Src IP	Dst IP	Dest Port	…	Response Time	…
1	10.3.3.100	10.1.1.100	80	…	10	…
2	10.3.3.100	10.1.1.100	80	…	100	…

Figure 10-20 *Performance Monitoring and Export with AppNav*

If NBAR is enabled on the WAN interface, and WAAS is enabled, AppNav automatically runs NBAR on the *AppNav-Uncompress* virtual interface.

Summary

Application Visibility is a key component of IWAN to add application-level visibility to branch and WAN aggregation routers in an IWAN domain. Application Visibility recognizes and classifies thousands of applications and uses this classification to perform per-application monitoring of bandwidth statistics, of transactional ART metrics, and of media application metrics such as latency and jitter. The per-application metrics are exported via NetFlow v9 and IPFIX for analysis, reporting, and visualization by partner network management systems. Application Visibility is integrated directly into the Cisco devices with a next-generation passive monitoring engine called *Performance Monitor*. ezPM is the recommended configuration to enable Performance Monitor because it provides the Cisco recommended parameters and deployment options as well as a drastically reduced configuration.

Further Reading

Cisco. “Cisco Application Visibility and Control (AVC).” www.cisco.com/go/avc.

Cisco. *Cisco Application Visibility and Control User Guide.* www.cisco.com.

Cisco. “AVC Metric Definitions.” www.cisco.com.

Claise, B., ed. RFC 3954. “Cisco Systems NetFlow Services Export Version 9. IETF, October 2004. www.ietf.org/rfc/rfc3954.txt.

Claise, B., P. Aitken, and N. Ben-Dvora. RFC 6759, “Cisco Systems Export of Application Information in IP Flow Information Export (IPFIX).” IETF, November 2012. https://tools.ietf.org/html/rfc6759.

Claise, B., B. Trammel, and P. Aitken, ed. RFC 7071, “Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information.” IETF, September 2013. www.ietf.org/rfc/rfc7011.txt.

IANA. “Real-Time Transport Protocol (RTP) Parameters .” www.iana.org/assignments/rtp-parameters.

Quittek, J., et al. RFC 5102, “Information Model for IP Flow Information Export.” IETF, January 2008. https://tools.ietf.org/html/rfc5102.

Chapter 11

Introduction to Application Optimization

This chapter covers the following topics:

- Application performance challenges
- Cisco Wide Area Application Services (WAAS)
- Optimization techniques
- Application-specific acceleration

Applications today are becoming increasingly robust and complex compared to applications from 15 years ago, making them more sensitive to network conditions.

The first application performance-limiting factors to examine are in the application stack on the endpoints. The second set of factors consists of those caused by the network. Many applications do not exhibit performance problems caused by these factors while operating in a LAN because they were not accounted for during design. When applications operate in a WAN environment, virtually all applications can be negatively affected from a performance perspective, because most were not designed with the WAN in mind.

The LAN illustrated in Figure 11-1 provides a reliable, low-latency, and high-bandwidth network, whereas the WAN has packet loss, high-latency, and low-bandwidth, which cause delivery challenges.

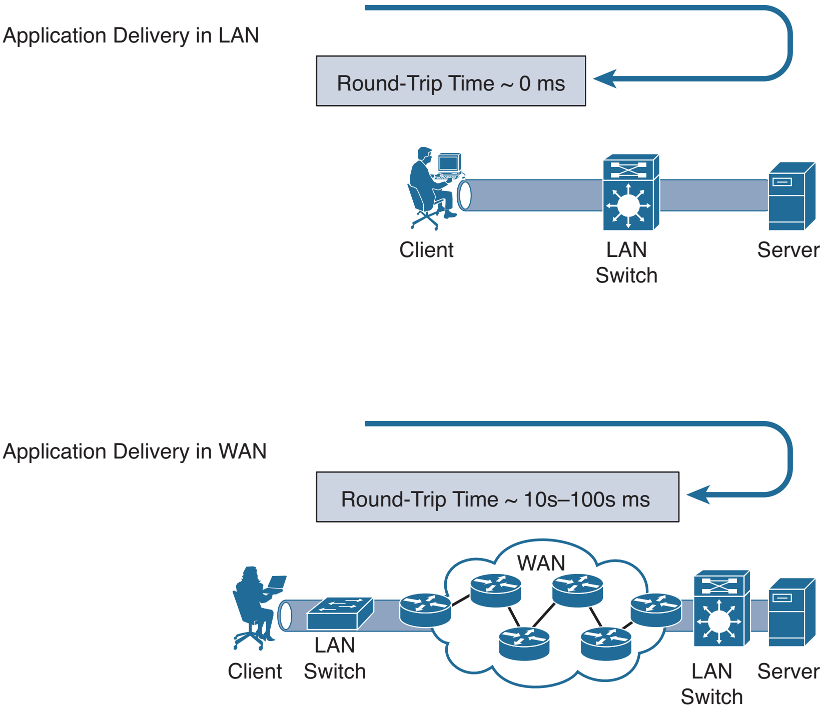

Application Delivery in LAN
Round-Trip Time ~ 0 ms
Client
LAN Switch
Server
Application Delivery in WAN
Round-Trip Time ~ 10s–100s ms
WAN
Client
LAN Switch
LAN Switch
Server

Figure 11-1 *Application Performance Challenges*

Application Behavior

Application behavior can be best understood by examining the *Open Systems Interconnection* (*OSI*) model. The OSI model describes host-to-host communication over a network. Each layer describes a specific function with the notion that a layer can be modified or changed without changes being made to the layer above or below it. The OSI model also provides a structured approach to intercompatibility between vendors.

The OSI model, illustrated in Figure 11-2, describes how data is sent and received over a network. It consists of seven layers: the physical layer, data link layer, network layer, transport layer, session layer, presentation layer, and application layer.

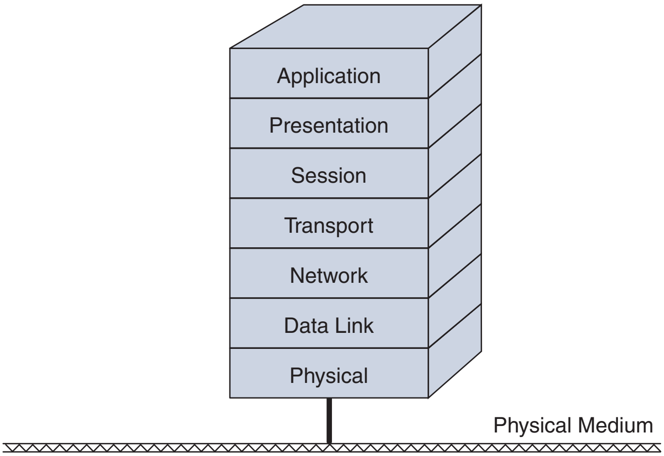


Figure 11-2 *The Open Systems Interconnection (OSI) Model*

Software and hardware can be developed for each layer separately. However, each layer must work with its adjacent layers to transport a message successfully. Each layer performs a specific function, as described in the following list:

- **Application layer (Layer 7):** Interacts with the operating system or application whenever a user chooses to transfer files, read messages, or perform other network-related activities.
- **Presentation layer (Layer 6):** Concerned with the presentation of data. This layer converts the data provided by the application layer into a standard format that the other layers can understand.
- **Session layer (Layer 5):** Handles the setup of the session, data exchanges, and the end of the session.
- **Transport layer (Layer 4):** Deals with the transmission of data between networks. This layer maintains flow control of data and provides error checking and recovery of data between nodes.
- **Network layer (Layer 3):** Splits long messages into smaller bits of data, referred to as packets. The network layer chooses the route data will take and addresses the data for delivery. It adds destination addressing and routing information to enable the packet to travel between nodes.
- **Data link layer (Layer 2):** Moves information from one node or network to another node or network. It performs three specific functions: controlling the physical layer and deciding when to transmit, formatting messages to indicate where they start and end, and detecting and correcting errors that occur during transmission.
- **Physical layer (Layer 1):** Provides a physical connection between nodes.

Any disruption encountered at any layer in the OSI model can affect the layer adjacent to (above or below) it, thus affecting application behavior. For example, changing a network (IP) address invokes a TCP reset in a Telnet session, even though the change occurs at the network layer. Interaction at this layer is complex because the number of operations that can be performed over a proprietary protocol or even a standards-based protocol can be literally in the hundreds or thousands.

This is generally a direct result of the complexity of the application itself and is commonly caused by the need for end-to-end state management between the client and the server to ensure that operations complete successfully. This leads to a high degree of overhead in the form of chatter (extra packets), which significantly affects performance in environments with high latency.

For example, when clients use the Common Internet File System (CIFS) to access a network file share, the client sends a large number of synchronous requests that require the client to wait for a response before sending the next request. Opening a 2 MB Microsoft Word document generates 1000 CIFS requests. If all these requests are sent over a 40 ms round-trip WAN, the response time is at least 52 seconds before the document is usable.

Bandwidth

Network bandwidth is not a limiting factor in most LAN environments, but this is often not the case in the WAN circuits. Service providers charge for bandwidth, and companies purchase bandwidth based on average usage to reduce costs. The lack of available network bandwidth coupled with application-layer inefficiencies creates performance barriers for applications.

Application responsiveness or performance barriers arise with applications that communicate inefficiently when WAN bandwidth is constrained. For instance, assume that 10 remote users connect to the corporate network on a T1 line (1.544 Mbps). If a user sends an email message with a 1 MB attachment to nine users at the same site via the email server located in a data center, the email is transferred over the WAN once for the originating user sending the email, and then nine times for the recipients. In essence, one email crosses the WAN circuit 10 times in total.

Network bandwidth can create constraints related to application performance. Bandwidth found in the LAN has evolved over the years from Ethernet (10 Mbps), to Fast Ethernet (100 Mbps), to Gigabit Ethernet (1 Gbps), to 10 Gigabit Ethernet (10 Gbps); eventually 40 or 100 Gigabit Ethernet (100 Gbps) will be deployed. In most cases, the bandwidth capacity on the LAN is not a limitation from an application performance perspective.

WAN bandwidth is not increasing as rapidly as LAN bandwidth, and the price of bandwidth in the WAN is significantly higher than the price of bandwidth in the LAN. In essence, the circuit cost increases depending upon variables such as distance and bandwidth needs.

The most common WAN circuits today are much smaller in bandwidth capacity than what can be deployed in enterprise LAN environments. The most common WAN link in remote office and branch office environments is the T1 (1.544 Mbps), which is roughly 1/64 the capacity of a Fast Ethernet connection and roughly 1/664 the capacity of a Gigabit Ethernet connection. Technologies such as DSL, broadband cable modems, and Metro Ethernet provide connectivity to branch sites and are gaining popularity by offering more bandwidth than a traditional T1 and at a lower price point.

When examining application performance in WAN environments, it is important to note the bandwidth disparity that exists between LAN and WAN environments. When the amount of traffic on the LAN awaiting service over the WAN increases beyond the capacity of the WAN, the link is said to be oversubscribed with the increasing probability of packet loss.

When oversubscription is encountered, traffic that is competing for available WAN bandwidth must be queued. When queues become full, packets must be dropped, because there is no memory available in the device to temporarily store the data while it is waiting to be serviced. Loss of packets affects the application's ability to achieve higher levels of throughput and, in the case of a connection-oriented transport protocol, causes the communicating nodes to adjust their rates of transmission to a level that allows them to use only their fair share of the available bandwidth or to be within the capacity limits of the network.

Like bandwidth inefficiencies, throughput limitations can significantly hinder performance. A throughput limitation refers to the inability of an application to take advantage of the network that is available to it and is commonly a direct result of latency and bandwidth inefficiencies. As application latency for a *send-and-wait* application increases, the amount of time that is spent waiting for an acknowledgment or a response from the peer directly translates into time when the application is unable to do any further useful work.

Most applications allow certain operations to function in a parallel or asynchronous manner. Many operations that are critical to data integrity, security, and coherency must be handled in a serial manner. In such cases, these operations are not parallelized, and before subsequent messages can be handled, these critical operations must be completed in a satisfactory manner.

Bandwidth inefficiency can be directly correlated to throughput limitations associated with a given application. As the amount of data exchanged increases, the probability of encountering congestion also increases—not only in the network, but also in the presentation-, session-, and transport-layer buffers. Congestion brings about packet loss caused by buffer exhaustion due to lack of memory to store the data, and it results in

- Retransmission of data between nodes (if encountered in the network)
- Repeated delivery of application data to lower layers (if encountered at or above the transport layer)

Packet loss and congestion have a negative effect on application throughput. Packet loss can be caused by anything from signal degradation to faulty hardware and cannot generally be proactively reported to the packet's sender. For example, when a router drops a packet, it does not notify a transmitting node that a specific packet was dropped because of a congested queue. A transmitting node handles packet loss reactively based on the acknowledgments that are received from the recipient.

In TCP, the lack of an acknowledgment causes the transmitter to resend and adjust the rate at which it was sending data. The loss of a segment causes TCP to adjust its window capacity to a lower value to cover scenarios where too much data is being sent:

- Too much data for the network to deliver (because of oversubscription of the network)
- Too much data for the recipient to receive (because of congested receive buffers)

Upon encountering packet loss and having to retransmit data, the overall throughput of the TCP connection might be decreased to try to find a rate that does not transmit too much data. *Congestion Avoidance* is when TCP adjusts its rate in an attempt to match the available capacity in the network and the recipient and is accomplished through constant manipulation of the congestion window.

Latency

Latency is the amount of time a message takes to traverse a system. In a computer network, it is an expression of how much time it takes for a packet of data to get from one designated point to another. Latency is categorized as either application or network latency.

Application Latency

Latency is a culmination of the processing delays introduced by each of the four upper layers of the OSI model—the application, presentation, session, and transport layers—that manage the exchange of application data from node to node.

Application-layer (Layer 7) latency is defined as the processing delay of an application protocol that is generally exhibited when applications have a send-and-wait type of behavior, that is, a high degree of chatter, where messages must execute in sequence and are not parallelized.

Presentation-layer (Layer 6) latency is defined as the amount of latency incurred by ensuring that data conforms to the appropriate representation and managing data that is not correctly conformed or cannot be correctly conformed.

Session-layer (Layer 5) latency is defined as the delay caused by the exchange or management of state-related messages between communicating endpoints. For applications and protocols where a session-layer protocol is used, such messages may be required before any usable application data is transmitted.

Transport-layer (Layer 4) latency is defined as the delay in moving data from socket buffers (the memory allocated to a socket, for either data to transmit or received data) in one node to another. This can be caused by delays in receiving message acknowledgments, lost segments and the retransmissions that follow, and inadequately sized buffers that lead to the inability of a sender to send or a receiver to receive.

Performance limitations encountered at a lower layer affect the performance of the upper layers; for instance, a performance limitation that affects TCP directly affects the performance of any application operating at Layers 5 through 7 that uses TCP.

Note The chatter found in applications and protocols may demand that information be exchanged multiple times over the network. This means that the latency effect is multiplied and leads to a downward spiral in application performance and responsiveness.

Network Latency

Network latency is the amount of time it takes for data to traverse a network between two communicating devices. Latency occurs because it takes some amount of time for light or electrons to transmit from one point and arrive at another, commonly called a *propagation delay*.

Propagation delay can be measured by dividing the speed at which light or electrons are able to travel by the distance that they are traveling. Although this seems extremely fast on the surface, when stretched over a great distance, the latency can be quite noticeable. When you factor in delays associated with segmenting, packetization, serialization delay, and framing on the sender side, along with processing and response times on the recipient side, the amount of perceived latency can quickly increase.

The reason network latency affects application performance is twofold. First, network latency introduces delays that affect mechanisms that control rate of transmission. The second is related to application-specific messages that must be exchanged using these latency-sensitive transport protocols.

Because of the latency found in the WAN network, additional bandwidth may never be used, and capacity improvements may not be recognized. The network's latency can have a significant effect on the maximum amount of network capacity that can be consumed by two communicating nodes—even if there is a substantial amount of unused bandwidth available.

Note LAN latency is generally under 1 ms, meaning that the amount of time for data transmitted by a node to be received by the recipient is less than 1 ms. As utilization and oversubscription increase, the probability of packets being queued for an extended period of time increases, thereby likely causing an increase in latency.

WAN latency is generally measured in tens or hundreds of milliseconds and is much higher than what is found in a LAN.

Cisco Wide Area Application Services (WAAS)

Cisco *Wide Area Application Services (WAAS)* overcomes performance barriers presented by the WAN by employing

- **Application-agnostic optimization (WAN optimization):** WAN optimization refers to employing techniques at the transport protocol that apply across any application protocol using that network or transport protocol.
- **Application-specific optimization (application acceleration):** Application acceleration refers to employing optimizations directly against an application or an application protocol that it uses. WAN optimization has broad applicability, whereas application acceleration has focused applicability.

WAN optimization and application acceleration are complementary, in that the performance improvements provided by WAN optimization complement those provided by application acceleration.

When combined, WAN optimization and application acceleration improve application performance substantially more than if only one of the two is used. Cisco WAAS is a transparent solution in that it does not affect operational behavior or other components that exist end to end.

Cisco WAAS is transparent in three domains:

- **Clients:** No changes are needed on a client node to benefit from the optimization provided by Cisco WAAS.
- **Servers:** No changes are needed on a server node to benefit from Cisco WAAS.
- **Network:** Cisco WAAS provides the strongest levels of interoperability with technologies deployed in the network, including QoS, NetFlow, IP SLAs, ACLs, and firewall policies, which ensures seamless integration into the network.

The Cisco WAAS system consists of a set of devices called *Wide Area Application Engines (WAEs)* that work together to optimize TCP traffic over the network. The difference between a WAE and a *WAVE (Wide Area Application Virtualization Engine)* is that a WAVE provides a hypervisor that allows running other x86 virtual machines. Cisco WAAS uses Kernel-based Virtual Machine (KVM) technology from Red Hat to allow the WAVE appliance to host third-party operating systems and applications. Microsoft Windows Server, versions 2003 and 2008, is supported for installation on the WAAS virtual blade (VB) architecture. This configuration includes Microsoft Windows Server 2008 Core, Active Directory read-only domain controller, DNS server, DHCP server, and print server. The WAAS VB architecture helps enable customers to further consolidate infrastructure by minimizing the number of physical servers required in the branch office for those applications that are not good candidates for centralization into a data center location.

Note Cisco WAAS development has veered back toward the WAE architecture versus that of the WAVE. In fact, VBs have been deprecated in WAAS version 6.x and later.

When client and server applications attempt to communicate with each other, the network intercepts and redirects this traffic to the WAEs so that they can act on behalf of the client application and the destination server. The WAEs examine the traffic and use built-in optimization policies to determine whether to optimize the traffic or allow it to pass through the network optimized.

This unique combination of the three domains (client, server, and network) allows Cisco WAAS to be the least disruptive addition to an enterprise IT infrastructure. This level of transparency provides compatibility with existing network capabilities, which also allows customers to create the most compelling end-to-end application performance management solutions involving many technologies that all help achieve performance objectives.

Cisco WAAS Architecture

The foundational layer of the Cisco WAAS software is the underlying Cisco Linux platform. The Cisco Linux platform is hardened to ensure that rogue services are not installed, and secured so that third-party software cannot be installed or other changes made.

The Cisco Linux platform hosts a CLI shell similar to that of Cisco IOS software, which, along with the Central Manager (CM) and other interfaces, forms the primary means of configuring, managing, and troubleshooting a device or system. All relevant configuration, management, monitoring, and troubleshooting subsystems are made accessible directly through this CLI as opposed to exposing the Linux shell.

The Cisco Linux platform hosts a variety of services for WAAS run-time operation. These include disk encryption, *Central Management Subsystem (CMS)*, interface manager, reporting facilities, network interception and bypass, application traffic policy engine, and kernel-integrated virtualization services, as shown in Figure 11-3.

Bandwidth shortcomings are addressed by the following caching and compression techniques:

- Copies of previously transmitted files from HTTP, CIFS, and Server Message Block (SMB) protocols are maintained locally in the branch. Subsequent requests are served locally.
- Data Redundancy Elimination (DRE) is a Cisco proprietary compression and byte-level cache algorithm.
- *Lempel-Ziv (LZ)* compression is a well-known and often-used compression algorithm.
- TCP transport flow optimization (TFO) addresses latency.
- Application optimizers (AOs) mitigate and optimize application behavior (including "chattiness" within an application protocol).

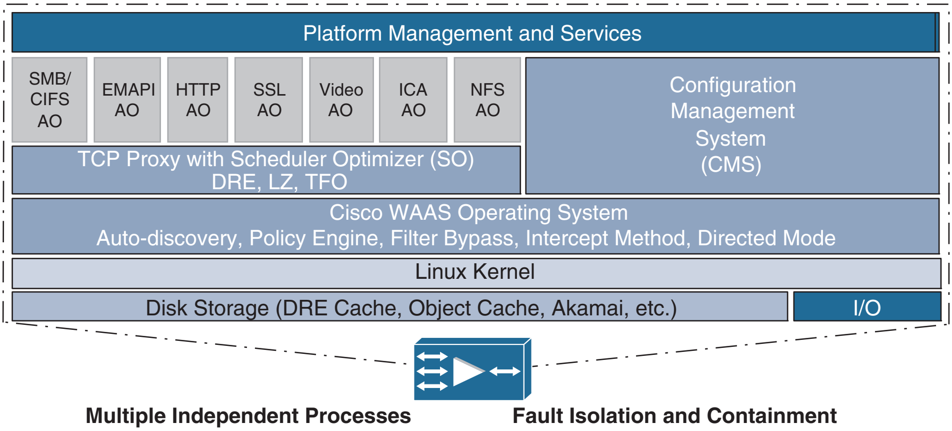

Platform Management and Services
SMB/ CIFS AO
EMAPI AO
HTTP AO
SSL AO
Video AO
ICA AO
NFS AO
Configuration Management System (CMS)
TCP Proxy with Scheduler Optimizer (SO) DRE, LZ, TFO
Cisco WAAS Operating System
Auto-discovery, Policy Engine, Filter Bypass, Intercept Method, Directed Mode
Linux Kernel
Disk Storage (DRE Cache, Object Cache, Akamai, etc.)
I/O
Multiple Independent Processes
Fault Isolation and Containment

Figure 11-3 *Cisco WAAS Architecture*

Application Optimizers

Application optimizers (AOs), also known as application accelerators, are the application-specific software that optimizes specific protocols at the seventh layer of the OSI model. An AO may be viewed as an "application" in the WAE system (in an OS analogy). The generic AO acts as a catchall for all traffic that has no protocol-specific AO and functions as a delegate if a protocol-specific AO decides not to apply optimization.

Configuration Management System

The *Configuration Management System (CMS)* consists of the WAAS Central Manager and its database for storing WAAS device configuration information. The CMS allows the configuration and management of WAE devices and device groups from a single Central Manager GUI interface.

Data Redundancy Elimination (DRE) with Scheduler

The *Data Redundancy Elimination (DRE)* with scheduler (SO-DRE) is the key module in the Layer 4 optimization space and is responsible for all data reduction techniques in the system, including DRE and persistent Lempel-Ziv (PLZ) compression. In addition to the system-wide algorithms for data reduction that are implemented here, this component includes a scheduling element that allows the system to better control the order and pace of using DRE for different AOs.

Storage

The storage system manages the system disks and the logical RAID volumes on systems that have multiple disks. Disk storage is used for system software, the DRE cache, the CIFS cache, and VB storage.

Network I/O

The network I/O component is responsible for all aspects that are related to handling data communication coming into or going out from a WAE, including WAE-to-WAE communication and WAE-to-client/server communication.

Interception and Flow Management

Interception and flow management consists of multiple submodules that, using policies configured by the user, intercept traffic, automatically discover peers, and start optimization on a TCP connection. Some of the key submodules are auto-discovery, policy engine, and filter bypass.

Auto-discovery

Auto-discovery allows peer devices to discover each other dynamically and does not require preconfiguration of WAE pairs. Auto-discovery is a multi-WAE end-to-end mechanism that defines a protocol between the WAEs that discovers a pair of peer WAEs for a given connection.

WAE devices automatically discover one another during the TCP three-way handshake that occurs when two nodes establish a TCP connection. This discovery is accomplished by adding a small amount of data to the TCP options field (0x21) in the SYN, SYN/ACK, and ACK messages. This TCP option allows WAE devices to understand which WAE is on the other end of the link and allows the two to describe which optimization policies they would like to employ for the flow. If intermediate WAEs exist in the network path, they simply pass through flows that are being optimized by other WAEs.

At the end of the auto-discovery process, the WAEs shift the sequence numbers in the TCP packets between the participating WAEs by increasing the sequence numbers to more than two billion, to mark the optimized segment of the connection.

Policy Engine

The policy engine module determines if traffic needs to be optimized, which AO to direct it to, and the level of data reduction (DRE), if any, that should be applied to it. The policy engine classifies traffic beyond connection establishment (for example, based on payload information) and changes the flow of a connection dynamically from unoptimized to optimized.

The elements of a policy include the following:

- **Application definition:** A logical grouping of traffic to help report statistics on the type of traffic
- **Traffic classifier:** An ACL that helps choose connections based on IP addresses, ports, and so on
- **Policy map (two types):** Binds the application and the classifier with an action, which specifies the type of optimization, if any, to be applied
  - **Static policy map:** Configured on the device through the CLI or GUI (or installed by default) and is persistent unless removed
  - **Dynamic policy map:** Automatically configured by the WAE and has a life span just long enough to accept a new connection

Filter Bypass

After interception, the filter bypass module acts as the mediator between the policy engine and auto-discovery. The filter bypass module tracks all optimized connections in a filtering table for the life of the connection. Traffic that should not be optimized can be bypassed.

TCP Optimization

Cisco WAAS uses a variety of TFO features to optimize TCP traffic intercepted by the WAAS devices. TFO protects communicating clients and servers from negative WAN conditions, such as bandwidth constraints, packet loss, congestion, and retransmission.

TFO includes the following optimization features:

- Windows scaling
- TCP initial window size maximization

- Increased buffering
- Selective acknowledgment (SACK)
- Binary increase congestion (BIC) TCP
- Compression

TCP Windows Scaling

The TCP receive window size determines the amount of space that the receiver has available for unacknowledged data. By default, TCP headers limit the receiver's window size to 64 KB, which can reduce the utilization of high-bandwidth, high-latency circuits to a fraction of the available bandwidth. On high-bandwidth, high-latency circuits it is common for the receiver's window to be transmitted in its entirety, and to require the sender to then wait for the receiver's TCP acknowledgment before sending the next window of packets.

TCP window scaling allows the TCP header to specify the size of the receive window (up to 1 GB). It allows TCP endpoints to take advantage of available bandwidth in the network and does not restrict network traffic to the default window size specified in the TCP header.

TCP Initial Window Size Maximization

WAAS increases the upper limit for TCP's initial window from one or two segments to two to four segments (approximately 4 KB). Increasing TCP's initial window size provides the following advantages:

- When the initial TCP window is only one segment, a receiver that uses delayed ACKs is forced to wait for a timeout before generating an ACK response. With an initial window of at least two segments, the receiver generates an ACK response after the second data segment arrives, eliminating the wait on the timeout.
- For connections that transmit only a small amount of data, a larger initial window reduces the transmission time. For many email (SMTP) and web page (HTTP) transfers that are less than 4 KB, the larger initial window reduces the data transfer time to a single round-trip time (RTT).
- For connections that use large congestion windows, the larger initial window eliminates up to three RTTs and a delayed ACK timeout during the initial slow-start phase.

Increased Buffering

Cisco WAAS enhances the buffering algorithm used by the TCP kernel so that WAEs can more aggressively pull data from branch office clients and remote servers. This increased buffering helps the two WAEs participating in the connection to keep the link between them full, increasing link utilization.

Selective Acknowledgment (SACK)

Selective acknowledgment (SACK) is an efficient packet loss recovery and retransmission feature that allows clients to recover from packet losses more quickly than the default recovery mechanism used by TCP.

By default, TCP uses a cumulative acknowledgment scheme that forces the sender to either wait for a round trip to learn if any packets were not received by the recipient or to unnecessarily retransmit segments that may have been correctly received.

SACK allows the receiver to inform the sender about all segments that have arrived successfully, so the sender needs to retransmit only the segments that have actually been lost.

Binary Increase Congestion (BIC) TCP

Binary Increase Congestion (BIC) TCP is a congestion management protocol that allows a network to recover more quickly from packet loss events.

When a network experiences a packet loss event, BIC TCP reduces the receiver's window size and sets that reduced size as the new value for the minimum window. BIC TCP then sets the maximum window size value to the size of the window just before the packet loss event occurred. Because packet loss occurred at the maximum window size, the network can transfer traffic without dropping packets whose size falls within the minimum and maximum window size values.

If BIC TCP does not register a packet loss event at the updated maximum window size, that window size becomes the new minimum. If a packet loss event does occur, that window size becomes the new maximum. This process continues until BIC TCP determines the new optimum minimum and maximum window size values.

Caching and Compression

Cisco WAAS offers DRE a context-aware byte-level cache that includes application intelligence which inspects TCP traffic to identify redundant data patterns at the *byte level* and then quickly replaces them with signatures if they have been previously seen so that the peer Cisco WAAS device can use them to reproduce the original data.

Cisco WAAS also offers LZ compression, which is a lossless compression algorithm that uses an extended compression history for each TCP connection to achieve higher levels of compression than standard LZ variants can achieve. LZ is helpful for data that DRE has not identified as redundant and can even provide additional compression for DRE encoded messages, because the DRE signatures are compressible. These techniques are discussed in detail in the following sections.

Compression

Cisco WAAS uses the following compression technologies to help reduce the size of data transmitted over a WAN:

- Data Redundancy Elimination (DRE)
- LZ compression byte caching

Data Redundancy Elimination (DRE)

Data Redundancy Elimination (DRE) is a bidirectional, lossless data deduplication algorithm that leverages both memory (high throughput and high I/O rates) and disk (persistent and large compression history). DRE examines data in flight for redundant patterns (patterns that have been previously identified) and works in an application-agnostic manner, meaning that redundant patterns found in traffic for one application can be leveraged for another application.

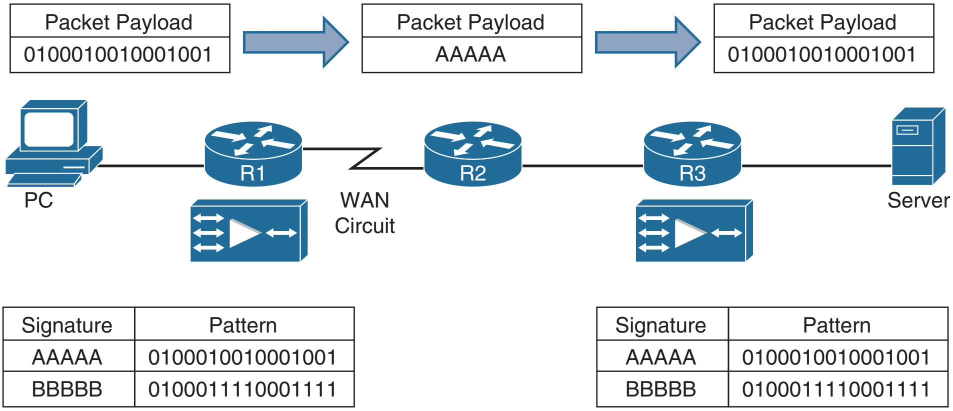
WAAS examines packets looking for patterns in 256-byte, 1 KB, 4 KB, and 16 KB increments and creates a signature for each of those patterns. WAAS maintains a local *data store* that contains a central repository for signatures with the actual data at a byte level.

If the pattern is sent a second time, the data is replaced with a signature by the first WAAS device. The signature is sent across the WAN link and replaced with the data by the second WAAS device. This drastically reduces the size of the packet as it crosses the WAN but still keeps the original payload between the devices communicating. DRE can provide significant levels of compression for flows containing data that has been previously identified, which helps minimize bandwidth consumption on the WAN.

DRE is bidirectional, meaning patterns identified during one direction of traffic flow can be leveraged for traffic flowing in the opposite direction. DRE is also application agnostic in that patterns identified in a flow for one application can be leveraged to optimize flows for a different application.

When a WAE uses compression to optimize TCP traffic, it replaces repeated data in the stream with a much shorter reference, then sends the shortened data stream out across the WAN. DRE is an advanced form of network compression that allows Cisco WAAS to maintain a database of byte sequences previously seen traversing the network. This information is used to prevent redundant transmission patterns from traversing the network. For repeated patterns, only the pattern identifiers are sent, and the original message is then rebuilt in its entirety by the distant appliance.

Figure 11-4 demonstrates this concept where the WAE device attached to R1 replaces the data pattern *0100010010001001* with the data signature *AAAAA*. R3 then replaces the data signature *AAAAA* with the data pattern *0100010010001001*. The data signature consumes less bandwidth than the actual data pattern.



Signature	Pattern
AAAAA	0100010010001001
BBBBB	0100011110001111

Signature	Pattern
AAAAA	0100010010001001
BBBBB	0100011110001111

Figure 11-4 *Data Redundancy Elimination Logic*

The DRE feature enables significant levels of compression and helps ensure message and application coherency because the remote Cisco WAAS device always rebuilds and verifies the original message. The receiving WAE uses its local redundancy library to reconstruct the data stream before passing it along to the destination client or server.

Because DRE is application agnostic and bidirectional, it is effective regardless of the direction of traffic flow. Data patterns identified for one application protocol can be reused by other applications, and patterns that have been identified for one direction of traffic flow can be reused to remove redundancy in traffic flowing in the other direction.

The WAAS compression scheme is based on a shared cache architecture where each WAE involved in compression and decompression shares the same redundancy library. When the cache that stores the redundancy library on a WAE becomes full, WAAS uses a FIFO algorithm (first in, first out) to discard old data and make room for new.

DRE is context aware, and adaptive DRE has two modes:

- **Unidirectional:** The data signature replaces the data pattern in only one direction.
- **Bidirectional:** The data signature can replace the data pattern in either direction.

Figure 11-5 demonstrates both modes. Steps 1–4 demonstrate the process of the WAE devices learning the data signature with the data patterns. However, Steps 5–6 differentiate the ways that the data signature can be used depending upon the DRE mode selected.

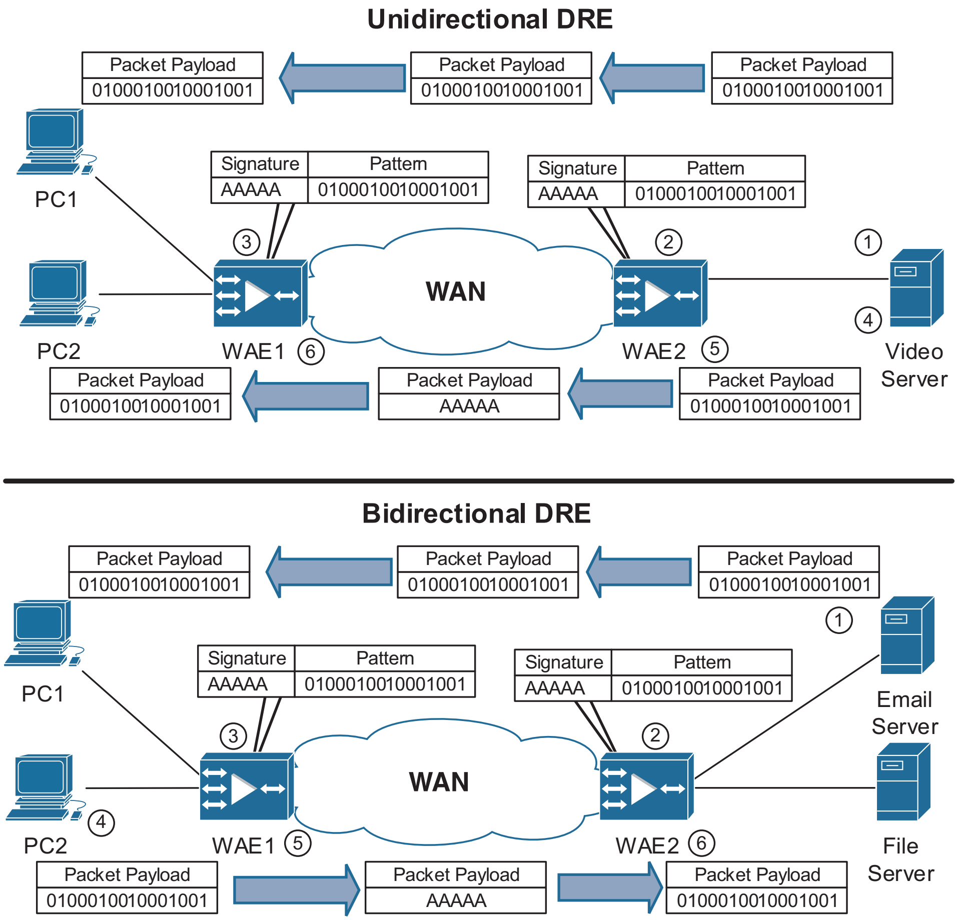


Figure 11-5 *Unidirectional and Bidirectional DRE Modes*

Unidirectional Mode

In unidirectional mode, traffic is traditionally deployed at the remote site because of the nature of the network traffic. There is no need to cache the data at the data center or head end. Only the signatures are needed at the head end in order to serve the second and subsequent requests for the data. Data is written only to the destination cache.

Unidirectional mode is typically deployed for video streams, video on demand, or cloud-based applications such as backup. In Figure 11-5, PC1 subscribes to a video stream. WAE2 does not have a local signature for the first video stream but creates a compression history that is updated with the data patterns from the flow. WAE2 then sends the complete packet across the WAN. WAE1 stores the signature and byte-level correspondence in its local store. When PC2 subscribes to the same video stream, WAE2 has a local signature and uses that instead of the original data to reduce the data transmitted over the WAN.

Bidirectional Mode

In bidirectional mode, patterns identified during one direction of traffic flow can be leveraged for traffic flowing in the opposite direction. DRE is application agnostic in that patterns identified in a flow for one application can be leveraged to optimize flows for a different application.

Bidirectional DRE is illustrated at the bottom of Figure 11-5. When PC1 downloads an email containing an attachment, the compression history on each of the WAAS devices in the connection path is updated with the data patterns contained in the flow. PC2 has a copy of that file and uploads the file to a different server with another application (CIFS file share). The compression history that was previously built from the email transfer can be leveraged to provide compression for the CIFS upload.

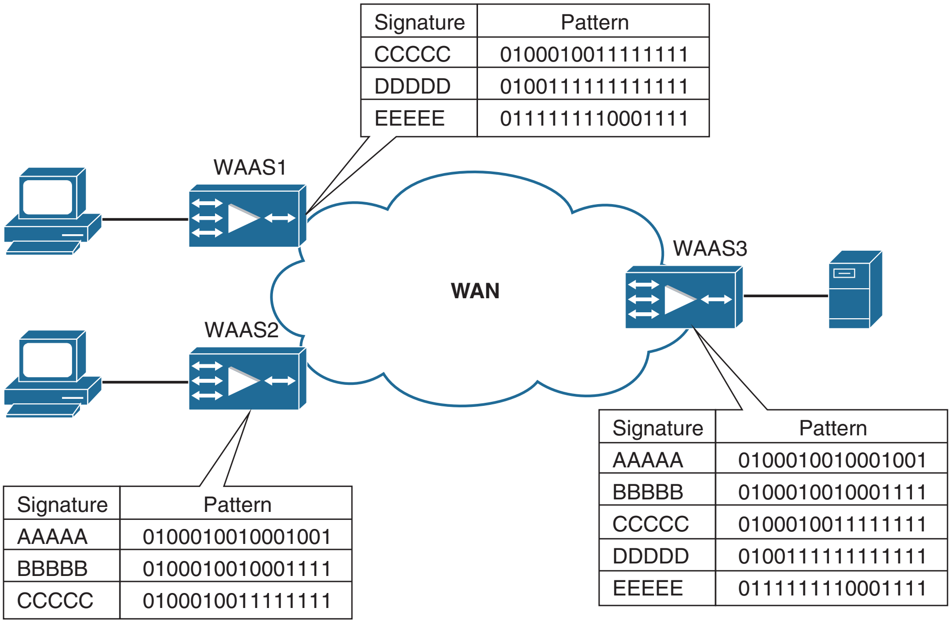
Unified Data Store

Prior to Cisco WAAS Release 4.4, a Cisco WAE's cache (at the data center) was segmented. Each branch was assigned a specific portion of the data center appliance's cache, and it could not be shared with other WAAS nodes. This meant that even though a particular block of data had already gone through the data center to one location, other nodes could not take advantage of this information. With the release of Cisco WAAS 4.4 software, Cisco WAAS's context-aware DRE does away with the segmented cache, opting for a single large unified cache in which all appliances can participate.

Context-aware DRE places signatures in the data center WAE on a per-peer basis, and the actual data chunks are shared (replicated) across peers. Synchronization of the peer signatures combined with shared chunks of data across peers helps provide consistent, reliable, and fair DRE performance for all peers. As you can see, the cache architecture is a hybrid (both per-peer cache and global cache). Per-peer caching is used for signatures. This maximizes lookup efficiency and helps prevent one branch from starving another of cache.

The central site data cache is global; there is only one copy of a given chunk of data. This maximizes storage efficiency and increases performance (data is not repeatedly written/read from multiple locations on disk). The central site's WAE cache is consolidated into a single data store for all peers.

Figure 11-6 illustrates the concept where all three WAAS appliances have a CCCCC signature. Only WAAS1 and WAAS3 have the DDDDD and EEEEE signatures, whereas WAAS2 and WAAS3 have the AAAAA and BBBBB signatures.



Signature	Pattern
CCCCC	0100010011111111
DDDDD	0100111111111111
EEEEE	0111111110001111

Signature	Pattern
AAAAA	0100010010001001
BBBBB	0100010010001111
CCCCC	0100010011111111

Signature	Pattern
AAAAA	0100010010001001
BBBBB	0100010010001111
CCCCC	0100010011111111
DDDDD	0100111111111111
EEEEE	0111111110001111

Figure 11-6 *Unified DRE Data Store*

Lempel-Ziv (LZ) Compression

LZ compression is a standards-based compression that can be applied to further reduce the amount of bandwidth consumed by a TCP flow. LZ compression can be used in conjunction with DRE or independently. LZ compression can provide from 2:1 to 4:1 compression, depending on the application being used and the data being transmitted.

LZ compression operates on smaller data streams and keeps limited compression history, whereas DRE operates on significantly larger streams (typically tens to hundreds of bytes or more) and maintains a much larger compression history. LZ compression is especially helpful for data that has not been previously seen and suppressed by DRE because the pattern identifiers are highly compressible.

These compression technologies reduce the size of transmitted data by removing redundant information before sending the shortened data stream over the WAN. By reducing the amount of transferred data, WAAS compression can reduce network utilization and application response times.

Object Caching

The CIFS object cache is a client-side function, so it caches/serves content only for a local client. Object and metadata caching are techniques employed by Cisco WAAS to allow a device to retain a history of previously accessed objects and their metadata. These techniques are leveraged in CIFS acceleration and metadata.

Unlike DRE, which maintains a history of previously seen data on the network (with no correlation to the upper-layer application), object caching and metadata caching are specific to the application being used, and the cache is built with pieces of an object or the entire object, along with its associated metadata. With caching, if a user attempts to access an object, a directory listing, or file attributes that are stored in the cache, such as a file previously accessed from a particular file server, the file can be safely served from the edge device.

Note With DRE there is always some type of traffic traversing the WAN. In contrast, object caching stores a copy of previously accessed objects to be reused by subsequent users. Object caching mitigates latency by serving objects locally. This saves bandwidth and improves application performance because no data is sent over the WAN.

In Figure 11-7, the latency between the PC and the server is 100 ms. The latency between the branch PC and the local object cache is 5 ms, which is a shorter delay than waiting for the file to be retrieved from the server, which is 100 ms. Only the initial file transfer takes 100 ms; subsequent requests for the same file are provided locally from the cache.

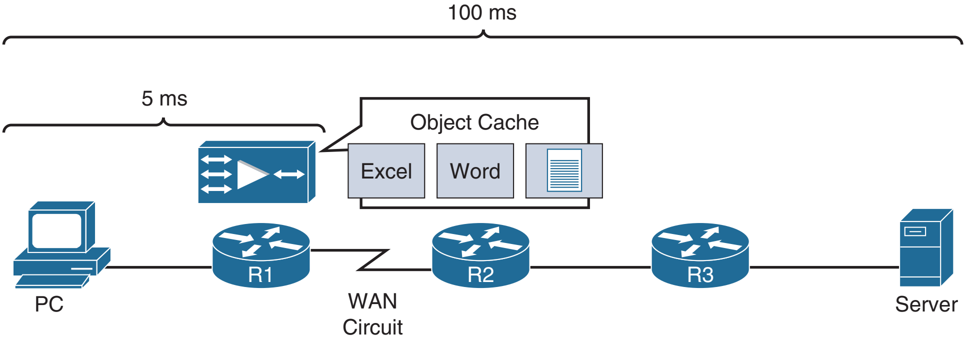


Figure 11-7 *Object Caching and Effect on Network Latency*

Application-Specific Acceleration

Application acceleration capabilities provided in Cisco WAAS work in conjunction with WAN optimization features and help mitigate the negative effects of the WAN by providing safe caching, protocol acceleration, message batching, read-ahead optimization, and more.

Cisco WAAS supports a broad range of applications accelerated through application-specific support, including CIFS, SMB, network file server (NFS), Messaging Application Programming Interface (MAPI), Encrypted MAPI (EMAPI), HTTP, secure HTTP (HTTPS), and Akamai Connect. Each of these technologies is explained further in the following sections.

Microsoft Exchange Application Optimization

Microsoft Exchange email relies on the MAPI messaging interface, used over remote-procedure calls (RPCs) to deliver email, calendaring, contacts, and more to Microsoft Outlook users for collaboration and productivity. As with many applications operating over a WAN, Microsoft Exchange performance is constrained by bandwidth limitations and latency found in the WAN.

Cisco WAAS provides a number of acceleration services for Microsoft Exchange to improve performance. Cisco WAAS acceleration for Microsoft Exchange was developed in conjunction with Microsoft to help ensure protocol correctness and compatibility with all major versions of Microsoft Exchange (including Microsoft Exchange 2000, 2003, 2007, 2010, and 2012), with native support for both encrypted and unencrypted traffic, without relying on reverse engineering of protocols.

Cisco WAAS provides the following acceleration capabilities for Microsoft Exchange:

- **Asynchronous write operations:** Write operations for sending email and attachments are acknowledged locally. Local generation of responses allows clients to fully utilize WAN bandwidth.
- **Object read-ahead:** Objects being fetched from the server, such as email, calendar items, and address books, are fetched at an accelerated rate; Cisco WAAS prefetches these objects on behalf of the user. This feature helps mitigate the send-and-wait behavior of Microsoft Exchange and Outlook.
- **Message decompression:** Cisco WAAS can automatically defer native compression provided by Microsoft Exchange Server and Outlook in favor of Cisco WAAS DRE and PLZ compression. Cisco WAAS can also natively decode messages encoded by Microsoft Exchange or Outlook to provide additional levels of compression. Full data coherency is preserved end to end.
- **DRE hints:** Cisco WAAS provides hints to the DRE compression process based on the message payload, resulting in better compression and overall improvement in DRE efficiency.
- **Payload aggregation:** Cisco WAAS recognizes many Microsoft Exchange messages that are small and can either batch these messages together for optimized delivery or dynamically adjust DRE and LZ compression to improve compression ratios for them.

EMAPI is supported with network, security, and application transparency and complies with Microsoft's Kerberos security negotiation. No configuration changes are required on the client or server to support Cisco WAAS acceleration.

HTTP Application Optimization

Cisco WAAS provides the following HTTP acceleration capabilities for enterprise applications:

- **Fast connection reuse:** Connection reuse decreases the load time for complex pages or pages with numerous embedded objects when the client or server cannot use persistent connections. Optimized connections on the WAN remain active for a short time period so that they can be reused if additional data between the client/server pair needs to be exchanged.
- **Connection multiplexing:** Rather than requiring that multiple connections be established between client/server pairs, connections established between clients and servers are reused. This feature eliminates the latency caused by the process of establishing multiple connections between clients and servers.
- **Local response:** The use of cached metadata allows Cisco WAAS branch office devices to respond locally to certain HTTP requests. These local responses are based on cached metadata from previously seen server responses and are continuously updated. This powerful HTTP optimization feature greatly reduces protocol chattiness and helps improve application response times through faster page downloads.

SharePoint Application Optimization

Cisco WAAS provides the following acceleration capabilities for Microsoft SharePoint:

- The Microsoft SharePoint optimization feature provides optimization by prefetching objects for Microsoft Word and Excel and storing them in the Cisco WAAS metadata cache. This optimization saves RTT for each successful fetch to reduce latency from the client's point of view and improve the overall user experience.
- The Cisco WAAS advanced HTTP parser provides intelligent recommendations that make DRE more effective and enable offloading of compression from the server resources.

SSL Application Optimization

Cisco WAAS provides SSL optimization capabilities that integrate transparently with existing data center key management and trust models that both WAN optimization and application acceleration components can use. WAAS can also optimize SSL-encrypted applications through SSL optimization. SSL optimization enables WAAS devices to become trusted intermediary devices to decrypt incoming encrypted data, apply the appropriate set of optimizations, and reencrypt data for further delivery over the WAN.

The certificate private keys are installed into the WAAS Central Manager secure store that is a passphrase-protected and -encrypted vault. The WAAS Central Manager deploys the certificates that will be used for interception to all other WAAS devices in the network. Disk encryption can be applied to all WAAS devices to ensure protection of stored data found in optimized TCP connections should a device or its disks be physically compromised.

SSL optimization allows the data center WAAS device to act as an SSL proxy for the origin server, enabling the WAAS device to control the SSL session. Session keys are securely distributed to the peer WAAS devices participating in optimizing the SSL session, which gives both devices the capability to decrypt, optimize, and reencrypt traffic.

SSL optimization provides full support for a number of critical security-related features, including *Online Certificate Status Protocol (OCSP)* and certificate revocation checks/validation, and it interoperates with web proxies. Figure 11-8 shows how WAAS becomes a trusted proxy to intercept and optimize SSL-encrypted sessions while preserving the security model in place and the trust boundary for private key data.

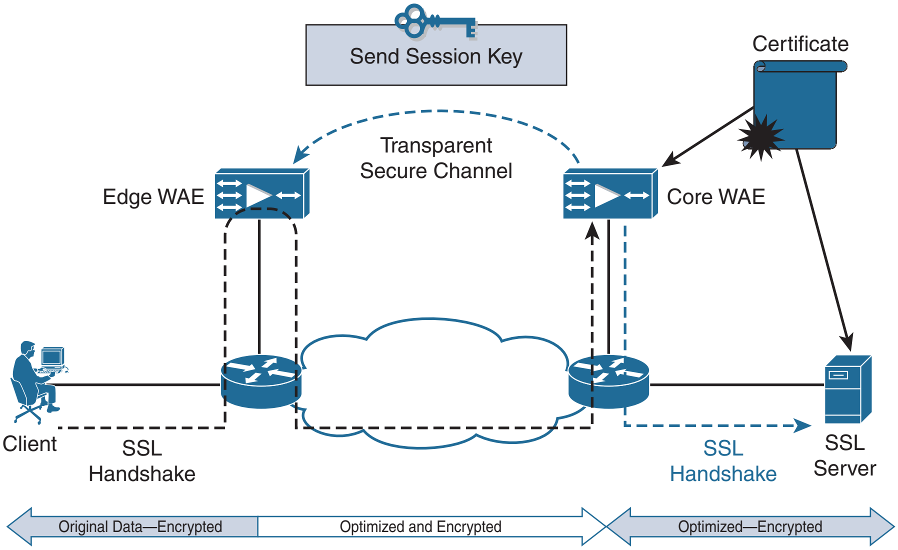


Figure 11-8 *SSL-Encrypted Data Flow Key Exchange and Optimization*

Citrix Application Optimization

Cisco WAAS is a Citrix-ready certified solution that helps ensure a high-quality user experience for Citrix XenDesktop and XenApp. The solution is jointly tested, validated, and supported by Cisco and Citrix, providing optimization transparently. No changes to existing Citrix infrastructure are required.

The Cisco WAAS with Citrix XenDesktop and XenApp solution offers the following benefits:

- **Full support for Citrix XenDesktop and XenApp:** Cisco WAAS optimization techniques are applicable to Citrix XenDesktop and XenApp deployments.
- **Transparency to Citrix encryption:** Cisco WAAS can be transparently inserted into encrypted communications (Basic and RC5) without requiring changes to the Citrix configuration or infrastructure.
- **Full compatibility with Citrix Multi-Stream Independent Computing Architecture (ICA; MSI):** Cisco WAAS also includes the capability to apply DSCP tagging of both MSI and non-MSI ICA and Citrix Common Gateway Protocol (CGP) traffic.
- **Full compatibility with Citrix Session Reliability (CGP):** Cisco WAAS parses the ICA and Citrix CGP traffic and learns the arbitrary TCP ports through the MSI negotiation. The Cisco WAAS ICA process then establishes dynamic Cisco WAAS policy engine rules to allow the additional ICA/CGP connections to be intercepted and passed to the Cisco WAAS ICA process for optimization.
- **Full compatibility with Citrix HDX MediaStream:** Citrix HDX MediaStream exposes the media stream, which enables Cisco WAAS optimization of the streaming media protocol.
- **Full compatibility with ICA over SSL:** Cisco WAAS identifies ICA inside SSL and provides the optimization while maintaining the SSL connection between the client and Citrix Access Gateway without the need for any configuration changes to the client or server.
- **Context-aware DRE:** Directional understanding of data enables best performance as well as increased bandwidth savings as a result of improved compression.

CIFS Application Optimization

Cisco WAAS can provide the following acceleration capabilities for CIFS:

- **Safe data and metadata caching:** By caching copies of objects and metadata, Cisco WAAS can reduce the transmission of CIFS data over the WAN, thereby providing tremendous levels of optimization for branch office users accessing file servers in the data center. All data is validated against the server for coherency to help ensure that a user never receives out-of-date (stale) data.
- **Read-ahead processing:** For situations in which objects are not cached or cannot be cached, Cisco WAAS employs read-ahead optimization to bring the data to the user more quickly. Read-ahead processing reduces the negative effects of latency on CIFS by requesting the data on behalf of the user. This data can then be used, when safe, to respond to the user on behalf of the server.

- **Message pipelining:** CIFS messages can be pipelined over the WAN to mitigate the effects of the send-and-wait behavior of CIFS. This feature allows operations to occur in parallel rather than serially, thus improving performance for the user.
- **Prepositioning:** File server data and metadata can be copied in a scheduled manner to improve performance for first-user access. This feature is helpful in environments in which large objects must traverse the WAN, including software distribution and desktop management applications.
- **Microsoft Windows printing acceleration:** Cisco WAAS can intelligently accelerate CIFS printing traffic over the WAN to allow centralization of print services in the data center. This feature helps reduce the branch office infrastructure without compromising printing performance and is transparent to the existing printer and queue management architectures.
- **Intelligent file server offloading:** Cisco WAAS CIFS acceleration reduces the burden placed on the origin file server through advanced caching techniques that can serve data locally to the requesting user when the user is authenticated and authorized and the cached contents are validated as coherent with the origin file server. Thus, file servers see fewer requests and are required to transfer less data, thereby enabling file server scalability and better economics.
- **Invalid file ID processing:** Cisco WAAS can deny requests to access files with invalid file handle values locally instead of having to send the requests to the server, thus improving performance for the user.
- **Batch close optimization:** Cisco WAAS can perform asynchronous file close optimization for file-sharing traffic.

SMB Application Optimization

The file services feature in Cisco WAAS maintains files locally, close to the clients. Changes made to files are immediately stored in the local branch WAE and then streamed to the central file server. Files stored centrally appear as local files to branch users, which improves access performance.

SMB caching includes the following features:

- **Local metadata handling and caching** allows metadata such as file attributes and directory information to be cached and served locally, optimizing user access.
- **Partial file caching** propagates only the segments of the file that have been updated on write requests rather than the entire file.
- **Write-back caching** facilitates efficient write operations by allowing the data center WAE to buffer writes from the branch WAE and to stream updates asynchronously to the file server without risking data integrity.
- **Advance file read** increases performance by allowing a WAE to read the file in advance of user requests when an application is conducting a sequential file read.

- **Negative caching** allows a WAE to store information about missing files to reduce round trips across the WAN.
- **Microsoft Remote Procedure Call (MSRPC) optimization** uses local request and response caching to reduce the round trips across the WAN.
- **Signaling message prediction and reduction** uses algorithms that reduce round trips over the WAN without loss of semantics.

NFS Acceleration

Cisco WAAS provides protocol acceleration for UNIX environments in which the NFS protocol is used for file exchange. In conjunction with the WAN optimization capabilities provided by Cisco WAAS, NFS acceleration helps improve file access performance—both interactive access and access during file transfer—by mitigating the negative effects of latency and bandwidth constraints.

Cisco WAAS provides the following NFS acceleration capabilities:

- **Metadata optimization:** Cisco WAAS pipelines interactive operations such as directory traversal to reduce the amount of time required to traverse directories and view file and directory metadata. Additionally, Cisco WAAS caches metadata when it is safe to do so, to reduce the number of performance-limiting operations that traverse the WAN.
- **Read-ahead optimization:** Cisco WAAS performs read-ahead optimization on behalf of the requesting node to prefetch data from the file being accessed. This feature makes the data readily available at the edge device for faster read throughput.
- **File write optimization:** Asynchronous write operations are used to batch write messages and eliminate the send-and-wait behavior of NFS file write operations while working in conjunction with existing NFS protocol semantics to help ensure file data integrity.

Akamai Connect

Cisco WAAS with Akamai Connect is a fully integrated solution from Cisco and Akamai that provides next-generation application and network optimization. The solution integrates both Cisco WAN optimization and Akamai intelligent caching techniques into one solution.

Akamai Connect provides faster content delivery regardless of device, connectivity, or cloud (including the corporate private cloud and the public Internet cloud). Akamai Connect integrates Akamai's HTTP caching technology into Cisco WAAS to further enhance users' application experiences.

Cisco WAAS can be deployed on both ends of the WAN or in a single-sided direct-to-Internet scenario to provide performance improvement regardless of where the HTTP traffic egresses.

The following sections explain Akamai Connect's HTTP caching techniques.

Transparent Cache

Akamai's HTTP object cache provides the capability to locally cache HTTP-based content, regardless of whether the web application was served from the private corporate cloud or the public Internet. Transparent caching has three modes, which can be configured at the global, domain, or host level:

- **Basic:** This mode caches only objects with explicit caching directives and obeys any client directives.
- **Standard:** This mode is the default mode. It obeys any server caching directives but also makes intelligent caching decisions for objects that do not have caching directives.
- **Advanced:** This mode also obeys any caching directives and makes intelligent caching decisions about the objects to be cached, but it provides a more aggressive algorithm than Standard mode and is typically well suited for media-intensive objects.

Akamai Connected Cache

Akamai's proprietary caching rules and connection with the edge servers of the Akamai Intelligent Platform provide the capability to cache and deliver content within a branch office for traffic that may otherwise be deemed uncacheable. This content may be an enterprise's own web content or any content that is delivered by the Akamai Intelligent Platform, which is up to 30 percent of all web traffic.

Dynamic URL HTTP Cache (Over-the-Top Cache)

The high-performance object-level cache from Akamai provides the capability to cache HTTP content served from dynamically generated URLs and content normally not cacheable, such as YouTube videos, often used today for product demonstrations and advertisements displayed in stores on digital signage, employee training, and other contexts.

Content Prepositioning for Enhanced End-User Experience

Content prepositioning allows organizations to define policies to proactively fetch content on a specific schedule. By warming the HTTP web cache during nonpeak times, organizations can improve application performance and increase network offload when the network is busiest.

Summary

IT organizations are challenged with the need to provide high levels of application performance for an increasingly distributed workforce. This chapter examined the most common causes of application performance challenges found in WAN environments, latency and bandwidth inefficiencies. With the understanding of the factors that affect application performance across a WAN, network engineers can now formulate an appropriate solution.

Cisco WAAS provides solutions to the performance barriers presented by the WAN by employing a series of application-agnostic optimizations in conjunction with a series of application-specific optimizations, so users can enjoy near-LAN performance when working with geographically dispersed applications and content.

Further Reading

Seils, Zach, Joel Christner, and Nancy Jin. *Deploying Cisco Wide Area Application Services, Second Edition*. Indianapolis: Cisco Press, 2010.

Chapter 12

Cisco Wide Area Application Services (WAAS)

This chapter covers the following topics:

- Cisco WAAS architecture
- Interception techniques and protocols
- WAAS platforms
- WAAS design and performance metrics
- Integration best practices

Cisco WAAS is a software component that resides on hardware devices deployed at each network site. WAAS can run on a hypervisor, a router-integrated network module, or a dedicated appliance.

This chapter covers the WAAS architecture and interception techniques and looks at the positioning for each of the hardware platforms, as well as the performance and scalability metrics for each platform.

Note “Application optimizers” and “application accelerators” are terms that are used interchangeably throughout this chapter.

Cisco WAAS Architecture

The Cisco WAAS architecture is based on the Cisco Linux OS. The Cisco Linux OS is hardened to ensure that rogue services are not installed, and secured to prevent third-party software installation and other changes. The Cisco Linux OS provides a CLI shell similar to Cisco IOS devices. This special CLI shell, along with the WAAS Central Manager, makes up the primary means of configuring, managing, and troubleshooting a WAAS device or system.

Note All relevant configuration, management, monitoring, and troubleshooting subsystems are made accessible directly through this CLI as opposed to exposing the actual Linux operating system.

The Cisco Linux platform hosts a variety of services for WAAS run-time operation such as disk encryption, Central Management Subsystem (CMS), interface manager, reporting facilities, network interception and bypass, and Application Traffic Policy (ATP) engine, as shown in Figure 12-1.

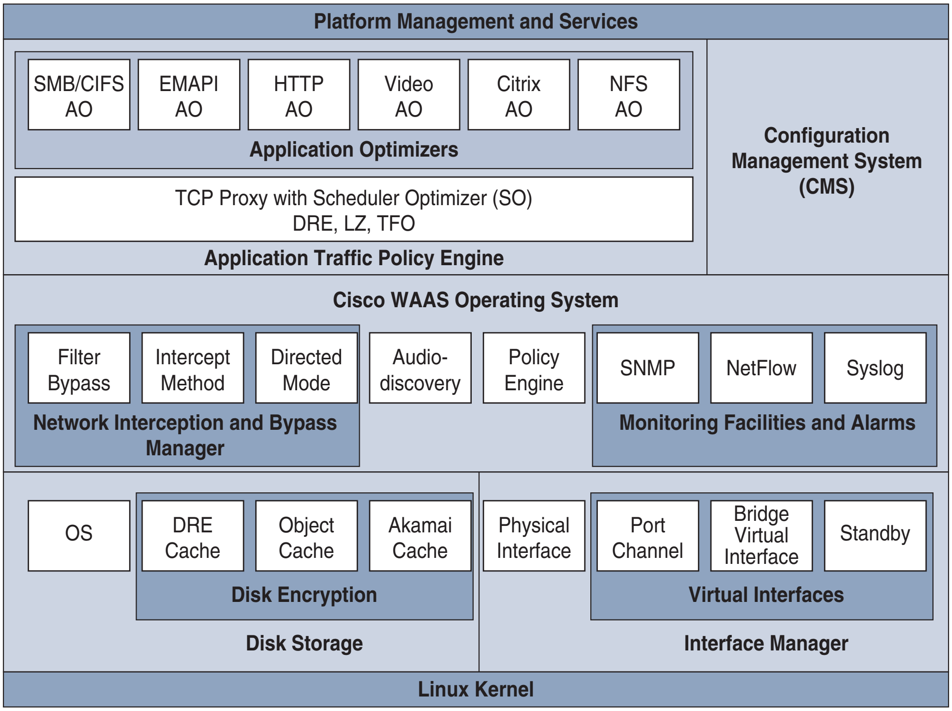


Figure 12-1 *Cisco WAAS Hardware and Software Architecture*

The following sections examine each of the Cisco WAAS architecture items. Cisco WAAS optimization components, including Data Redundancy Elimination (DRE), persistent LZ (PLZ) compression, transport flow optimization (TFO), and application accelerators, were discussed in Chapter 11, “Introduction to Application Optimization,” and are not discussed in this chapter.

Central Management Subsystem

CMS is a process that runs on each WAAS device, including accelerators and Central Managers. This process manages the configuration and monitoring components of a WAAS device and ensures that each WAAS device is synchronized with the Central Manager based on a scheduler known as the *Local Central Manager (LCM)* cycle. The LCM cycle is responsible for synchronizing the Central Manager CMS process with the remote WAAS devices. The CMS process exchanges configuration data, fetches health and status information, and gathers monitoring and reporting data. The CMS process is tied to a management interface known as the *primary interface*. The primary interface is configured on the WAAS device CLI prior to registration to the Central Manager. Any communication that occurs between WAAS devices for CMS purposes uses SSL for security.

Interface Manager

The Cisco WAAS device interface manager manages the physical and logical interfaces that are available on the WAAS device. Each WAAS device includes two integrated Gigabit Ethernet interfaces. Each WAAS appliance has expansion slots to support one or more additional feature cards, such as the inline module.

The interface manager provides management of logical interfaces. The first logical interface is the port channel interface, which can be used to aggregate (802.3ad) WAAS device interfaces together for the purposes of high availability and load balancing. The second logical interface is the standby interface. A standby interface has multiple physical interfaces associated to it, where one physical interface is active and a second interface is used as a backup in the event the active interface fails.

Note Standby interfaces are used when WAAS device interfaces connect to separate switches, whereas port channel interfaces are used when the WAAS device interfaces connect to the same switch. Port channels can also be used between multiple switches that support virtual port channel (vPC) or virtual switching system (VSS).

Monitoring Facilities and Alarms

Cisco WAAS supports SNMP versions 1, 2c, and 3 and a variety of MIBs that provide complete health reporting of each individual WAAS device. Cisco WAAS provides support for NetFlow and the definition of up to four syslog servers, which can be used as alarm recipients when syslog messages are generated.

The WAAS Central Manager has an alarm dashboard. The Central Manager offers an API that is available for third-party visibility and monitoring systems. Transaction logs can be configured to be stored on each of the accelerator devices in the network for persistent retention of connection statistics, which may be useful for troubleshooting, debugging, or analytics purposes.

Network Interception and Bypass Manager

The Cisco WAAS device uses the network interception and bypass manager to establish relationships with intercepting devices where necessary and ensure low-latency bypass of traffic that the WAAS device is not intended to handle.

The *Web Cache Communication Protocol (WCCP)* version 2 (WCCPv2) is a protocol managed by the network interception and bypass manager to allow the WAAS device to successfully join a WCCPv2 service group with one or more adjacent routers, switches, or other WCCPv2-capable server devices. Other network interception options include policy-based routing (PBR) and physical inline interception. As flows are intercepted by the WAAS device and determined to be candidates for optimization, they are handed to the ATP engine to identify what level of optimization and acceleration should be applied based on the configured policies and classifier matches.

Application Traffic Policy Engine

The foundational optimization layer of the Cisco WAAS software is the ATP engine. The ATP engine is responsible for examining details of each incoming flow (after being handled by the interception and bypass mechanisms) in an attempt to identify the application or protocol associated with the flow. This association is done by comparing the packet headers from each flow against a set of classifiers that identify network traffic based upon one or more match conditions in the protocol fields. A classifier can be predefined, administratively configured, or dynamic.

WAAS policies are evaluated in priority order, and the first classifier and policy match determine the action taken against the flow and where the statistics for that flow are aggregated. Flows that do not have a match with an existing classifier are considered "other" traffic and are handled according to the policy defined for other traffic, which indicates that there are no classifier matches and that the default policy should be used.

Optimization class maps associate an application classification with a policy map that provides the action on a particular flow.

When a classifier match is found, the ATP engine examines the policy configuration for that classifier to determine how to optimize the flow. The ATP engine also notes the application group to which the classifier belongs to route statistics gathered to the appropriate application group for proper charting (visualization) and reporting. The configured policy dictates which optimization and acceleration components are enacted upon the flow and how the packets within the flow are handled. The list of configurable elements within a policy include the following:

- **Type of policy:** Defines whether the policy is a basic policy (optimize, accelerate, and apply a marking), *Wide Area File Services (WAFS) Software* transport (used for legacy mode compatibility with WAAS version 4.0 devices), or *endpoint mapper (EPM)*, which is used to identify universally unique identifiers for classification and policy.

- **Application:** Defines which application group the statistics should be collected into, including byte counts, compression ratios, and other data, which are then accessible via the WAAS device CLI or Central Manager.
- **Action:** Defines the WAN optimization policy that should be applied to flows that match the classifier match conditions. These include
  - **Pass-through:** Take no optimization action on this flow.
  - **TFO only:** Apply only TCP optimization to this flow, but no compression or data deduplication.
  - **TFO with LZ compression:** Apply TCP optimization to this flow, in conjunction with PLZ compression.
  - **TFO with DRE:** Apply TCP optimization to this flow, in conjunction with data deduplication.
  - **Full optimization:** Apply TCP optimization, PLZ compression, and data duplication to this flow.
- **Position:** Specifies the priority order of this policy. Policies are evaluated in priority order, and the first classifier and policy match determine the action taken against the flow and where the statistics for that flow are aggregated.
- **Accelerate:** Accelerates the traffic from within this flow using one of the available application accelerators. This provides additional performance improvement above and beyond that provided by the WAN optimization components defined in Action.

Settings configured in the policy are employed in conjunction with one another. For instance, the CIFS policy is, by default, configured to leverage the CIFS accelerator prior to leveraging the *full optimization* (DRE, PLZ, TFO) capabilities of the underlying WAN optimization layer. This can be coupled with a configuration that applies a specific DSCP marking to the packets within the flow. This is defined in a single policy, thereby simplifying overall system policy management.

Classifiers within the ATP engine can be defined based on source or destination IP addresses or ranges, TCP port numbers or ranges, or universally unique identifiers (UUIDs). The ATP engine is consulted only during the establishment of a new connection, which is identified through the presence of the TCP synchronize (SYN) flag, which occurs within the first packet of the connection. By making a comparison against the ATP using the SYN packet of the connection being established, the ATP engine does not need to be consulted for traffic flowing in the reverse direction, because the context of the flow is established by all WAAS devices in the path between the two endpoints and applied to all future packets associated with that particular flow. In this way, classification performed by the ATP engine is done once against the three-way handshake (SYN, SYN/ACK packets) and is applicable to both directions of traffic flow.

Note DSCP markings are used to group packets together based upon business relevance, application characteristics, and performance requirements. Routers use DSCP to prioritize traffic based on QoS policies. WAAS can either preserve the existing DSCP markings or apply a specific marking to the packets matching the flow based on the configuration of this setting.

Disk Encryption

Cisco WAAS devices can encrypt the data, swap, and spool partitions on the hard disk drives using encryption keys that are stored on and retrieved from the Central Manager. The disk encryption uses the strongest commercially available encryption (AES-256).

During the device boot process, WAAS retrieves the encryption keys from the Central Manager, which are then stored locally in nonpersistent device memory. When power is removed from the device, the copy of the key is not retained. This protects WAAS devices from being physically compromised or having a disk stolen. Without the encryption keys, the encrypted data is unusable and scrambled.

The encryption keys are stored in the Central Manager database (which can be encrypted too) and synchronized among all Central Manager devices for high availability. The encryption key is fetched from the Central Manager over the SSL-encrypted session that is used for message exchanges between the WAAS devices and the Central Manager devices.

Cisco WAAS Platforms

The current Cisco WAAS platform consists of router-integrated network modules, appliance models, ISR 4000 Series routers integrated with WAAS (ISR-WAAS), and virtual appliance models. With such a diverse hardware portfolio, Cisco WAAS can be deployed in each location with the appropriate amount of optimization capacity for the needs of the users or servers in that particular location.

Every Cisco WAAS device, regardless of form factor, uses two different types of storage that correlate to either *boot time* or *run time*. The boot-time storage, used for booting the WAAS OS, maintains configuration files and is typically a compact flash card for physical devices. The run-time storage is hard disks for optimization data (including object cache and DRE), swap space, and a software image storage repository. Separating the two types of storage allows the device to remain accessible on the network for troubleshooting in the event of a hard drive failure.

This section explains the current platforms and positioning of each. Performance and scalability metrics for each platform are examined later in this section, along with a methodology for accurately sizing a Cisco WAAS deployment.

Router-Integrated Network Modules

The Cisco WAAS router-integrated network modules are designed to provide optimization services for the remote branch office or enterprise edge. These modules occupy an available network module slot in a Cisco Integrated Services Router (ISR). The ISR is an ideal platform for the branch office in that it provides a converged service platform for the remote office, including routing, switching, wireless connectivity, voice, security, and WAN optimization in a single chassis (platform, software version, and slot capacity dependent).

The ISR is a strong foundational hardware component for Cisco IWAN solutions. Table 12-1 shows the Cisco UCS-E family of network modules.

Appliances

The Cisco WAAS appliance platforms accommodate deployment scenarios of any size, such as small branch offices, campus networks, or the largest of enterprise data center networks. The Cisco WAVE appliance platform includes models 294, 594, 694, 7541, 7571, and 8541. WAVE appliance models 294 and 694, along with WAVE appliance model 594, are targeted toward branch office deployments, whereas the WAVE appliance models 7541, 7571, and 8541 are targeted toward large regional office and data center deployments. The WAVE-694 is a hybrid device that is commonly used for larger branch offices or small data centers.

Each of the WAVE appliance models 294, 594, 694, 7541, and 7571 has externally accessible hard disk drives and RAID support (some models support hot-swappable disk drives). Each WAVE appliance has two built-in Gigabit Ethernet interfaces, which can be deployed independently of one another or as a pair in either an active/standby configuration or port channel configuration. Each WAVE appliance can be deployed using a variety of network connectivity, interception modules, and techniques, including physical inline interception, WCCPv2, PBR, or AppNav. WAAS appliance selection should be based upon performance, and scalability recommendations should be followed.

WAVE Model 294

The Cisco WAVE model 294 (WAVE-294) can contain either 4 or 8 GB of RAM and a single 250 GB SATA hard disk drive with the option to use a 200 GB SSD instead. The WAVE-294 has two built-in Gigabit Ethernet ports that can be used for interception or management. The WAVE-294 supports an optional inline card (with support for two WAN links) that has either four or eight Gigabit Ethernet ports.

WAVE Model 594

The Cisco WAVE model 594 (WAVE-594) can contain 8 to 12 GB of RAM. The WAVE-594 has a maximum usable storage of 400 GB (one 400 GB SSD drive) or 500 GB (two 500 GB SATA drives in a RAID). The WAVE-594 supports a larger number of optimized TCP connections and higher levels of WAN bandwidth than the WAVE-294.

Table 12-1 *UCS-E Platforms*

	UCS EN120E	UCS EN140N	UCS EN120S-M2	UCS E140S M1 and M2	UUCS E160DM2, UCS-E180D M2, CS E140D M1, E160D M1	E140DP M1 and E160DP M1
Form Factor	EHWIC	NIM	Single-wide	Single-wide	Double-wide	Double-wide servers with PCIe cards
Processor	Intel Atom Processor C2358	Intel Atom Processor C2518	Intel Pentium Processor B925C	Intel Xeon Processor	Intel Xeon Processor	Intel Xeon Processor
	(2 cores, 1 M cache, 1.70 GHz)	(4 cores, 2 M cache, 1.70 GHz)	(4 M cache, 2 cores)	E3-1105C v2 (6 MB cache, 1.8 GHz, and 4 cores for M2)	E5-2428L v1, V2 (15 MB cache, 1.8 GHz, and 6 or 8 cores)	E5-2428L (15 MB cache, 1.8 GHz, and 6 cores)
				E3-1105C (6 MB cache, 1.00 GHz, and 4 cores for M1)	E5-2418L v1, V2 (10 MB cache, 2.0 GHz, and 4 or 6 cores)	E5-2418L (10 MB cache, 2.0 GHz, and 4 cores)
Memory	8 GB	8 GB	Default: one 4 GB DIMM and up to 16 GB (two 8 GB DIMMs)	Default: one 8 GB DIMM and up to 16 GB (two 8 GB DIMMs)	8 GB (default) and up to 48 GB (three 16 GB DIMMs)	8 GB (default) and up to 48 GB (three 16 GB DIMMs)
Storage	One hard drive:	One hard drive:	Up to two:	Up to two:	Up to three :	Up to two:
	64 GB (50 GB) MSATA	64 GB (50 GB) MSATA	7200 RPM SATA: 1 TB	7200 RPM SATA: 1 TB	7200 RPM SATA: 1 TB	7200 RPM SATA: 1 TB
	128 GB (100 GB) MSATA	128 GB (100 GB) MSATA	10,000 RPM SAS: 900 GB	10,000 RPM SAS: 900 GB	10,000 RPM SAS: 900 GB	10,000 RPM SAS: 900 GB
	256 GB (200 GB) MSATA	256 GB (200 GB) MSATA		10,000 RPM SAS SED: 600 GB	10,000 RPM SAS SED: 600 GB	10,000 RPM SAS SED: 600 GB

	UCS EN120E	UCS EN140N	UCS EN120S-M2	UCS E140S M1 and M2	UUCS E160DM2, UCS-E180D M2, CS E140D M1, E160D M1	E140DP M1 and E160DP M1
				SAS SSD SLC: 200 GB	SSD SLC: 200 GB	SSD SLC: 200 GB
				SSD eMLC: 200 GB and 400 GB	SSD eMLC: 200 GB and 400 GB	SSD eMLC: 200 GB and 400 GB
RAID Options	N/A	N/A	Hardware RAID 0 and 1	Hardware RAID 0 and 1	Hardware RAID 0, 1, and 5	Hardware RAID 0, 1, and 5
Network Interface Cards	Two internal and one external Gigabit Ethernet ports	Two internal and one external Gigabit Ethernet ports	Two internal and one external Gigabit Ethernet ports	Two internal and one external Gigabit Ethernet ports	Two internal and two external Gigabit Ethernet ports	Two internal and two external Gigabit Ethernet ports
PCIe	None	None	None	None	None	1 Gigabit Ethernet
						One 10 Gigabit Ethernet, FCoE SFP+
Supported Cisco ISRs	Cisco 1921, 1941, 2901, 2911, 2921, 2951, 3925, 3925E, 3945, and 3945E	Cisco ISR 4321, 4331, 4351, 4431, and 4451	Cisco 2911, 2921, 2951, 3925, 3925E, 3945, 3945E, 4331, 4351, and 4451	Cisco 2911, 2921, 2951, 3925, 3925E, 3945, 3945E, 4331, 4351, and 4451	Cisco UCS E140D M1 and UCS E160D M2: 2921, 2951, 3925, 3925E, 3945, 3945E, 4331, 4351, and 4451	Cisco UCS E140DP: 2921, 2951, 3925, 3925E, 3945, 3945E, 4331, 4351, and 4451
					Cisco UCS E160D M1 and UCS E180D M2: 3925, 3925E, 3945, 3945E, 4351, and 4451	Cisco UCS E160DP: 3925, 3925E, 3945, 3945E, 4331, 4351, and 4451

The 594 has two onboard Gigabit Ethernet ports that can be used for interception or management. The 594 supports optional inline cards with four-port Gigabit Ethernet copper inline, eight-port Gigabit Ethernet copper inline, or four-port Gigabit Ethernet SX fiber module.

WAVE Model 694

The Cisco WAVE model 694 (WAVE-694) contains 16 or 24 GB of RAM. The WAVE-694 supports two 600 GB SATA hard disk drives, which are configured for software RAID 1. The 694 has two onboard Gigabit Ethernet ports that can be used for interception or management. I/O modules (IOMs) supported are the four-port Gigabit Ethernet copper inline, eight-port Gigabit Ethernet copper inline, or four-port Gigabit Ethernet SX fiber.

WAVE Model 7541

The Cisco WAVE model 7541 (WAVE-7541) contains 24 GB of RAM and 2.2 TB of storage (six 450 GB SATA drives). The 7541 has two onboard Gigabit Ethernet ports that can be used for interception or management. This platform also supports the following optional modules: eight-port Gigabit Ethernet copper inline, four-port Gigabit Ethernet SX fiber inline, or two-port 10 Gigabit Ethernet Enhanced Small Form-Factor Pluggable (SFP+) module.

WAVE Model 7571

The Cisco WAVE model 7571 (WAVE-7571) contains 48 GB of RAM and supports eight 450 GB hard disk drives in a RAID 5 for a total of 3.2 TB of storage. The 7571 has two onboard Gigabit Ethernet ports that can be used for interception or management. The platform offers an optional eight-port Gigabit Ethernet copper inline, four-port Gigabit Ethernet SX fiber inline, or two-port 10 Gigabit Ethernet Enhanced Small Form-Factor Pluggable (SFP+) module.

WAVE Model 8541

The Cisco WAVE model 8541 (WAVE-8541) contains 96 GB of RAM and eight 600 GB hot-swappable hard drives in a RAID 5 (4.2 TB of storage). The 8541 has two onboard Gigabit Ethernet ports that can be used for interception or management. The platform supports an optional eight-port Gigabit Ethernet copper inline, four-port Gigabit Ethernet SX fiber inline, or two-port 10 Gigabit Ethernet Enhanced Small Form-Factor Pluggable (SFP+) module.

Interception Modules

The WAVE appliances listed also support a variety of deployment and connectivity options:

- Four-port Gigabit Ethernet copper module (WAVEINLN-GE-4T)
  - Fail-to-wire capability
  - Support for inline and WCCP deployments
- Eight-port Gigabit Ethernet copper module (WAVEINLN-GE-8T)
  - Fail-to-wire capability
  - Support for inline and WCCP deployments
- Four-port Gigabit Ethernet (SX) fiber module (WAVE-INLN-GE-4SX)
  - Fail-to-wire capability
  - Support for inline and WCCP deployments
- Two-port 10 Gigabit Ethernet module (WAVE-10GE-2SFP)
  - Support for Cisco SFP+ short reach (SR) transceivers
  - Support for WCCP interception only

Virtual WAAS

The Cisco *Virtual WAAS (vWAAS)* platform is designed to be deployed where physical WAVE devices cannot be deployed. vWAAS software can be installed on VMware ESXi 5.0 and later and is provided as an *Open Virtual Appliance (OVA)* that is prepackaged with disk, memory, CPU, NICs, and other VMware-related configuration in an *Open Virtualization Format (OVF)* file format. Cisco vWAAS OVA files are provided based on vWAAS models. Table 12-2 provides a matrix of vWAAS models.

Table 12-2 *Virtual WAAS Matrix*

OVA Name	Connection Capacity	Similar Appliance	CPU	RAM (GB)	Disk (GB)	Optimized Connections	WAN Bandwidth	LAN Throughput (Mbps)	Maximum Number of Peers (Fan-out)
vWAAS-200	200	294-4G	1	2	160	200	10	20	100
vWAAS-750	750	594-8G	2	4	250	750	50	100	100
vWAAS-1300	1300	594-12G	2	6	300	1300	80	150	200
vWAAS-2500	2500	694-16G	4	8	400	2500	200	400	300
vWAAS-6000	6000	694-24G	4	8	500	6000	200	500	300
vWAAS-12000	12,000	N/A	4	12	750	12,000	310	1000	1,400
vWAAS-50000	50,000	N/A	8	48	1500	50,000	1000	2000	2,800

Also, Cisco provides a Virtual Central Manager (vCM) with the appropriate capacity guidelines provided in Table 12-3.

Table 12-3 *WAAS Virtual Central Manager Matrix*

OVA Name	Capacity	Similar Appliance	CPU	RAM (GB)	Disk (GB)
vCM-100N	Manages 100 nodes		2	2	250
vCM-500N	Manages 500 nodes		2	2	300
vCM-1000N	Manages 1000 nodes	594-8G	2	4	400
vCM-2000N	Manages 2000 nodes	694-16G	4	8	600

ISR-WAAS

ISR-WAAS is a virtualized WAAS instance that runs on an ISR 4000 Series router using a Cisco IOS XE integrated container. The term “container” refers to the KVM hypervisor that runs virtualized applications on IOS XE–based platforms. The term “host” refers to the primary operating system running on a system. For example, in ISR-WAAS on Cisco ISR 4451-X, the host is defined as a Cisco ISR 4451-X running on Cisco IOS XE. ISR-WAAS provides WAN optimization functions to the Cisco ISR 4451-X in this example.

Architecture

The Cisco ISR 4000 Series router uses Cisco IOS XE software. IOS XE uses key architectural components of IOS while adding stability through platform abstraction and the use of a Linux kernel. The Linux kernel allows processes to execute across multiple CPUs. All the routing protocols exist within a process called IOSd. Programming of the device hardware runs in a different process. Platform management can access all the necessary components of the IOS XE platform as part of the configuration management process.

APIs allow an application to run as a *service container* on the same hardware as the router. The Linux-based OS facilitates separation of the data and control planes and uses dedicated CPUs for services. Because the services plane is separate from the data and control planes, the router can handle more services on a single platform, allowing an office to consolidate functions onto a single device.

Service containers offer dedicated virtualized computing resources that include CPU, disk storage, and memory for each service. A hypervisor presents the underlying infrastructure to the application or service. This scenario offers better performance than a tightly coupled service, deployment with zero footprint, security through fault isolation, and the flexibility to upgrade network services independent of the router software.

Figure 12-2 shows both the Cisco 4300 (right) and 4400 (left) Series architecture, which includes physical separation of the control and data planes in the 4400. Feature support is identical.

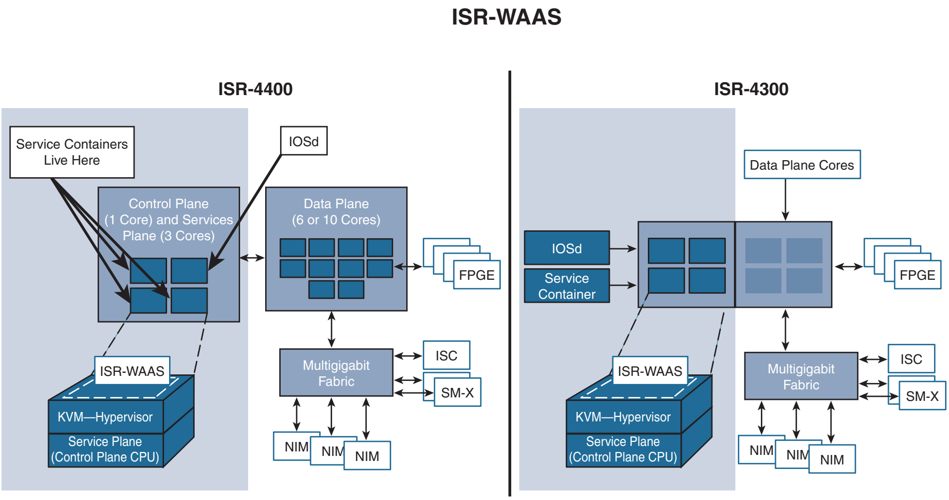


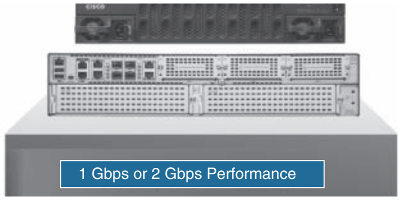
Figure 12-2 *Cisco ISR 4000 Architecture*

Sizing

ISR-WAAS sizing is based on several factors, including concurrent TCP connections, target WAN bandwidth, and router platform. Optimized connections range from 750 to 2500 connections. Figures 12-3 through 12-6 include the resources and sizing metrics for ISR-WAAS per platform including the platform's throughput.

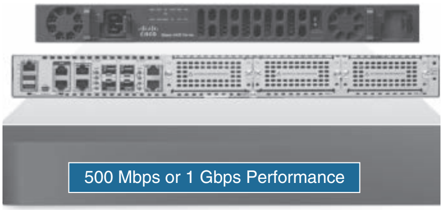
The ISR 4451-X offers 1 Gbps performance, upgradable to 2 Gbps via performance license, two physical processors, a four-core processor (one control and three services), and a 10-core data plane processor.

The 4431, shown in Figure 12-4, offers 500 Mbps performance, upgradable to 1 Gbps via performance license. It has two physical processors, four core processors (one control and three services), and six core data plane CPUs.



	ISR-WAAS-2500	ISR-WAAS-1300	ISR-WAAS-750
TCP Connections	2500	1300	750
Target WAN Bandwidth	150 Mbps	100 Mbps	75 Mbps
DRE Disk Capacity	210 GB	70 GB	70 GB
Maximum Peer	200	200	200
Akamai Connect	50 GB	30 GB	30 GB
Optimized LAN Throughput	400	300	200
vCPU	6	4	2
Virtual Memory	8 GB	6 GB	4 GB
Data Storage	360 GB	170 GB	170 GB
Number of SSDs Required	2	1	1

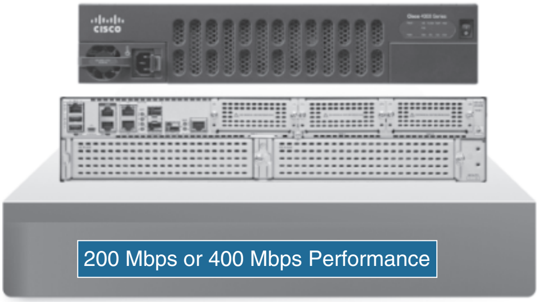
Figure 12-3 *ISR 4451WAAS*



	ISR-WAAS-1300	ISR-WAAS-750
TCP Connections	1300	750
Target WAN Bandwidth	50 Mbps	25 Mbps
DRE Disk Capacity	70 GB	70 GB
Maximum Peer	200	200
Akamai Connect	30 GB	30 GB
Optimized LAN Throughput	250 Mbps	200 Mbps
vCPU	4	2
Virtual Memory	6 GB	4 GB
Data Storage	170 GB	170 GB
Number of SSDs Required	1	1

Figure 12-4 *ISR 4431 ISR-WAAS*

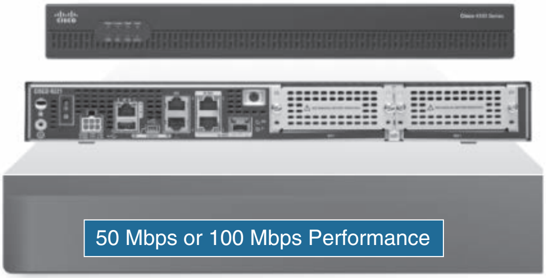
The Cisco 4351, shown in Figure 12-5, offers 200 Mbps performance, upgradable to 400 Mbps, one physical eight-core CPU with four data plane cores, one control plane core, and three cores dedicated to services. The Cisco 4331 offers 100 Mbps performance, upgradable to 300 Mbps with the same CPU configuration as the 4351.



	ISR-WAAS-750
TCP Connections	750
Target WAN Bandwidth	25 Mbps
DRE Disk Capacity	70 GB
Maximum Peer	200
Akamai Connect	30 GB
Optimized LAN Throughput	100
vCPU	2
Virtual Memory	4 GB
Data Storage	170 GB
Number of SSDs Required	1

Figure 12-5 *ISR 4351 and 4331 ISR-WAAS*

The 4321, shown in Figure 12-6, offers 50 Mbps performance, upgradable to 100 Mbps, a single four-core CPU with two data plane cores, one control plane core, and one core dedicated to services.



	ISR-WAAS-200
TCP Connections	200
Target WAN Bandwidth	15 Mbps
DRE Disk Capacity	70 GB
Maximum Peer	200
Akamai Connect	30 GB
Optimized LAN Throughput	50 Mbps
vCPU	1
Virtual Memory	2 GB
Data Storage	170 GB
Number of SSDs Required	1

Figure 12-6 *ISR 4321 ISR-WAAS*

WAAS Performance and Scalability Metrics

The design of a Cisco WAAS solution involves many factors, but performance and scalability metrics are the cornerstone for the solution as a whole while taking into account every individual location. Every component in an end-to-end system has a series of static and dynamic system limits. For instance, a typical application server may be limited in terms of the number of connections it can support, disk I/O throughput, network throughput, CPU speed, or number of transactions per second. Likewise, each Cisco WAAS device has static and dynamic system limits that dictate how and when a particular WAAS device is selected for a location within an end-to-end design.

This section examines the performance and scalability metrics of the Cisco WAAS platform and provides a definition of what each item is and how it is relevant to a localized (per-location) design and an end-to-end system design. The static and dynamic limits referred to are used as a means of identifying which device is best suited to providing services to a particular location in the network. The device may be deployed as an edge device, where it connects to potentially many peer devices in one or more data center locations, or as a core device, where it serves as an aggregation point for many connected edges. WAAS devices can also be deployed as devices to optimize links between data center locations, where devices on each side are realistically core devices.

A fundamental understanding of the performance and scalability metrics is paramount in ensuring a proper design. Although WAAS devices have no concept of the network core or edge, the deployment position within the network has an effect on the type of workload handled by a device and should be considered—primarily as it relates to TCP connection count and peer fan-out (how many peers can connect to a device for the purpose of optimization).

This section also examines each of the performance and scalability system limits, both static and dynamic, that should be considered. These include device memory, disk capacity, the number of optimized TCP connections, WAN bandwidth and LAN throughput, the number of peers (fan-out), and the number of devices managed.

WAAS Design and Performance Metrics

This section discusses design and deployment considerations in deploying WAAS over data center and branch architectures.

Device Memory

The amount of memory installed in a WAAS device dictates the level of performance and scalability the device can provide. As the memory capacity increases, the ability of a WAAS device to handle a larger number of connections, a larger addressable index space for compression, or a longer history of compression data also increases. Having larger amounts of memory also enables the WAAS device to run additional services, such as application acceleration or disk encryption, and positions the device to accept additional features that might be introduced in future software releases. For devices that support flexible memory configuration (such as the WAVE-294, WAVE-594, and WAVE-694), higher levels of WAN bandwidth can be realized, along with an increase in the number of optimized TCP connections that the device can handle concurrently.

Disk Capacity

Optimization services in the Cisco WAAS hardware platforms leverage both memory and disk. From a disk perspective, the larger the amount of available capacity, the larger the amount of optimization history that can be leveraged by the WAAS device during runtime operation. For instance, a WAVE-294 (with 4 GB of DRAM) has 200 GB of physical disk capacity, of which 40 GB is available for use by DRE for compression history. With 40 GB of compression history, one can estimate the length of the compression history given WAN conditions, expected network utilization, and assumed redundancy levels.

Table 12-4 shows how the length of the compression history can be calculated for a particular WAAS device, along with an example. This example assumes a T1 WAN that is 75 percent utilized during business hours (75 percent utilization over 8 hours per day) and 50 percent utilized during nonbusiness hours (16 hours per day). It also assumes that data traversing the network is 75 percent redundant (highly compressible by DRE). This table also assumes a WAVE-294 with 40 GB of allocated capacity for DRE compression history.

Table 12-4 *Calculating Compression History*

Step Logic	Calculations
Convert WAN capacity to bytes (divide the number of bits per second by 8)	(T1 = 1.544 Mbps)/8 = 193 Kbps
Identify maximum WAN throughput for a given day (convert seconds to minutes, to hours, to a day)	193 Kbps * 60 sec/min 11.58 MB/min * 60 min/hr 694.8 MB/hr * 24 hr per day Total = 16.68 GB per day
Identify WAN throughput given utilization (multiply by the number of hours and utilization per hour)	694.8 MB/hr * 8 hr * 75% utilization = 4.168 GB 694.8 MB/hr * 16 hr * 50% utilization = 5.56 GB Total = 9.72 GB/day
Identify WAN throughput given utilization and expected redundancy (multiply daily throughput by expected redundancy or compressibility)	9.72 GB/day * .25 (.75 is 75% redundancy) = 2.43 GB/day
Calculate compression history (divide capacity by daily throughput)	Storage capacity of unit divided by daily throughput 40 GB divided by 2.43 GB/day = 16.5 days of history

The disk capacity available to a WAAS device is split among five major components:

- **DRE compression history:** This capacity is used to store DRE chunk data and signatures.
- **CIFS cache:** This capacity is pre-allocated.
- **Platform services:** This capacity is pre-allocated for operating system image storage, log files, and swap space.
- **Print services:** This capacity is pre-allocated for print spool capacity.
- **Akamai Connect Object Cache:** This capacity is pre-allocated for AKC HTTP object cache capacity.

Number of Optimized TCP Connections

Each WAAS device has a static number of TCP connections that can be optimized concurrently. Each TCP connection is allocated memory and other resources within the system, and if the concurrently optimized TCP connection static limit is met, additional connections are handled in a pass-through fashion. Adaptive buffering (memory allocation) is used to ensure that more active connections are allocated additional memory, and less active connections are allocated only the memory they require.

The TCP connection limit of each WAAS device can be roughly correlated to the number of users supported by a given WAAS device model, but note that the number of TCP connections open on a particular node can vary based on user productivity, application behavior, time of day, and other factors. It is commonly assumed that a user has 10 to 15 connections open at any given time. If necessary, policies can be adjusted on the WAAS Central Manager to pass through certain applications that may realize only a small benefit from WAAS. This type of change could potentially help increase the number of users who can be supported by a particular WAAS device.

Table 12-5 displays a list of models and the number of optimized connections that are supported on it.

Table 12-5 *Optimized TCP Connection Capacity per Device Model*

Appliance Model	Optimized Connections
WAVE-294	200
WAVE-294-8GB	400
WAVE-594	750
WAVE-594-12GB	1300
WAVE-694	2500
WAVE-694-24GB	6000

(*Continued*)

Table 12-5 *Continued*

Appliance Model	Optimized Connections
WAVE-7541	18,000
WAVE-7571	60,000
WAVE-8541	150,000

The number of connections a typical user has in a location can be determined by using tools that exist in the operating system of the user's workstation. Although the estimate of six to ten optimized TCP connections is accurate for the majority of customers, those who wish to more accurately determine how many connections a typical user has open at any given time can do so.

For the data center, the sum of all remote office TCP connections should be considered one of the key benchmarks by which the data center sizing should be done. Note that the largest Cisco WAAS device supports up to 150,000 optimized TCP connections—which is approximately 15,000 users (assuming 10 TCP connections per user). For organizations that need to support a larger number of users or want to deploy the data center devices in a high-availability manner, multiple devices can be used. The type of network interception used determines the aggregate number of optimized TCP connections that can be supported by a group of Cisco WAAS devices deployed at a common place within the data center.

Recommended practice dictates that sites that require high availability be designed with *N* + 1 availability relative to the number of maximum optimized TCP connections—that is, if 100,000 optimized TCP connections must be supported, the location should have a minimum of two WAVE-7571 devices to support the workload, a third WAVE-7571 device to handle failure of one of the devices, and an interception mechanism such as WCCP or AppNav that supports load balancing of the workload across all three devices.

WAN Bandwidth and LAN Throughput

WAAS devices are not restricted in software or hardware in terms of the amount of WAN bandwidth or LAN throughput supported. However, recommendations are in place to specify which WAAS device should be considered for a specific WAN environment. WAN bandwidth is defined as the amount of WAN capacity that the WAAS device can fully use when employing the full suite of optimization capabilities (this includes DRE, PLZ, TFO, and the other application acceleration capabilities). LAN throughput is defined as the maximum amount of application-layer throughput (throughput as perceived by users and servers) that can be achieved with the particular WAAS hardware model and an equivalent or more powerful peer deployed at the opposite end of the network.

For some deployment scenarios, such as data replication, it is desirable to use the Cisco WAAS devices only for TCP optimization. Cisco WAAS TFO provides a suite of optimizations to better allow communicating nodes to "fill the pipe" (that is, fully leverage the available WAN bandwidth capacity) when the application protocol is not restricting throughput because of application-induced latency. Each Cisco WAAS device has a TFO-only throughput capacity that can be considered when WAAS devices are deployed strictly for TCP optimization. TFO-only optimization is recommended only for situations where compression, redundancy elimination, and application acceleration are not required, and the application throughput has been validated to be hindered only by the performance of the TCP implementation in use. This is common in some data-center-to-data-center applications, such as data replication or data protection, where the traffic that is sent is previously compressed, redundancy eliminated, or encrypted. TFO attempts to fully utilize the available bandwidth capacity but may be hindered by congestion in the network (not enough available bandwidth) or performance impedance caused by application protocol chatter.

Table 12-6 shows the WAN bandwidth supported by each WAAS device model and the maximum LAN-side throughput and targeted WAN bandwidth. Note that other factors can influence these values, and throughput levels can be achieved only when the link capacity available supports such a throughput level. For instance, a LAN throughput maximum of 150 Mbps is not possible on a Fast Ethernet connection; rather, a Gigabit Ethernet connection is required. Similarly, for throughput speeds of more than 1 Gbps, multiple 1 Gbps interfaces must be used.

Table 12-6 *WAN Bandwidth and LAN Throughput Capacity per WAAS Device*

Appliance Model	Optimized LAN Throughput	Target Bandwidth
WAVE-294	200	10 Mbps
WAVE-294-8GB	400	20 Mbps
WAVE-594	750	50 Mbps
WAVE-594-12GB	1300	100 Mbps
WAVE-694	2500	200 Mbps
WAVE-694-24GB	6000	200 Mbps
WAVE-7541	18,000	500 Mbps
WAVE-7571	60,000	1000 Mbps
WAVE-8541	150,000	2000 Mbps

The amount of bandwidth required per site is the sum of available WAN capacity that can be used at that site and not the sum of all WAN bandwidth for every connected peer. For instance, if a branch office has four bundled T1 links (totaling 6 Mbps of aggregate WAN throughput) but only two are used at any given time (high-availability configuration), a device that supports 3 Mbps or more is sufficient to support the location. Similarly, if a data center has four DS-3 links (totaling 180 Mbps of aggregate WAN throughput) but uses only three at a time (*N* + 1 configuration), a device that supports 135 Mbps of WAN bandwidth or more is sufficient to support that location.

The WAN throughput figures given in Table 12-6 are (as discussed previously) not limited in hardware or software. In some cases, the WAN throughput that a device achieves may be higher than the values specified here. Any WAAS system can optimize up to the maximum of its capacity until overload conditions arise. During overload conditions, new connections are not optimized. Existing connections are optimized to the greatest degree possible by the system. Should scalability beyond the capacity of a single device be required, multiple devices can be deployed. The maximum number of optimized TCP connections assumes that MAPI AO is disabled, and therefore the connection reservation requirement is removed. If MAPI AO is enabled, a small number of connections are put into a reservation pool to account for the multiple connections associated with a single user flow. Target WAN bandwidth is not limited in software or by any other system limit but rather is provided as guidance for deployment sizing purposes.

Target WAN bandwidth is a measure of the optimized/compressed throughput WAAS can support; this value is taken at approximately 70 percent compression. Maximum optimized LAN throughput numbers are measured with five to ten high-throughput connections optimized over a Gigabit network with minimal latency. LAN throughput may be affected by factors outside of WAAS; for example, the router that is doing the redirection may become the bottleneck. Actual results depend on the use case.

Number of Peers and Fan-out Each

Cisco WAAS devices have a static system limit in terms of the number of concurrent peers with which they can actively communicate at any given time. When designing for a particular location where the number of peers exceeds the maximum capacity of an individual device, multiple devices can be deployed, assuming that an interception mechanism that uses load balancing is employed (such as WCCPv2 or AppNav). In cases where load balancing is used, TCP connections are distributed according to the interception configuration, thereby allowing for near-linear scalability increases in connection count, peer count, and WAN bandwidth as devices are added to the pool. Load-balancing interception techniques are recommended when multiple devices are used in a location. Peer relationships are established between Cisco WAAS devices during the automatic discovery process on the first connection optimized between the two devices. These peer relationships time out after 10 minutes of inactivity (that is, no active connections are established and optimized between two peers for 10 minutes). Each WAAS device supports a finite number of active peers, and when the peer relationship is timed out, that frees up peering capacity that can be reused by another peer. Data stored in the DRE compression history remains intact even if a peer becomes disconnected because of inactivity, unless the DRE compression history becomes full. In cases where the DRE compression history becomes full, an eviction process is initiated to remove the oldest set of data to make room for new data.

Table 12-7 shows the maximum number of concurrent peers supported per WAAS platform. If peers are connected beyond the allocated limit, the WAVE permits the connections to be established and gracefully degrades performance as needed. Connections associated with peers in excess of the maximum fan-out ratio are able to use the existing compression history but are not able to add new chunks of data to it.

The result is lower effective compression ratios for the connections using peers that are in excess of the specified fan-out ratio.

Table 12-7 *Maximum Supported Peers per WAAS Device*

Appliance Model	Maximum Supported Peers
WAVE-294	100
WAVE-294-8GB	100
WAVE-594	100
WAVE-594-12GB	100
WAVE-694	150
WAVE-694-24GB	300
WAVE-7541	700
WAVE-7571	1400
WAVE-8541	2800

The number of peers supported by a device is typically the last factor that should be considered when sizing a solution for a particular location. The primary reason is that the WAN capacity or number of connections supported on the device (at the maximum concurrent peers specification) is generally higher than what the device can support. For instance, although a WAVE-294 can support up to 100 peers, even if those peers are the WAVE-694 (each supporting 2500 optimized TCP connections), it is not able to handle the 2500 possible optimized TCP connections that all WAVE-694s are attempting to optimize with it. It is best to size a location first based on WAN bandwidth capacity and TCP connections, and in most cases only a simple validation that the number of peers supported is actually required.

Central Manager Sizing

Each Cisco WAAS deployment must have at least one Cisco WAAS device deployed as a Central Manager. The Central Manager is responsible for system-wide policy definition, synchronization of configuration, device monitoring, alarming, and reporting.

The Central Manager can be deployed as appliances or a virtual instance and can be deployed in an active/standby fashion. When a certain type of WAAS device is configured as a Central Manager, it is able, based on the hardware or virtual platform selected for the Central Manager, to manage a maximum number of WAAS devices within the topology.

In high-availability configurations, each Central Manager WAVE or virtual instance should be of the same hardware configuration or managed node count. Although hardware disparity between Central Manager WAVEs works, it is not a recommended practice given the difference in the number of devices that can be managed among the WAVE hardware models. It should be noted that standby Central Managers receive information in a synchronized manner identically to how accelerator WAAS devices do. Table 12-8 shows the maximum number of managed nodes that can be supported by each WAAS appliance when configured as a Central Manager.

Table 12-8 *Central Manager Scalability*

Appliance Model (as a Central Manager)	Number of Managed Devices
WAVE-294	250
WAVE-294-8GB	250
WAVE-594	1000
WAVE-594-12GB	1000
WAVE-694	2000
WAVE-694-24GB	2000

Use of multiple WAAS devices configured as Central Manager devices does not increase the overall scalability in terms of the number of devices that can be managed. To manage a number of devices greater than the capacities given in Table 12-8, multiple autonomous Central Managers are needed. For instance, in an environment with 3000 devices, two separate instances of Central Manager are required, and each instance can be composed of a single device or multiple devices deployed in a high-availability primary/standby configuration.

Licensing

Licenses are not enforced in WAAS; however, licenses can be applied only to platforms that support the particular license in question. The Enterprise License is included with all WAAS device purchases. The Enterprise License allows a WAAS device to apply all the WAN and application acceleration techniques. WAAS devices that act as Central Managers also require the Enterprise License. To deploy both the ISR-WAAS and the AppNav-XE components, the Application Experience (appxk9 package) License is required.

Note Akamai Connect Licensing is an add-on license based on connection count of the WAAS node on which it is being deployed.

Cisco WAAS Operational Modes

By default, WAAS transparently (by preserving the packet's original source/destination IP addresses and TCP ports) sets up a new TCP connection to a peer WAVE, which can cause firewall traversal issues when a WAAS device tries to optimize traffic. If a WAVE device is behind a firewall that prevents traffic optimization, Directed Mode can be used. In Directed Mode, all TCP traffic that is sent to a peer WAVE is encapsulated in UDP, which allows a firewall to either bypass the traffic or inspect the traffic (by adding a UDP inspection rule).

Transparent Mode

By default, WAAS handles traffic transparently by preserving the packet's original source/destination IP addresses and TCP ports. The Transparent Mode of operation allows for end-to-end traffic visibility, which eases interoperability with existing network-based QoS, access control, and performance management/reporting capabilities.

Directed Mode

WAAS version 4.1 added an alternative mode of operation called Directed Mode. Directed Mode transports optimized connections using a nontransparent (UDP-encapsulated) mechanism between two WAAS devices. The source and destination IP addresses of the encapsulated packet are the IP addresses of the WAEs themselves.

Directed Mode relies on the auto-discovery process to establish the peer relationship between two WAEs. This means that Directed Mode does not bypass any security measures. Initial TCP traffic flows between client and server must pass through any firewalls before traffic can be optimized by WAAS in Directed Mode. After the auto-discovery process succeeds, the server-side WAE sends a *TCP reset (RST)* packet toward the client-side WAE to clear out the connection state on any intermediate devices between the WAEs. Future traffic for the connection is then encapsulated in a UDP header with a configurable source and destination port of 4050.

Interception Techniques and Protocols

There are two approaches to leveraging the network infrastructure to intercept and redirect traffic to WAAS for optimization. The first method relies on interception protocols or routing configuration used by the networking components (routers and switches) to selectively intercept traffic and redirect it to the WAAS infrastructure. This method is referred to as off-path interception. The most common method for off-path network interception is WCCPv2.

The second method places the WAVE physically inline between two network devices, most commonly a router and a LAN switch. All traffic between the two network devices passes through the WAVE, which can then selectively intercept traffic for optimization. This method is referred to as in-path interception, because the WAVE is physically placed in the data path between the clients and servers.

This section discusses both off-path (WCCPv2) and in-path (inline) interception in detail. It also discusses other interception options for specific use cases, such as PBR and AppNav. These additional interception options add to the flexibility with which WAAS can be integrated into existing network infrastructures of all sizes.

Web Cache Communication Protocol

This section does not provide an exhaustive reference for the WCCPv2 protocol. Rather, it provides enough information about the protocol background and concepts to enable you to understand the WCCPv2 implementation in Cisco WAAS.

WCCP is a transparent interception protocol developed by Cisco Systems, Inc., in 1997. WCCP is a control plane protocol that runs between devices running Cisco IOS and WCCP "clients" such as WAAS. The protocol enables the network infrastructure to selectively intercept traffic based on IP protocol and transport protocol port numbers and redirect that traffic to a WCCP client. WCCP is considered transparent, because it allows for local interception and redirection of traffic without any configuration changes to the clients or servers. WCCP has built-in load-balancing, scalability, fault tolerance, and service assurance (fail-open) mechanisms.

The current version, WCCPv2, is used by Cisco WAAS to transparently intercept and redirect all TCP traffic, regardless of port. The following section describes the basic WCCPv2 concepts and how they are specifically used by Cisco WAAS.

WCCP Service Groups

The routers and WAVEs participating in the same service constitute a service group. A service group defines a set of characteristics about what types of traffic should be intercepted, as well as how the intercepted traffic should be handled. There are two types of service groups:

- **Well-known services**, also referred to as static services, have a fixed set of characteristics that are known by both IOS and WCCPv2 client devices. There is currently a single well-known service called Web-Cache. This service redirects all TCP traffic with a destination port of 80.
- **Dynamic services** are initially known only to the WCCPv2 clients within the service group.

The characteristics of the service group are communicated to the IOS devices by the first WCCPv2 client device to join the service group. A unique service ID, which is a number from 0 to 255, identifies service groups. Service IDs 0 to 50 are reserved for well-known services.

The WCCPv2 implementation in WAAS supports a single dynamic WCCPv2 service, the TCP-Promiscuous service. Although referred to in WAAS as a single service, the TCP-Promiscuous service is in fact two different services. The two service IDs enabled with the TCP-Promiscuous service are 61 and 62. These are the two service group IDs that are configured in IOS when using WCCPv2 with WAAS.

Two different service groups are used because by default both directions (client to server and server to client) of a TCP connection must be transparently intercepted. To optimize a connection, WAAS must see both directions of the connection on the same WAE. Not only does WAAS intercept the connection in both directions, but it also intercepts the connection on both sides of the WAN link. Because the packet Layer 3 and Layer 4 headers are preserved, transparent interception is used on both sides of the WAN in both directions to redirect connections to the WAAS infrastructure for optimization. Figure 12-7 shows a basic topology with WCCPv2 interception configured for WAAS.

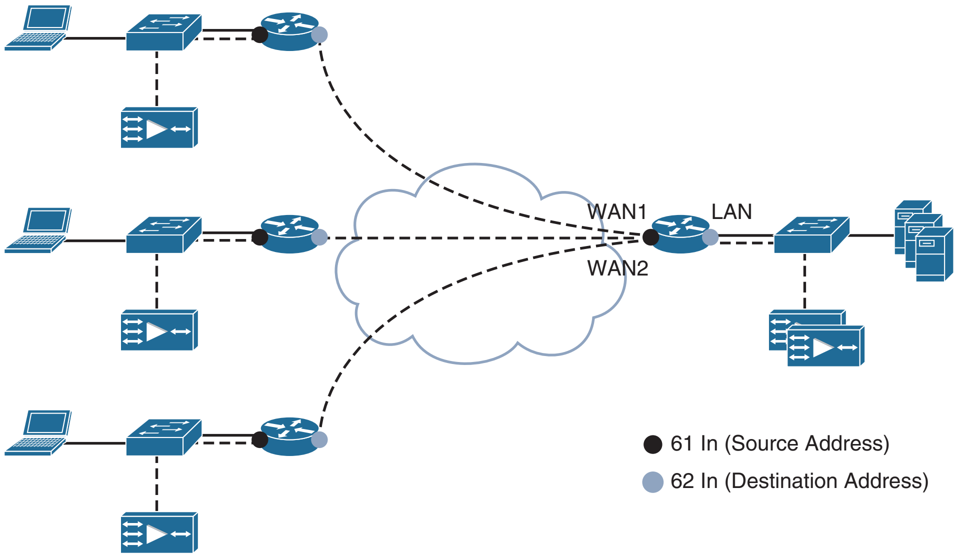


Figure 12-7 *Basic Network Topology with WCCP*

By default, service group 61 hashes on the source IP address and service group 62 hashes on the destination IP address. Later, this chapter discusses the significance of the hash key used in each service group. By default, the spoof-client-ip feature is enabled for both services. This is the WCCPv2 feature that allows WAAS to handle optimized traffic transparently. Traffic forwarded to the WAE uses the same source and destination IP addresses and TCP ports as when it entered the WAVE.

Forwarding and Return Methods

WCCPv2 supports different methods of forwarding redirected traffic from an IOS router or switch to a WAVE, and for the WAVE to return traffic to the IOS router/switch for forwarding. These methods are referred to as the forwarding and return methods and are negotiated between IOS and the WAVE when a WAVE joins the service group. The forwarding method defines how traffic that is being redirected from IOS to the WAVE is transmitted across the network.

- The first forwarding method, *GRE forwarding*, encapsulates the original packet in a WCCP GRE header with the destination IP address set to the target WAVE and the source IP address set to the WCCPv2 router ID of the redirecting router. When the WAVE receives the GRE-encapsulated packet, the GRE header is removed, and the packet is processed.

- The second forwarding method, *L2 forwarding*, simply rewrites the destination MAC address of the packet being redirected to equal the MAC address of the target WAVE. This forwarding method assumes that the WAE is Layer 2 adjacent to the redirecting router. One of the benefits of L2 forwarding is that it allows the WCCPv2 redirection to occur in hardware.

The return method defines how traffic should be returned from the WAVE to the redirecting router/switch for normal forwarding. As with the forwarding method, there are two different return methods:

- In *GRE return*, egress traffic from the WAE using GRE return is encapsulated using WCCP GRE, with a destination IP address of the WCCPv2 router ID and a source IP address of the WAVE itself. When the WCCPv2-enabled router receives the returned packet, the IP GRE header is removed and the packet is forwarded normally. WCCPv2 in IOS knows not to re-intercept traffic returned to it using GRE return.
- *L2 return* returns traffic to the WCCPv2-enabled router by rewriting the destination MAC address of the packet to equal the MAC address of the WCCPv2-enabled router. Whether the negotiated return method is used by WAAS to inject traffic back into the network infrastructure is determined by the configured egress method.

Load Distribution

When multiple WAVEs exist in a service group, WCCPv2 automatically distributes redirected traffic across all WAVEs in the service group. When traffic passes through an IOS device with WCCPv2 redirection configured, the IOS device assigns traffic for that connection to a bucket. Each bucket is assigned to a specific WAVE. The method that determines to which bucket traffic is assigned, which determines how traffic is distributed across multiple WAEs within a service group, is called the assignment method.

The bucket assignments are communicated from the lead WAE to all IOS devices in the service group. The assignment method can use either a hashing or a masking scheme and is negotiated between IOS and WAVE during the formation of the service group.

Hash assignment, which is the default assignment method, performs a bitwise hash on a key identified as part of the service group. In WAAS, the hash key used for service group 61 is the source IP address, whereas the hash key used for service group 62 is the destination IP address. The hash is not configurable and is deterministic in nature. This means that all the routers within the same service group make the same load-balancing decision given the same hash key. This deterministic behavior is what allows WCCPv2 to support asymmetric traffic flows, so long as both directions of the flow pass through WCCPv2-enabled IOS devices in the same service group. Hash assignment uses 256 buckets.

The second assignment method is called *mask assignment*. With mask assignment, the source IP address, destination IP address, source port, and destination port are concatenated and ANDed with a 96-bit mask to yield a value. The resulting 96-bit value is compared to a list of mask/value pairs. Each mask/value pair is associated with a bucket, and each bucket is in turn assigned to a WAVE. Unlike hash assignment, the number of buckets used with mask assignment depends on the number of bits used in the mask. By default, WAAS uses a mask of 0x1741. This results in 26 buckets that can be assigned across the WAVEs in a service group.

Failure Detection

After a WAVE has successfully joined a service group, a periodic keepalive packet is sent every 10 seconds (by default) from the WAVE to each router in the service group. The keepalive mechanism occurs independently for each configured service group. If a router in the service group has not received a keepalive packet from the WAVE in 2.5 times the keepalive interval, the router unicasts a *removal query (RQ)* message to that WAVE requesting that it immediately respond. If no response is received within 5 seconds, for a total of 30 seconds (by default) since the last keepalive message from the WAVE, the WAVE is considered offline and is removed from the service group.

Flow Protection

When a WAVE (re)joins the service group, a new *redirect assignment (RA)* message is generated by the lead WAVE. The RA message instructs routers how to reallocate load redistribution between other WAVE devices. When the new WAVE begins receiving redirected traffic from the routers in the service group, it does one of two things, depending on whether or not the redirected traffic is associated with a new TCP connection or part of an existing connection.

Traffic associated with newly established connections is evaluated against the ATP and processed normally by the WAVE. Traffic associated with existing connections is forwarded directly to the WAE that previously owned the bucket for that connection. This WCCPv2 mechanism is called flow protection and is enabled by default. Flow protection allows existing connections to continue to be optimized even when the traffic assignments for the WAEs in a service group change.

Scalability

With WCCPv2, each service group can support up to 32 routers and 32 WAEs. This means that a single service group can support $N \times 32$ concurrent optimized TCP connections, where N is the number of concurrent optimized TCP connections supported by the largest WAE model. Each WAE in the service group is manually configured with the IP address of each router in the service group. The WAE then uses unicast packets to exchange WCCPv2 messages with each router. It is not required that the routers in the service be manually configured with the IP address of each WAE in the service group. Each router listens passively for WCCPv2 messages from the WAEs in the service group and responds only as a result of receiving those messages.

The WAVE in the service group with the lowest IP address is elected as the “lead” WAVE. The lead WAVE is responsible for communicating the list, or view, of the routers in the service group to the service group routers. The lead WAVE is also responsible for informing the routers how traffic should be distributed across WAVEs in the service group through the use of RA messages. Upon receiving the view of the routers in the service group from the lead WAVE, each router responds individually with a router view. The router view contains a list of each WAVE with which the router is currently communicating. What is implied is that the routers in the service group do not communicate directly with each other; they learn about each other through the router view advertised by the WAVE. Likewise, the WAEs in a service group do not communicate directly; they learn about each other from the WAVE view advertised by the routers.

Redirect Lists

WCCPv2 redirect lists are used for deployments that may want to limit redirection to specific types of traffic. WCCP redirect lists are also useful for restricting transparent interception during proof of concept or pilot testing to a limited set of hosts and/or applications.

A WCCPv2 redirect list is a standard or extended IOS access list that is associated with a WCCPv2 service. Traffic passing through an interface on the router with WCCPv2 redirection configured must match not only the protocol/port specified as part of the service group, but also a permit entry in the redirect list. Packets that match the service group protocol/port criteria but do not match a permit entry in the redirect list are forwarded normally.

Service Group Placement

The placement of service groups 61 and 62 should not be overlooked in a WAAS deployment. The placement refers to which IOS interfaces are configured with service group 61 and which interfaces are configured with service group 62.

The direction in which interception occurs on the interfaces is important. Interception is configured in either the inbound or outbound direction. Inbound redirection evaluates traffic against the service group criteria as it enters the interface of a router, and outbound redirection evaluates traffic after it has already been switched through the router and is exiting the egress (based on routing table lookup) interface. In most deployments, service group 61 should be configured on the client-facing interfaces. The client-facing interfaces may differ depending on whether you are configuring WCCP in a remote branch office or in the data center.

For example, when deploying WCCPv2 on a remote office WAN router, service group 61 is configured to intercept a client request. Configuring group 61 inbound on the router’s LAN interface or outbound on the router’s WAN interface accomplishes this. By using service group 61 to intercept traffic in the client-to-server direction, WCCP performs load balancing in the service group based on the client IP address.

For the reverse direction of the connection, service group 62 is used. Service group 62 is configured in the opposite direction of service group 61. Because traffic is flowing in the reverse direction (server to client), the load balancing also occurs on the client IP address.

Egress Methods

Cisco WAAS provides several options for handling egress traffic received on intercepted connections. These options allow for flexibility when determining where to integrate WAAS into the existing network infrastructure and help preserve the original path selection for traffic flows.

The first egress method available in Cisco WAAS is IP forwarding. Egress traffic received on intercepted connections is forwarded based on the configuration of the local WAVE routing table, which typically means that traffic is forwarded to the configured default gateway. In addition to supporting a single default gateway, WAAS supports up to 1024 static routes. Static routes are configured with a next-hop IP address of a directly connected interface; recursive next-hop IP addresses are not supported. Although it is possible to configure multiple static routes for the same destination, there is no support for ECMP. Only a single route is installed in the routing table at a time.

Note Traffic originating from the WAVE itself also uses IP forwarding, regardless of the egress method configuration. The IP forwarding egress method is suited for basic topologies where only a single egress path for traffic exists.

The second egress method option available is called WCCP *generic routing encapsulation (GRE)*. This technique makes it possible to support redundant routers and router load balancing. The GRE tunnels are created automatically to process outgoing GRE-encapsulated traffic for WCCP. They appear when a WAVE requests GRE redirection. The GRE tunnel is not created directly by WCCP but indirectly via a tunnel API. Packet redirection is handled entirely by the redirecting device in software. WCCP has no direct knowledge of these tunnel interfaces but knows enough to cause packets to be redirected to them. This results in the appropriate encapsulation being applied, after which the packet is then sent to the WAVE device. WAAS makes a best effort to return frames back to the router from which they arrived.

The third option available is called *generic GRE*. Generic GRE functions in a similar manner to WCCP GRE return but leverages a traditional GRE tunnel interface in IOS to receive egress traffic from one or more WAVEs in the service group. The generic GRE egress method is designed specifically to be used in deployments where the router or switch has hardware-accelerated processing of GRE packets. To use the generic GRE egress method, a GRE tunnel interface must be created on each router.

With generic GRE return, a tunnel interface is configured in IOS on all WCCP-enabled routers in the service group. The tunnel interface is configured as either a point-to-point or a point-to-multipoint interface.

Policy-Based Routing (PBR)

Policy-based routing provides another alternative for transparent interception with WAAS, although it is less commonly deployed than WCCPv2 and inline interception. PBR can be used in situations where customers are unable to run WCCPv2 or inline interception.

PBR functions in a similar manner to WCCPv2, in that a router/switch running Cisco IOS is configured to intercept interesting traffic and redirect it to a WAE. Unlike WCCPv2, no configuration is required on the WAE to support interception using PBR via a routing policy. The following steps are used to configure PBR:

1. Create an access list to define interesting traffic for redirection.
2. Create a route map that matches the ACL created in Step 1 and sets an IP next-hop address of the target WAE.
3. Apply the route map to interfaces through which client and server traffic traverses.

Example 12-1 provides the basic PBR configuration used to redirect all TCP traffic to a single WAVE.

Example 12-1 *Policy-Based Routing Configuration*

```
access-list 175 permit tcp any any
!
route-map WAAS-Redirect 10
 match ip address 175
 set ip next-hop 10.10.20.2
!
interface Tunnel 100
 description ** WAN Interface **
 ip add 192.168.40.1 255.255.255.252
 ip policy route-map WAAS-Redirect

!
interface GigabitEthernet0/0
 no ip address
 duplex auto
 speed auto
!
interface GigabitEthernet0/0.1
 description ** Branch Client VLAN **
 encapsulation dot1q 10
 ip address 10.10.10.1 255.255.255.0
 ip policy route-map WAAS-Redirect
!
interface GigabitEthernet0/0.20
 description ** Branch WAVE VLAN **
 ip address 10.10.20.1 255.255.255.0
```

Because PBR evaluates only traffic entering an interface, the route map entries are configured on both the ingress and egress interfaces. This is the equivalent of using only inbound redirection with WCCPv2. The **set ip next-hop** command in the route map is configured with the IP address of the WAVE. By default, PBR does not validate the availability of the IP address specified as the next-hop address. As long as the next-hop address exists in the routing table, the route map entry is applied. On some platforms and software versions, Cisco *Service Assurance Agent (SAA)* can be used to track the availability of the next-hop IP address.

If the next-hop address becomes unreachable, traffic matching the route map entry is forwarded normally using the routing table. Another difference between WCCPv2 and PBR is that PBR does not perform automatic load distribution and failover when multiple WAEs exist. The first next-hop IP address configured in the route map is used until it becomes unavailable. Only at that point is traffic redirected to a secondary next-hop IP address in the route map.

Inline Interception

An alternative to the various off-path interception mechanisms is to place the WAE physically inline between two network elements, such as a WAN access router and a LAN switch. Figure 12-8 shows a basic topology with the WAE deployed physically inline.

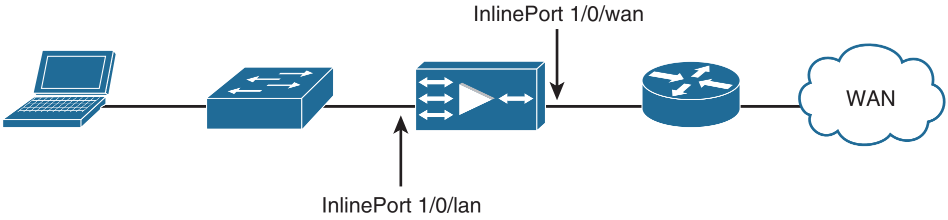


Figure 12-8 *WAE Physical In-Path Deployment*

Physical inline interception is an attractive option for situations where it is not possible or ideal to run WCCPv2. Common scenarios for inline interception are when access to the networking equipment at a site is provided and managed by a managed service provider (MSP). Because a logical interception technique cannot be configured on the device, physical inline interception can be used to provide the benefits.

To support physical inline interception, the WAE requires a separate inline module. The inline module is a two- or four-port, fail-to-wire network interface card (NIC) with each pair of ports in a unique inline group. Each inline group has a synchronous pair of inline ports that interconnect two network elements. Traffic entering one inline port is optimized by WAAS (when applicable) and switched out the opposite inline port in the same group. The inline group functions like a transparent Layer 2 bridge.

On platforms that support a four-port inline module, the WAE can support designs where multiple paths out of a site exist for redundancy and load sharing. Each unique path is connected to the WAE through a separate inline group. Figure 12-9 shows a sample remote site topology with multiple WAN routers and a single WAE deployed with inline interception.

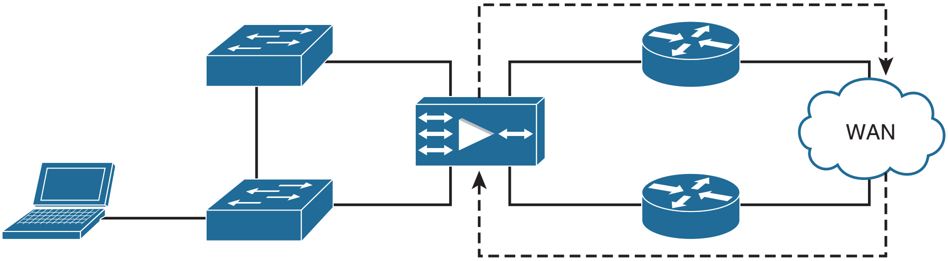


Figure 12-9 *Physical In-Path Deployment Using Multiple Routers*

As the arrows in Figure 12-9 indicate, traffic can enter or leave the site through either router. Even though the same flow enters the site through one inline group and exits the site through another inline group, the connection is still optimized. The optimized connection state is not tied to a physical interface but is tracked for the WAE as a whole independent of the interfaces traversed by the traffic.

Each inline group functions in one of two operating modes:

- **Interception:** Traffic entering the inline group is evaluated against the ATP for optimization.
- **Bypass:** All traffic entering the inline group is bridged without any optimization.

The bypass operating mode is designed to enable the WAVE to continue passing traffic if the WAVE loses power. A keepalive mechanism between the network drivers and the inline module determines if the WAVE is functioning properly and can optimize connections.

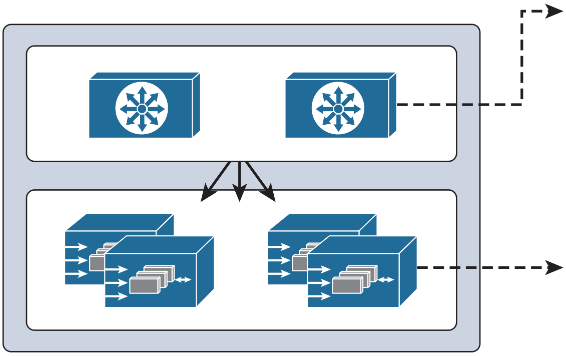
The keepalive frequency is configurable between 1 and 10 seconds. The default failover timer is set to 3 seconds. The transition between intercept operating mode and bypass operating mode does cause a momentary loss of line protocol. If one or more of the inline ports are connected to a LAN switch, this transition in interface state can cause the *Spanning Tree Protocol (STP)* recalculation. To prevent the STP calculation from interrupting traffic forwarding, the switchport connected to the inline module on the WAE should have the STP PortFast feature enabled.

AppNav Overview

The AppNav model is well suited to data center deployments and addresses many of the challenges of WAN optimization in this type of environment. Cisco AppNav technology enables customers to virtualize WAN optimization resources in the data center by pooling them into one elastic resource.

The AppNav solution has the ability to scale up to available capacity by taking into account WAAS device utilization as it distributes traffic among nodes. It integrates transparently with Cisco WAAS physical and virtual network infrastructure, supporting more than a million connections, providing significant investment protection for existing network design objectives as well as the capability to expand the WAN optimization service to meet future demands. Also, the solution provides for high availability of optimization capacity by monitoring node overload and liveliness and by providing configurable failure and overload policies. Because the Cisco AppNav solution enables organizations to pool elastic resources, it lays the foundation for migration to cloud services as well.

AppNav is a hardware and software solution that simplifies network integration of WAN optimization. WAAS version 5.0 introduces a new AppNav deployment model that greatly reduces dependency on the intercepting switch or router by taking on the responsibility of distributing traffic among WAAS devices for optimization. An AppNav device can be either a WAAS appliance with a Cisco AppNav Controller (ANC) Interface Module or a Cisco router with Cisco IOS XE Release 3.9 (or later) running AppNav-XE (known as an AppNav-XE device). Figure 12-10 illustrates the basic components of an AppNav Cluster.



AppNav Controllers (ANC)

- Hardware-or software-based network integration and interception component
  - AppNav IOM
  - IOS XE (ASR/CSR/ISR 4000)
- Provides service-aware flow distribution, to direct traffic to the WAAS nodes within the cluster

WAAS nodes (WNs)

- Current WAAS components responsible for traffic optimization and acceleration.
- Any current WAAS appliance version 5.0 and above can be a WN, including WAAS appliances and vWAAS.

Both roles can coexist on the same device: an ANC can also optimize.

Figure 12-10 *AppNav Cluster Components*

WAAS appliances with AppNav Controller Interface Modules operate in a special AppNav Controller mode with AppNav policies controlling traffic flow to WAAS devices doing optimization. Cisco AppNav technology is deployed on the new-generation Cisco WAVE appliances in the data center or private cloud. The 1 Gbps solution is offered with two types of IOMs in copper and Small Form-Factor Pluggable (SFP) form factors that can be plugged into Cisco WAVE-8541, 8571, 7541, and 694 devices. These solutions can run purely as Cisco AppNav solutions or as Cisco AppNav and Cisco WAAS on the same platform. The 10 Gbps package runs exclusively in Cisco AppNav mode and consists of the chassis (WAVE-594) as well as AppNav IOM. The AppNav 10 Gbps appliance provides high performance and high availability (dual power supply) in a single form factor. Both 1 Gbps and 10 Gbps solutions run Cisco WAAS software version 5.0 or later.

An AppNav-XE device, which is a Cisco router or virtual Cloud Services Router with virtual AppNav capability, can interoperate with other WAAS devices that are acting as WAAS nodes. An AppNav-XE device acts as an AppNav Controller that distributes traffic to other WAAS devices acting as WAAS nodes that optimize the traffic. AppNav-XE can run on router platforms that use IOS XE. These platforms are currently the ISR 4000, Cloud Services Router (CSR) 1000V, and Aggregation Services Router (ASR) 1000 Series routers, starting in IOS XE version 3.9.

Note ANCs on different platforms cannot be in the same AppNav Cluster.

AppNav Cluster Components

The following section describes the components of the AppNav Cluster:

- **AppNav Controller Group** (ANCG, or ACG on an XE-based router): A group of AppNav Controllers that together provide the necessary intelligence for handling asymmetric flows and providing high availability. The ANCG is configured on the ANC. An ANCG can have up to eight WAAS appliance-based ANCs. An ACG can have four AppNav-XE-based ANCs (which must be on the same router platform with the same memory configuration). For example, an ASR1002 and ASR1004 cannot be in the same ANC.
- **WAAS node (WN)**, or service node (SN) on the router: A WAAS optimization engine (WAE or WAVE appliance, network module, or vWAAS instance) that optimizes and accelerates traffic according to the optimization policies configured on the device. There can be up to 32 WNs in the cluster. (In the CLI and on the router, a WAAS node is also known as a service node.)
- **AppNav Cluster:** The group of all ANC and WN devices within a cluster.
- **AppNav context:** The topmost entity that groups together one ANCG, one or more WAAS node groups (WNGs), and an associated AppNav policy. The AppNav context is configured on the ANC. When using a WAAS appliance ANC, there is only one AppNav context, but when using an AppNav-XE ANC, you can define up to 32 AppNav contexts, which are associated with different VRF instances defined on the router.

Class Maps

AppNav class maps classify traffic according to one or more of the following match conditions:

- Peer device ID matches traffic from one peer WAAS device, which can be handling traffic from a single site or a group of sites.
- Source IP, and/or destination IP, and/or destination port matches traffic from a specific application.

- The class default class map (or APPNAV-class-default on AppNav-XE clusters) is a system-defined default class map that is defined to match any traffic. By default, it is placed in the last rule in each policy to handle any traffic that is not matched by other classes.

AppNav Policies

The AppNav policy is a flow distribution policy that allows you to control how ANCs distribute traffic to the available WNs. The AppNav policy consists of class maps that classify traffic according to one or more match conditions and a policy that contains rules that specify distribution actions to WNGs for each of the classes.

AppNav Site Versus Application Affinity

A WNG can be provisioned for serving specific peer locations (site affinity) or applications (application affinity) or a combination of the two. Using a WNG for site or application affinity provides the following advantages:

- Site affinity gives you the ability to always send all traffic from one site to a specific WNG, which allows you to reserve optimization capacity for critical sites and to improve compression performance through better utilization of the DRE cache. Traffic from any location, not just a single site, can be matched in a class map and associated with a WNG.
- Application affinity gives you the ability to always send certain application traffic to a specific WNG, which allows you to reserve optimization capacity for different applications depending on business priorities.

AppNav IOM

Cisco AppNav enables you to easily add WAN optimization capacity to the data center without having to make any changes to network configurations. AppNav allows for the expansion of WAN optimization capacity without any service disruption, regardless of whether it is deployed inline or off the path. It integrates transparently with Cisco WAAS physical and virtual network infrastructure. Because the Cisco AppNav solution enables organizations to pool elastic resources, it lays the foundation for migration to cloud services as well.

AppNav Controller Deployment Models

AppNav Controllers can be deployed in two ways:

- **In-path:** The ANC is physically placed between one or more network elements, enabling traffic to traverse a bridge group configured on the device in inline mode. A bridge group consists of two or more physical or logical (port channel) interfaces.

- **Off-path:** The ANC works with the network infrastructure to intercept traffic through the WCCP. The ANC operates in WCCP interception mode with one physical or logical (standby or port channel) interface configured with an IP address.

The ANC provides the same features in both in-path and off-path deployments. In either case, only ANCs participate in interception from the switch or router. The ANCs then distribute flows to WNs using a consistent and predictable algorithm that considers configured policies and WN utilization.

Cisco AppNav technology is a complementary technology to WCCP. WCCP provides both redirection and load distribution capabilities and is the technology most widely used to deploy WAN optimization today. The Cisco AppNav module allows customers to offload the load distribution function from the WCCP router, allowing the router to focus on traffic interception and redirection to a single WAN optimization pool.

AppNav Controller Interface Modules

A WAAS appliance operating as an ANC requires a Cisco AppNav Controller Interface Module, which is similar to a standard WAVE appliance interface module but contains additional hardware, including a network processor and high-speed ternary content addressable memory (TCAM), to provide intelligent and accelerated flow handling.

The following AppNav Controller Interface Modules (IOMs) are supported:

- 1 GB copper 12-port AppNav Controller Interface Module (APNV-GE-12T)
- 1 GB SFP 12-port AppNav Controller Interface Module (APNV-GE12SFP)
- 10 GB SFP+ four-port AppNav Controller Interface Module (APNV-10GSFP)

Table 12-9 provides additional details about the AppNav IOMs that are available.

Table 12-9 *Characteristics of an AppNav IOM*

	1 GB Factor Modules		**10 GB Appliance**
AppNav IOM	APNV-GE-12T	APNV-GE12SFP	APNV-10GSFP
Maximum Device Throughput	12,288 Mbps	12,288 Mbps	12,288 Mbps
Optimized TCP Connections	1,000,000	1,000,000	1,000,000
Pass-Through TCP Connections	1,000,000	1,000,000	1,000,000
Maximum AppNav Controllers (AppNav IOMs)	8	8	8
Maximum WAAS Accelerator Nodes	32	32	32

	1 GB Factor Modules		10 GB Appliance
Supported Hardware Platforms	WAVE-694	WAVE-694	Bundled with 594 appliance only
	WAVE-7541	WAVE-7541	
	WAVE-7571	WAVE-7571	
	WAVE-8541	WAVE-8541	

AppNav IOM Interfaces

Interfaces on the AppNav Controller Interface Module can have three functions:

- **Interception:** Used to receive traffic intercepted from the network and egress traffic to the network. The interception interface is implied based on the AppNav Controller placement and does not require explicit configuration for this function.
- **Distribution:** Used to distribute traffic to the WNs and receive egressed traffic from the WNs. The distribution interface is explicitly configured as the cluster interface for intra-cluster traffic and must be assigned an IP address.
- **Management:** A management interface can be optionally and exclusively designated for management traffic and isolated from the normal data path. Cisco recommends using one of the appliance's built-in interfaces for management traffic and reserving the high-performance interfaces on the AppNav Controller Interface Module for interception and distribution.

Guidelines and Limitations

You should use separate interfaces for interception and distribution for best performance, but you can use the same interface for both functions. AppNav Controller Interface Modules support port channel and standby logical interfaces. A port channel allows you to increase the bandwidth of a link by combining multiple physical interfaces into a single logical interface. A standby interface allows you to designate a backup interface in case of a failure.

Interfaces on the AppNav Controller Interface Module support the following:

- A maximum of seven port channels with up to eight physical interfaces combined into a single port channel group
- A maximum of five bridge groups configured over the physical or logical interfaces

Interfaces on the AppNav Controller Interface Module do not support the following:

- Fail-to-wire capability
- Bridge virtual interfaces (BVIs)

AppNav-XE

The AppNav-XE component is made up of a distribution unit called the AppNav Controller and service nodes. The AppNav Controller distributes flows and the service nodes process the flows. Additionally, up to four AppNav Controllers can be grouped together to form an AppNav Controller Group to support asymmetric flows and high availability.

Note All the routers in the AppNav Controller Group need to be on the same platform and have the same memory capacity.

Advantages of Using the AppNav-XE Component

The advantages of using the AppNav-XE component are:

- AppNav-XE can intelligently redirect new flows based on the load on each service node. This includes loads of individual L7 application accelerators.
- For flows that do not require any optimization, service nodes can inform the AppNav Controller to directly pass through the packets, thereby minimizing the latency and resource utilization.
- There is minimal impact to traffic when adding or removing service nodes.
- The AppNav-XE component supports VRF so that VRF information is preserved when traffic returns from a service node.
- You can use an AppNav Controller Group to optimize asymmetric flows.
  An asymmetric flow is when the traffic in one direction goes through one AppNav Controller and the return traffic goes through a different AppNav Controller, but both AppNav Controllers redirect the traffic to the same service node.
- AppNav-XE provides inter-router high availability, where if one router goes down, the traffic can be rerouted to a different router within the AppNav Controller Group, keeping the traffic flows uninterrupted.
- AppNav-XE provides intra-router high availability of the AppNav Controller on Cisco ASR 1000 Series platforms that have dual route processors (RPs) or dual forwarding processors (FPs). This means that if the active RP fails, the standby RP takes over, or if the active FP fails, the standby FP takes over and the flows continue uninterrupted. The intra-router high-availability feature is available only on the Cisco ASR 1000 Series platforms.

Note The AppNav-XE component can interoperate with the following router-based features: QoS, NAT, AVC 2.0, IPsec, GET-VPN (ASR 1000 Series only), EZVPN, DMVPN, ACL, VRF, MPLS, WCCP, PBR, and PfR.

Guidelines and Limitations

- Identify the WAN interfaces for the router that is running the AppNav Controller. The AppNav Controller intercepts packets on both ingress and egress of the WAN interface. Only configure the AppNav Controller on WAN interfaces, including all WAN interfaces that will be load balancing.
- Use port channels between the AppNav Controller and the service nodes to increase AppNav Controller-to-service node bandwidth.
- On AppNav-XE devices, all ANCs in the cluster must have an identical AppNav configuration (class maps, policy maps, and VRFs).
- On AppNav-XE devices, do not use VRF to access the WNs from the ANCs.
- On AppNav-XE devices, do not use a port channel between the ANCs and the WNs, because traffic is transmitted over a GRE tunnel and all traffic is switched on one link.
- An AppNav-XE device cannot intercept Overlay Transport Virtualization (OTV) traffic that is configured on the interception interface.
- Deploy only one service context if you are using WAAS appliances. If only AppNav-XE devices are deployed, the design can use up to 32 service contexts.
- The following maximum policy entity constraints exist for an AppNav-XE Cluster:
  - 32 match conditions per class map
  - 16,384 AppNav class maps
  - 1000 rules per AppNav policy
  - 1024 AppNav policies
- Mixing ANCs on different platforms is not allowed in an AppNav Cluster.

Note The AppNav-XE component introduces the concept of a virtual interface, which allows users to configure features specific to compressed or uncompressed traffic. For instance, to monitor the traffic that is being redirected to the service node and the traffic that is returning from the service node, you can configure the FNF feature on the *AppNav-Uncompress* and *AppNav-Compress* virtual interfaces. Note that these AppNav-XE virtual interfaces appear to the user just like any other interface.

WAAS Interception Network Integration Best Practices

The following network integration best practices are recommended for the majority of WAAS deployments:

- Leave the physical WAE interfaces set to auto-sense. Because it is possible that some of your WAEs are able to run at 1 Gbps speed, leaving all the WAEs set to auto-sense simplifies the configuration and deployment. In addition, an alarm is raised in the Central Manager if an interface negotiates to half-duplex.
- Use port channel for interface redundancy when both physical WAE interfaces connect to the same LAN switch. Improve performance by providing twice the available LAN bandwidth.
- Use a standby interface for interface redundancy when both physical WAE interfaces connect to different LAN switches. Increase WAE availability in the event of a problem with the primary interface or connected LAN switch.

Summary

The Cisco Wide-Area Application Engine family includes Virtual WAAS, ISR-WAAS, and appliance models. This breadth of portfolio provides customers with the flexibility necessary to allocate the right platform for each network location where WAN optimization, application acceleration, and virtualization capabilities are needed. Two licenses are available for Cisco WAAS. The Enterprise License includes all application accelerators except Akamai Connect. Akamai Connect is an advanced license that can be added to Cisco WAAS. Sizing of a Cisco WAAS solution requires consideration of a number of factors, including network conditions (WAN bandwidth and LAN throughput), number of users and concurrent optimized TCP connections, disk capacity and compression history, memory, concurrently connected peers, and virtualization requirements. By following the recommended guidelines for performance and scalability, a robust Cisco WAAS design can be realized, thereby allowing administrators to deploy the solution confidently to improve application performance over the WAN while enabling centralization and consolidation of infrastructure.

This chapter provided a detailed examination of the various methods of integrating WAAS into the network infrastructure. Various techniques for physical connectivity, including options for increased interface bandwidth and high availability, were reviewed. The chapter also previewed the network interception techniques that are used to transparently redirect traffic to the WAAS infrastructure for optimization. Particular focus was given to WCCPv2 and inline interception, which are the two most common interception methods. The interception method you choose is a site-specific decision. For example, you can use WCCPv2 at some locations and inline at other locations. Finally, the chapter discussed AppNav and AppNav-XE traffic distribution, which provides control over how intercepted traffic is distributed to a WAVE or group of WAVEs. You should now have a good feel for the flexibility of the WAAS solution when it comes to network integration. The techniques available enable Cisco WAAS to integrate into network infrastructures of any size and complexity.

Further Reading

Seils, Zach, Joel Christner, and Nancy Jin. *Deploying Cisco Wide Area Application Services, Second Edition.* Indianapolis: Cisco Press: 2010.

This page intentionally left blank

Chapter 13

Deploying Application Optimizations

This chapter covers the following topics:

- Deployment of the Central Manager
- Deployment of WAAS devices
- Deployment of AppNav-XE
- Customer case study

This chapter examines how to configure basic *Wide Area Application Services (WAAS)* and AppNav-XE features. In the default configuration, all the WAN optimization facilities are automatically enabled, minimizing the complexity associated with initially deploying systems.

A successful WAAS deployment begins with good design and planning. The content of this chapter will equip an administrator with the knowledge to design, integrate, and deploy WAAS in an enterprise network. The previous chapters presented an introduction to the WAN optimization capabilities provided by Cisco WAAS; this chapter explains the configuration of these features.

Because all applications and networks are not created equal, even with a well-planned and carefully designed WAAS network, there can be occasional and unexpected performance challenges. Although the symptoms of the performance challenges may be similar (such as slow application response time), the causes can come from different sources. Having a solid understanding of the application data flow is important to troubleshooting the source of performance degradation.

The capabilities of Cisco WAAS are commonly associated with the terms "WAN optimization" and "application acceleration." On the surface, these two terms seem similar, but in fact they are fundamentally different. WAN optimization refers to a set of capabilities that operate in an application-agnostic manner, at either the transport or the network layer, making the transfer of information over the WAN more efficient.

In the case of Cisco WAAS, WAN optimization is implemented in the transport layer. Application acceleration, on the other hand, refers to a set of capabilities that operate in an application-specific manner and interact at the application protocol layer to improve performance.

The previous chapters described how application acceleration and WAN optimization can combine with functions within the network to help IT organizations achieve a number of goals, including the following:

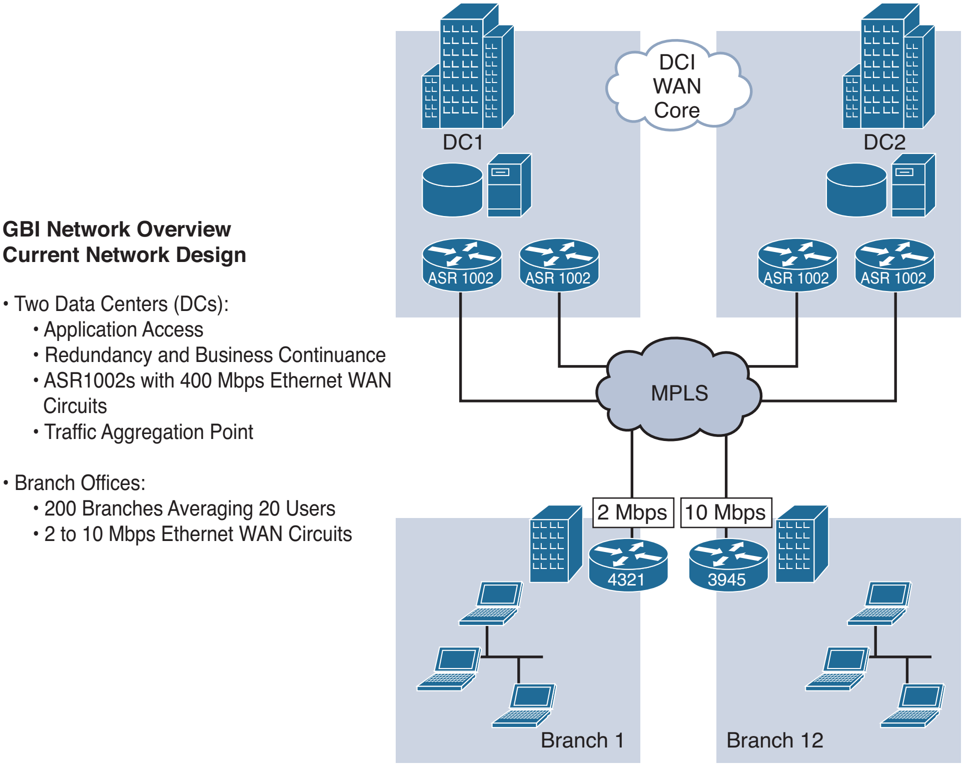
- Improve application performance over the WAN
- Enable more efficient utilization of existing network capacity

This chapter uses a customer case study of a fictional company called GBI to provide a scenario where an accelerator technology is employed to achieve these goals. The case study discusses the challenges the customer faces, the accelerator solution that addresses the challenges, the integration and deployment of the solution, and the effect the solution had on the environment.

GBI: Saving WAN Bandwidth and Replicating Data

GBI, which manufactures widgets, has two primary virtual data centers that are geographically dispersed, and the company needs to replicate data between the two locations while also minimizing bandwidth consumption and improving performance for users who access applications and data in the data centers. Application acceleration and WAN optimization are introduced into GBI's network to help address these challenges.

GBI has 4000 employees. All business-critical systems are deployed in two data centers, and these data centers are approximately 250 miles from each other. In most cases, users access the majority of their applications in the data center closer to them, but some applications require users to access a distant data center. Data in each of the data centers is replicated between data centers to ensure that it is available in case of disaster. The replication causes high levels of utilization on the WAN connection between the data centers and slow response time for users accessing applications hosted at the data centers. Figure 13-1 illustrates GBI's network as it is today.



GBI Network Overview
Current Network Design

- Two Data Centers (DCs):
  - Application Access
  - Redundancy and Business Continuance
  - ASR1002s with 400 Mbps Ethernet WAN Circuits
  - Traffic Aggregation Point
- Branch Offices:
  - 200 Branches Averaging 20 Users
  - 2 to 10 Mbps Ethernet WAN Circuits

Figure 13-1 *GBI Initial Network Diagram*

GBI faces challenges on two fronts:

- The amount of WAN bandwidth is no longer sufficient to keep up with the replication of data between the two data centers. GBI has found that its users' data is becoming increasingly rich and robust in size and complexity. Consequently, the time taken to replicate data continues to increase.
- Application access is affected for the many users attempting to access the data that is located across an already congested WAN.

WAN Optimization Solution

GBI decided to deploy an accelerator solution to minimize the amount of bandwidth consumed and time required to replicate data, and to provide faster access to distant data for its staff when working on data across the WAN. The compression capabilities of the accelerator will remove repeated patterns from network transmission, which will provide a tangible increase in the available WAN capacity. Compression will help to ensure that non-repeated patterns of data are shrunk to also alleviate bandwidth consumption.

Application acceleration capabilities will be employed to ensure that staff has fast access to data in both data centers. Given that GBI employs VoIP and QoS and its own security solution within its network, the company opted for a transparent accelerator solution, which allowed it to retain compatibility with these services.

For GBI's design, a set of common requirements exist that apply to every location or the solution as a whole. Those common requirements are:

- No significant changes should be made to the existing network topology or routing policy.
- No *Wide-Area Application Virtualization Engine (WAVE)* redundancy is required for the remote office deployments.

Deploying Cisco WAAS

Chapter 12, "Cisco Wide Area Application Services (WAAS)," covered design and sizing of WAAS. Remembering the flexibility of the portfolio is necessary to allocate the right platform for each network location where WAN optimization and application acceleration are deployed. Sizing of a Cisco WAAS solution requires consideration of a number of factors, including network conditions: WAN bandwidth, LAN throughput, and number of concurrent optimized TCP connections.

WAAS Data Center Deployment

This section examines the key design considerations for deploying WAAS in a data center environment, including considerations for environments with multiple data centers. Sample design models and configurations are provided throughout this section.

GBI Data Centers

GBI's existing WAN is built on an MPLS-based transport service from two national service providers. Both data centers included in this design support 200 remote offices. Figure 13-2 shows a high-level overview of the existing WAN topology.

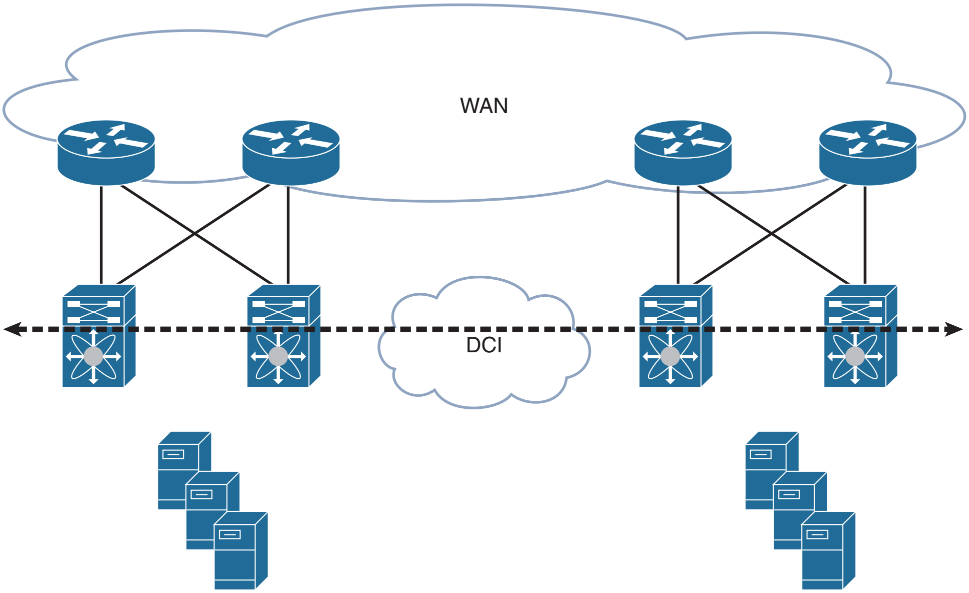


Figure 13-2 *Data Center High-Level Network Design*

Note This design assumes that the data centers are well connected with a *data center interconnect (DCI)*. The connectivity between data centers is high bandwidth, greater than 100 Mbps, and low latency, less than 10 msec. In essence, the AppNav-XE controllers and WAVEs in both data centers act as a single logical cluster.

The following key aspects of the WAN topology need to be considered for the design:

- Remote offices should be grouped based on common characteristics, such as business function, topology, networking equipment used, user community characteristics, and so on.
- The dual MPLS-based WAN service provides any-to-any connectivity. The location of client/server resources and the traffic flow effect on connection resources and fan-out should be taken into consideration for WAVE sizing.
- Some remote sites and the data center have multiple connections to the WAN. The routing policy and path preference need to be understood and accounted for in the design.

Data Center Device Selection and Placement

Starting from the top of the topology shown in Figure 13-2, traffic enters the data center through two Cisco Cloud Services Routers, per data center, at the WAN edge. All the WAN edge routers are aggregated to a pair of WAN distribution switches that connect to the core of the data center network infrastructure. Each data center has two 400 Mbps MPLS circuits that provide connectivity to 200 remote sites.

Based on the existing design of the data centers, GBI's network engineers decide to deploy AppNav-XE on the existing Cloud Services Router in a single AppNav Cluster across data centers with two node groups for application acceleration. The first logical location within the data center to consider deploying WAAS is at the WAN edge, or the point where traffic from remote branch offices first enters the data center from the WAN. The benefits of deploying WAAS at this location include the following:

- The WAN edge is a natural aggregation point for traffic destined for or sourced from a remote branch office.
- The configuration required to support this deployment model is kept simple, because transparent interception needs to be configured and maintained only in a single location.

The node groups' minimum requirements are to optimize 40,000 concurrent TCP connections, a fan-out of 200 peers, and WAN bandwidth of 800 Mbps. The two node groups consist of two WAVE-7571 appliances for each node group. One node group handles remote branch user traffic optimization, and a second node group handles data replication traffic between data centers. Figure 13-3 shows the data center topology with AppNav-XE and WAVEs deployed at the WAN edge.

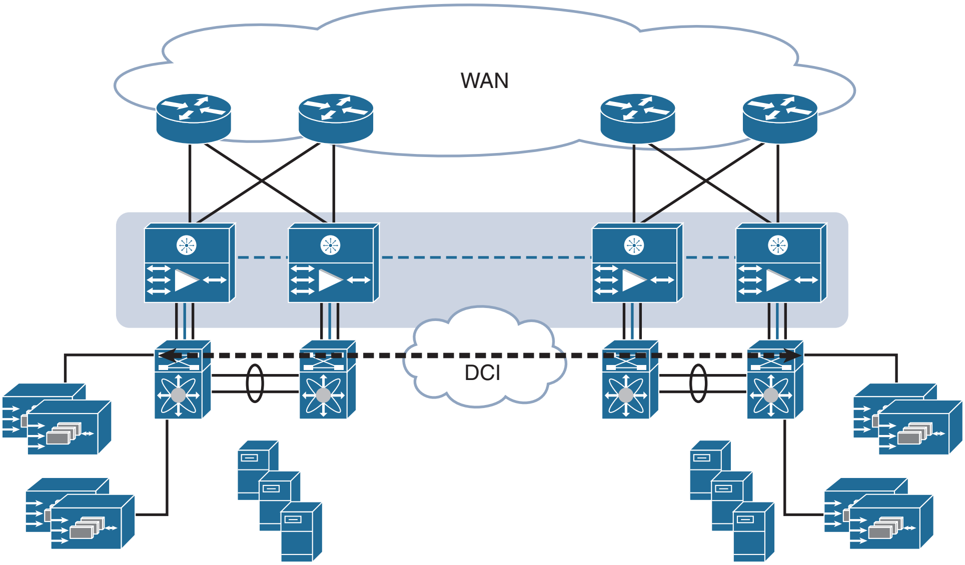


Figure 13-3 *WAN Edge with AppNav-XE and WAVE Placement*

Primary Central Manager

Every WAAS network must have one primary Cisco WAAS Central Manager that is responsible for managing the other WAAS devices in the network. The WAAS Central Manager hosts the WAAS Central Manager GUI, a web-based interface that facilitates the configuring, managing, and monitoring of the WAAS devices in the network. The WAAS Central Manager resides on a dedicated WAVE appliance or as a vWAAS instance (a WAAS running as a virtual machine).

Note The Cisco WAAS Central Manager is *not* critical for system operations, so if the Cisco WAAS Central Manager is unavailable or not accessible, there is no degradation of WAN optimization performance. The administrator only loses the ability to monitor or administer WAVE devices.

It is important to use the appropriate Cisco WAVE device (or Cisco vWAAS) from Table 13-1 for the Cisco WAAS Central Manager function at the primary location in order to provide graphical management, configuration, and reporting for the Cisco WAAS network. In order to initially configure the WAAS Central Manager, terminal access to the console port for basic configuration options and IP address assignment is required. For all Cisco WAVE devices, the factory default username is *admin* and the factory default password is *default*.

GBI's administrator chooses a WAVE-594 for the central management platform based on Table 13-1.

Table 13-1 *Central Manager Scalability*

Appliance Model (as a Central Manager)	Number of Managed Devices
WAVE-294	250
WAVE-294-8GB	250
WAVE-594	1000
WAVE-594-12GB	1000
WAVE-694	2000
WAVE-694-24GB	2000

Initial Primary Central Manager Configuration

The WAAS Central Manager device must be initially configured with network-based settings before the remaining configuration can be done via a GUI. A console connection is required during the initial configuration of the WAVE appliances. After the network settings have been defined, a Telnet session for subsequent CLI sessions can be used.

The following steps initialize the WAAS Central Manager device:

Step 1. **Choose the device mode.**

Choose the device mode with the global configuration command **device mode** *central-manager*.

Step 2. **Define the device's host name.**

Define the device's host name with the global configuration command **hostname** *hostname*.

Step 3. **Configure an interface IP address.**

An IP address must be assigned to an interface. The command **interface** *interface-id* identifies the interface that will be used for initial network connectivity. The IP address is then associated to the interface with the command **ip address** *ip-address subnet-mask*.

Step 4. **Configure the device's default gateway.**

Configure the default gateway with the global configuration command **ip default-gateway** *ip-address*.

Step 5. **Define the management (communication) interface.**

Configure the management (communication) interface with the global configuration command **primary-interface** *interface-id*.

Step 6. **Enable the management service.**

Enable the management service with the global configuration command **cms enable.**

Step 7. **Save the running configuration to the startup configuration.**

Save the running configuration to the startup configuration with the exec command **copy running-config startup-config.**

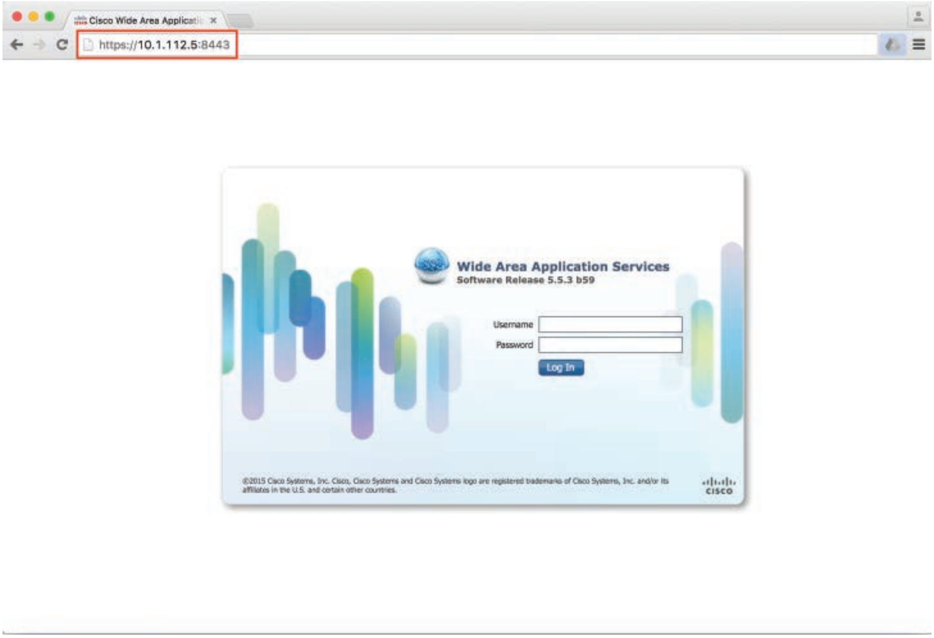
Example 13-1 demonstrates the sample configuration for the Central Manager for DC1.

Example 13-1 *Central Manager Configuration*

```
device mode central-manager
! Output omitted for brevity
!
hostname Central-Mgr-DC1
!
interface GigabitEthernet 0/0
 ip address 10.1.112.5 255.255.255.0
 exit
ip default-gateway 10.1.112.1
!
```

```
primary-interface GigabitEthernet 0/0
!
cms enable
!
copy running-config startup-config
```

After the network is initialized, the WAAS Central Manager's GUI can be accessed via a web browser at port 8443. The CM's URL resembles the format **https://***wave-ip-address***:8443** as shown in Figure 13-4. The GUI is used to continue the configuration of the Central Manager, beginning with the login.


https://10.1.112.5:8443

Cisco Wide Area Applicati
Wide Area Application Services
Software Release 5.5.3 b59
Username
Password
Log In
©2015 Cisco Systems, Inc. Cisco, Cisco Systems and Cisco Systems logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and certain other countries.
cisco

Figure 13-4 *Primary Central Manager Login Page*

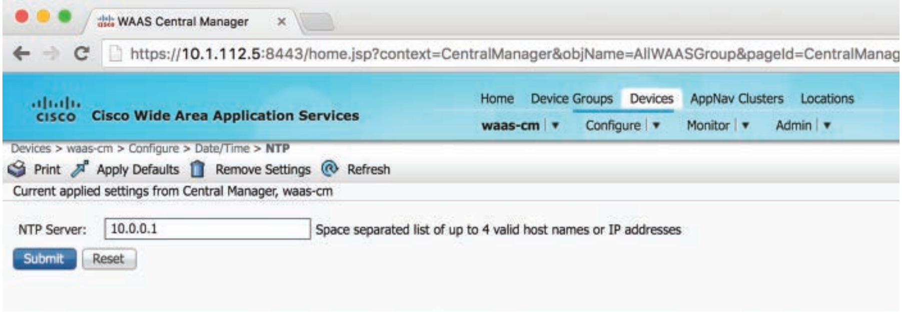
When prompted, enter the default administrator username (**admin**) and default password (**default**) and press **Enter**. NTP and DNS must be configured as the last two steps of the Central Manager configuration.

Note The Central Manager services are dependent on time and name resolution for reporting, accurate statistics, and Akamai Connect functions.

Configuring the Primary Central Manager’s NTP Settings

The WAAS Central Manager GUI allows you to configure the time and date settings using an NTP host on your network. NTP allows the synchronization of time and date settings for the devices in your WAAS network, which is important for proper system operation and monitoring. To configure NTP settings, follow these steps:

1. From the WAAS Central Manager menu, choose **Devices >** *device-name*.
2. Choose **Configure > NTP**. The NTP Settings window appears.
3. In the **NTP Server** field, enter up to four host names or IP addresses, separated by spaces. Figure 13-5 displays the NTP configuration screen.



https://10.1.112.5:8443/home.jsp?context=CentralManager&objName=AllWAASGroup&pageId=CentralManag

Devices > waas-cm > Configure > Date/Time > **NTP**

Current applied settings from Central Manager, waas-cm

NTP Server: 10.0.0.1 Space separated list of up to 4 valid host names or IP addresses

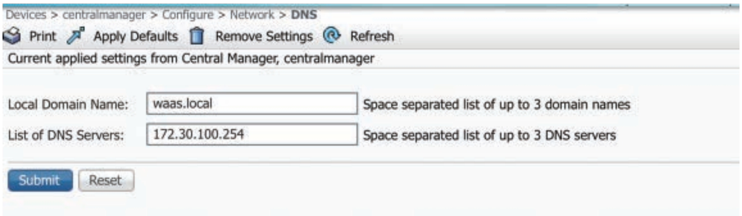
Figure 13-5 *Central Manager NTP Configuration*

4. Click **Submit**.

Configuring the Primary Central Manager’s DNS Settings

The primary Central Manager’s DNS settings are configured with the following steps:

1. From the WAAS Central Manager menu, choose **Devices >** *device-name*.
2. Choose **Configure > DNS**. The DNS Settings window appears.
3. In the **List of DNS Servers** field, enter up to three host names or IP addresses, separated by spaces.
4. In the **Local Domain Name** field, enter up to three host names separated by spaces, as shown in Figure 13-6. Click **Submit**.



Devices > centralmanager > Configure > Network > **DNS**

Current applied settings from Central Manager, centralmanager

Local Domain Name: waas.local Space separated list of up to 3 domain names

List of DNS Servers: 172.30.100.254 Space separated list of up to 3 DNS servers

Figure 13-6 *Central Manager DNS Configuration*

The command **show cms info** displays the device ID, function (such as primary), and CMS status as shown in Example 13-2.

Example 13-2 *Verifying the Primary Central Manager's Availability*

```
dc1-cm# show cms info
! Output omitted for brevity

Device registration information :
Device Id                              = 239
Device registered as                   = WAAS Central Manager
Current WAAS Central Manager role      = Primary

CMS services information :
Service cms_httpd is running
Service cms_cdm is running
```

Configuring WAAS Group Settings

A usability feature of the WAAS CM is the ability to logically group devices into configuration groups. These groups, called device groups, allow an administrator to apply a configuration change across multiple WAAS devices simultaneously by applying a change to the group.

The CM provides a default device group, called the AllWAASGroup, to which all Cisco WAAS devices are automatically joined after registration. Initial configuration can be done across the AllWAASGroup. Central Manager devices cannot and do not belong to a device group.

To make universal changes to multiple accelerators in a WAAS deployment, choose the **AllWAASGroup** as depicted in Figure 13-7.

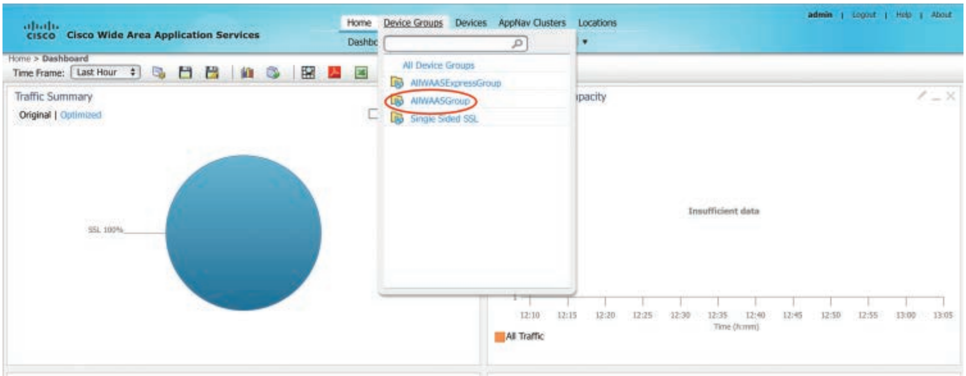
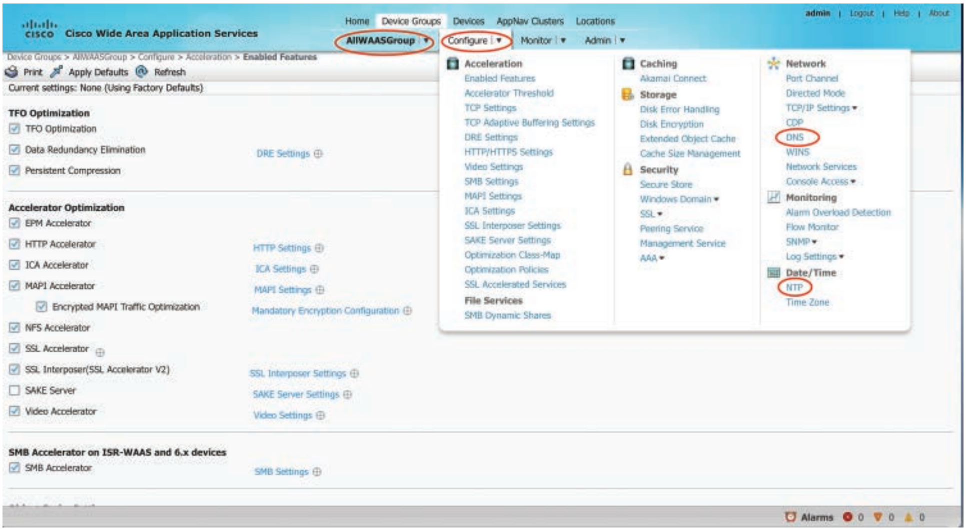


Figure 13-7 *Selecting the AllWAASGroup*

Device Group Basic Settings

The WAAS Central Manager GUI allows you to configure the time and date settings on every WAAS device using an NTP host on your network, which is important for proper system operation and monitoring. This setting is needed on every WAAS device to keep the clocks synchronized. In addition, DNS should be configured for host resolution via the AllWAASGroup, as seen in Figure 13-8.



Device Groups > AllWAASGroup > Configure > Acceleration > **Enabled Features**

Current settings: None (Using Factory Defaults)

TFO Optimization

☑ TFO Optimization

☑ Data Redundancy Elimination

☑ Persistent Compression

Accelerator Optimization

☑ EPM Accelerator

☑ HTTP Accelerator

☑ ICA Accelerator

☑ MAPI Accelerator

☑ Encrypted MAPI Traffic Optimization

☑ NFS Accelerator

☑ SSL Accelerator

☑ SSL Interposer(SSL Accelerator V2)

□ SAKE Server

☑ Video Accelerator

SMB Accelerator on ISR-WAAS and 6.x devices

☑ SMB Accelerator

Figure 13-8 *Device Group DNS and NTP Settings*

Standby Central Manager

For high availability, the Cisco WAAS Central Manager can also be deployed in an active/standby configuration. One Cisco WAAS device acts as the primary Cisco WAAS Central Manager, and the second device acts as a backup Cisco WAAS Central Manager.

In this mode, the Cisco WAAS device serving as the backup Central Manager automatically receives configuration changes and monitoring data from the primary Central Manager, just like any other WAAS device in the Cisco WAAS topology. If the primary Central Manager fails, the Cisco WAAS devices in the topology automatically redirect themselves to the secondary Central Manager. The managed devices receive the IP address of the standby Central Manager in their databases after registration. The registered devices automatically try to contact the standby CM if the primary is not reachable. When the primary Central Manager is back online, the standby rejects requests and the registered devices then query the originally configured primary Central Manager.

The standby WAVE device (or Cisco vWAAS) should be sized in the same way as the primary device. In essence, both boxes should use Table 13-1 for sizing to ensure that either appliance can handle the full load. The backup Cisco WAAS Central Manager should be placed at a different location for resiliency purposes.

GBI's administrator chooses a WAVE-594 as the standby central management platform to match the primary Central Manager platform.

Note The device configuration needs to be modified only in case the primary CM IP address is changed. The monitoring data collected on the primary Cisco WAAS Central Manager will be available on the secondary Cisco WAAS Central Manager device.

Standby Central Manager's Configuration

The configuration of the standby Central Manager is similar to the initial configuration of the primary Central Manager with the additional command to define the primary Central Manager. The configuration is done using a console connection to provide settings for network connectivity. After a console connection has been used to define the device network settings, a Telnet session for subsequent CLI sessions can be used. The following steps initialize the secondary WAAS Central Manager device:

Step 1. **Choose the device mode.**

Choose the device mode with the global configuration command **device mode** *central-manager*.

Step 2. **Configure the Central Manager role.**

Configure the Central Manager role as standby using the Central Manager command **central-manager** *role standby*.

Step 3. **Define the device's host name.**

Define the device's host name with the global configuration command **hostname** *hostname*.

Step 4. **Configure the interface IP address.**

An IP address must be assigned to an interface. The command **interface** *interface-id* identifies the interface that will be used for initial network connectivity. The IP address is then associated to the interface with the command **ip address** *ip-address subnet-mask*.

Step 5. **Configure the device's default gateway.**

Configure the default gateway with the global configuration command **ip default-gateway** *ip-address*.

Step 6. **Define the management (communication) interface.**

Configure the management (communication) interface with the global configuration command **primary-interface** *interface-id*.

Step 7. **Define the primary Central Manager's IP address.**

Configure the primary CM's address with the global configuration command **central-manager** *address cm-primary-address*.

Step 8. **Enable the management service.**

Enable the management service with the global configuration command **cms enable.**

Step 9. **Save the running configuration to the startup configuration.**

Save the running configuration to the startup configuration with the exec command **copy running-config startup-config.**

Example 13-3 demonstrates the sample configuration for the standby Central Manager that is located in DC2.

Example 13-3 *Configuration of the Standby Central Manager*

```
device mode central-manager
!
hostname Central-Mgr-DC2
!
interface GigabitEthernet 0/0
 ip address 10.1.113.6 255.255.255.0
 exit
ip default-gateway 10.1.113.1
!
primary-interface GigabitEthernet 0/0
!
central-manager role standby
central-manager address 198.19.0.3
!
cms enable
!
copy running-config startup-config
```

The preceding process allows the WAAS Central Manager to maintain a copy of the WAAS network configuration on a second WAAS Central Manager device. If the primary WAAS Central Manager fails, the standby can replace it.

Note For interoperability, when a standby WAAS Central Manager is used, it must be at the same software version as the primary WAAS Central Manager to maintain the full WAAS Central Manager configuration. Otherwise, the standby WAAS Central Manager detects this status and does not process any configuration updates that it receives from the primary WAAS Central Manager until the problem is corrected.

Example 13-4 verifies that the newly configured standby CM is currently acting as the standby CM using the command **show cms info**.

Example 13-4 *Verification of the Standby Central Manager*

```
dc2-cm# show cms info
! Output omitted for brevity

Device registration information :
Device Id                            = 2176
Device registered as                 = WAAS Central Manager
Current WAAS Central Manager role    = Standby
Current WAAS Central Manager         = 198.19.0.3

CMS services information :
Service cms_httpd is running
Service cms_cdm is running
```

AppNav-XE

Cisco WAAS was deployed with off-path traffic redirection with *Web Cache Communication Protocol (WCCP)* for many years. In many ways, *Application Navigation (AppNav)* may be considered as a next-generation WCCP. It was first implemented as a pluggable component in WAAS to provide an over-the-top *in-path insertion solution* for data center deployments. Subsequently, Cisco ported the functions of the distribution component to IOS XE, making AppNav a network-integrated insertion offering rather than just over the top. The AppNav-XE component of Cisco IOS XE Release 3.9/3.10 and beyond provides an easy way for Cisco IOS XE *Aggregation Services Router (ASR)*, *Cloud Services Router (CSR)*, and *Integrated Services Router (ISR)* to intercept and intelligently distribute flows to service nodes (that is, WAAS).

The AppNav-XE solution is made up of a distribution unit called the *AppNav Controller (ANC)* and *service nodes (SNs)*. The AppNav Controller distributes flows, and the service nodes process the flows. The AppNav intelligent flow distribution incorporates knowledge and policy to direct flows to groups of SNs based on where the flow came from (downstream branch or WAVE), where the flow is going (specific applications), the WAVE availability and load, and the WAVE *Application Optimizer (AO)* availability and load. Thus AppNav uniquely offers the ability to dynamically scale a heterogeneous farm of accelerators while incorporating physical and virtual devices.

Initial GBI AppNav-XE Deployment

In GBI's topology, both data centers use dedicated DCI links, and each data center WAN edge has two CSRs with AppNav-XE. Four WAVEs are connected to data center distribution switches. Routing is configured so that inter-data-center traffic (both data traffic and control plane cluster traffic) goes across the DCI link between these two data centers.

Note The DCI link is a Layer 3 connection point.

The WAAS Central Manager can manage AppNav-XE devices via the HTTPS protocol. To establish communication between a WAAS Central Manager and a Cisco IOS XE router device, register the Cisco IOS XE router device with the Central Manager. Using the Central Manager GUI to register a Cisco IOS router device is the easiest method.

Note SSH and the secure server must be enabled on the IOS XE device.

To register IOS XE devices to the WAAS Central Manager, follow these steps:

Step 1. **Open the registration window.**

From the WAAS Central Manager menu, choose **Admin > Registration > Cisco IOS Routers** as shown in Figure 13-9. The Cisco IOS Router Registration window appears.

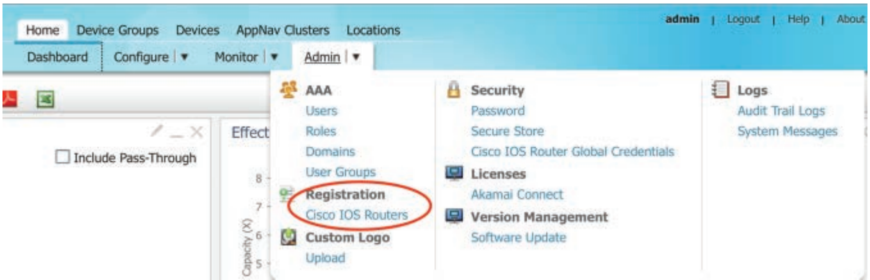
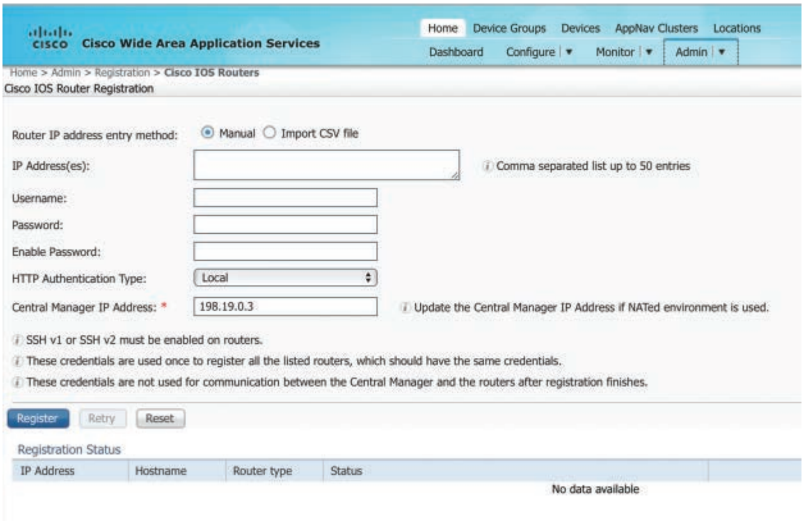


Figure 13-9 *Navigating to Cisco IOS Router Registration*

Step 2. **Register the router IP address.**

In the **IP Address(es)** field, enter the router IP address to register, separating addresses with commas if there is more than one device being registered. The IP address, host name, router type, and status are displayed in the Registration Status table.



Cisco IOS Router Registration

Router IP address entry method: ● Manual ○ Import CSV file

IP Address(es): Comma separated list up to 50 entries

Username:

Password:

Enable Password:

HTTP Authentication Type: Local

Central Manager IP Address: * 198.19.0.3 Update the Central Manager IP Address if NATed environment is used.

SSH v1 or SSH v2 must be enabled on routers.

These credentials are used once to register all the listed routers, which should have the same credentials.

These credentials are not used for communication between the Central Manager and the routers after registration finishes.

Registration Status

IP Address	Hostname	Router type	Status
			No data available

Figure 13-10 *Enter the Router Credentials and IP Address*

Step 3. **Configure the router login credentials.**

Fill in the **Username**, **Password**, and **Enable password** fields as shown in Figure 13-10.

Step 4. **Choose the HTTP authentication type.**

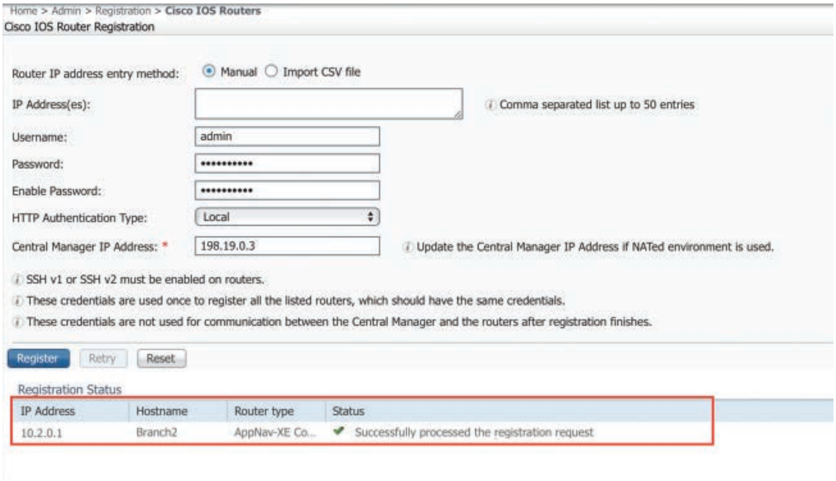
The choices are **Local** or **AAA**.

Step 5. **Enter the IP address the Central Manager will use.**

In the **Central Manager IP Address** field, enter the router's IP address that the Central Manager will use. This field is initially filled in with the current Central Manager IP address.

Step 6. **Complete the registration.**

Click the **Register** button and verify that the registration status was successful, as shown in Figure 13-11. Notice that the device's status shows *Successfully processed the registration request*.



Home > Admin > Registration > **Cisco IOS Routers**

Cisco IOS Router Registration

Router IP address entry method: ● Manual ○ Import CSV file

IP Address(es): ⓘ Comma separated list up to 50 entries

Username: admin

Password: ••••••••••

Enable Password: ••••••••••

HTTP Authentication Type: Local

Central Manager IP Address: * 198.19.0.3 ⓘ Update the Central Manager IP Address if NATed environment is used.

ⓘ SSH v1 or SSH v2 must be enabled on routers.

ⓘ These credentials are used once to register all the listed routers, which should have the same credentials.

ⓘ These credentials are not used for communication between the Central Manager and the routers after registration finishes.

Register Retry Reset

Registration Status

IP Address	Hostname	Router type	Status
10.2.0.1	Branch2	AppNav-XE Co...	✓ Successfully processed the registration request

Figure 13-11 *Successful Registration*

For the WAAS Central Manager to access a Cisco IOS XE router to monitor and administer AppNav-XE, the WAAS Central Manager must have global credentials configured for router access.

On the Central Manager, the global credentials are defined for all applicable Cisco IOS XE router devices, or you can define credentials at the device group or individual device level by using the **Admin > Security > Cisco IOS Router Global Credentials** menu item. The Cisco IOS router registration process must be completed before the global credentials for the router are configured.

To configure global router credentials, follow these steps:

Step 1. **Open the credentials window.**

From the WAAS Central Manager menu, choose **Admin > Security > Cisco IOS Router Global Credentials** as shown in Figure 13-12. The Cisco IOS Router Global Credentials window appears.

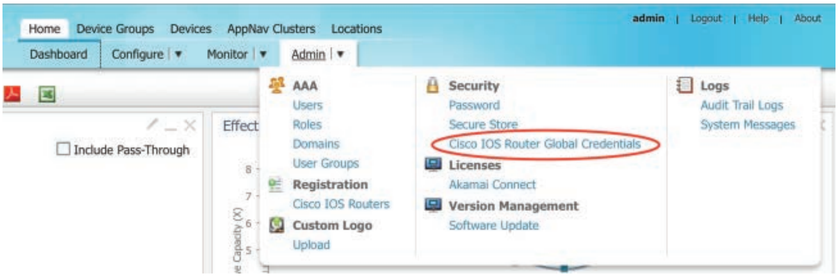


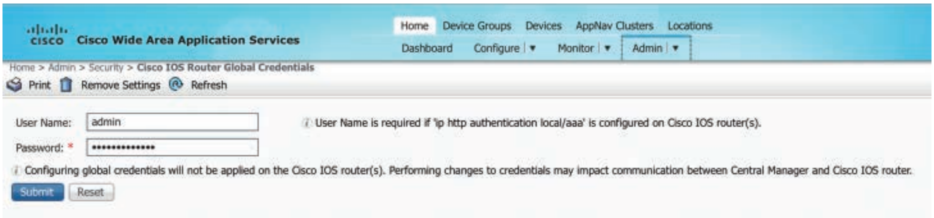
Figure 13-12 *Cisco IOS Router Global Credentials Navigation*

Step 2. **Enter the username.**

In the **User Name** field, enter a username that is defined on the router.

Step 3. **Add the password.**

In the **Password** field, enter the password for the specified username as shown in Figure 13-13.


cisco Cisco Wide Area Application Services
Home Device Groups Devices AppNav Clusters Locations
Dashboard Configure ▾ Monitor ▾ Admin ▾
Home > Admin > Security > Cisco IOS Router Global Credentials
Print Remove Settings Refresh

User Name: admin

Password: * •••••••••••••

User Name is required if 'ip http authentication local/aaa' is configured on Cisco IOS router(s).

Configuring global credentials will not be applied on the Cisco IOS router(s). Performing changes to credentials may impact communication between Central Manager and Cisco IOS router.

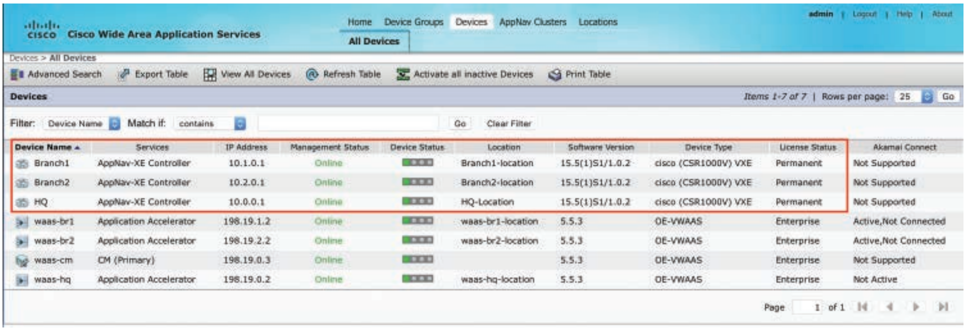
Figure 13-13 *Configuring Cisco IOS Router Global Credentials*

Step 4. **Complete the configuration.**

Click **Submit** to complete the configuration.

Note The user is required to have privilege level 15 access to the IOS XE device. The device will appear as offline on the **Devices > All Devices** page if the wrong privilege level or no privilege level is defined.

After successfully registering a Cisco IOS XE router device, the Central Manager displays it in the All Devices list. The device should have a device status of green, as shown in Figure 13-14.


cisco Cisco Wide Area Application Services
Home Device Groups Devices AppNav Clusters Locations
All Devices
admin | Logout | Help | About
Devices > All Devices
Advanced Search Export Table View All Devices Refresh Table Activate all inactive Devices Print Table
Devices
Items 1-7 of 7 | Rows per page: 25 Go
Filter: Device Name Match if: contains Go Clear Filter

Device Name ▲	Services	IP Address	Management Status	Device Status	Location	Software Version	Device Type	License Status	Akamai Connect
Branch1	AppNav-XE Controller	10.1.0.1	Online	🟩	Branch1-location	15.5(1)S1/1.0.2	cisco (CSR1000V) VXE	Permanent	Not Supported
Branch2	AppNav-XE Controller	10.2.0.1	Online	🟩	Branch2-location	15.5(1)S1/1.0.2	cisco (CSR1000V) VXE	Permanent	Not Supported
HQ	AppNav-XE Controller	10.0.0.1	Online	🟩	HQ-Location	15.5(1)S1/1.0.2	cisco (CSR1000V) VXE	Permanent	Not Supported
waas-br1	Application Accelerator	198.19.1.2	Online	🟩	waas-br1-location	5.5.3	OE-VWAAS	Enterprise	Active,Not Connected
waas-br2	Application Accelerator	198.19.2.2	Online	🟩	waas-br2-location	5.5.3	OE-VWAAS	Enterprise	Active,Not Connected
waas-cm	CM (Primary)	198.19.0.3	Online	🟩		5.5.3	OE-VWAAS	Enterprise	Not Supported
waas-hq	Application Accelerator	198.19.0.2	Online	🟩	waas-hq-location	5.5.3	OE-VWAAS	Enterprise	Not Active

Figure 13-14 *Verification of Successful IOS XE Registration*

Deploying a Data Center Cluster

To create a new AppNav Cluster using the AppNav Cluster Wizard, follow these steps:

Step 1. **Navigate to the All AppNav Clusters page.**

From the WAAS Central Manager menu, choose **AppNav Clusters > All AppNav Clusters** as shown in Figure 13-15.

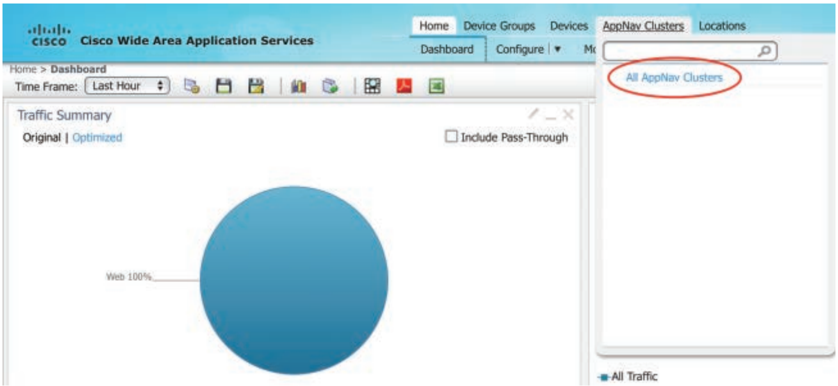


Figure 13-15 *Opening AppNav Clusters*

Step 2. **Start the AppNav Cluster Wizard.**

Click the **AppNav Cluster Wizard** as shown in Figure 13-16.

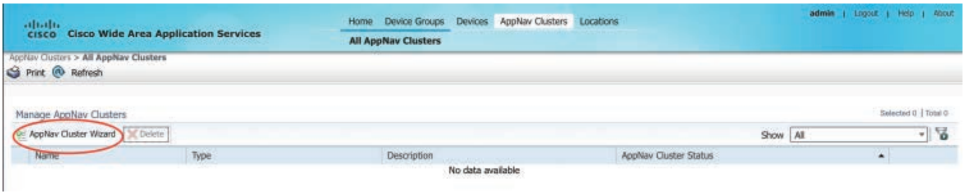
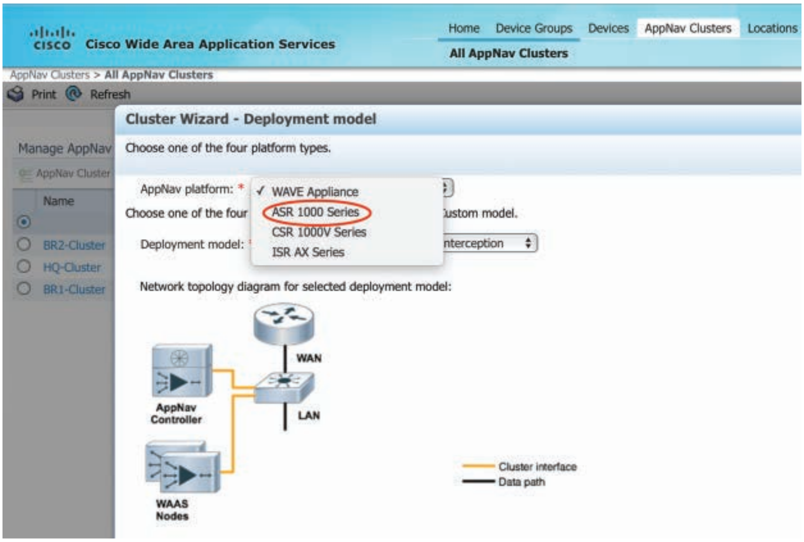


Figure 13-16 *Launching the AppNav Cluster Wizard*

Step 3. **Select the deployment model.**

In the Cluster Wizard screen, under the **AppNav Platform** drop-down list, choose the deployment model that matches your deployment. In this chapter's example, the **ASR 1000 Series** has been selected, as shown in Figure 13-17.



Cluster Wizard - Deployment model

Choose one of the four platform types.

AppNav platform: * ✓ WAVE Appliance / ASR 1000 Series / CSR 1000V Series / ISR AX Series

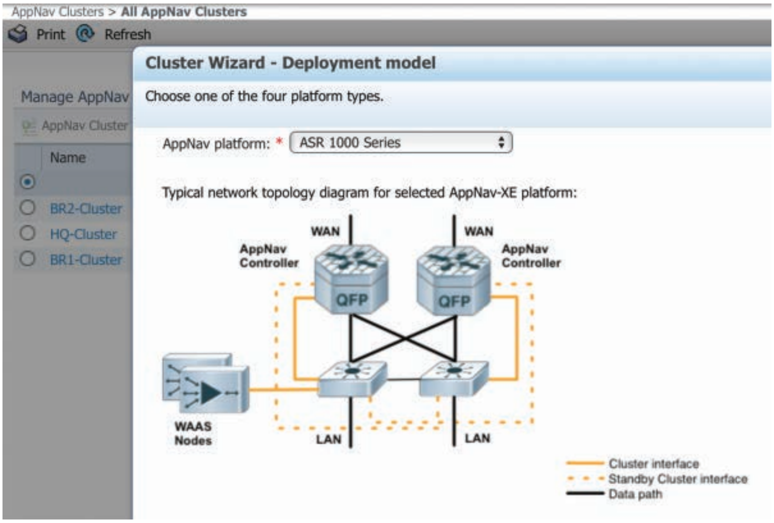
Choose one of the four ... ustom model.

Deployment model: * ... nterception

Network topology diagram for selected deployment model:

Figure 13-17 *AppNav Platform Selection*

Notice that the diagram changes for the typical network topology based on the platform selected. Figure 13-18 displays the topology for the ASR 1000 routers.



Cluster Wizard - Deployment model

Choose one of the four platform types.

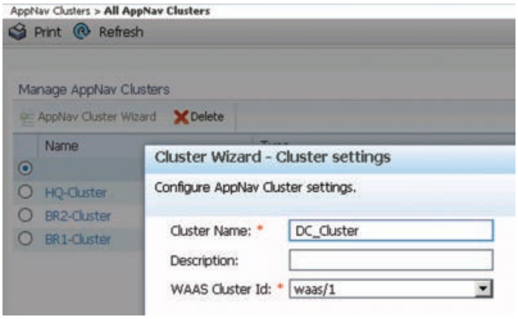
AppNav platform: * ASR 1000 Series

Typical network topology diagram for selected AppNav-XE platform:

Figure 13-18 *ASR 1000 Deployment Model Selected*

Step 4. **Define the cluster settings.**

Enter the cluster name and select the WAAS cluster ID. In the **Name** field, enter a name for the cluster. Use only letters, numbers, hyphen, and underscore, up to a maximum of 32 characters and beginning with a letter. This book's example uses the name **DC_Cluster** as shown in Figure 13-19.



AppNav Clusters > **All AppNav Clusters**

Manage AppNav Clusters

	Name
●	
○	HQ-Cluster
○	BR2-Cluster
○	BR1-Cluster

Cluster Wizard - Cluster settings

Configure AppNav Cluster settings.

Cluster Name: * DC_Cluster

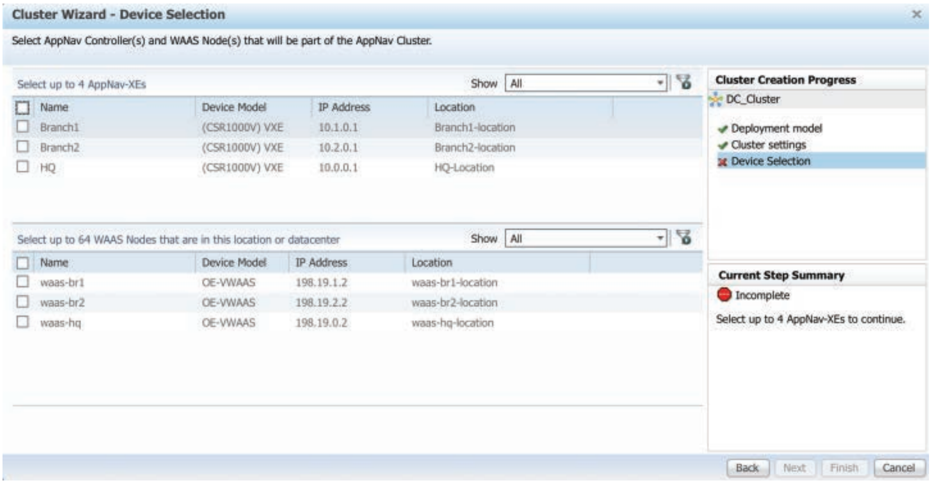
Description:

WAAS Cluster Id: * waas/1

Figure 13-19 *Define the Cluster Settings*

Step 5. **Choose the AppNav Controllers and nodes.**

Select the AppNav Controllers and WAAS nodes that will be part of the AppNav Cluster, as shown in Figure 13-20, and then click **Next.**



Cluster Wizard - Device Selection

Select AppNav Controller(s) and WAAS Node(s) that will be part of the AppNav Cluster.

Select up to 4 AppNav-XEs — Show: All

□ Name	Device Model	IP Address	Location
□ Branch1	(CSR1000V) VXE	10.1.0.1	Branch1-location
□ Branch2	(CSR1000V) VXE	10.2.0.1	Branch2-location
□ HQ	(CSR1000V) VXE	10.0.0.1	HQ-Location

Select up to 64 WAAS Nodes that are in this location or datacenter — Show: All

□ Name	Device Model	IP Address	Location
□ waas-br1	OE-VWAAS	198.19.1.2	waas-br1-location
□ waas-br2	OE-VWAAS	198.19.2.2	waas-br2-location
□ waas-hq	OE-VWAAS	198.19.0.2	waas-hq-location

Cluster Creation Progress

DC_Cluster

✓ Deployment model

✓ Cluster settings

✗ Device Selection

Current Step Summary

Incomplete

Select up to 4 AppNav-XEs to continue.

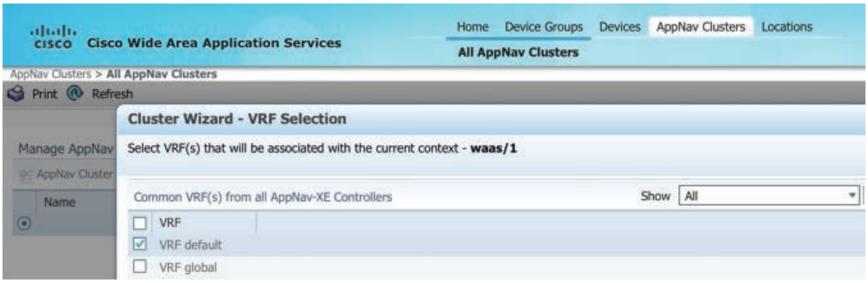
Figure 13-20 *AppNav Controllers and WAAS Nodes Selection*

Step 6. **Select the VRF.**

Select the VRF available as shown in Figure 13-21 and click **Next.** More than one VRF name may be entered, up to 64 VRF names. VRF global is the same as the other VRF definitions except that it identifies traffic with no VRF.

If you do not configure a VRF in the service context, the system automatically applies the default configuration of VRF Default. The purpose of VRF Default is to match traffic that does not match a configured VRF name or VRF global.

The following logic is used to pick the correct service context for a packet: The system compares the VRF on the LAN interface traversed by the packet against the VRF names (or VRF global) that are configured in the service contexts. If there is a match, the system picks the corresponding service context. If there is no match, the system picks a service context with VRF Default. If there is no such service context, the system passes through the packet.



Cluster Wizard - VRF Selection

Select VRF(s) that will be associated with the current context - **waas/1**

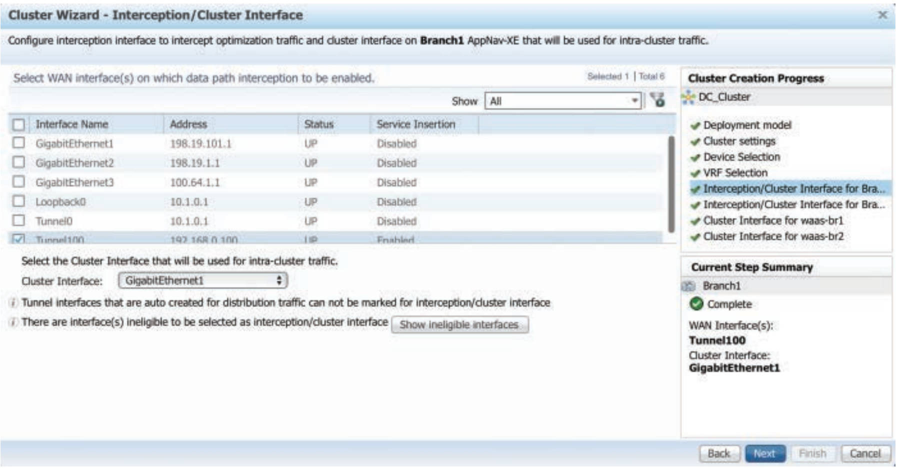
Common VRF(s) from all AppNav-XE Controllers   Show: All

- ☐ VRF
- ☑ VRF default
- ☐ VRF global

Figure 13-21 *Choosing the VRF*

Step 7. Choose the interception interfaces.

Select the interception interfaces and cluster interfaces as shown in Figure 13-22, and then click **Next**.



Cluster Wizard - Interception/Cluster Interface

Configure interception interface to intercept optimization traffic and cluster interface on **Branch1** AppNav-XE that will be used for intra-cluster traffic.

Select WAN interface(s) on which data path interception to be enabled.   Selected 1 | Total 6

Show: All

	Interface Name	Address	Status	Service Insertion
☐	GigabitEthernet1	198.19.101.1	UP	Disabled
☐	GigabitEthernet2	198.19.1.1	UP	Disabled
☐	GigabitEthernet3	100.64.1.1	UP	Disabled
☐	Loopback0	10.1.0.1	UP	Disabled
☐	Tunnel0	10.1.0.1	UP	Disabled
☑	Tunnel100	192.168.0.100	UP	Enabled

Select the Cluster Interface that will be used for intra-cluster traffic.

Cluster Interface: GigabitEthernet1

Tunnel interfaces that are auto created for distribution traffic can not be marked for interception/cluster interface

There are interface(s) ineligible to be selected as interception/cluster interface

Cluster Creation Progress

- ✓ Deployment model
- ✓ Cluster settings
- ✓ Device Selection
- ✓ VRF Selection
- ✓ Interception/Cluster Interface for Bra...
- ✓ Interception/Cluster Interface for Bra...
- ✓ Cluster Interface for waas-br1
- ✓ Cluster Interface for waas-br2

Current Step Summary

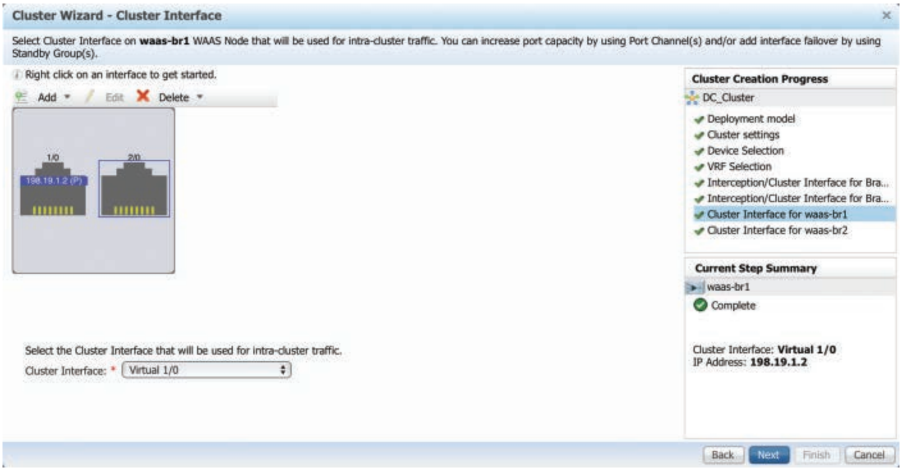
Complete

WAN Interface(s):
Tunnel100
Cluster Interface:
GigabitEthernet1

Figure 13-22 *Interception Interface Selection*

Step 8. **Verify the device configuration.**

Verify the cluster interface, IP address, and netmask for each device in the cluster as shown in Figure 13-23. The wizard automatically selects recommended cluster interfaces that should be configured. To edit the IP address and netmask settings for a device, choose the device and click the **Edit** taskbar icon, and then click **Finish.**


Cluster Wizard - Cluster Interface

Select Cluster Interface on **waas-br1** WAAS Node that will be used for intra-cluster traffic. You can increase port capacity by using Port Channel(s) and/or add interface failover by using Standby Group(s).

Right click on an interface to get started.

Select the Cluster Interface that will be used for intra-cluster traffic.

Cluster Interface: * Virtual 1/0

Cluster Creation Progress

DC_Cluster

- ✓ Deployment model
- ✓ Cluster settings
- ✓ Device Selection
- ✓ VRF Selection
- ✓ Interception/Cluster Interface for Bra...
- ✓ Interception/Cluster Interface for Bra...
- ✓ Cluster Interface for waas-br1
- ✓ Cluster Interface for waas-br2

Current Step Summary

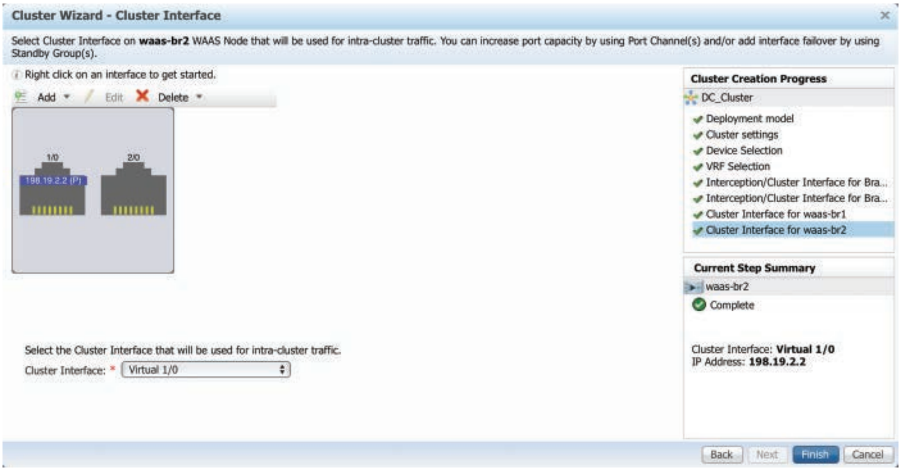
waas-br1

Complete

Cluster Interface: **Virtual 1/0**
IP Address: **198.19.1.2**

Figure 13-23 *Verifying the Device Information*

After all devices have been configured, click **Finish** as shown in Figure 13-24.


Cluster Wizard - Cluster Interface

Select Cluster Interface on **waas-br2** WAAS Node that will be used for intra-cluster traffic. You can increase port capacity by using Port Channel(s) and/or add interface failover by using Standby Group(s).

Right click on an interface to get started.

Select the Cluster Interface that will be used for intra-cluster traffic.

Cluster Interface: * Virtual 1/0

Cluster Creation Progress

DC_Cluster

- ✓ Deployment model
- ✓ Cluster settings
- ✓ Device Selection
- ✓ VRF Selection
- ✓ Interception/Cluster Interface for Bra...
- ✓ Interception/Cluster Interface for Bra...
- ✓ Cluster Interface for waas-br1
- ✓ Cluster Interface for waas-br2

Current Step Summary

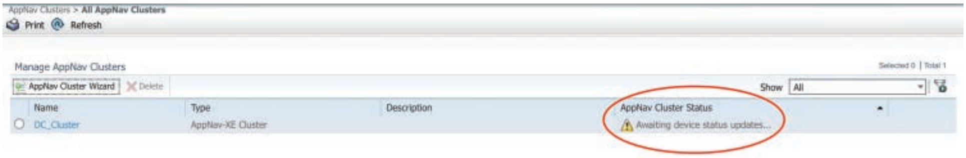
waas-br2

Complete

Cluster Interface: **Virtual 1/0**
IP Address: **198.19.2.2**

Figure 13-24 *End of the AppNav Cluster Wizard*

After the **Finish** button in the AppNav Cluster Wizard has been clicked, an AppNav Cluster Status window appears. Initially the status displays *Awaiting device status updates* as shown in Figure 13-25, but the message will change to *AppNav Cluster is operational*.



Show: All

	Name	Type	Description	AppNav Cluster Status
○	DC_Cluster	AppNav-XE Cluster		⚠ Awaiting device status updates...

Figure 13-25 *Viewing the AppNav Cluster Status*

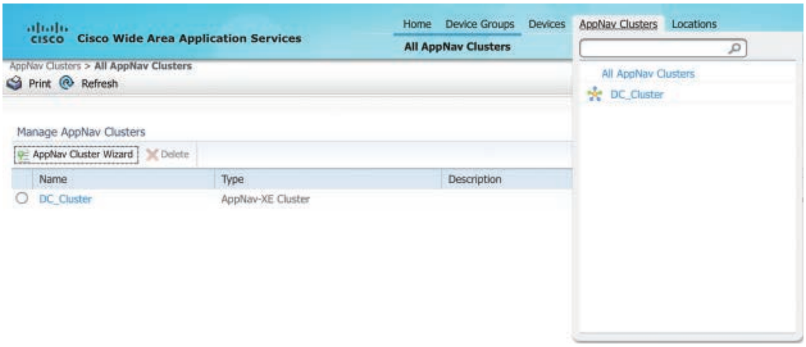
Deploying a Separate Node Group and Policy for Replication

AppNav reduces the dependency on an intercepting switch or router by distributing traffic among WAAS devices for optimization using class and policy matching. The AppNav solution has the ability to scale up to available capacity by taking into account WAAS device utilization as it distributes traffic among nodes. The solution provides for high availability of optimization capacity by monitoring node overload and liveliness and by providing configurable failure and overload policies.

To configure a new WAAS node group and a policy for data center replication on the ANC, follow these steps:

Step 1. **Select the configured cluster.**

From the WAAS Central Manager menu, choose **AppNav Clusters** and select the previously configured cluster. This chapter's example is **DC_Cluster** as shown in Figure 13-26.

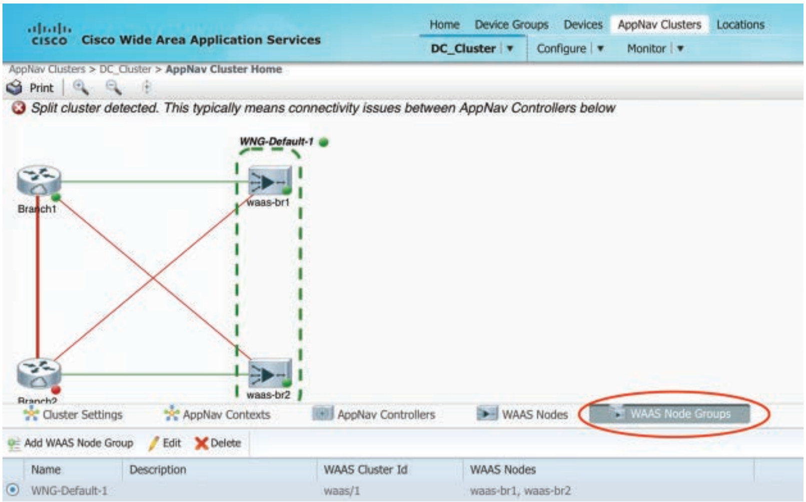


	Name	Type	Description
○	DC_Cluster	AppNav-XE Cluster	

Figure 13-26 *Cluster Selection*

Step 2. **Select the node group.**

Click the **WAAS Node Groups** tab below the topology diagram as shown in Figure 13-27.



Split cluster detected. This typically means connectivity issues between AppNav Controllers below

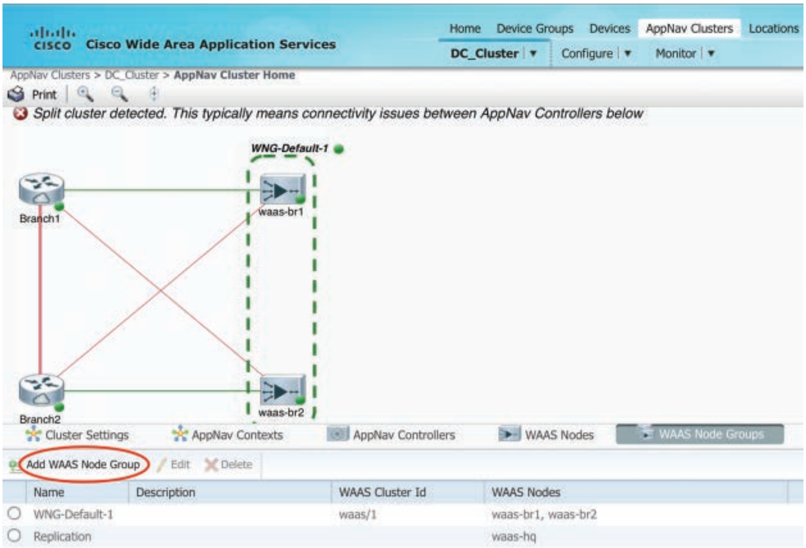
	Name	Description	WAAS Cluster Id	WAAS Nodes
◉	WNG-Default-1		waas/1	waas-br1, waas-br2

Figure 13-27 *WAAS Node Group Selection*

Note There is a convergence waiting period of up to two minutes before the new WAAS group is available.

Step 3. **Add a node group.**

Click the **Add WAAS Node Group** taskbar icon as shown in Figure 13-28.



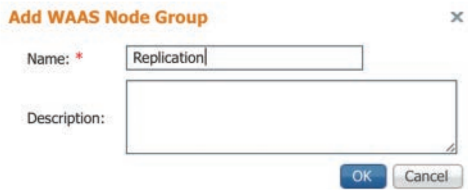
Split cluster detected. This typically means connectivity issues between AppNav Controllers below

	Name	Description	WAAS Cluster Id	WAAS Nodes
○	WNG-Default-1		waas/1	waas-br1, waas-br2
○	Replication			waas-hq

Figure 13-28 *Adding the WAAS Node Group*

Step 4. Name the group.

In the **Name** field, enter the name of the WAAS node group, and then click **OK** to save the settings. Figure 13-29 demonstrates this using the name **Replication**.



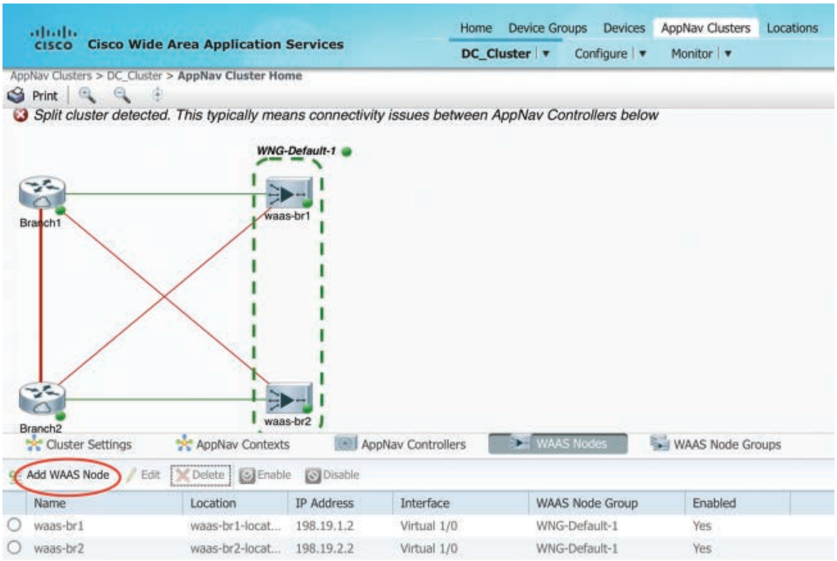
Name: * Replication

Description:

Figure 13-29 *Adding the Name to the WAAS Node Group*

Step 5. Add the node to the group.

Click the **WAAS Nodes** tab below the topology diagram, and then click **Add WAAS Node** as shown in Figure 13-30.



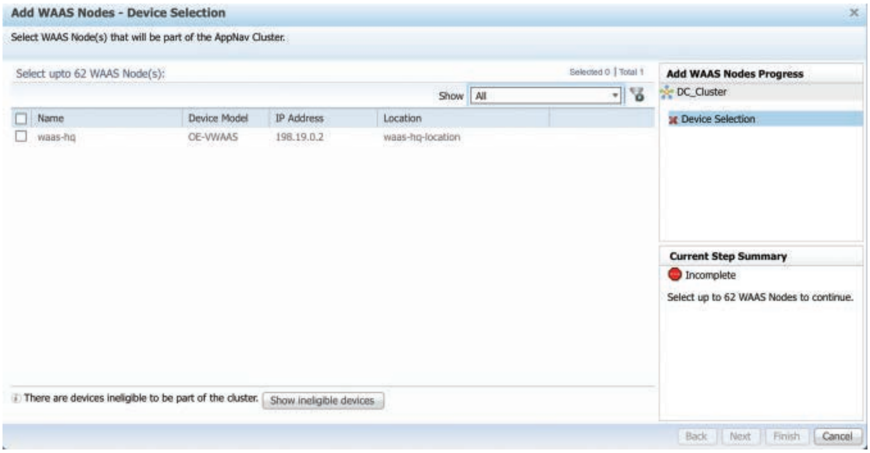
Split cluster detected. This typically means connectivity issues between AppNav Controllers below

	Name	Location	IP Address	Interface	WAAS Node Group	Enabled
○	waas-br1	waas-br1-locat...	198.19.1.2	Virtual 1/0	WNG-Default-1	Yes
○	waas-br2	waas-br2-locat...	198.19.2.2	Virtual 1/0	WNG-Default-1	Yes

Figure 13-30 *Adding the WAAS Node to a Group*

Step 6. Select the nodes to add.

Select the WAAS nodes in the device list that should be added to that WAAS node group as shown in Figure 13-31.



Select WAAS Node(s) that will be part of the AppNav Cluster.

Select upto 62 WAAS Node(s):

□	Name	Device Model	IP Address	Location
□	waas-hq	OE-VWAAS	198.19.0.2	waas-hq-location

Incomplete

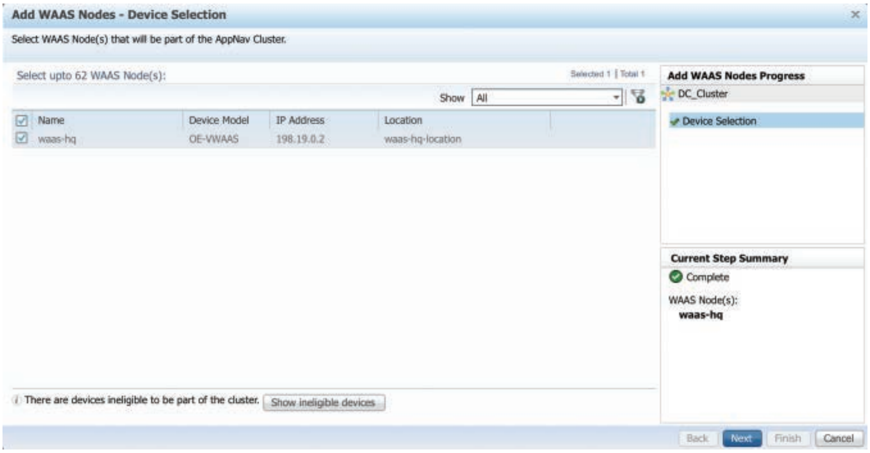
Select up to 62 WAAS Nodes to continue.

There are devices ineligible to be part of the cluster.

Figure 13-31 *List of Available WAAS Nodes*

Step 7. **Add additional nodes to be added.**

After selecting one or more WAAS nodes in the WAAS nodes device list by checking the check boxes next to the device names, click **Next** as shown in Figure 13-32.



Add WAAS Nodes - Device Selection

Select WAAS Node(s) that will be part of the AppNav Cluster.

Select upto 62 WAAS Node(s):

Selected 1 | Total 1

Show: All

☑ Name	Device Model	IP Address	Location
☑ waas-hq	OE-VWAAS	198.19.0.2	waas-hq-location

There are devices ineligible to be part of the cluster.

Add WAAS Nodes Progress

DC_Cluster

✓ Device Selection

Current Step Summary

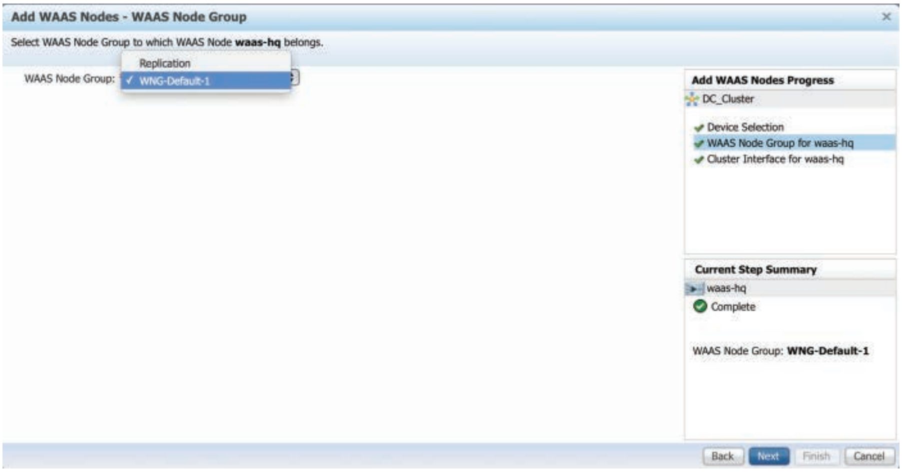
Complete

WAAS Node(s): **waas-hq**

Figure 13-32 *WAAS Node Device Selection*

Step 8. **Add the nodes to the node group.**

From the **WAAS Node Group** drop-down list, choose the WAAS node group to which you want to add the new WAAS nodes. Figure 13-33 displays the selection screen.



Add WAAS Nodes - WAAS Node Group

Select WAAS Node Group to which WAAS Node **waas-hq** belongs.

WAAS Node Group: ✓ WNG-Default-1

Add WAAS Nodes Progress

DC_Cluster

✓ Device Selection

✓ WAAS Node Group for waas-hq

✓ Cluster Interface for waas-hq

Current Step Summary

waas-hq

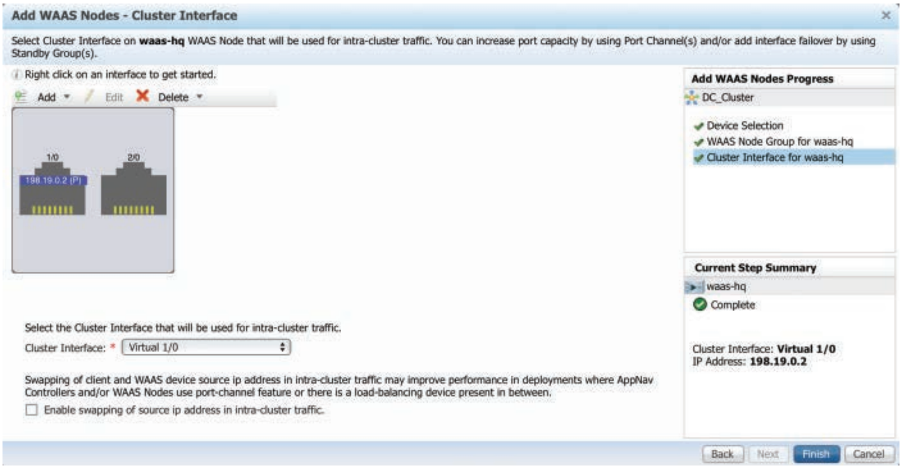
Complete

WAAS Node Group: **WNG-Default-1**

Figure 13-33 *WAAS Node Group Association*

Step 9. **Identify the communication interface.**

The Cluster Interface Wizard needs to identify the WAAS node interface for cluster communications. After selecting the interface, click **Finish** as shown in Figure 13-34.



Add WAAS Nodes - Cluster Interface

Select Cluster Interface on **waas-hq** WAAS Node that will be used for intra-cluster traffic. You can increase port capacity by using Port Channel(s) and/or add interface failover by using Standby Group(s).

Right click on an interface to get started.

Select the Cluster Interface that will be used for intra-cluster traffic.

Cluster Interface: * Virtual 1/0

Swapping of client and WAAS device source ip address in intra-cluster traffic may improve performance in deployments where AppNav Controllers and/or WAAS Nodes use port-channel feature or there is a load-balancing device present in between.

□ Enable swapping of source ip address in intra-cluster traffic.

Add WAAS Nodes Progress

DC_Cluster

- ✓ Device Selection
- ✓ WAAS Node Group for waas-hq
- ✓ Cluster Interface for waas-hq

Current Step Summary

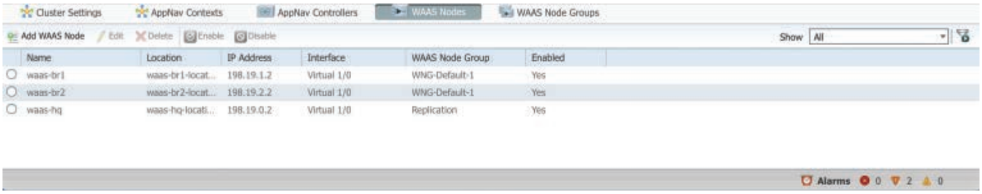
waas-hq

✓ Complete

Cluster Interface: **Virtual 1/0**
IP Address: **198.19.0.2**

Figure 13-34 *Choosing the Node Cluster Interface*

After the WAAS node group has been configured, verify that all WAAS nodes appear under the group WAAS Node Group as shown in Figure 13-35.



	Name	Location	IP Address	Interface	WAAS Node Group	Enabled
○	waas-br1	waas-br1-locat...	198.19.1.2	Virtual 1/0	WNG-Default-1	Yes
○	waas-br2	waas-br2-locat...	198.19.2.2	Virtual 1/0	WNG-Default-1	Yes
○	waas-hq	waas-hq-locati...	198.19.0.2	Virtual 1/0	Replication	Yes

Show: All

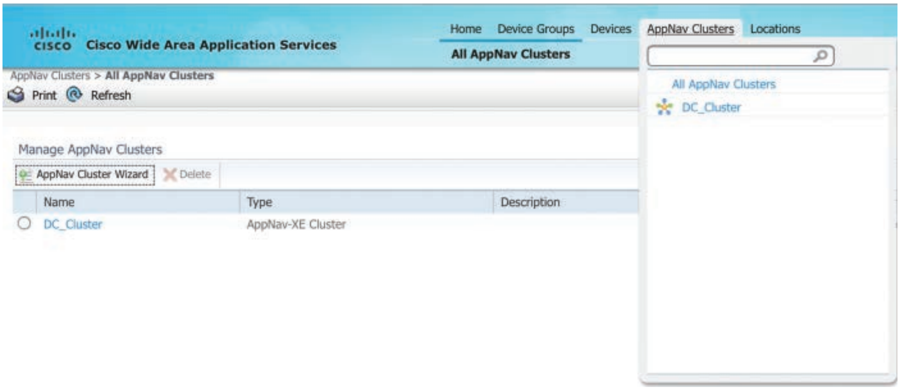
Figure 13-35 *Verification of WAAS Nodes and WAAS Node Groups*

Deploying a New Policy for Data Center Replication

Creating a policy for data center replication for the AppNav Cluster involves the following steps:

Step 1. **Select the configured cluster.**

From the WAAS Central Manager menu, choose **AppNav Clusters** and select the previously configured cluster. This chapter's example is **DC_Cluster** as shown in Figure 13-36.

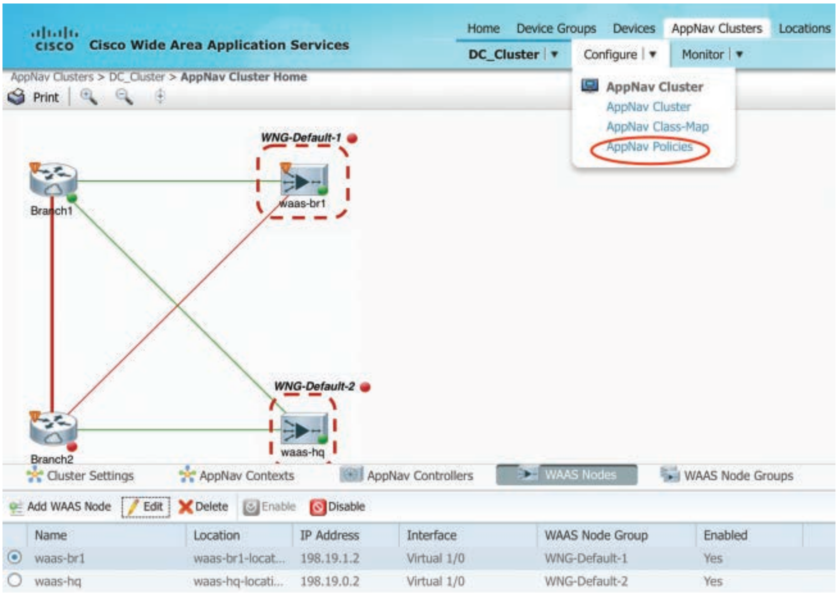


	Name	Type	Description
○	DC_Cluster	AppNav-XE Cluster	

Figure 13-36 *Cluster Selection for Data Center Replication*

Step 2. Navigate to the policies area.

Click **Configure > AppNav Cluster > AppNav Policies** as shown in Figure 13-37.

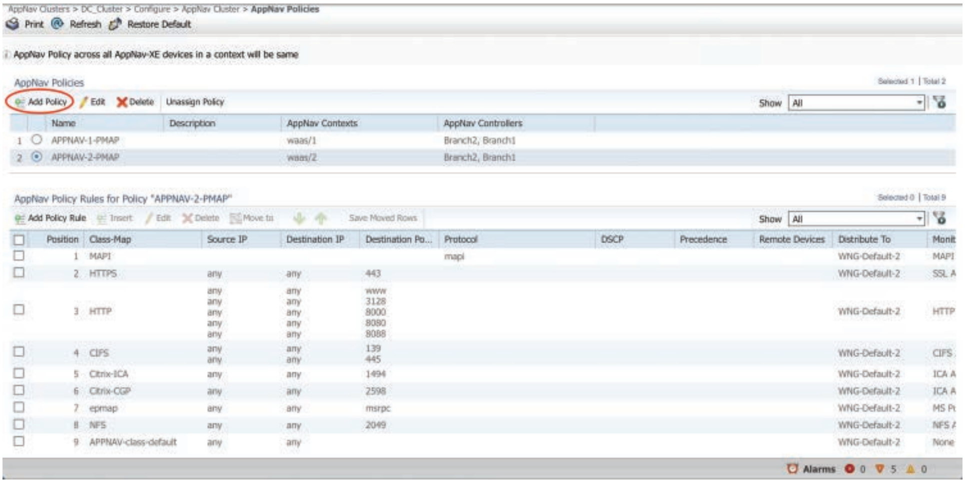


	Name	Location	IP Address	Interface	WAAS Node Group	Enabled
●	waas-br1	waas-br1-locat...	198.19.1.2	Virtual 1/0	WNG-Default-1	Yes
○	waas-hq	waas-hq-locati...	198.19.0.2	Virtual 1/0	WNG-Default-2	Yes

Figure 13-37 *Navigating to AppNav Policies*

Step 3. Add a policy.

Click the **Add Policy** taskbar icon as shown in Figure 13-38.



AppNav Clusters > DC_Cluster > Configure > AppNav Cluster > **AppNav Policies**

Print Refresh Restore Default

AppNav Policy across all AppNav-XE devices in a context will be same

AppNav Policies

Selected 1 | Total 2

Add Policy Edit Delete Unassign Policy Show All

		Name	Description	AppNav Contexts	AppNav Controllers
1	○	APPNAV-1-PMAP		waas/1	Branch2, Branch1
2	●	APPNAV-2-PMAP		waas/2	Branch2, Branch1

AppNav Policy Rules for Policy "APPNAV-2-PMAP"

Selected 0 | Total 9

Add Policy Rule Insert Edit Delete Move to Save Moved Rows Show All

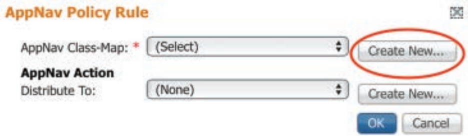
□	Position	Class-Map	Source IP	Destination IP	Destination Po...	Protocol	DSCP	Precedence	Remote Devices	Distribute To	Monit
□	1	MAPI				mapi				WNG-Default-2	MAPI
□	2	HTTPS	any	any	443					WNG-Default-2	SSL A
□	3	HTTP	any any any any any	any any any any any	www 3128 8000 8080 8088					WNG-Default-2	HTTP
□	4	CIFS	any any	any any	139 445					WNG-Default-2	CIFS
□	5	Citrix-ICA	any	any	1494					WNG-Default-2	ICA A
□	6	Citrix-CGP	any	any	2598					WNG-Default-2	ICA A
□	7	epmap	any	any	msrpc					WNG-Default-2	MS Pi
□	8	NFS	any	any	2049					WNG-Default-2	NFS A
□	9	APPNAV-class-default	any	any						WNG-Default-2	None

Alarms 0 5 0

Figure 13-38 *Selecting Add Policy*

Step 4. Create a new policy.

Next to the **AppNav Class-Map** drop-down list, click **Create New** as shown in Figure 13-39.



AppNav Policy Rule

AppNav Class-Map: * (Select) Create New...

AppNav Action

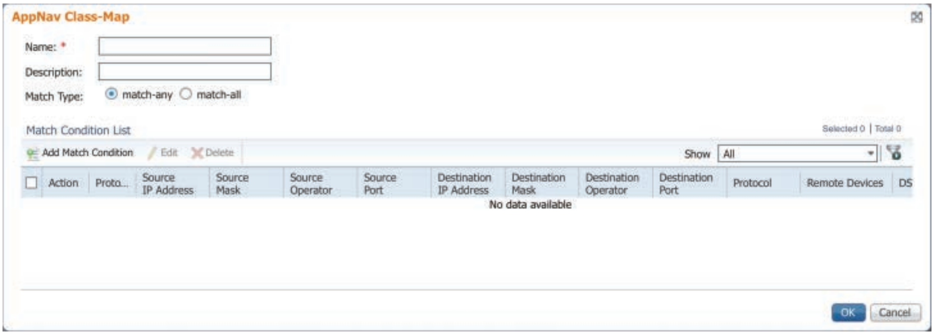
Distribute To: (None) Create New...

OK Cancel

Figure 13-39 *Creation of a New AppNav Policy Rule*

Step 5. Name the policy.

In the **Name** field, enter a name for the class map as shown in Figure 13-40. This chapter uses the name **Replication**. Then click **OK**.



AppNav Class-Map

Name: *

Description:

Match Type: ● match-any ○ match-all

Match Condition List

Selected 0 | Total 0

Add Match Condition Edit Delete Show All

□	Action	Proto...	Source IP Address	Source Mask	Source Operator	Source Port	Destination IP Address	Destination Mask	Destination Operator	Destination Port	Protocol	Remote Devices	DS

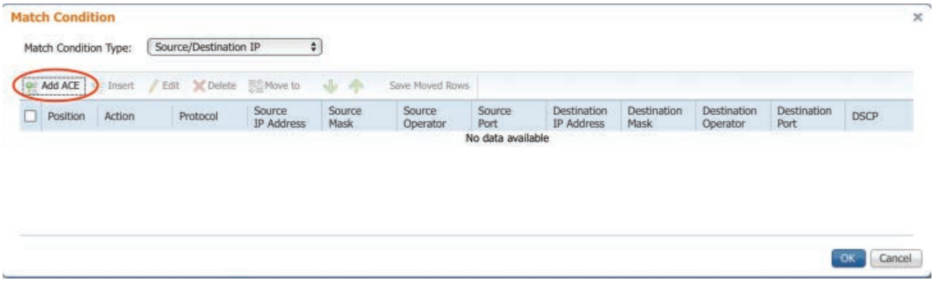
No data available

OK Cancel

Figure 13-40 *Creation of the AppNav Class Map*

Step 6. Create an ACE.

The WAAS class maps use ACL-like functions to identify traffic based on Layer 3 and Layer 4 characteristics. An *Access Control Entry (ACE)* is created to match traffic based on the specific Layer 3 and Layer 4 patterns. Click the **Add ACE** taskbar icon as shown in Figure 13-41.



Match Condition

Match Condition Type: Source/Destination IP

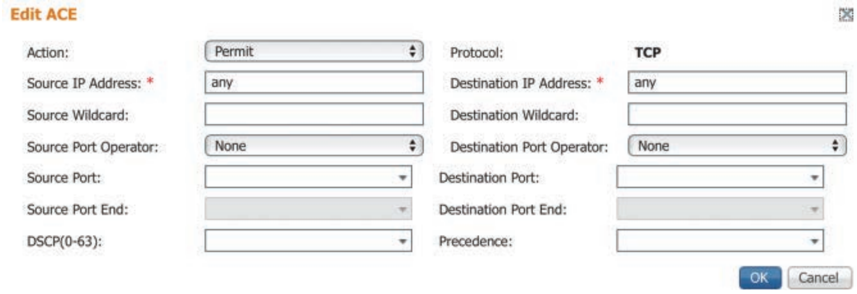
☐	Position	Action	Protocol	Source IP Address	Source Mask	Source Operator	Source Port	Destination IP Address	Destination Mask	Destination Operator	Destination Port	DSCP

No data available

Figure 13-41 *Adding Match Conditions*

Step 7. Define the ACE.

Add the match criteria (source/destination IP ranges, DSCP/precedence, and so on) as shown in Figure 13-42. After completing the ACE match criteria, click **OK**.

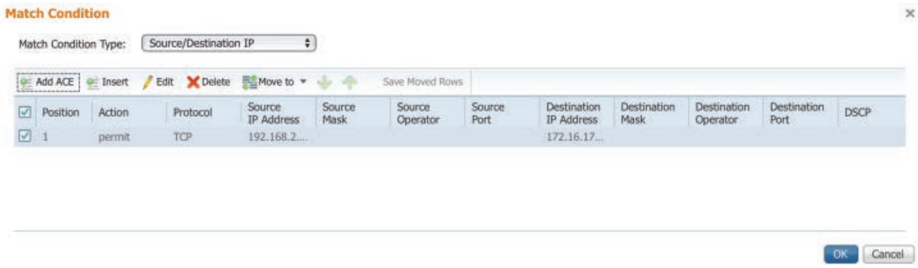


Edit ACE

Action:	Permit	Protocol:	TCP
Source IP Address: *	any	Destination IP Address: *	any
Source Wildcard:		Destination Wildcard:	
Source Port Operator:	None	Destination Port Operator:	None
Source Port:		Destination Port:	
Source Port End:		Destination Port End:	
DSCP(0-63):		Precedence:	

Figure 13-42 *Entering Data in the Class Map ACE Fields*

Figure 13-43 displays the ACE entries associated to the class map. After filling in as many of the class map ACE entries as needed, click **OK**.



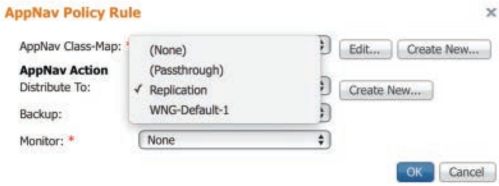
Match Condition Type: Source/Destination IP

☑	Position	Action	Protocol	Source IP Address	Source Mask	Source Operator	Source Port	Destination IP Address	Destination Mask	Destination Operator	Destination Port	DSCP
☑	1	permit	TCP	192.168.2....				172.16.17....				

Figure 13-43 *Review/Addition of Class Map ACE Entries*

Step 8. **Choose the AppNav action.**

In the AppNav Policy Rule window, use the AppNav **Distribute To** drop-down list to select the **Distribution** action. Then select the AppNav WAAS Device Group as shown in Figure 13-44.



AppNav Class-Map: *

AppNav Action

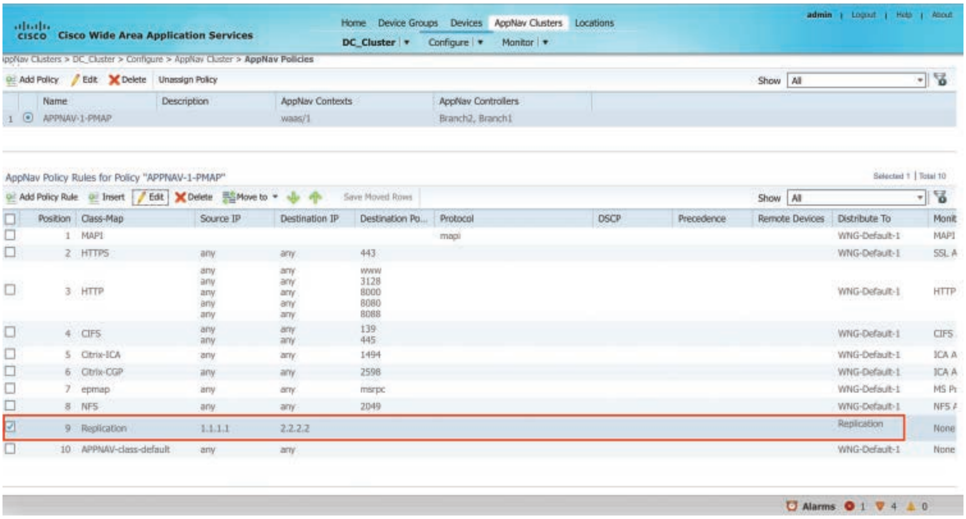
Distribute To: Replication

Backup:

Monitor: * None

Figure 13-44 *Distribute Action of the Class Map*

Now verify that the new class map has been added to the AppNav policy. Notice that in Figure 13-45 the new Replication class map has been added.



ppNav Clusters > DC_Cluster > Configure > AppNav Cluster > **AppNav Policies**

		Name	Description	AppNav Contexts	AppNav Controllers
1	●	APPNAV-1-PMAP		waas/1	Branch2, Branch1

AppNav Policy Rules for Policy "APPNAV-1-PMAP"

Selected 1 | Total 10

□	Position	Class-Map	Source IP	Destination IP	Destination Po...	Protocol	DSCP	Precedence	Remote Devices	Distribute To	Monit
□	1	MAPI				mapi				WNG-Default-1	MAPI
□	2	HTTPS	any	any	443					WNG-Default-1	SSL A
□	3	HTTP	any any any any any	any any any any any	www 3128 8000 8080 8088					WNG-Default-1	HTTP
□	4	CIFS	any any	any any	139 445					WNG-Default-1	CIFS
□	5	Citrix-ICA	any	any	1494					WNG-Default-1	ICA A
□	6	Citrix-CGP	any	any	2598					WNG-Default-1	ICA A
□	7	epmap	any	any	msrpc					WNG-Default-1	MS Pr
□	8	NFS	any	any	2049					WNG-Default-1	NFS A
☑	9	Replication	1.1.1.1	2.2.2.2						Replication	None
□	10	APPNAV-class-default	any	any						WNG-Default-1	None

Figure 13-45 *Verify the Updated AppNav Policy*

GBI Branch Deployment

Now that the process for deploying Cisco WAAS clusters has been explained, the GBI case study will describe the process for deploying two different branches based upon Figure 13-1.

Branch 1 Sizing

GBI's Branch 1 has 20 users and a 2 Mbps WAN circuit. Sizing Branch 1 with 200 concurrent TCP connections and a 2 Mbps WAN circuit has resulted in the use of an ISR 4331 router. The ISR 4331 router supports up to 750 concurrent connections using ISR-WAAS and is rated for 25 Mbps of bandwidth. This allows for more services and a smaller equipment footprint in the branch.

Branch 1 Deployment

Branch 1's ISR-WAAS deployment will use the EZConfig program. The EZConfig program is a single CLI command **service waas enable** that launches an interactive mode for enabling ISR-WAAS. The program walks you through a series of questions and enables the corresponding AppNav Controller, container, interface, and WAAS configurations. Privilege level 15 is required to execute WAAS EZConfig.

Example 13-5 demonstrates the use of WAAS EZConfig.

Example 13-5 *Running EZConfig Programming*

```
! Output omitted for brevity
!
router# service waas enable

**************************************************************************
**** Entering WAAS service interactive mode. ****
**** You will be asked a series of questions, and your answers ****
**** will be used to modify this device's configuration to ****

**** enable a WAAS Service on this router. ****
**************************************************************************
Continue? [y]: y

 At any time: ? for help, CTRL-C to exit.

Only one WAAS image found locally (harddisk:/ISR-WAAS-5.3.5a.5.ova) - using as
  default
Extracting profiles from harddisk:/ISR-WAAS-5.3.5a.5.ova, this may take a couple of
  minutes ...
These are the available profiles
1. ISR-WAAS-750
```

```
Select option [1]:1
An internal IP interface and subnet is required to deploy a WAAS service on this
  router.
This internal subnet must contain two usable IP addresses that can route and
  communicate with the WAAS Central Manager (WCM).

Enter the IP address to be configured on the WAAS service: 10.5.36.8

The following IP interfaces are currently available on the router:

Enter a WAN interface to enable WAAS interception
Enter additional WAN interface (blank to finish) []: Tunnel 100
Enter additional WAN interface (blank to finish) []: Tunnel 200
******************* ** Configuration Summary: ** ********************
a)  WAAS Image and Profile Size:
harddisk:/ISR-WAAS-5.3.5a.5.ova (941127680) bytes
ISR-WAAS-750

b)  Router IP/mask:
Using ip unnumbered from interface Port-channel1.64

WAAS Service IP:
10.5.36.8

c)  WAAS Central Manager:
198.19.0.3

Router WAN Interfaces:
Tunnel200
Tunnel100

Choose one of the letter from ‘a-d’ to edit, ‘v’ to view config script, ‘s’ to apply
  config [s]:s
```

Now that the ISR-WAAS has been deployed, the status is verified with the command **show cms info** as shown in Example 13-6.

Example 13-6 *Branch 1 ISR-WAAS Status*

```
isr-waas1# show cms info
Device registration information :
Device Id                            = 8704
Device registered as                 = WAAS Application Engine
Current WAAS Central Manager          = 198.19.0.3
Registered with WAAS Central Manager = 198.19.0.3
```

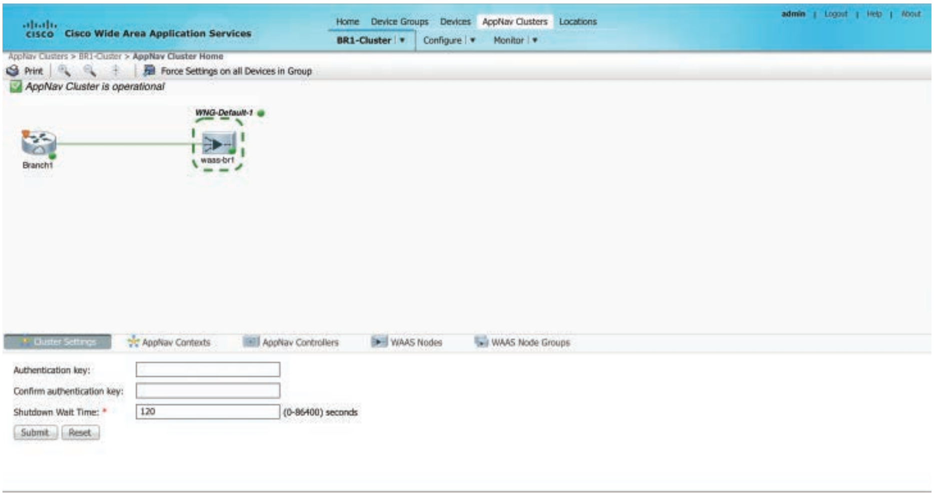
```
Status                                  = Online
Time of last config-sync                = Thu Jul  7 12:12:30 2016

CMS services information :
Service cms_ce is running
```

The health of the cluster is verified with the command **show service-insertion service-context** as shown in Example 13-7.

Example 13-7 *Verification of the Cluster Health*

```
router# show service-insertion service-context
Service Context : waas/1

Cluster protocol ICIMP version
    : 1.1

Cluster protocol DMP version
    : 1.1

Time service context was enabled
    : Mon Jun 23 19:54:39 2016

Current FSM state
    : Operational

Time FSM entered current state
    : Mon Jun 23 19:54:53 2016

Last FSM state
    : Converging

Time FSM entered last state
    : Mon Jun 23 19:54:39 2016

Cluster operational state
    : Operational

Stable AppNav controller View:
10.5.36.1

Stable SN View:
10.5.36.8
```

Every AppNav Cluster has a 360-degree view that verifies that the cluster is operational and online. Using the Central Manager as shown in Figure 13-46 provides an additional verification step.



☑ AppNav Cluster is operational

Authentication key:

Confirm authentication key:

Shutdown Wait Time: * 120 (0-86400) seconds

Figure 13-46 *Verification of Site 1's AppNav Cluster*

Branch 12 Sizing

GBI's Branch 12 has 150 users and a 10 Mbps WAN circuit. Sizing Branch 12 with 1500 concurrent TCP connections and a 10 Mbps WAN circuit has resulted in the use of an ISR 4451. The ISR 4451 router supports up to 2500 concurrent connections using ISR-WAAS and is rated for 50 Mbps of bandwidth. This allows for more services and a smaller equipment footprint in the branch.

Branch 12 WAAS Deployment

Branch 12's ISR-WAAS deployment will use the EZConfig program. The EZConfig program is a single CLI command that launches an interactive mode for enabling ISR-WAAS. The program walks you through a series of questions and enables the corresponding AppNav Controller, container, interface, and WAAS configurations.

Example 13-8 displays Branch 12's deployment of WAAS.

Example 13-8 *Branch 12's WAAS Deployment*

```
! Output omitted for brevity
!
router# service waas enable

**************************************************************************
**** Entering WAAS service interactive mode. ****
**** You will be asked a series of questions, and your answers ****
**** will be used to modify this device's configuration to ****

**** enable a WAAS Service on this router. ****
**************************************************************************
Continue? [y]: y

 At any time: ? for help, CTRL-C to exit.

Only one WAAS image found locally (harddisk:/ISR-WAAS-5.3.5a.5.ova) - using as
  default
Extracting profiles from harddisk:/ISR-WAAS-5.3.5a.5.ova, this may take a couple of
  minutes ...
These are the available profiles
1. ISR-WAAS-2500
2. ISR-WAAS-1300
3. ISR-WAAS-750

Select option [1]:2
An internal IP interface and subnet is required to deploy a WAAS service on this
  router.
This internal subnet must contain two usable IP addresses that can route and
  communicate with the WAAS Central Manager (WCM).

Enter the IP address to be configured on the WAAS service: 10.6.36.8

The following IP interfaces are currently available on the router:

Enter a WAN interface to enable WAAS interception
Enter additional WAN interface (blank to finish) []: Tunnel 100
Enter additional WAN interface (blank to finish) []: Tunnel 200
******************* ** Configuration Summary: ** ********************
a)  WAAS Image and Profile Size:
harddisk:/ISR-WAAS-5.3.5a.5.ova (941127680) bytes
ISR-WAAS-2500

b)  Router IP/mask:
Using ip unnumbered from interface Port-channel1.64
```

```
WAAS Service IP:
10.6.36.8

c)  WAAS Central Manager:
198.19.0.3

Router WAN Interfaces:
Tunnel200
Tunnel100

Choose one of the letter from 'a-d' to edit, 'v' to view config script, 's' to apply
  config [s]:s
```

The next step is to verify that ISR-WAAS is registered to the Central Manager by running the command **show cms info** as shown in Example 13-9.

Example 13-9 *Verification of ISR-WAAS Registration*

```
waas-br1# show cms info
Device registration information :
Device Id                               = 8704
Device registered as                    = WAAS Application Engine
Current WAAS Central Manager            = 198.19.0.3
Registered with WAAS Central Manager = 198.19.0.3
Status                                  = Online
Time of last config-sync                = Thu Jul  7 12:12:30 2016
CMS services information :
Service cms_ce is running
```

Last, the cluster status is verified with the command **show service-insertion service-context** as shown in Example 13-10.

Example 13-10 *Verification of Cluster Status*

```
router# show service-insertion service-context
Service Context : waas/1

Cluster protocol ICIMP version
  : 1.1

Cluster protocol DMP version
  : 1.1

Time service context was enabled
  : Mon Jun 23 19:58:39 2016
```

```
Current FSM state
  : Operational

Time FSM entered current state
  : Mon Jun 23 19:58:53 2016

Last FSM state
  : Converging

Time FSM entered last state
  : Mon Jun 23 19:58:39 2016

Cluster operational state
  : Operational

Stable AppNav controller View:
10.6.36.1

Stable SN View:
10.6.36.8
```

Summary

This chapter described the practical application of the knowledge gained throughout this book. It looked at the design considerations that are key to a successful WAAS deployment in the data center environment and branch office. A two-site deployment was analyzed, with slightly different implementation solutions chosen for each site. The example design in this chapter highlights the ease of the overall deployment. Example configurations were provided for all the key components in the design.

At this point, you should have a solid understanding of the design and network integration options available with Cisco WAAS. This chapter explored various options for integrating Cisco WAAS into the branch office network infrastructure. Different topology and configuration scenarios were discussed, including both in-path and off-path interception options. The material in this chapter provides you with a solid set of options for integrating WAAS into different network topologies.

This page intentionally left blank

Chapter 14

Intelligent WAN Quality of Service (QoS)

This chapter covers the following topics:

- Classification and queueing
- Hierarchical QoS
- DMVPN Per-Tunnel QoS
- QoS and IPsec interaction

Quality of Service (QoS) is a vital component of any network design. It is a method of granting preference to one type of network traffic over a different traffic type. Traffic is typically identified and marked close to the packet's origin within a trusted environment. Most organizations use the *Differentiated Services (DiffServ)* QoS model that queues, shapes, and polices network traffic on a *per-hop basis (PHB)* after evaluating the QoS markings. If all the devices have the same policy, network traffic will have a consistent end-to-end QoS policy.

Cisco IWAN offers QoS capabilities that are not available with a traditional WAN. IWAN provides the following QoS benefits to an organization:

- **Consistent QoS markings:** MPLS VPN service providers typically limit the number of QoS markings to specific DSCPs, and they sometimes remark network traffic as it travels across their infrastructure. This means that a customer's network traffic may need to be marked to match the service provider's QoS settings and then reclassified (changed back) once the traffic has crossed the WAN.
- **Per-Tunnel QoS:** With Per-Tunnel QoS a spoke router identifies the QoS shaping policy that the hub router should apply when communicating explicitly with that spoke. The QoS policy specifies the ingress WAN bandwidth capabilities of the remote site, which in turn shapes the rate at which the hub router transmits packets to the spoke router. This prevents the hub site from flooding a spoke site (because

hub sites typically have more bandwidth than branch sites) and protects the hub's available bandwidth from greedy spokes.

The goal of this chapter is to supplement the reader's existing QoS knowledge with specific concepts that are extended within Cisco IWAN.

QoS Overview

QoS configuration begins with the classification of network traffic via a class map that defines what traffic matches and how it is classified. The QoS classification of traffic is traditionally based on a value in the traffic itself. The traffic can be classified by DSCP, IP precedence, source IP address, destination IP address, protocol, source port, destination port, and/or application ID, as explained in Chapter 6, "Application Recognition."

A QoS policy map is then used to define queueing actions upon the classes. Within the policy map, real-time traffic can be prioritized to manage delay and minimize jitter while bandwidth is assigned to that class. This provides guaranteed service for a classification of network traffic. Most organizations have deployed a QoS policy in their LAN environment that matches a 12-class policy defined in RFC 4594, an example of which is shown in Table 14-1. The primary goal is to elevate priority for applications that are sensitive to delay or packet retransmission. Using a 12-class policy as opposed to an eight-class policy provides more granularity among all applications that are present on a network.

Table 14-1 *Sample 12-Class QoS Policy*

Service Class Name	Admission Control	DSCP Name	DSCP Value	Application Function
Network Control	N/A	CS6	110000	Routing protocols (EIGRP, OSPF, BGP, etc.)
Telephony	Required	EF	101110	Voice calls
Signaling	Required	CS5	101000	Video surveillance
Multimedia Conferencing	Required	AF41, AF42, AF43	100010 100100 100110	Cisco Jabber, WebEx
Real-Time Interactive	Required	CS4	100000	Cisco TelePresence
Multimedia Streaming	Recommended	AF31, AF32, AF33	011010 011100 011110	Video on demand
Broadcast Video	N/A	CS3	011000	Skinny Client Control Protocol (SCCP), Session Initiation Protocol (SIP)

Service Class Name	Admission Control	DSCP Name	DSCP Value	Application Function
Low-Latency Data	N/A	AF21, AF22, AF23	001010 001100 001110	ERP/CRM apps or databases
OAM	N/A	CS2	010000	SNMP, SSH, syslog
High-Throughput Data	N/A	AF11, AF12, AF13	001010 001100 001110	Email, FTP
Best Effort	N/A	DF	000000	Default class
Low-Priority Data	N/A	CS1	001000	Scavenger (Netflix, iTunes)

It is important to understand that as part of the GRE and IPsec RFCs, the QoS *type of service (TOS)* byte from the original packet is copied to the outer packet header. Copying the TOS byte allows the QoS policy to continue to match on both DSCP and IP precedence. However, it does not offer a method to match on any of the additional values commonly used to classify traffic.

In environments where traffic is not classified and marked according to an end-to-end policy this can lead to trouble. Because the packet's original information has been encapsulated in the payload, the packet's original source and destination IP addresses cannot be differentiated along with other common fields used to define a QoS policy.

To accomplish classification based on the other common fields in the network traffic (source/destination IP address and/or source/destination protocol/port), the command **qos pre-classify** must be configured on the tunnel interface to allow classification based on the inner packet header. The command **qos pre-classify** allows router-sourced traffic to be remarked and classified if desired.

Unfortunately, **qos pre-classify** matches only on the header information and does not allow *deep packet inspection (DPI)* required by NBAR to support matching on an application ID. Changing the logic of the router's QoS policy and separating classification and queueing into two separate processes allow traffic to be classified using NBAR. The processes are as follows:

- Classifying network traffic on the ingress LAN interface based on application ID and configuring the QoS policy to mark the DSCP value for egress classification based on the new DSCP value
- Queueing on the egress WAN interface using the newly set DSCP value

The IWAN QoS model is based upon this concept of separating classification and queuing into two separate processes.

Note The two-stage QoS model removes the need to use **qos pre-classify** for traffic traversing the router.

Ingress QoS NBAR-Based Classification

Not all network traffic is properly marked in the enterprise network. Most network engineers think that matching traffic based on common Layer 3 and Layer 4 fields is sufficient. However, this technique lacks visibility to the applications on their network, and the relevance that they provide to the business can change. Administrators can implement policies more effectively after identifying the types of applications and protocols that are running on a network. These policies do not change when a new server is brought up or moved to a different location, because the application classification remains the same. If traffic is not properly marked on a network segment, it should be classified and marked based on the application. NBAR provides an accurate identification of network traffic using multiple engines.

The first step for configuring an NBAR-based class map is to define the class map with the command **class-map** [**match-all** | **match-any**] *class-map-name*. Multiple different packet attributes can be used as a conditional match. If more than one conditional match attribute is used, a differentiation must be identified between the **match-all** and **match-any** keywords.

Note If the packet must match all the conditional match attributes, the **match-all** keyword should be used. If the packet must match at least one conditional match attribute, the keyword **match-any** should be used. If neither keyword is specified, **match-any** is implied.

The next step requires that conditional attribute matches be defined. Some of the more common fields are as follows:

- **match protocol attribute business-relevance** {**business-relevant** | **default** | **business-irrelevant**}: Business relevance directly refers to applications that are identified as follows:
  - **Business relevant:** These applications directly benefit the business objectives and should be classified according to RFC 4594–based rules.
  - **Default:** The application may or may not support business objectives (for example, HTTP). Alternatively, the network engineers may not know the application or how it is being used in the organization.

  - **Business irrelevant:** These applications are known and do not directly support any business objectives. This type of traffic includes all personal and consumer applications. Applications in this class should be marked CS1 and provisioned with a less-than-best-effort service as outlined in RFC 3662.

- **match protocol attribute traffic-class** *traffic-class-name*: This is an attribute defined by the Solution Reference Network Design (SRND) policy model that simplifies identification of traffic for QoS policies. Ten traffic classes have been defined that apply to business-relevant applications.

- **match protocol** *protocol-name sub-classification value*: The *sub-classification* keyword depends on the specified *protocol-name*. The **http** protocol allows wildcard matching and is covered in depth in Chapter 6.

- **match access-group** {*1-2799* | **name** *named-acl*}: This allows the use of an ACL for conditional matching on Layer 3 and Layer 4 information.

Note A list of the various NBAR protocol attributes and the technique to modify them is provided in Chapter 6.

Example 14-1 provides the configuration for the NBAR-based QoS class maps used for the classification of LAN traffic as it enters the IWAN routers. Most of the class maps use the **match-all** keyword with a conditional match on **traffic-class** and **business-relevance** as part of the SRND guidelines.

Example 14-1 *NBAR-Based Class Maps for Ingress Marking*

```
R11, R12, R21, R22, R31, R41, R51, and R52
class-map match-all CLASS-NBAR-VOICE
   match protocol attribute traffic-class voip-telephony
   match protocol attribute business-relevance business-relevant
class-map match-all CLASS-NBAR-BROADCAST-VIDEO
   match protocol attribute traffic-class broadcast-video
   match protocol attribute business-relevance business-relevant
class-map match-all CLASS-NBAR-REAL-TIME-INTERACTIVE
   match protocol attribute traffic-class real-time-interactive
   match protocol attribute business-relevance business-relevant
class-map match-all CLASS-NBAR-MULTIMEDIA-CONFERENCING
   match protocol attribute traffic-class multimedia-conferencing
   match protocol attribute business-relevance business-relevant
class-map match-all CLASS-NBAR-MULTIMEDIA-STREAMING
   match protocol attribute traffic-class multimedia-streaming
   match protocol attribute business-relevance business-relevant
 class-map match-all CLASS-NBAR-SIGNALING
   match protocol attribute traffic-class signaling
   match protocol attribute business-relevance business-relevant
```

```
class-map match-all CLASS-NBAR-NETWORK-CONTROL
   match protocol attribute traffic-class network-control
   match protocol attribute business-relevance business-relevant
class-map match-all CLASS-NBAR-NETWORK-MANAGEMENT
   match protocol attribute traffic-class ops-admin-mgmt
   match protocol attribute business-relevance business-relevant
class-map match-all CLASS-NBAR-TRANSACTIONAL-DATA
   match protocol attribute traffic-class transactional-data
   match protocol attribute business-relevance business-relevant
class-map match-all CLASS-NBAR-BULK-DATA
   match protocol attribute traffic-class bulk-data
   match protocol attribute business-relevance business-relevant
class-map match-all CLASS-NBAR-SCAVENGER
   match protocol attribute business-relevance business-irrelevant
```

The policy in Example 14-1 accurately categorizes most applications. However, some applications may not match the needs of your organization. For example, HTTP is defined by the NBAR Protocol Pack with the *transactional-data* traffic class and a *default* business relevance. The *business-relevance* value can be changed to *business-relevant*, but that would make all HTTP traffic business relevant. Business-irrelevant traffic can then be included in that QoS model because many applications use HTTP as a transport protocol. Many network engineers consider HTTP to be "the new TCP."

This scenario is overcome by using a nested approach to traffic classification that involves the creation of two additional class maps. Creating a class map that provides explicit identification of the business-relevant HTTP traffic can be accomplished by using an ACL based on server and client IP addresses or through the host URL. Example 14-2 demonstrates both techniques for classification of business-relevant HTTP traffic.

Example 14-2 *Classifying Business-Relevant HTTP Traffic*

```
! ACL Method
ip access-list extended ACL-BUSINESS-RELEVANT-HTTP
 permit tcp 10.0.0.0 0.255.255.255 10.1.0.0 0.0.255.255 eq 80
 permit tcp 10.1.0.0 0.0.255.255 eq 80 10.0.0.0 0.255.255.255
 permit tcp 10.0.0.0 0.255.255.255 10.2.0.0 0.0.255.255 eq 80
 permit tcp 10.2.0.0 0.0.255.255 eq 80 10.0.0.0 0.255.255.255
!
class-map match-all CLASS-BUSINESS-RELEVANT-HTTP-ACL
 match access-group name ACL-BUSINESS-RELEVANT-HTTP
```

```
! URL Method
class-map match-all CLASS-BUSINESS-RELEVANT-HTTP-URL
 match protocol http host *.cisco.com
```

Now a nested class map is created that references the NBAR-based class map and the two custom class maps that were built earlier in Example 14-2. The class map configuration command **class** *class-map-name* allows the nesting of other defined class maps. This class map uses the **match-any** keyword so traffic has to match only one of the child class maps or the other child class map, but not both child class maps, to be considered a part of the parent class map. Example 14-3 provides the sample configuration of the new nested class map.

Example 14-3 *Creating a Nested Class Map*

```
class-map match-any CLASS-MIXED-TRANSACTIONAL-DATA
 class CLASS-NBAR-TRANSACTIONAL-DATA
 class CLASS-BUSINESS-RELEVANT-HTTP-URL
 class CLASS-BUSINESS-RELEVANT-HTTP-ACL
```

The new class map CLASS-MIXED-TRANSACTIONAL-DATA would then be used in any ingress QoS policy maps so that the business-relevant HTTP traffic is prioritized appropriately.

Note Another solution is to create a new custom NBAR application as defined in Chapter 6. This would not require the additional class maps and would provide visibility in any reporting.

Ingress LAN Policy Maps

The policy map is a basic component within the Cisco QoS structure. The ingress policy map identifies the type of network traffic by using the NBAR class maps previously defined, and then remarking the packet's DSCP which can be shaped later as part of the egress QoS policy.

Example 14-4 demonstrates the QoS policy map used for remarking the traffic as it enters the IWAN router on the LAN interface GigabitEthernet0/3.

Example 14-4 *Ingress Policy Map for IWAN Routers*

```
R11, R12, R21, R22, R31, R41, R51, and R52
policy-map POLICY-INGRESS-LAN-MARKING
 class CLASS-NBAR-VOICE
  set dscp ef
 class CLASS-NBAR-BROADCAST-VIDEO
  set dscp cs5
 class CLASS-NBAR-REAL-TIME-INTERACTIVE
  set dscp cs4
 class CLASS-NBAR-MULTIMEDIA-CONFERENCING
  set dscp af41
 class CLASS-NBAR-MULTIMEDIA-STREAMING
  set dscp af31
 class CLASS-NBAR-SIGNALING
  set dscp cs3
 class CLASS-NBAR-NETWORK-CONTROL
  set dscp cs6
 class CLASS-NBAR-NETWORK-MANAGEMENT
  set dscp cs2
 class CLASS-NBAR-TRANSACTIONAL-DATA
  set dscp af21
 class CLASS-NBAR-BULK-DATA
  set dscp af11
 class CLASS-NBAR-SCAVENGER
  set dscp cs1
 class class-default
  set dscp default
!
interface GigabitEthernet0/3
 description LAN Interface
 service-policy input POLICY-INGRESS-LAN-MARKING
```

Egress QoS DSCP-Based Classification

Now that the network traffic has been marked upon receipt with the appropriate DSCP markings, an egress class map is created for the egress QoS policy. The first step for configuring a DSCP-based class map is to define the class map with the command **class-map** [**match-all** | **match-any**] *class-map-name*.

The DSCP values should then be defined with the command **match dscp** *value1 value2 … value8*. The command accepts up to eight DSCP values, which are separated by a white space. The command **match ip dscp** was not used so that this class map is protocol agnostic, matching IPv4 and IPv6 packets simultaneously.

Example 14-5 provides a sample configuration for classifying network traffic based upon a 12-class queueing model that matches the RFC 4594 policy definition.

Example 14-5 *Classification of Traffic for RFC 4594*

```
R11, R12, R21, R22, R31, R41, R51, and R52
class-map match-all CLASS-DSCP-VOICE
 match dscp ef
class-map match-all CLASS-DSCP-BROADCAST
 match dscp cs5
class-map match-all CLASS-DSCP-REALTIME
 match dscp cs4
class-map match-all CLASS-DSCP-MULTIMEDIA-CONF
 match dscp af41
class-map match-all CLASS-DSCP-MULTIMEDIA-STREAM
 match dscp af31
class-map match-all CLASS-DSCP-CONTROL
 match dscp cs6
class-map match-all CLASS-DSCP-SIGNALING
 match dscp cs3
class-map match-all CLASS-DSCP-OAM
 match dscp cs2
class-map match-all CLASS-DSCP-TRANSACTIONAL
 match dscp af21
class-map match-all CLASS-DSCP-BULK
 match dscp af11
class-map match-all CLASS-DSCP-SCAVENGER
 match dscp cs1
```

Egress QoS Policy Map

Now the egress policy map needs to be defined that will define within the class a queuing priority, bandwidth reservations, and remarking of the DMVPN header's DSCP. Service providers typically support only four to six class QoS models and remark (clear) the QoS markings on traffic that does not match their defined policies. Finding a service provider that supports a full 12-class QoS model can be a challenging task. If your SP does not support a 12-class QoS model, all egress network traffic must be reclassified before packets are sent to the service provider routers.

Example 14-6 provides a sample QoS policy map that correlates an enterprise's 12-class QoS policy with a service provider's six-class QoS policy. Notice that the queueing policy is defined to provide low-latency queueing for real-time traffic and appropriate bandwidth for classified applications. It is unnecessary to use the command **set dscp** *dscp* in all instances because the matched traffic already has the correct marking. Example 14-6 uses the command **set dscp** *dscp* in all classes for easy validation of how the traffic is marked when provided to the service provider's network.

Example 14-6 *12-Class QoS Policy Definition*

```
R31, R41, R51 and R52
policy-map POLICY-DSCP-12CLASS-TO-MPLS6CLASS
 class CLASS-DSCP-VOICE
  police cir percent 10
  priority level 1
  set dscp ef
 class CLASS-DSCP-REALTIME
  police cir percent 10
  priority level 2
  set dscp af41
 class CLASS-DSCP-BROADCAST
  police cir percent 10
  priority level 2
  set dscp af31
 class CLASS-DSCP-MULTIMEDIA-CONF
  bandwidth remaining percent 15
  random-detect dscp-based
  set dscp af41
 class CLASS-DSCP-MULTIMEDIA-STREAM
  bandwidth remaining percent 20
  random-detect dscp-based
  set dscp af31
 class CLASS-DSCP-CONTROL
  bandwidth remaining percent 5
  set dscp af41
 class CLASS-DSCP-SIGNALING
  bandwidth remaining percent 5
  set dscp af21
 class CLASS-DSCP-OAM
  bandwidth remaining percent 5
  set dscp af21
 class CLASS-DSCP-TRANSACTIONAL
  bandwidth remaining percent 19
  random-detect dscp-based
  set dscp af21
 class CLASS-DSCP-BULK
  bandwidth remaining percent 10
  random-detect dscp-based
  set dscp af21
 class CLASS-DSCP-SCAVENGER
  bandwidth remaining percent 1
  set dscp af11
 class class-default
  bandwidth remaining percent 25
  random-detect dscp-based
  set dscp default
```

Note Most of the branch network's WAN traffic is destined for the data centers that reside behind the DMVPN hub routers. Spoke-to-spoke network traffic typically consists of voice and video applications. Incorporating *Call Admission Control (CAC)* as part of the overall network QoS policy ensures that voice and video quality is maintained in the overall QoS policy.

Note Fair queueing is not possible with an implementation on the encrypted side of user flows because only a single flow exists between two routers. Configuration of **fair-queue** in a class creates four hash buckets. This limits the single encrypted flow to one of the hash buckets and reduces all traffic to one-fourth of the configured queue depth.

Hierarchical QoS

In the past, typical branch speeds were based on the available transport, T1 (1.5 Mbps) or E1 (2.0 Mbps), and incremented by multiples through the use of a Multilink Point-to-Point Protocol bundle. With the proliferation of Ethernet-based services and/or transports the support for subrate services has become prominent.

Service providers typically hand off broadband connectivity via an external modem for cable, DSL, or fiber-based services. The external modem provides a consistent Ethernet transport handoff to the customer devices. Regardless of the service provider's Ethernet handoff link speed (10 Mbps, 100 Mbps, 1 Gbps, or higher), it is possible to order service at a lower or intermediate speed. This ability to order a subrate service (bandwidth) allows for greater flexibility in transitioning to the next level of service, without the need for a technician to visit the branch site. Deploying a 5 Mbps service via a Fast or Gigabit Ethernet provider handoff can easily be achieved via a *hierarchical QoS (HQoS)* policy.

An HQoS policy is composed of two different policy maps that are linked with a *parent-child* relationship. The parent policy map contains only the **class-default** configuration with the desired subrate shaper and then associates a child policy map to it.

The parent policy provides the necessary back pressure for the child policy to provide appropriate prioritization and queueing of critical traffic. Within the child policy, network traffic is classified based on the desired requirements of the application traffic. Voice and video traffic is prioritized and defined with either strict bandwidth requirements or a percentage of available total bandwidth. Other classes of network traffic are allocated bandwidth and queueing to meet the desired application and user experience.

Example 14-7 demonstrates an HQoS policy using the already defined policy map POLICY-DSCP-12CLASS-TO-MPLS6CLASS. This HQoS policy shapes all network traffic down to 5 Mbps, and then the child policy allocates bandwidth to the classes in percentages of the parent policy's bandwidth.

Example 14-7 *Hierarchical Shaper Configuration*

```
R31
policy-map HQoS-POLICY-5MBPS
 class class-default
  shape average 5000000
   service-policy POLICY-DSCP-12CLASS-TO-MPLS6CLASS
```

The HQoS policy map is applied to the physical interface defined as the tunnel source interface as configured in Example 14-8.

Example 14-8 *Service Policy Physical Interface Application*

```
R31
interface GigabitEthernet0/1
 description MPLS01-TRANSPORT
 service-policy output HQoS-POLICY-5MBPS
```

Verification and validation of the QoS policy are done via the **show policy-map interface** *interface-name* **output** command. Example 14-9 displays the total traffic per the parent shaper, and queueing structure and traffic load per the child policy.

Example 14-9 *HQoS Policy Verification*

```
R31-Spoke# show policy-map interface GigabitEthernet0/1 output
 GigabitEthernet0/1

  Service-policy output: HQoS-POLICY-5MBPS

    Class-map: class-default (match-any)
      104396 packets, 17966335 bytes
      5 minute offered rate 225000 bps, drop rate 0000 bps
      Match: any
      Queueing
      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 104396/18315061
      shape (average) cir 5000000, bc 20000, be 20000
      target shape rate 5000000

      Service-policy : POLICY-DSCP-12CLASS-TO-MPLS6CLASS

        queue stats for all priority classes:
          Queueing
          priority level 1
```

```
   queue limit 64 packets
   (queue depth/total drops/no-buffer drops) 0/0/0
   (pkts output/bytes output) 70569/11741542

 queue stats for all priority classes:
   Queueing
   priority level 2
   queue limit 64 packets
   (queue depth/total drops/no-buffer drops) 0/0/0
   (pkts output/bytes output) 0/0

Class-map: CLASS-DSCP-VOICE (match-all)
   70569 packets, 11718162 bytes
   5 minute offered rate 151000 bps, drop rate 0000 bps
   Match:  dscp ef (46)
   police:
       cir 10 %
       cir 500000 bps, bc 15625 bytes
     conformed 70569 packets, 11741542 bytes; actions:
       transmit
     exceeded 0 packets, 0 bytes; actions:
       drop
     conformed 151000 bps, exceeded 0000 bps
   Priority: Strict, b/w exceed drops: 0

   Priority Level: 1
   QoS Set
     dscp ef
       Packets marked 70569

 Class-map: CLASS-DSCP-REALTIME (match-all)
   0 packets, 0 bytes
   5 minute offered rate 0000 bps, drop rate 0000 bps
   Match:  dscp cs4 (32)
   police:
       cir 10 %
       cir 500000 bps, bc 15625 bytes
     conformed 0 packets, 0 bytes; actions:
       transmit
     exceeded 0 packets, 0 bytes; actions:
       drop
     conformed 0000 bps, exceeded 0000 bps
   Priority: Strict, b/w exceed drops: 0
   Priority Level: 2
   QoS Set
     dscp af41
       Packets marked 0
Class-map: CLASS-DSCP-BROADCAST (match-all)
   0 packets, 0 bytes
   5 minute offered rate 0000 bps, drop rate 0000 bps
   Match:  dscp cs5 (40)
   police:
       cir 10 %
       cir 500000 bps, bc 15625 bytes
     conformed 0 packets, 0 bytes; actions:
       transmit
     exceeded 0 packets, 0 bytes; actions:
       drop
     conformed 0000 bps, exceeded 0000 bps
   Priority: Strict, b/w exceed drops: 0

   Priority Level: 2
   QoS Set
     dscp af31
       Packets marked 0

 Class-map: CLASS-DSCP-MULTIMEDIA-CONF (match-all)
   0 packets, 0 bytes
   5 minute offered rate 0000 bps, drop rate 0000 bps
   Match:  dscp af41 (34)
   Queueing
   queue limit 64 packets
   (queue depth/total drops/no-buffer drops) 0/0/0
   (pkts output/bytes output) 0/0
   bandwidth remaining 15%
     Exp-weight-constant: 9 (1/512)
     Mean queue depth: 0 packets
     dscp       Transmitted         Random drop      Tail drop
       Minimum        Maximum     Mark
             pkts/bytes            pkts/bytes       pkts/bytes
               thresh         thresh     prob

  QoS Set
     dscp af41
       Packets marked 0
```

```
Class-map: CLASS-DSCP-MULTIMEDIA-STREAM (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0000 bps, drop rate 0000 bps
  Match:  dscp af31 (26)
  Queueing
  queue limit 64 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 0/0
  bandwidth remaining 20%
    Exp-weight-constant: 9 (1/512)
    Mean queue depth: 0 packets
    dscp       Transmitted         Random drop      Tail drop
      Minimum        Maximum     Mark
            pkts/bytes            pkts/bytes       pkts/bytes
              thresh         thresh     prob

  QoS Set
    dscp af31
      Packets marked 0

Class-map: CLASS-DSCP-CONTROL (match-all)
  152 packets, 22148 bytes
  5 minute offered rate 0000 bps, drop rate 0000 bps
  Match:  dscp cs6 (48)
  Queueing
  queue limit 64 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 152/24080
  bandwidth remaining 5%
  QoS Set
    dscp af41
      Packets marked 152

Class-map: CLASS-DSCP-SIGNALING (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0000 bps, drop rate 0000 bps
  Match:  dscp cs3 (24)
  Queueing
  queue limit 64 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 0/0
  bandwidth remaining 5%
  QoS Set
    dscp af21
      Packets marked 0
```

```
 Class-map: CLASS-DSCP-OAM (match-all)
   0 packets, 0 bytes
   5 minute offered rate 0000 bps, drop rate 0000 bps
   Match:  dscp cs2 (16)
   Queueing
   queue limit 64 packets
   (queue depth/total drops/no-buffer drops) 0/0/0
   (pkts output/bytes output) 0/0
   bandwidth remaining 5%
   QoS Set
     dscp af21
       Packets marked 0

 Class-map: CLASS-DSCP-TRANSACTIONAL (match-all)
   10985 packets, 2150400 bytes
   5 minute offered rate 29000 bps, drop rate 0000 bps
   Match:  dscp af21 (18)
   Queueing
   queue limit 64 packets
   (queue depth/total drops/no-buffer drops) 0/0/0
   (pkts output/bytes output) 10985/2161894
   bandwidth remaining 19%
     Exp-weight-constant: 9 (1/512)
     Mean queue depth: 0 packets
     dscp       Transmitted         Random drop      Tail drop
       Minimum        Maximum     Mark
           pkts/bytes            pkts/bytes       pkts/bytes
             thresh         thresh     prob

     af21       10985/2161894         0/0              0/0
       32            40  1/10
   QoS Set
     dscp af21
       Packets marked 10985

Class-map: CLASS-DSCP-BULK (match-all)
   0 packets, 0 bytes
   5 minute offered rate 0000 bps, drop rate 0000 bps
   Match:  dscp af11 (10)
   Queueing
   queue limit 64 packets
   (queue depth/total drops/no-buffer drops) 0/0/0
   (pkts output/bytes output) 0/0
   bandwidth remaining 10%
```

```
    Exp-weight-constant: 9 (1/512)
    Mean queue depth: 0 packets
    dscp       Transmitted         Random drop      Tail drop
      Minimum        Maximum     Mark
            pkts/bytes            pkts/bytes       pkts/bytes
              thresh         thresh     prob

  QoS Set
    dscp af21
      Packets marked 0

Class-map: CLASS-DSCP-SCAVENGER (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0000 bps, drop rate 0000 bps
  Match:  dscp cs1 (8)
  Queueing
  queue limit 64 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 0/0
  bandwidth remaining 1%
  QoS Set
    dscp af11
      Packets marked 0

Class-map: class-default (match-any)
  22690 packets, 4075625 bytes
  5 minute offered rate 52000 bps, drop rate 0000 bps
  Match: any
  Queueing
  queue limit 64 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 22690/4387545
  bandwidth remaining 25%
    Exp-weight-constant: 9 (1/512)
    Mean queue depth: 0 packets
    dscp       Transmitted         Random drop      Tail drop
      Minimum        Maximum     Mark
            pkts/bytes            pkts/bytes       pkts/bytes
              thresh         thresh     prob

    default    22690/4387545         0/0              0/0
      20            40  1/10
  QoS Set
    dscp default
      Packets marked 22500
```

Note The rate displayed is calculated based on a 30-second window. This is modified from the default of 300 seconds (5 minutes) by configuring the interface parameter command **load-interval 30**. This configuration gives a more realistic visualization of current load, with an insignificant increase in CPU resources.

DMVPN Per-Tunnel QoS

Per-Tunnel QoS offers the ability to apply a specific hierarchical QoS policy on the DMVPN hub routers for all network traffic destined for a specific spoke (branch site). The hierarchical policy sets the parent shaper to the branch router's transport bandwidth at the remote site, and the child policy specifies the queuing behavior for the parent's policy-shaped bandwidth. In addition, a grandparent policy can be specified on the hub router's physical egress interface that the tunnel will traverse to support subrate interfaces. This grandparent policy can have only a class default specifying a shape rate.

Figure 14-1 helps demonstrate this concept. R11 (hub router) connects to the MPLS provider with a 300 Mbps circuit, R31 connects to the same provider with a 2 Mbps circuit, R41 connects with a 5 Mbps circuit, and R51 connects with a 100 Mbps circuit. R11's circuit contains enough capacity to talk to all the circuits at the same time but can easily congest all three sites because of its size. With the help of Per-Tunnel QoS, communication between R11 and R31 is shaped to 2 Mbps, and R11 and R51's communication is shaped to 100 Mbps.

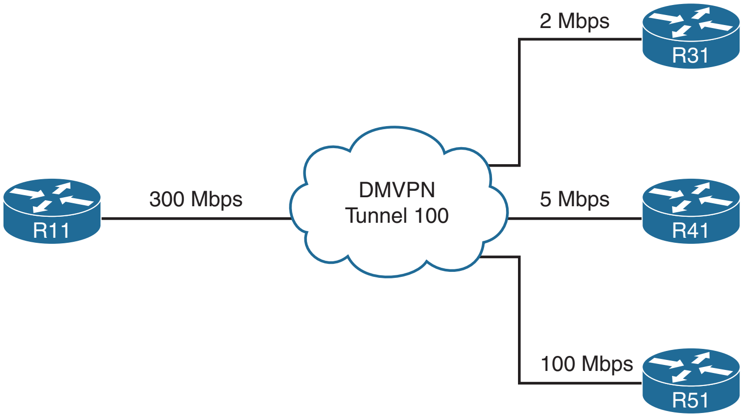


Figure 14-1 *MPLS Transports with Various Circuit Capacities*

Per-Tunnel QOS is based upon the following logic:

- All QoS policies are maintained on the DMVPN hub router. A group name is associated in the DMVPN tunnel interface with a specific HQoS policy map.

- The branch router configures its DMVPN tunnel interface with the group name that infers the desired policy associated on the DMVPN hub router. The branch router provides the group name to the hub router during the initial NHRP registration for the DMVPN tunnel. The spokes contain only the name of the group name, not the HQoS policy or the name of the HQoS policy map.
- The hub router receives the group name and then correlates that group name to the HQoS policy map, which is then used when communicating with that spoke.

Note Per-Tunnel QoS is applied from the hub toward the spoke. The spoke router uses a traditional HQoS policy as shown in Examples 14-6 to 14-8.

Per-Tunnel QoS Tunnel Markings

When defining a policy map for a DMVPN tunnel interface, the use of the command **set dscp** *dscp* is not a valid configuration option when deployed within the IWAN architecture. The policy map must use the command **set dscp tunnel** *dscp* in order to remark only the tunnel header while leaving the encapsulated traffic as originally marked. Specifying the tunnel QoS markings with the command **set tunnel dscp** *dscp* allows the hubs to have an end-to-end QoS model without having to reclassify traffic after it crosses the WAN. The command **qos pre-classify** is not used with the Per-Tunnel QoS policy because it is implied.

Note Using only the command **set dscp** *dscp* in a DMVPN Per-Tunnel QoS policy map leads to network traffic and PfR smart probes being remarked, so that PfR channels never establish bidirectionally.

With a per-tunnel policy map the same DSCP-based traffic classification model as shown in Example 14-6 is used. The DMVPN header's QoS markings have remarked the enterprise's 12-class QoS model to a six-class model for transport. In both scenarios, only the DMVPN header's QoS markings have been modified, not the original packet's QoS markings. The Per-Tunnel QoS policy map must use the **set dscp tunnel** command, not **set dscp** alone. Example 14-10 provides a sample policy map for a DMVPN tunnel interface.

Example 14-10 *DMVPN Per-Tunnel Policy Definition*

```
R11, R12, R21 and R22
policy-map POLICY-PER-TUNNEL-12CLASS-TO-6CLASS
 class CLASS-DSCP-VOICE
  police cir percent 10
  priority level 1
  set dscp tunnel ef
 class CLASS-DSCP-REALTIME
  police cir percent 10
  priority level 2
  set dscp tunnel af41
 class CLASS-DSCP-BROADCAST
  police cir percent 10
  priority level 2
  set dscp tunnel af31
 class CLASS-DSCP-MULTIMEDIA-CONF
  bandwidth remaining percent 15
  random-detect dscp-based
  set dscp tunnel af41
 class CLASS-DSCP-MULTIMEDIA-STREAM
  bandwidth remaining percent 20
  random-detect dscp-based
  set dscp tunnel af31
 class CLASS-DSCP-CONTROL
  bandwidth remaining percent 5
  set dscp tunnel af41
 class CLASS-DSCP-SIGNALING
  bandwidth remaining percent 5
  set dscp tunnel af21
 class CLASS-DSCP-OAM
  bandwidth remaining percent 5
  set dscp tunnel af21
 class CLASS-DSCP-TRANSACTIONAL
  bandwidth remaining percent 19
  random-detect dscp-based
  set dscp tunnel af21
 class CLASS-DSCP-BULK
  bandwidth remaining percent 10
  random-detect dscp-based
  set dscp tunnel af21
 class CLASS-DSCP-SCAVENGER
  bandwidth remaining percent 1
  set dscp tunnel af11
 class class-default
  bandwidth remaining percent 25
  random-detect dscp-based
  set dscp tunnel default
```

Note Per-Tunnel QoS queueing and shaping remain post-encryption. Therefore, IPsec overhead needs to be accounted for when designing a policy to support application patterns.

Bandwidth-Based QoS Policies

Hierarchical QoS is a key component of Per-Tunnel QoS. The hierarchical shaper restricts the amount of traffic sent from the DMVPN hub to the spoke routers. The hub can then provide back pressure to the network traffic's source in the data center. For example, assume that all the branch sites consume applications in the data center at Site 1. Back pressure on the hub router protects all branches from the bandwidth available in the data center (10 Gbps) which could exhaust the bandwidth on the spoke router.

A dedicated QoS policy is created for the bandwidth available at a site. If an organization provides only 2 Mbps or 5 Mbps of connectivity at a branch site, only two policies need to be created regardless of the number of branch sites deployed.

Example 14-11 provides sample Per-Tunnel HQoS policies for an IWAN environment that supports 2 Mbps, 5 Mbps, 10 Mbps, 20 Mbps, or 50 Mbps of bandwidth at the branch sites. It is important to note that the child policy uses a percentage-based policy, which should contain appropriate ratios for all sites. As bandwidth values change, the percentages may not be able to support the required policy at different levels. It may be necessary to have multiple child policies for the various HQoS policies to meet the real-world policy requirements.

Example 14-11 *Hierarchical Per-Tunnel Shaper*

```
R11, R12, R21 and R22
policy-map HQoS-POLICY-2MBPS
 class class-default
  shape average 2000000
   service-policy POLICY-PER-TUNNEL-12CLASS-TO-6CLASS
!
policy-map HQoS-POLICY-5MBPS
 class class-default
  shape average 5000000
   service-policy POLICY-PER-TUNNEL-12CLASS-TO-6CLASS
!
policy-map HQoS-POLICY-10MBPS
 class class-default
  shape average 10000000
   service-policy POLICY-PER-TUNNEL-12CLASS-TO-6CLASS
!
policy-map HQoS-POLICY-20MBPS
 class class-default
```

```
  shape average 20000000
   service-policy POLICY-PER-TUNNEL-12CLASS-TO-6CLASS
!
policy-map HQoS-POLICY-50MBPS
 class class-default
  shape average 50000000
   service-policy POLICY-PER-TUNNEL-12CLASS-TO-6CLASS
```

Bandwidth Remaining QoS Policies

It is import to understand how bandwidth is allocated in a QoS policy when using the command **bandwidth remaining percent** which is shown in Example 14-11. At times of network congestion on an interface (that is, DMVPN tunnel), traffic is allocated in the following order:

1. All priority classes are aggregated together with the physical interface, and priority queue traffic is always serviced first across any of the subinterfaces (per-tunnel policies for all branch routers).
2. Any bandwidth remaining is then allocated equally by default to a subinterface (per-tunnel policies for all branch routers).
3. Then traffic in the subinterface is allocated based on the child policy's configuration that uses the **bandwidth remaining percent** command.

Figure 14-2 illustrates this logic. Notice that R1's and R2's real-time network traffic is being serviced equally out of the priority queue Level 2. All other routers (R1–R10) are equally distributed out of the bandwidth remaining.

To understand this logic, imagine a topology with 10 remote branch routers (R1–R10) that are serviced by one hub router with a 100 Mbps circuit. All 10 branch sites have a 50 Mbps connection. If all 10 branch sites request 20 Mbps of traffic using only nonpriority traffic (such as FTP, SSH, web browsing), that would generate 200 Mbps of traffic, which exceeds the 100 Mbps total bandwidth available on the DMVPN hub.

The hub router would service the priority traffic first (there is none in this example). Because Per-Tunnel QoS has been deployed, each Per-Tunnel QoS policy to a branch router is considered to be a subinterface by the hub router. Using the second step of the preceding logic, the 100 Mbps circuit would be divided equally for every router with the formula as shown below:

Hub Bandwidth/Number of Sites = Average Site Bandwidth (BW)

Completing this formula, 100 Mbps/10 sites yields an *average branch site bandwidth* of 10 Mbps. Within each router's 10 Mbps of traffic, additional queuing would occur based on the type of network traffic and the classification in the child policy.

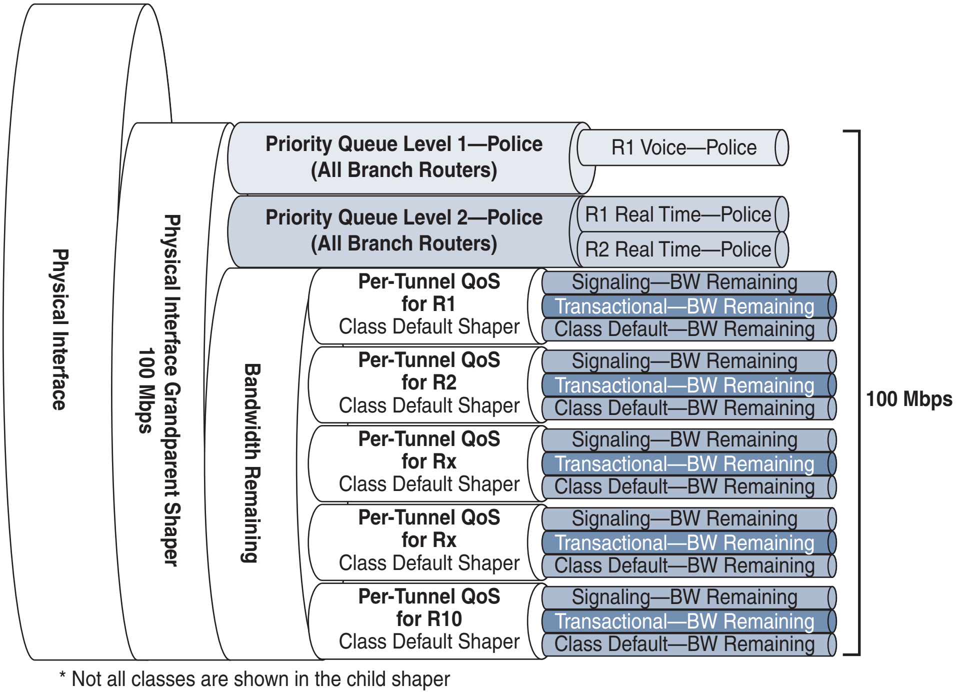


Figure 14-2 *Bandwidth Remaining Default Behavior*

Using an average branch site bandwidth across all site types gives smaller sites an advantage over larger sites. For simplicity, small branch sites are the sites with bandwidth that is less than the average branch site bandwidth, and large branch sites are the sites with bandwidth that is equal to or greater than that of the average branch site. A different set of formulas are required when small branch sites are intermixed with large branch sites. For small branch sites, the bandwidth allocated matches the shaper in the class default parent shaper. For the bandwidth allocated to large branch sites, the formula is shown in Figure 14-3.

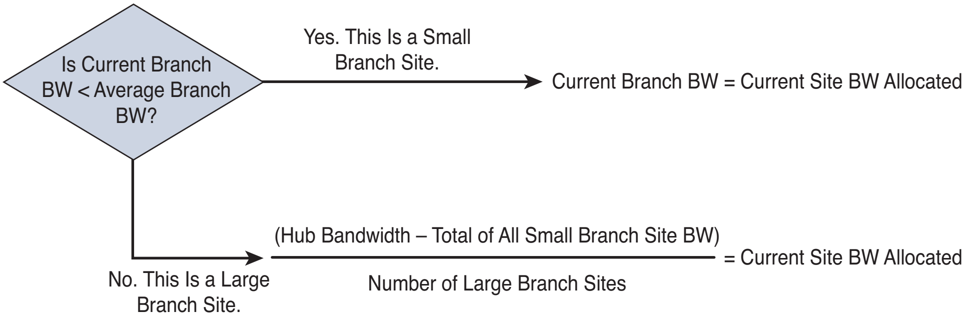
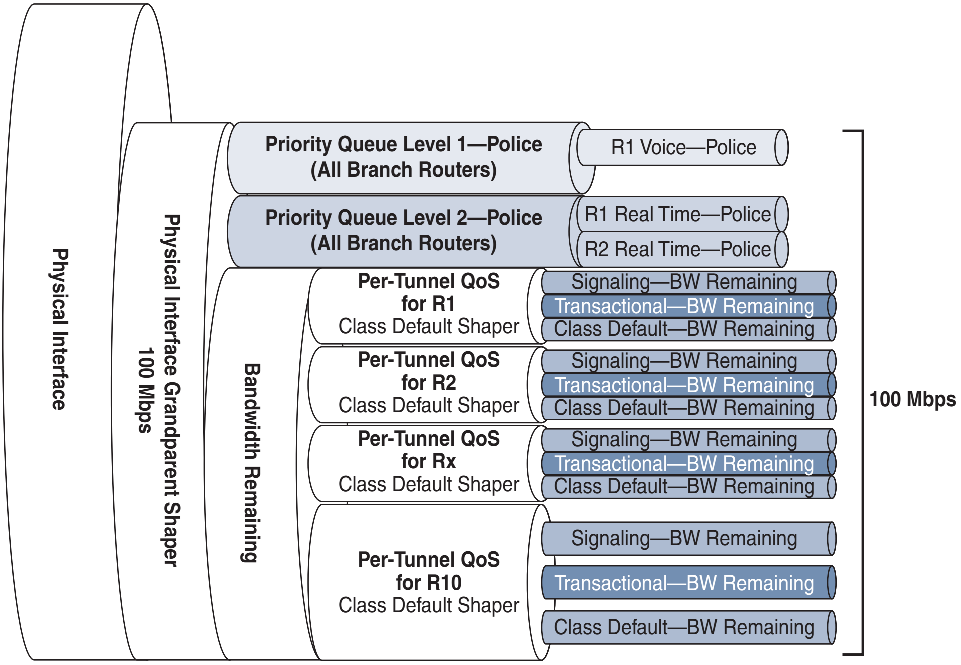

Is Current Branch
BW < Average Branch
BW?
Yes. This Is a Small
Branch Site.
Current Branch BW = Current Site BW Allocated
(Hub Bandwidth – Total of All Small Branch Site BW)
Number of Large Branch Sites
= Current Site BW Allocated
No. This Is a Large
Branch Site.

Figure 14-3 *Bandwidth Remaining Formula with Intermixed Bandwidth Sites*

For example, imagine that the first seven sites (R1–R7) have 2 Mbps circuits, and the last three sites have 50 Mbps circuits. If all 10 branch site requests exceed the 100 Mbps total bandwidth, the first seven sites (R1–R7) can accept only 2 Mbps of traffic because of circuit limitations. The seven 2 Mbps sites can be allocated a total of only 2 Mbps each, consuming a maximum of 14 Mbps of the 100 Mbps connection at the hub, thereby allowing 86 Mbps of traffic to be divided among the last three sites (R7–R10) that have 50 Mbps circuits. The average large branch site bandwidth is calculated according to the formula (100 Mbps – 14 Mbps)/3 large branch sites, which yields 28.6 Mbps for R7–R10.

Figure 14-4 represents the change in bandwidth allocation when smaller sites are mixed with larger sites. Notice that the large branch sites (R7–R10) have been allocated more bandwidth in their parent shaper out of the bandwidth remaining.


Physical Interface
Physical Interface Grandparent Shaper
100 Mbps
Bandwidth Remaining
Priority Queue Level 1—Police (All Branch Routers)
R1 Voice—Police
Priority Queue Level 2—Police (All Branch Routers)
R1 Real Time—Police
R2 Real Time—Police
Per-Tunnel QoS for R1 Class Default Shaper
Signaling—BW Remaining
Transactional—BW Remaining
Class Default—BW Remaining
Per-Tunnel QoS for R2 Class Default Shaper
Signaling—BW Remaining
Transactional—BW Remaining
Class Default—BW Remaining
Per-Tunnel QoS for Rx Class Default Shaper
Signaling—BW Remaining
Transactional—BW Remaining
Class Default—BW Remaining
Per-Tunnel QoS for R10 Class Default Shaper
Signaling—BW Remaining
Transactional—BW Remaining
Class Default—BW Remaining
100 Mbps

* Not all classes are shown in the child shaper

Figure 14-4 *Bandwidth Remaining Default Behavior with Ratios Set to Match Bandwidth*

This may not produce the desired effect. Because the physical interface is oversubscribed, providing a preferred ratio to sites may be required. By default, all sites are provided a ratio of one in the second step of the logic presented at the beginning of this section. On the IOS XE platforms (ASR 1000 and ISR 4000 Series), it is possible to change the distribution ratio to a value between 1 and 1000 with the command **bandwidth remaining ratio** [*1-1000*]. Calculating the bandwidth allocated to a router is based on the formula in Figure 14-5.

$$\frac{\text{Current Site BW Ratio}}{\text{Total of All Sites BW Ratio}} * \text{Hub Bandwidth} = \text{Current Site Bandwidth Allocated}$$

Figure 14-5 *Bandwidth Remaining Formula with Ratios*

With this new capability, a ratio can be associated to the bandwidth at a site, which provides a methodology for traffic being equally distributed on the hub router. Using the same scenario as before, the first seven sites (R1–R7) have 2 Mbps circuits, and the last three sites (R7–R10) have 50 Mbps circuits. The aggregate of all routers' bandwidth ratios equals 164 ((7 sites * 2 Mbps) + (3 sites * 50 Mbps) = 164).

Each of the small branch sites (R1–R7) is assigned 2/164 of the 100 Mbps, which equals 1.2 Mbps. The larger sites (R7–R10) are allocated a value of 50/164 of the 100Mbps, which equals 30.5 Mbps of traffic, providing a more consistent distribution of bandwidth, given 100 percent **bandwidth remaining percent** usage.

Example 14-12 provides sample QoS policies for an environment that supports 2 Mbps, 5 Mbps, 10 Mbps, 20 Mbps, or 50 Mbps of bandwidth at the branch sites. The configuration contains the enhanced command to match the given bandwidth in megabits per second.

Example 14-12 *Hierarchical Per-Tunnel Bandwidth Remaining Ratio*

```
R11, R12, R21 and R22
policy-map HQoS-POLICY-2MBPS
 class class-default
  bandwidth remaining ratio 2
  shape average 2000000
   service-policy POLICY-PER-TUNNEL-12CLASS-TO-6CLASS
!
policy-map HQoS-POLICY-5MBPS
 class class-default
  bandwidth remaining ratio 5
  shape average 5000000
   service-policy POLICY-PER-TUNNEL-12CLASS-TO-6CLASS
!
policy-map HQoS-POLICY-10MBPS
 class class-default
  bandwidth remaining ratio 10
  shape average 10000000
   service-policy POLICY-PER-TUNNEL-12CLASS-TO-6CLASS
!
policy-map HQoS-POLICY-20MBPS
 class class-default
  bandwidth remaining ratio 20
  shape average 20000000
   service-policy POLICY-PER-TUNNEL-12CLASS-TO-6CLASS
!
policy-map HQoS-POLICY-50MBPS
 class class-default
  bandwidth remaining ratio 50
  shape average 50000000
   service-policy POLICY-PER-TUNNEL-12CLASS-TO-6CLASS
```

Subrate Physical Interface QoS Policies

Assuming that the encapsulating DMVPN tunnel interface connects to a circuit that does not match that interface speed, only a class default shaping policy map should be configured to provide the necessary back pressure. With this policy map, the router now has a three-level QoS policy (physical interface grandparent shaper, per-tunnel parent shaper, and per-tunnel child policy). All priority classes are aggregated within the physical interface grandparent shaper ahead of all class-based queues.

Example 14-13 displays the configuration for shaping network traffic on the encapsulating interface for the MPLS hub routers. The same configuration should be applied to the Internet hub routers with the value for Internet connectivity

Example 14-13 *Per-Tunnel Hub Subrate Physical Interface Grandparent Shaper*

```
R11 and R21
policy-map POLICY-PHYSICAL-100MBPS
 class class-default
  shape average 100000000
!
interface GigabitEthernet0/1
 description MPLS01-TRANSPORT
 service-policy output POLICY-PHYSICAL-100MBPS
```

Note When applying a shaper to interfaces that use the Internet as a transport, it is important to realize that the circuit bandwidth may not match the throughput of the circuit. The designated circuit throughput may not always match the speed available and is covered in some SPs' disclaimers of SLAs. Configuring the Internet shaper at a value where traffic is consistently transmitted 100 percent of the time and meeting your application requirements is considered "goodput."

If every day your 20 Mbps circuit at 4:00 p.m. is only capable of transmitting 16 Mbps of traffic because of carrier contention, this will cause packet retransmission resulting in only 10 Mbps of application data being transferred. Configuring the Internet shaper to 16 Mbps minimizes retransmissions and provides full utilization of bandwidth for application goodput.

The configuration of the physical interface shaper must be completed prior to the configuration and registration of branches for Per-Tunnel QoS policy installation. Otherwise the configuration and installation of the service policy will not be accepted, as demonstrated in Example 14-14.

Example 14-14 *Error Message When Per-Tunnel QoS Has Already Been Applied*

```
R31(config)# interface GigabitEthernet0/1
R31(config-if)# service-policy output POLICY-PHYSICAL-100MBPS
Service_policy with queueing features on this interface is not allowed
if dmvpn based queuing policy is already installed.
```

Association of Per-Tunnel QoS Policies

As stated earlier, the policies reside on the DMVPN hub routers. All hierarchical QoS policies (such as Example 14-11) must be configured on the hub routers. The NHRP options must be defined on the tunnel interface so that the hub router can match the policy during NHRP registration. The policies are configured on a per-tunnel basis on the DMVPN hub routers, using the command **nhrp map group** *group-name* **service-policy output** *policy-map-name*. Example 14-15 demonstrates the configuration on R11.

Example 14-15 *Association of Group Name to HQoS Policy Map*

```
R11 and R21
interface Tunnel100
 nhrp map group 2MBPS service-policy output HQoS-POLICY-2MBPS
 nhrp map group 5MBPS service-policy output HQoS-POLICY-5MBPS
 nhrp map group 10MBPS service-policy output HQoS-POLICY-10MBPS
 nhrp map group 20MBPS service-policy output HQoS-POLICY-20MBPS
 nhrp map group 50MBPS service-policy output HQoS-POLICY-50MBPS
```

```
R12 and R22
interface Tunnel200
 nhrp map group 2MBPS service-policy output HQoS-POLICY-2MBPS
 nhrp map group 5MBPS service-policy output HQoS-POLICY-5MBPS
 nhrp map group 10MBPS service-policy output HQoS-POLICY-10MBPS
 nhrp map group 20MBPS service-policy output HQoS-POLICY-20MBPS
 nhrp map group 50MBPS service-policy output HQoS-POLICY-50MBPS
```

The branch router needs to be configured with the appropriate group name that is configured on the hub router that associates the policy map with the receivable bandwidth at the branch site. The group name is configured on the DMVPN tunnel with the command **nhrp group** *group-name*. In the event of asymmetric bandwidth at a site, the command **bandwidth receive** *bandwidth* should be configured on the router interface to provide appropriate load calculation.

Note Although it is recommended that the hub configuration be completed prior to branch configuration, this is not absolutely necessary. The NHRP group is part of each NHRP registration message and is provided during the registration within 200 seconds per the recommended NHRP holdtime of 600 seconds.

Example 14-16 demonstrates the Per-Tunnel QoS configuration on a branch.

Example 14-16 *Branch Per-Tunnel QoS Configuration*

```
R31, R41, and R51
interface Tunnel100
 description DMVPN-MPLS
 bandwidth 2000
 nhrp group 2MBPS
```

```
R31, R41, and R52
interface Tunnel200
 description DMVPN-INET
 bandwidth 5000
 bandwidth receive 20000
 nhrp group 20Mbps
```

Note The **nhrp group** and **nhrp map group** commands are protocol-agnostic replacements for the **ip nhrp group** and **ip nhrp map group** commands previously used. The **ip nhrp group** and **ip nhrp map group** commands have been made hidden commands so that they are still accepted for upgrade purposes. The first use of **nhrp group** or **nhrp map group** converts all **ip nhrp group** or **ip nhrp map group** configuration lines.

Per-Tunnel QoS Verification

Verification of registration is completed using the **show ip nhrp detail**, **show dmvpn detail**, or **show nhrp group-map** command, all of which display which branch is using which NHRP group. The command **show nhrp group-map** displays group-ID-to-NBMA address mapping. The command **show ip nhrp detail** displays NBMA address and next-hop address with the NHRP group ID, and the command **show dmvpn detail** displays everything about a branch to include the NHRP group ID and service policy applied. The **show dmvpn** and **show nhrp group-map** commands are protocol agnostic, displaying both IPv4 and IPv6 NBMA addressed branches. The **show { ip | ipv6 } nhrp** command requires that the transport network protocol used be specified as part of the command.

Example 14-17 displays output of the **show ip nhrp detail** command. Notice the highlighted group name.

Example 14-17 *Output of* **show ip nhrp detail**

```
R11# show ip nhrp detail
192.168.100.31/32 via 192.168.100.31
   Tunnel100 created 20:31:24, expire 00:08:36
   Type: dynamic, Flags: unique registered used nhop
   NBMA address: 172.16.31.1
   Preference: 255
   Group: 2MBPS
192.168.100.41/32 via 192.168.100.41
   Tunnel100 created 18:47:11, expire 00:09:29
   Type: dynamic, Flags: unique registered used nhop
   NBMA address: 172.16.41.1
   Preference: 255
   Group: 2MBPS
192.168.100.51/32 via 192.168.100.51
   Tunnel100 created 19:31:47, expire 00:08:13
   Type: dynamic, Flags: unique registered used nhop
   NBMA address: 172.16.51.1
   Preference: 255
   Group: 2MBPS
```

Example 14-18 displays output from the **show nhrp group-map** command. Notice that it shows which DMVPN spoke routers are associated to a specific NHRP group.

Example 14-18 *Output of* **show nhrp group-map**

```
R11# show nhrp group-map
Interface: Tunnel100
 NHRP group: 2MBPS
  QoS policy: HQoS-POLICY-2MBPS
  Transport endpoints using the qos policy:
  172.16.31.1
  172.16.41.1
  172.16.51.1

 NHRP group: 5MBPS
  QoS policy: HQoS-POLICY-5MBPS
  Transport endpoints using the qos policy: None

 NHRP group: 10MBPS
  QoS policy: HQoS-POLICY-10MBPS
  Transport endpoints using the qos policy: None
 NHRP group: 20MBPS
  QoS policy: HQoS-POLICY-20MBPS
  Transport endpoints using the qos policy: None

 NHRP group: 50MBPS
  QoS policy: HQoS-POLICY-50MBPS
  Transport endpoints using the qos policy: None
```

Example 14-19 displays output from the **show dmvpn detail** command. Notice that it shows the NHRP group name requested by the spoke routers and the associated QoS policy that is applied for that spoke.

Example 14-19 *Output of* **show dmvpn detail**

```
R11-Hub# show dmvpn detail
! Output omitted for brevity

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1 172.16.31.1      192.168.100.31   UP 20:33:50     D  192.168.100.31/32
NHRP group: 2MBPS
 Output QoS service-policy applied: HQoS-POLICY-2MBPS
    1 172.16.41.1      192.168.100.41   UP 18:50:07     D  192.168.100.41/32
NHRP group: 2MBPS
 Output QoS service-policy applied: HQoS-POLICY-2MBPS
    1 172.16.51.1      192.168.100.51   UP 19:34:43     D  192.168.100.51/32
NHRP group: 2MBPS
 Output QoS service-policy applied: HQoS-POLICY-2MBPS
```

Now that the NBMA address is known, and the association of a policy to that destination has been verified, the specific destination site policy can be reviewed. The command **show policy-map multipoint** *tunnel-interface nbma-address* **output** is used to review the installed Per-Tunnel QoS policy as shown in Example 14-20.

Example 14-20 *Output of* **show policy-map multipoint** *tunnel-interface nbma-address* **output**

```
R11-Hub# show policy-map multipoint tunnel100 172.16.41.1 output

Interface Tunnel100 <--> 172.16.41.1

  Service-policy output: HQoS-POLICY-2MBPS

    Class-map: class-default (match-any)
      546486 packets, 62524311 bytes
      30 second offered rate 47000 bps, drop rate 0000 bps
      Match: any
      Queueing
      queue limit 500 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 546486/96054212
      shape (average) cir 2000000, bc 8000, be 8000
      target shape rate 2000000

      Service-policy : POLICY-PER-TUNNEL-12CLASS-TO-6CLASS

        queue stats for all priority classes:
          Queueing
          priority level 1
          queue limit 50 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 340429/58316142

        queue stats for all priority classes:
          Queueing
          priority level 2
          queue limit 50 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 0/0

       Class-map: CLASS-DSCP-VOICE (match-all)
          340429 packets, 36754020 bytes
          30 second offered rate 29000 bps, drop rate 0000 bps
          Match:  dscp ef (46)
          police:
              cir 10 %
              cir 200000 bps, bc 17900 bytes
           conformed 340567 packets, 36771132 bytes; actions:
              transmit
            exceeded 0 packets, 0 bytes; actions:
              drop
            conformed 29000 bps, exceeded 0000 bps
          Priority: Strict, b/w exceed drops: 0

          Priority Level: 1
          QoS Set
            dscp tunnel ef
              Packets marked 340567

        Class-map: CLASS-DSCP-REALTIME (match-all)
          0 packets, 0 bytes
          30 second offered rate 0000 bps, drop rate 0000 bps
          Match:  dscp cs4 (32)
          police:
              cir 10 %
              cir 200000 bps, bc 17900 bytes
            conformed 0 packets, 0 bytes; actions:
              transmit
            exceeded 0 packets, 0 bytes; actions:
              drop
            conformed 0000 bps, exceeded 0000 bps
          Priority: Strict, b/w exceed drops: 0

          Priority Level: 2
          QoS Set
            dscp tunnel af41
              Packets marked 0

        Class-map: CLASS-DSCP-BROADCAST (match-all)
          0 packets, 0 bytes
          30 second offered rate 0000 bps, drop rate 0000 bps
          Match:  dscp cs5 (40)
          police:
              cir 10 %
              cir 200000 bps, bc 17900 bytes
            conformed 0 packets, 0 bytes; actions:
              transmit
            exceeded 0 packets, 0 bytes; actions:
              drop
            conformed 0000 bps, exceeded 0000 bps
          Priority: Strict, b/w exceed drops: 0
```

```
  Priority Level: 2
  QoS Set
    dscp tunnel af31
      Packets marked 0

Class-map: CLASS-DSCP-MULTIMEDIA-CONF (match-all)
  0 packets, 0 bytes
  30 second offered rate 0000 bps, drop rate 0000 bps
  Match:  dscp af41 (34)
  Queueing
  queue limit 49 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 0/0
  bandwidth remaining 15%
    Exp-weight-constant: 9 (1/512)
    Mean queue depth: 0 packets
    dscp       Transmitted         Random drop      Tail drop
      Minimum        Maximum     Mark
            pkts/bytes            pkts/bytes       pkts/bytes
              thresh         thresh     prob

  QoS Set
    dscp tunnel af41
      Packets marked 0

Class-map: CLASS-DSCP-MULTIMEDIA-STREAM (match-all)
  0 packets, 0 bytes
  30 second offered rate 0000 bps, drop rate 0000 bps
  Match:  dscp af31 (26)
  Queueing
  queue limit 49 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 0/0
  bandwidth remaining 20%
    Exp-weight-constant: 9 (1/512)
    Mean queue depth: 0 packets
    dscp       Transmitted         Random drop      Tail drop
      Minimum        Maximum     Mark
            pkts/bytes            pkts/bytes       pkts/bytes
              thresh         thresh     prob

  QoS Set
    dscp tunnel af31
      Packets marked 0
```

```
Class-map: CLASS-DSCP-CONTROL (match-all)
  3183 packets, 609662 bytes
  30 second offered rate 1000 bps, drop rate 0000 bps
  Match:  dscp cs6 (48)
  Queueing
  queue limit 17 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 3183/810874
  bandwidth remaining 5%
  QoS Set
    dscp tunnel af41
      Packets marked 3186

Class-map: CLASS-DSCP-SIGNALING (match-all)
  0 packets, 0 bytes
  30 second offered rate 0000 bps, drop rate 0000 bps
  Match:  dscp cs3 (24)
  Queueing
  queue limit 10 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 0/0
  bandwidth remaining 5%
  QoS Set
    dscp tunnel af21
      Packets marked 0

Class-map: CLASS-DSCP-OAM (match-all)
  0 packets, 0 bytes
  30 second offered rate 0000 bps, drop rate 0000 bps
  Match:  dscp cs2 (16)
  Queueing
  queue limit 14 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 0/0
  bandwidth remaining 5%
  QoS Set
    dscp tunnel af21
      Packets marked 0

Class-map: CLASS-DSCP-TRANSACTIONAL (match-all)
  0 packets, 0 bytes
  30 second offered rate 0000 bps, drop rate 0000 bps
  Match:  dscp af21 (18)
  Queueing
  queue limit 66 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 0/0
  bandwidth remaining 19%
    Exp-weight-constant: 9 (1/512)
    Mean queue depth: 0 packets
    dscp       Transmitted         Random drop      Tail drop
      Minimum        Maximum     Mark
            pkts/bytes            pkts/bytes       pkts/bytes
              thresh         thresh     prob

  QoS Set
    dscp tunnel af21
      Packets marked 0

Class-map: CLASS-DSCP-BULK (match-all)
  0 packets, 0 bytes
  30 second offered rate 0000 bps, drop rate 0000 bps
  Match:  dscp af11 (10)
  Queueing
  queue limit 17 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 0/0
  bandwidth remaining 10%
    Exp-weight-constant: 9 (1/512)
    Mean queue depth: 0 packets
    dscp       Transmitted         Random drop      Tail drop
      Minimum        Maximum     Mark
            pkts/bytes            pkts/bytes       pkts/bytes
              thresh         thresh     prob

  QoS Set
    dscp tunnel af21
      Packets marked 0

Class-map: CLASS-DSCP-SCAVENGER (match-all)
  0 packets, 0 bytes
  30 second offered rate 0000 bps, drop rate 0000 bps
  Match:  dscp cs1 (8)
  Queueing
  queue limit 4 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 0/0
  bandwidth remaining 1%
```

```
      QoS Set
        dscp tunnel af11
          Packets marked 0

    Class-map: class-default (match-any)
      202874 packets, 25160629 bytes
      30 second offered rate 18000 bps, drop rate 0000 bps
      Match: any
      Queueing
      queue limit 122 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 202874/36927196
      bandwidth remaining 25%
        Exp-weight-constant: 9 (1/512)
        Mean queue depth: 0 packets
        dscp       Transmitted         Random drop      Tail drop
          Minimum        Maximum     Mark
                pkts/bytes            pkts/bytes       pkts/bytes
                  thresh         thresh     prob

        default   202921/36931622        0/0              0/0
          20            40  1/10
        cs6           91/20690           0/0              0/0
          32            40  1/10
      QoS Set
        dscp tunnel default
          Packets marked 203012
```

Note Per-Tunnel QoS provides visibility for oversubscribed or dropped traffic toward a specific branch site. Typically dropped packets occur when the hub sends traffic at a higher rate than the rate at which the branch router is provisioned to receive from the branch router's circuit to the SP network.

The QoS policy for a branch site is examined with two different commands. Egress traffic is viewed with the command **show policy-map interface** *interface-id* **output** where the interface specified is the encapsulating interface with the HQoS policy. Ingress traffic from a hub router is examined with the command **show policy-map multipoint** *tunnel-interface nbma-address* **output.**

Per-Tunnel QoS Caveats

Because the QoS policy is not directly configured on an interface, it is possible to configure a policy with options assigned that do not present a valid configuration. If any of the **show** commands do not present a policy being assigned, or a syslog message is received (every registration timeout of 200 seconds) stating that a policy has not been associated, proper debugging is required to determine the reason. The following tips are helpful for identifying the reason:

- Examine the output of **show dmvpn detail** on the hub routers. Verify that the NHRP group names that are being requested are configured and associated to a QoS policy. If the "*Output QoS service-policy applied: none*" message is displayed, the group name requested by the spoke router does not exist on the hub router. NHRP group names are case sensitive.
- At the time of this writing, Per-Tunnel QoS is not supported on DMVPN encapsulating interfaces that are a port channel or scenarios with ECMP routes in the FVRF to the branch sites. The latter scenario involves the use of terminating DMVPN tunnels on loopback interfaces.
- Per-Tunnel QoS is not supported on the hub routers when the encapsulating interface already has an HQoS policy associated to it.
- The command **debug tunnel qos** shows the policy installation and reason for failure.
- The **debug nhrp group** command shows what group is being registered to verify that the branch is properly configured to match the hub configuration.

In Example 14-21, the configured policy is specifically broken by specifying a custom sustained bit rate (BC) and excess burst (BE) bits per interval values. The configuration of these values is not recommended per the configuration documentation; they are used here only for demonstration purposes.

Example 14-21 *Per-Tunnel Branch Registration*

```
%NHRP-3-QOS_POLICY_APPLY_FAILED: Failed to apply QoS policy HQoS-POLICY-10MBPS-
  broken mapped to NHRP group broken on interface Tunnel100, to tunnel 172.16.41.1
  due to policy installation failure


R11-Hub# debug tunnel qos
Tunnel-QoS: qos info attached to adj of Tunnel100 o_addr: 192.168.100.41
Tunnel-QoS: returning out-phy-idb GigabitEthernet0/1 if_number 4 to hqf Configured
  shape BC value results in the interval being out of range.
 Allowed BC value for given CIR of 1000000 should be at least 4000.
Tunnel-QoS: returning out-phy-idb GigabitEthernet0/1 if_number 4 to hqf
Tunnel-QoS-TARGET: Failed to activate HQoS-POLICY-10MBPS-broken to Tunnel100 addr =
  172.16.41.1
Tunnel-QoS: failed to attach qos policy HQoS-POLICY-10MBPS-broken to tunnel
  172.16.41.1
Tunnel-QoS: qos info detached from adjacency of Tunnel100 addr: 192.168.100.41


R11-Hub# debug nhrp group
NHRP-GROUP: rcvd group name: old 'broken' new 'broken'
NHRP-GROUP: failed to apply qos policy 'HQoS-POLICY-10MBPS-broken' to tunnel
  172.16.41.1, error 5
```

Another issue that may be seen is the logging of "*%QOS-4-QLIMIT_HQUEUE _VALUE_SYNC_ISSUE*" as more Per-Tunnel QoS sessions are established. This is an informational message that can be safely ignored and will be corrected in an upcoming release. Physical interface hold queue is not relevant for Per-Tunnel QoS–based policy configuration.

QoS and IPSec Packet Replay Protection

IPsec tunnel protection includes an anti-replay mechanism that protects a router from various types of intrusions on encrypted conversations. The anti-replay window is a method that provides a form of partial sequence integrity, per RFC 2401, and was explained in Chapter 5, "Securing DMVPN Tunnels and Routers."

The Cisco implementation for the anti-replay window defaults to a 64-packet window of where packets belong. On high-speed networks configured with QoS (especially those with priority queues, where packets may be reordered or dropped after being encrypted), the default window size is not large enough to support the application profile.

Although the anti-replay sequence number implementation is mandatory, it is valuable only if the receiver does not decide to disable the check. In order to validate the anti-replay check and provide for QoS, the ability to increase the anti-replay window size is paramount to proper operation.

The configuration of the anti-replay window is accomplished with the command **crypto ipsec security-association replay window-size** {**64** | **128** | **256** | **512** | **1024**}. Without maximizing the anti-replay window, the router displays syslog messages such as "*%CRYPTO-4-PKT_REPLAY_ERR: decrypt: replay check failed connection id=#, sequence number=#,*." Application performance suffers because of the dropped packets, which were processed by QoS on the sending side and then dropped on the receiving side because of the QoS policy. Configuring the maximum anti-replay window size supported is recommended for optimal application performance.

With IWAN the utilization of IPsec tunnel protection over any unsecured transport is mandatory. Deploying IPsec tunnel protection everywhere simplifies deployment and operational support by providing a consistent environment that makes every transport look exactly the same. This translates to overhead on a per-packet basis, which is easy to account for with large TCP flows but may seem excessive with small UDP/RTP voice flows. QoS prioritization and queueing occur post-IPsec encapsulation, therefore requiring that the IPsec overhead be accounted for when specifying the amount of bandwidth reserved for specific classes.

A simple example would be a g.711 call that consumes 80 Kbps, 64 Kbps payload and 16 Kbps header. This g.711 voice call has to account for the IPsec overhead when deploying a priority queue. The overhead is on a per-packet basis and was discussed in Chapter 3, "Dynamic Multipoint VPN." G.711 utilizes a 20 ms payload rate at 50 pps. With 160 bytes of voice payload, 40 bytes of IP/UDP/RTP packet overhead, 64 bytes of IPsec overhead using ESP-GCM 256, we have added 32 percent overhead to the g.711

packet, going from 80 Kbps per call to 106 Kbps per call with IPsec overhead. Additional information can be found in the Cisco “IPsec Overhead Calculator,” https://cway.cisco.com/tools/ipsec-overhead-calc/ipsec-overhead-calc.html.

Complete QoS Configuration

This section provides a sample of the complete QoS configuration for a DMVPN hub and spoke routers. Example 14-22 has configurations for a DMVPN hub (R11) and DMVPN spoke (R31). In both configurations, it is assumed that network traffic is properly marked earlier in the network. If needed, a policy map can be added on the LAN-facing interfaces to classify and remark QoS traffic.

Example 14-22 *Complete QoS Configuration for R11 and R31*

```
R11-Hub
! Ingress Classification
class-map match-all CLASS-NBAR-VOICE
   match protocol attribute traffic-class voip-telephony
   match protocol attribute business-relevance business-relevant
class-map match-all CLASS-NBAR-BROADCAST-VIDEO
   match protocol attribute traffic-class broadcast-video
   match protocol attribute business-relevance business-relevant
class-map match-all CLASS-NBAR-REAL-TIME-INTERACTIVE
   match protocol attribute traffic-class real-time-interactive
   match protocol attribute business-relevance business-relevant
class-map match-all CLASS-NBAR-MULTIMEDIA-CONFERENCING
   match protocol attribute traffic-class multimedia-conferencing
   match protocol attribute business-relevance business-relevant
class-map match-all CLASS-NBAR-MULTIMEDIA-STREAMING
   match protocol attribute traffic-class multimedia-streaming
   match protocol attribute business-relevance business-relevant
 class-map match-all CLASS-NBAR-SIGNALING
   match protocol attribute traffic-class signaling
   match protocol attribute business-relevance business-relevant
class-map match-all CLASS-NBAR-NETWORK-CONTROL
   match protocol attribute traffic-class network-control
   match protocol attribute business-relevance business-relevant
class-map match-all CLASS-NBAR-NETWORK-MANAGEMENT
   match protocol attribute traffic-class ops-admin-mgmt
   match protocol attribute business-relevance business-relevant
class-map match-all CLASS-NBAR-TRANSACTIONAL-DATA
   match protocol attribute traffic-class transactional-data
   match protocol attribute business-relevance business-relevant
class-map match-all CLASS-NBAR-BULK-DATA
   match protocol attribute traffic-class bulk-data
   match protocol attribute business-relevance business-relevant
```

```
class-map match-all CLASS-NBAR-SCAVENGER
   match protocol attribute business-relevance business-irrelevant
!
policy-map POLICY-INGRESS-LAN-MARKING
 class CLASS-NBAR-VOICE
  set dscp ef
 class CLASS-NBAR-BROADCAST-VIDEO
  set dscp cs5
 class CLASS-NBAR-REAL-TIME-INTERACTIVE
  set dscp cs4
 class CLASS-NBAR-MULTIMEDIA-CONFERENCING
  set dscp af41
 class CLASS-NBAR-MULTIMEDIA-STREAMING
  set dscp af31
 class CLASS-NBAR-SIGNALING
  set dscp cs3
 class CLASS-NBAR-NETWORK-CONTROL
  set dscp cs6
 class CLASS-NBAR-NETWORK-MANAGEMENT
  set dscp cs2
 class CLASS-NBAR-TRANSACTIONAL-DATA
  set dscp af21
 class CLASS-NBAR-BULK-DATA
  set dscp af11
 class CLASS-NBAR-SCAVENGER
  set dscp cs1
 class class-default
  set dscp default
!
interface GigabitEthernet0/3
 description LAN Interface
 service-policy input POLICY-INGRESS-LAN-MARKING
! Egress Shaping Policy
class-map match-all CLASS-DSCP-TRANSACTIONAL
 match dscp af21
class-map match-all CLASS-DSCP-OAM
 match dscp cs2
class-map match-all CLASS-DSCP-CONTROL
 match dscp cs6
class-map match-all CLASS-DSCP-SIGNALING
 match dscp cs3
class-map match-all CLASS-DSCP-BROADCAST
 match dscp cs5
```

```
class-map match-all CLASS-DSCP-SCAVENGER
 match dscp cs1
class-map match-all CLASS-DSCP-VOICE
 match dscp ef
class-map match-all CLASS-DSCP-REALTIME
 match dscp cs4
class-map match-all CLASS-DSCP-BULK
 match dscp af11
class-map match-all CLASS-DSCP-MULTIMEDIA-CONF
 match dscp af41
class-map match-all CLASS-DSCP-MULTIMEDIA-STREAM
 match dscp af31
!
policy-map POLICY-PER-TUNNEL-12CLASS-TO-6CLASS
 class CLASS-DSCP-VOICE
  police cir percent 10
  priority level 1
  set dscp tunnel ef
 class CLASS-DSCP-REALTIME
  police cir percent 10
  priority level 2
  set dscp tunnel af41
 class CLASS-DSCP-BROADCAST
  police cir percent 10
  priority level 2
  set dscp tunnel af31
 class CLASS-DSCP-MULTIMEDIA-CONF
  bandwidth remaining percent 15
  random-detect dscp-based
  set dscp tunnel af41
 class CLASS-DSCP-MULTIMEDIA-STREAM
  bandwidth remaining percent 20
  random-detect dscp-based
  set dscp tunnel af31
 class CLASS-DSCP-CONTROL
  bandwidth remaining percent 5
  set dscp tunnel af41
 class CLASS-DSCP-SIGNALING
  bandwidth remaining percent 5
  set dscp tunnel af21
class CLASS-DSCP-OAM
  bandwidth remaining percent 5
  set dscp tunnel af21
 class CLASS-DSCP-TRANSACTIONAL
  bandwidth remaining percent 19
  random-detect dscp-based
  set dscp tunnel af21
 class CLASS-DSCP-BULK
  bandwidth remaining percent 10
  random-detect dscp-based
  set dscp tunnel af21
 class CLASS-DSCP-SCAVENGER
  bandwidth remaining percent 1
  set dscp tunnel af11
 class class-default
  bandwidth remaining percent 25
  random-detect dscp-based
  set dscp tunnel default
!
policy-map HQoS-POLICY-5MBPS
 class class-default
  shape average 5000000
   service-policy POLICY-PER-TUNNEL-12CLASS-TO-6CLASS
!
policy-map HQoS-POLICY-2MBPS
 class class-default
  shape average 2000000
   service-policy POLICY-PER-TUNNEL-12CLASS-TO-6CLASS
!
policy-map HQoS-POLICY-50MBPS
 class class-default
  shape average 50000000
   service-policy POLICY-PER-TUNNEL-12CLASS-TO-6CLASS
!
policy-map HQoS-POLICY-20MBPS
 class class-default
  shape average 20000000
   service-policy POLICY-PER-TUNNEL-12CLASS-TO-6CLASS
!
policy-map HQoS-POLICY-10MBPS
 class class-default
  shape average 10000000
   service-policy POLICY-PER-TUNNEL-12CLASS-TO-6CLASS
!
policy-map POLICY-PHYSICAL-100MBPS
 class class-default
  shape average 100000000
```

```
!
interface Tunnel100
 load-interval 30
 nhrp map group 2MBPS service-policy output HQoS-POLICY-2MBPS
 nhrp map group 5MBPS service-policy output HQoS-POLICY-5MBPS
 nhrp map group 10MBPS service-policy output HQoS-POLICY-10MBPS
 nhrp map group 20MBPS service-policy output HQoS-POLICY-20MBPS
 nhrp map group 50MBPS service-policy output HQoS-POLICY-50MBPS
!
interface GigabitEthernet0/1
 service-policy output POLICY-PHYSICAL-100MBPS
```

R31-Spoke

```
! Ingress Classification
class-map match-all CLASS-NBAR-VOICE
   match protocol attribute traffic-class voip-telephony
   match protocol attribute business-relevance business-relevant
class-map match-all CLASS-NBAR-BROADCAST-VIDEO
   match protocol attribute traffic-class broadcast-video
   match protocol attribute business-relevance business-relevant
class-map match-all CLASS-NBAR-REAL-TIME-INTERACTIVE
   match protocol attribute traffic-class real-time-interactive
   match protocol attribute business-relevance business-relevant
class-map match-all CLASS-NBAR-MULTIMEDIA-CONFERENCING
   match protocol attribute traffic-class multimedia-conferencing
   match protocol attribute business-relevance business-relevant
class-map match-all CLASS-NBAR-MULTIMEDIA-STREAMING
   match protocol attribute traffic-class multimedia-streaming
   match protocol attribute business-relevance business-relevant
 class-map match-all CLASS-NBAR-SIGNALING
   match protocol attribute traffic-class signaling
   match protocol attribute business-relevance business-relevant
class-map match-all CLASS-NBAR-NETWORK-CONTROL
   match protocol attribute traffic-class network-control
   match protocol attribute business-relevance business-relevant
class-map match-all CLASS-NBAR-NETWORK-MANAGEMENT
   match protocol attribute traffic-class ops-admin-mgmt
   match protocol attribute business-relevance business-relevant
class-map match-all CLASS-NBAR-TRANSACTIONAL-DATA
   match protocol attribute traffic-class transactional-data
   match protocol attribute business-relevance business-relevant
class-map match-all CLASS-NBAR-BULK-DATA
   match protocol attribute traffic-class bulk-data
   match protocol attribute business-relevance business-relevant
class-map match-all CLASS-NBAR-SCAVENGER
   match protocol attribute business-relevance business-irrelevant
```

```
!
policy-map POLICY-INGRESS-LAN-MARKING
 class CLASS-NBAR-VOICE
  set dscp ef
 class CLASS-NBAR-BROADCAST-VIDEO
  set dscp cs5
 class CLASS-NBAR-REAL-TIME-INTERACTIVE
  set dscp cs4
 class CLASS-NBAR-MULTIMEDIA-CONFERENCING
  set dscp af41
 class CLASS-NBAR-MULTIMEDIA-STREAMING
  set dscp af31
 class CLASS-NBAR-SIGNALING
  set dscp cs3
 class CLASS-NBAR-NETWORK-CONTROL
  set dscp cs6
 class CLASS-NBAR-NETWORK-MANAGEMENT
  set dscp cs2
 class CLASS-NBAR-TRANSACTIONAL-DATA
  set dscp af21
 class CLASS-NBAR-BULK-DATA
  set dscp af11
 class CLASS-NBAR-SCAVENGER
  set dscp cs1
 class class-default
  set dscp default
!
interface GigabitEthernet0/3
 description LAN Interface
 service-policy input POLICY-INGRESS-LAN-MARKING
!
! Egress Shaping Policy
class-map match-all CLASS-DSCP-TRANSACTIONAL
 match dscp af21
class-map match-all CLASS-DSCP-OAM
 match dscp cs2
class-map match-all CLASS-DSCP-CONTROL
 match dscp cs6
class-map match-all CLASS-DSCP-SIGNALING
 match dscp cs3
class-map match-all CLASS-DSCP-BROADCAST
 match dscp cs5
class-map match-all CLASS-DSCP-SCAVENGER
 match dscp cs1
```

```
class-map match-all CLASS-DSCP-VOICE
 match dscp ef
class-map match-all CLASS-DSCP-REALTIME
 match dscp cs4
class-map match-all CLASS-DSCP-BULK
 match dscp af11
class-map match-all CLASS-DSCP-MULTIMEDIA-CONF
 match dscp af41
class-map match-all CLASS-DSCP-MULTIMEDIA-STREAM
 match dscp af31
!
policy-map POLICY-DSCP-12CLASS-TO-MPLS6CLASS
 class CLASS-DSCP-VOICE
  police cir percent 10
  priority level 1
  set dscp ef
 class CLASS-DSCP-REALTIME
  police cir percent 10
  priority level 2
  set dscp af41
 class CLASS-DSCP-BROADCAST
  police cir percent 10
  priority level 2
  set dscp af31
 class CLASS-DSCP-MULTIMEDIA-CONF
  bandwidth remaining percent 15
  random-detect dscp-based
  set dscp af41
 class CLASS-DSCP-MULTIMEDIA-STREAM
  bandwidth remaining percent 20
  random-detect dscp-based
  set dscp af31
 class CLASS-DSCP-CONTROL
  bandwidth remaining percent 5
  set dscp af41
 class CLASS-DSCP-SIGNALING
  bandwidth remaining percent 5
  set dscp af21
class CLASS-DSCP-OAM
  bandwidth remaining percent 5
  set dscp af21
 class CLASS-DSCP-TRANSACTIONAL
  bandwidth remaining percent 19
  random-detect dscp-based
  set dscp af21
 class CLASS-DSCP-BULK
  bandwidth remaining percent 10
  random-detect dscp-based
  set dscp af21
 class CLASS-DSCP-SCAVENGER
  bandwidth remaining percent 1
  set dscp af11
 class class-default
  bandwidth remaining percent 25
  random-detect dscp-based
  set dscp default
!
policy-map HQoS-POLICY-5MBPS
 class class-default
  shape average 5000000
   service-policy POLICY-DSCP-12CLASS-TO-MPLS6CLASS
policy-map HQoS-POLICY-2MBPS
 class class-default
  shape average 2000000
   service-policy POLICY-DSCP-12CLASS-TO-MPLS6CLASS
!
interface Tunnel100
 load-interval 30
 nhrp group 2MBPS
!
interface Tunnel200
 load-interval 30
 nhrp group 20MBPS
!
interface GigabitEthernet0/1
 description MPLS01-TRANSPORT
 service-policy output HQoS-POLICY-2MBPS
!
interface GigabitEthernet0/2
 description INET01-TRANSPORT
 service-policy output HQoS-POLICY-5MBPS
```

Note The provided configurations do not include the **bandwidth remaining ratio** [*1-1000*] command. The command needs to be added if that is a portion of your QoS strategy.

Summary

QoS is an integral part of the IWAN solution, providing appropriate application responsiveness and experience across the WAN. The IWAN architecture is based upon two separate processes:

- Inbound network traffic classification
- Egress QoS queueing and marking of DMVPN tunnels

Branch routers apply an HQoS policy on the encapsulating interface that shapes traffic based on the DMVPN QoS markings. The policy can remark the tunnel's QoS markings to match the service provider's QoS markings, but the original packet's QoS markings are not modified. Hub routers use Per-Tunnel QoS to provide additional support for specifying the rate at which traffic can be sent to a specific site, protecting availability for all sites, and providing back pressure to appropriate sources of network traffic.

Appropriate interaction between QoS and IPsec must be understood, while still meeting security mandates and controls. Today the IWAN solution is rigidly controlled to provide a very prescriptive solution that is both easily deployed and highly serviceable. As new features are implemented and released, they will be integrated into the IWAN solution for enhancement of the end-user experience.

Per-Tunnel QoS policies can be modified easily on the hub routers as time goes on. Data modeling and analysis can be performed based on the information collected with Performance Monitor and NetFlow as explained in Chapter 10, "Application Visibility." Quality of service is a key to maintaining critical application responsiveness and appropriate user experience.

Further Reading

Babiarz, J., K. Chan, and F. Baker. RFC 4594, "Configuration Guidelines for DiffServ Service Classes." IETF, August 2006. https://tools.ietf.org/html/rfc4594.

Fleshner, Kelly. "IWAN AVC/QoS Design." Presented at Cisco Live, Berlin, 2016.

Hanks, S., T. Li, D. Farinacci, and P. Traina. RFC 1702, "Generic Routing Encapsulation over IPv4 Networks." IETF, October 2004. http://tools.ietf.org/html/rfc1702.

Kent, S., and R. Atkinson. RFC 2401, "Security Architecture for the Internet Protocol." IETF, November 1998. https://tools.ietf.org/html/rfc2401.

This page intentionally left blank

Chapter 15

Direct Internet Access (DIA)

This chapter covers the following topics:

- Direct Internet access
- Guest access
- Internal access
- Cloud web security

Internet access is an essential component of any network deployment. Employees' productivity may be affected adversely if they have unfettered Internet access, or productivity may flourish because of employee happiness. Either way, Internet access has become a requirement for the workplace.

The Internet has become an integral part of the business environment through sales, marketing, and applications. Internet connectivity has expanded from corporate environments into other markets such as retail. Providing guest Internet access attracts customers and can increase sales. As more applications move to the cloud, providing fast and responsive Internet access is essential for business productivity.

Previously, all Internet access was provided in a centralized manner, because branch locations did not have an Internet connection. Branch locations that did have Internet connectivity used the Internet for a point-to-point IPsec tunnel that acted as a backup path if the primary MPLS transport was down. Locations that had Internet connectivity but used it strictly as a backup transport wasted money because that bandwidth could be consumed. The available Internet bandwidth could offload both best-effort and critical cloud-based applications.

Providing direct Internet access to branch sites presents many obstacles to a typical branch network deployment. Centralized Internet access provides a minimal number of touchpoints to the outside world, whereas distributed Internet access provides the lowest latency and the most bandwidth to Internet-based applications.

The centralized Internet access model provides the fewest control points in the network. Bringing all Internet access back to this centralized point in the network places more stress on this infrastructure. In this model, Internet traffic is backhauled across the WAN (using the private MPLS or Internet VPN tunnels), requiring sufficient bandwidth for Internet traffic and internal traffic. This does not account for the actual Internet circuit's bandwidth at the central site.

Internet connectivity has a much lower cost than other transport choices, and organizations now pay twice for Internet bandwidth, once for the branch to reach the security infrastructure, and then again to access the Internet itself. This does not take into consideration the return of all this traffic to the branch and the additional latency.

For example, users on the West Coast of the United States (for example, Los Angeles, California) frequently access local online resources such as news websites, weather reports, or restaurant menus to order lunch for delivery. Their company uses a centralized Internet access model, so Internet connectivity for this branch location is provided via a data center on the East Coast (Ashburn, Virginia). The additional latency for these services is negligible, but it is always present for all Internet traffic. As more users, services, and applications use the Internet, the latency and additional bandwidth requirements have a demonstrable effect on productivity.

The distributed Internet access model provides the best access to the applications by using Internet access locally at the branch site to reach those applications. This does distribute the number of touchpoints to every branch. Being able to maintain a centralized policy with a distributed Internet access model is critical.

Figure 15-1 displays the centralized and distributed Internet access models. It is important to note that all traffic between the branch sites (R31 and R41) to R12 is transmitted across the DMVPN tunnels.

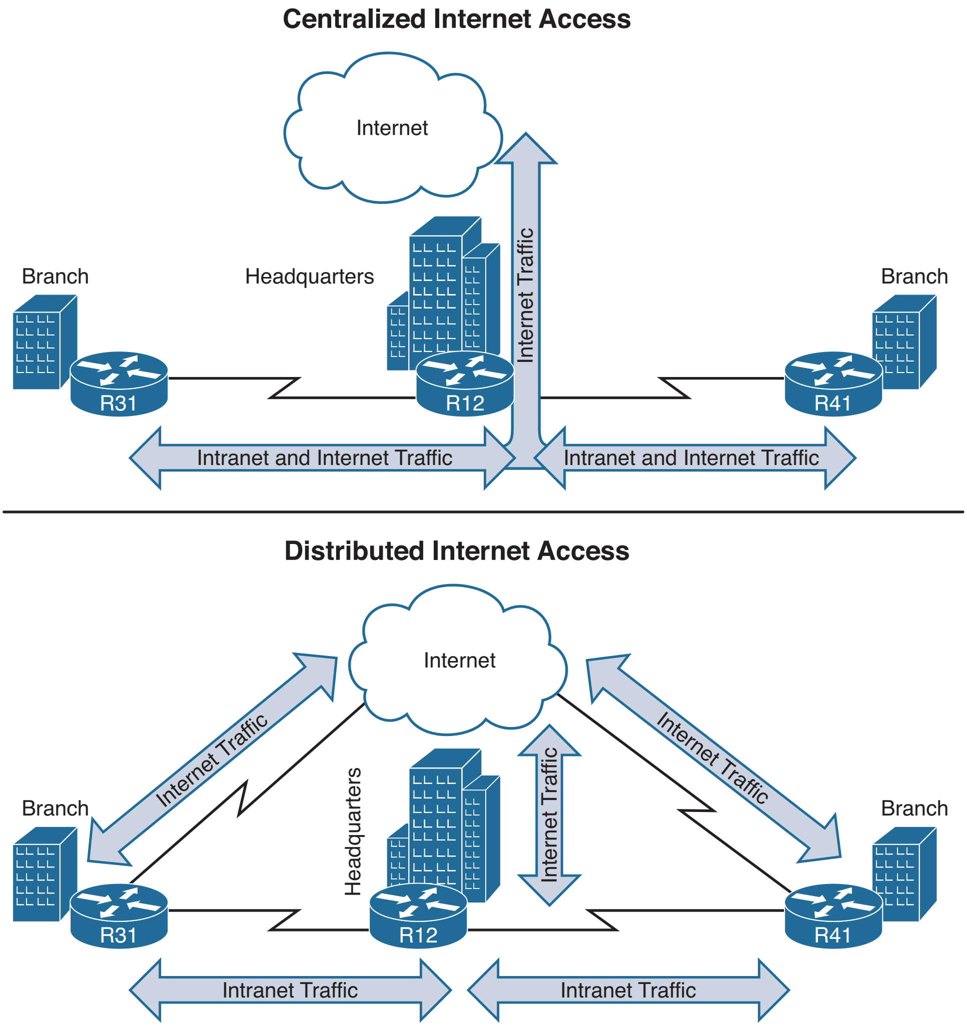


Figure 15-1 *Centralized and Distributed Internet Access Models*

Guest Internet Access

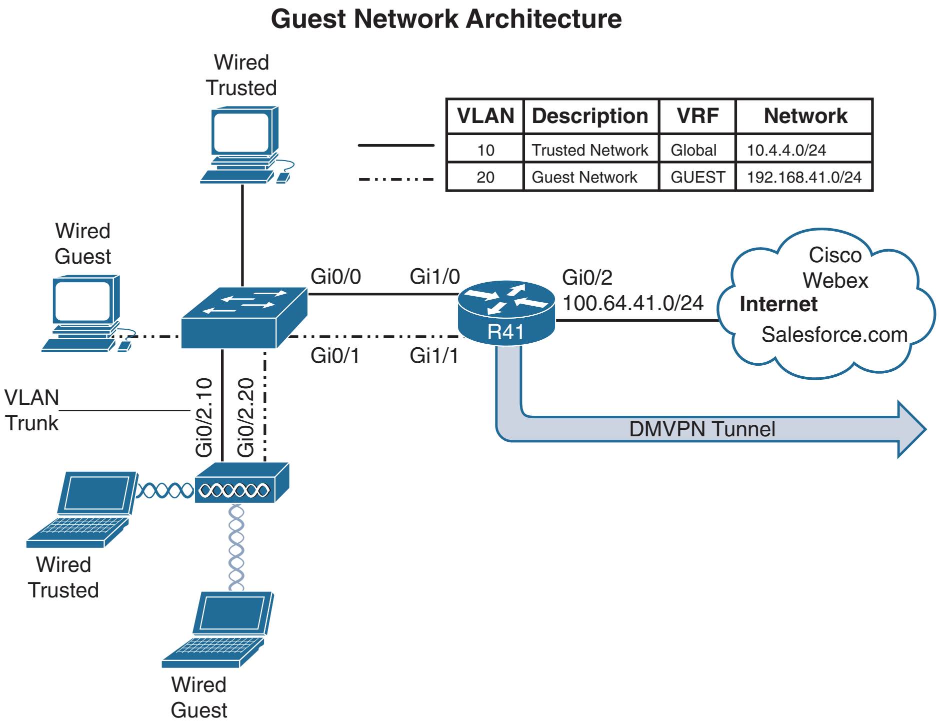
One of the simplest use cases for *direct Internet access (DIA)* is in organizations that provide guest access to nonemployees (customers, partners, or vendors). Guest Internet access is provided typically with wireless (WiFi), is not filtered, and may require an acceptable use policy. The acceptable use policy can define guidelines for use, state what monitoring may (or may not) occur, and require acceptance to remove liability from the organization providing Internet access.

Placing the guest access network in the same network as the corporate network requires proper planning to ensure that internal resources are secured. Placing the guest access network within a *Virtual Route Forwarding (VRF)* instance simplifies the task because the guest access network is logically separate from the internal network in the global routing table. This places the guest users outside of the internal global routing table, which can then be combined with *Zone-Based Firewall (ZBFW)* to limit guest traffic types and deploy QoS to limit the amount of bandwidth usable by the guest users as a pool.

Note Remember that a VRF provides a logical separation of interfaces and routing tables on a router. Placing the guest access network in the same FVRF as the one used for sourcing the Internet DMVPN tunnel eliminates the need for leaking routes between different VRFs.

Figure 15-2 illustrates a sample network architecture that provides guest access to the Internet. Internet connectivity is provided by R41's GigabitEthernet0/2 interface, which is attached to an Internet service provider (ISP) that is using DHCP to assign an IP address on the 100.64.41.0/24 network. The internal LAN network (10.4.4.0/24) is attached to GigabitEthernet1/0, and the guest network (192.168.41.0/24) is attached to GigabitEthernet1/1. The Layer 2 switch connects each of these cables and associates each link into the proper VLAN. In essence, VLAN 10 belongs to the default VRF (global), and VLAN 20 provides guest connectivity through the Internet FVRF. The wireless users are locally connected at the branch and not anchored to a wireless controller.

Another common method of segmenting guest and corporate user traffic is to use 802.1Q tagged subinterfaces on the router, and then the port on the switch is configured as a VLAN trunk. This reduces the number of physical interfaces that are required.



VLAN	Description	VRF	Network
10	Trusted Network	Global	10.4.4.0/24
20	Guest Network	GUEST	192.168.41.0/24

Figure 15-2 *Guest Network Architecture*

Following is the process for creating an FVRF:

Step 1. Create the front-door VRF (FVRF).

The VRF instance is created with the command **vrf definition** *vrf-name*.

Step 2. Identify the address family.

Initialize the appropriate address family for the transport network with the command **address-family** {**ipv4** | **ipv6**}. The address family can be IPv4, IPv6, or both.

Step 3. Associate the FVRF to the interface.

Enter interface configuration submode and specify the interface to be associated with the VRF with the command **interface** *interface-id*. The VRF is linked to the interface with the interface parameter command **vrf forwarding** *vrf-name*.

Step 4. Configure an IP address on the interface.

Configure an IPv4 address with the command **ip address** {*ip-address subnet-mask* | **dhcp**} or with an IPv6 address with the command **ipv6 address** *ipv6-address/prefix-length*.

Example 15-1 provides a sample configuration for the branch LAN network and branch guest network that is hosted on R41. Notice that the alternative configuration with subinterface configuration is provided as well. The subinterface number does not determine the VLAN tag used, but the command **encapsulation dot1q** *vlan* specifies the VLAN tag that is used on the Layer 2 switch to correlate with that subinterface on R41.

Example 15-1 *Branch LAN and Guest Network Configuration*

```
R41 (Interface for Each Function)
interface GigabitEthernet1/0
 description SITE-LAN
 ip address 10.4.4.1 255.255.255.0
!
interface GigabitEthernet1/1
 description GUEST-USERS
 vrf forwarding INET01
 ip address 192.168.41.1 255.255.255.0
```

```
R41 (Sub-Interface for Each Function)
interface GigabitEthernet1/0.10
 encapsulation dot1q 10
 description SITE-LAN
 ip address 10.4.4.1 255.255.255.0
!
interface GigabitEthernet1/0.20
 encapsulation dot1q 20
 description GUEST-USERS
 vrf forwarding INET01
 ip address 192.168.41.1 255.255.255.0
```

> **Note** The topology in Figure 15-2 does not differentiate between wired and wireless clients. Larger sites can differentiate as needed and add more VLANs into the appropriate VRF context.

Dynamic Host Configuration Protocol (DHCP)

Most networks today dynamically assign IP addresses to computers and other host devices. They use a *Dynamic Host Configuration Protocol (DHCP)* server to allocate IP addresses from a block of IP addresses. This allows the DHCP server to track who has the IP address and for how long, and to provide any additional options.

The following steps outline the process for defining a DHCP pool that is used to dynamically assign IP addresses to guest users at the branch:

Step 1. **Create a DHCP address pool.**

The DHCP address pool is the fundamental building block for a DHCP server. The DHCP address pool contains basic IP addressing functions such as DNS servers, default gateways, and networks that are to be assigned. The DHCP address pool is created with the command **ip dhcp pool** *dhcp-pool-name*.

Step 2. **Associate the FVRF to the address pool.**

The DHCP server must know which VRF context it should listen to for DHCP request packets. Any VRF besides the default VRF must be specified with the command **vrf** *vrf-name*.

Step 3. **Define the network range.**

The network out of which IP addresses will be assigned is defined with the command **network** *network* [{*mask* | /*prefix-length*}].

Step 4. **Specify the default router.**

The devices need to know what IP to use as the default gateway. The DHCP server can define the default router with the command **default-router** *ip-address*. Typically the IP address is the IP address to the guest interface.

Step 5. **Define the DNS servers.**

The DNS servers are responsible for performing FQDN-to-IP-address name resolution. The command **dns-server** *ip-address* [*ip-address*] provides the DNS settings to the DHCP client. Alternatively, the command **import all** places DHCP information received from the ISP into its DHCP messages.

Another option for DNS is to provide use of free DNS server offerings such as Google DNS or OpenDNS. Future integration of OpenDNS authentication may provide additional controls via an enterprise account.

Step 6. **Specify the lease duration (optional).**

In essence, the DHCP server assigns (lends) the IP address to a host for a specific amount of time. The DHCP client then starts to ask the DHCP server to renew the IP address it was assigned at the 50 percent mark of the lease. The lease for a DHCP reservation is defined with the command **lease** {*days* [*hours*] [*minutes*] | **lifetime**}.

If the lease duration is not specified, the default is one day.

Step 7. **Specify the list of IP addresses that should be excluded.**

IP addresses that are known to be in use for a DHCP-enabled network segment should be excluded so that they are not assigned. The command **ip dhcp excluded-address** [**vrf** *vrf-name*] *starting-ip-address* [*ending-ip-address*] excludes the entered IP addresses from being assigned to a DHCP client. At a minimum, the IP address for the router should be identified.

Example 15-2 displays the DHCP server configuration for the guest network segment.

Example 15-2 *DHCP Server Configuration for the Guest Network*

```
ip dhcp excluded-address vrf INET01 192.168.41.0 192.168.41.9
!
ip dhcp pool GUEST
! Use the DNS server information provided by Service Providers DHCP server
 import all
 vrf INET01
 network 192.168.41.0 255.255.255.0
 default-router 192.168.41.1
```

Network Address Translation (NAT)

Network Address Translation (NAT) is a technology that rewrites packet headers as packets cross through a NAT device. NAT was created to address concerns of IP address depletion, and it provides the benefit of IP address conservation while allowing internal users to access the Internet.

NAT allows multiple devices on one network to connect to another network segment, all appearing to come from only one IP address. In the context of DIA, the guest access network segment is the internal network segment, and the Internet network is the external network segment. NAT allows the internal network to use only one external IP address to connect to the Internet. This provides additional security by effectively hiding the entire internal network behind a single service-provider-assigned address.

NAT is configured to provide the guest access network connectivity to the Internet using the DHCP address learned from the service provider. The guest access network is identified as the network assigned to the inside network for NAT.

Step 1. **Define the outside interface for NAT.**

Enter interface configuration submode for the guest access network with the command **interface** *interface-id*. Identify the outside interface for NAT with the command **ip nat outside.**

Step 2. **Define the inside interface for NAT.**

Enter interface configuration submode for the guest access network with the command **interface** *interface-id*. Identify the inside interface for NAT with the command **ip nat inside**.

Step 3. **Create an ACL to define the traffic to be translated.**

A standard or extended ACL can be configured to identify the traffic that should be translated. A standard ACL is used because all network traffic on the guest access network should be translated. The standard ACL is defined with the command **ip access-list standard** *access-list-name*. Within the ACL, the guest network should be identified with the command **permit** *guest-network guest-wildcard-mask*.

Step 4. Enable NAT.

NAT is enabled on the branch router with the command **ip nat inside source list** *access-list-name* **interface** *interface-id* [**vrf** *vrf-name* [**match-in-vrf**]] [**overload**] [**no-payload**].

The *interface-id* is the outside interface attached to the Internet. The *vrf-name* is the name of the FVRF, and the **match-in-vrf** keyword allows the inside and outside networks to reside in the same VRF. The **overload** keyword enables many-to-one addressing through port address translation. The **no-payload** option disables payload translation and is used because there will be no static translations or protocols.

Example 15-3 demonstrates the NAT configuration on R41 for the guest access network.

Example 15-3 *R41 Guest in FVRF INET01*

```
interface GigabitEthernet0/2
 description Internet Link
 vrf forwarding INET01
! The DHCP server is assigning the IP address of 100.64.41.1 to this interface
 ip address dhcp
 ip nat outside
! Automatically enabled when NAT is configured
 ip virtual-reassembly in
!
interface GigabitEthernet1/1
 description GUEST-USERS
 vrf forwarding INET01
 ip address 192.168.41.1 255.255.255.0
 ip nat inside
! Automatically enabled when NAT is configured
 ip virtual-reassembly in
!
! The following two lines are a single configuration statement
ip nat inside source list GUEST interface GigabitEthernet0/2 vrf INET01 match-in-vrf
  overload no-payload
!
ip access-list standard GUEST
 permit 192.168.41.0 0.0.0.255
```

Note The configuration contains **ip virtual-reassembly in** on the NAT-enabled interfaces which is automatically enabled to support fragmented packet reassembly as required by NAT. The feature protects the router and clients from a number of fragmentation threats.

Verification of NAT

The command **show ip nat translations** [**vrf** *vrf-name*] [**verbose**] displays the inside local (guest) IP addresses. Example 15-4 demonstrates the use of the command to see the traffic that was translated by NAT. In this example a guest has initiated a traceroute to 192.0.2.1.

Example 15-4 *Verification of NAT*

```
R41-Spoke# show ip nat translations vrf INET01
Pro Inside global      Inside local       Outside local      Outside global
udp 100.64.41.1:49157  192.168.41.10:49157 192.0.2.1:33437   192.0.2.1:33437
udp 100.64.41.1:49158  192.168.41.10:49158 192.0.2.1:33438   192.0.2.1:33438
udp 100.64.41.1:49159  192.168.41.10:49159 192.0.2.1:33439   192.0.2.1:33439
udp 100.64.41.1:49160  192.168.41.10:49160 192.0.2.1:33440   192.0.2.1:33440
udp 100.64.41.1:49161  192.168.41.10:49161 192.0.2.1:33441   192.0.2.1:33441
udp 100.64.41.1:49162  192.168.41.10:49162 192.0.2.1:33442   192.0.2.1:33442
udp 100.64.41.1:49163  192.168.41.10:49163 192.0.2.1:33443   192.0.2.1:33443
udp 100.64.41.1:49164  192.168.41.10:49164 192.0.2.1:33444   192.0.2.1:33444
```

```
R41-Spoke# show ip nat translations verbose
! Output omitted for brevity
Pro Inside global      Inside local       Outside local      Outside global
udp 100.64.41.1:49157  192.168.41.10:49157 192.0.2.1:33437   192.0.2.1:33437
    create 00:00:58, use 00:00:58 timeout:300000, left 00:04:01, Map-Id(In): 1,
    flags:
extended, no-payload, match-in-vrf, use_count: 0, VRF : INET01, entry-id: 1,
  lc_entries: 0
```

Zone-Based Firewall (ZBFW) Guest Access

A stateful firewall secures the network so that only approved network traffic is returned to the client device. The branch routers use Cisco ZBFW for the guest access network to provide stateful firewall support because specific types of traffic are allowed to guest users.

Some organizations provide the guest access network with unfettered access to the Internet, but other organizations may provide connectivity to only web-based applications.

The following steps explain how to configure access for only web-based applications on the guest access network. The configuration provides DNS connectivity for domain-name-to-IP-address resolution to find destinations.

Step 1. **Define the security zones.**

Zones are configured using the command **zone security** *zone-name*. One zone needs to be created for the guest access network, and one for the outside interface.

Step 2. **Define the inspection class map.**

The class map for inspection defines a method for classification of traffic. The class map is configured using the command **class-map type inspect [match-all | match-any]** *class-name*. The **match-all** keyword requires that network traffic match all the conditions listed to qualify (Boolean AND), whereas the **match-any** keyword requires that network traffic match only one of the conditions listed to qualify (Boolean OR). If neither keyword is specified, the **match-all** function is selected.

Step 3. **Define the inspection policy map.**

The inspection policy map applies firewall policy actions to the class maps defined in the policy map. The policy map is then associated to a zone pair.

The inspection policy map is defined with the command **policy-map type inspect** *policy-name*. After the policy map is defined, the various class maps are defined with the command **class type inspect** *class-name*. Under the class map, the firewall action is defined with the following commands:

- **drop** [**log**]: This is the default action which silently discards packets that match the class map. The **log** keyword adds syslog information that includes source and destination information (IP address, port, and protocol).
- **pass** [**log**]: This action makes the router forward packets from the source zone to the destination zone. Packets are forwarded only in one direction. A policy must be applied for traffic to be forwarded in the opposite direction. The **pass** action is useful for protocols such as IPsec ESP and other inherently secure protocols with predictable behavior. The optional **log** keyword adds syslog information that includes the source and destination information.
- **inspect**: The **inspect** action offers state-based traffic control. The router maintains connection/session information and permits return traffic from the destination zone without the need to specify it in a second policy.

The inspection policy map has an implicit class default that uses a default **drop** action. This provides the same implicit "deny all" that is found in an ACL. Adding it to the configuration may simplify troubleshooting for junior network engineers

Step 4. **Define the zone pairs.**

A policy map is now applied to a traffic flow source to a destination configured as **zone-pair security** *zone-pair-name* **source** *source-zone-name* **destination** *destination-zone-name*. The inspection policy map is then applied to the zone pair with the command **service-policy type inspect** *policy-name*. Traffic will be statefully inspected between the source and destination, with return traffic allowed.

Step 5. **Apply the security zones to the appropriate interfaces.**

An interface is assigned to the appropriate zone by entering interface configuration submode with the command **interface** *interface-id* and associating the interface to the correct zone with the command **zone-member security** *zone-name* as defined in Step 1.

Example 15-5 is the deployed configuration for our guest users. Guest users are allowed DNS access to find a destination address and access to HTTP and HTTPS websites and applications.

Example 15-5 *R41 Guest in FVRF INET01 Zone-Based Firewall Configuration*

```
zone security OUTSIDE
 description OUTSIDE Zone used for Internet Interface
zone security GUEST
 description GUEST Zone used for limited Guest Internet Access
!
ip access-list extended ACL-DENY-ANY-IP
 deny ip any any
!
class-map type inspect match-any CLASS-GUEST-TO-OUTSIDE
 match protocol dns
 match protocol http
 match protocol https
 match access-group name ACL-DENY-ANY-IP
!
policy-map type inspect POLICY-GUEST-TO-OUTSIDE
 class type inspect CLASS-GUEST-TO-OUTSIDE
  inspect
 class class-default
  drop
!
zone-pair security GUEST-TO-OUTSIDE source GUEST destination OUTSIDE
 service-policy type inspect POLICY-GUEST-TO-OUTSIDE
!
interface GigabitEthernet0/2
 description Internet Link
 vrf forwarding INET01
```

```
! The DHCP server is assigning the IP address of 100.64.41.1 to this interface
 ip address dhcp
 zone-member security OUTSIDE
!
interface GigabitEthernet1/1
 description GUEST-USERS
 vrf forwarding INET01
 ip address 192.168.41.1 255.255.255.0
 zone-member security GUEST
```

Example 15-6 provides the ZBFW configuration from Chapter 5, “Securing DMVPN Tunnels and Routers,” that allows for DMVPN tunnel establishment and basic connectivity tests. This configuration is provided for reference so that readers can see the complete ZBFW configuration in one place.

Example 15-6 *Reference ZBFW Configuration for DMVPN, Ping, and Traceroute*

```
zone security OUTSIDE
 description OUTSIDE Zone used for Internet Interface
!
ip access-list extended ACL-DHCP-IN
 permit udp any eq bootps any eq bootpc
ip access-list extended ACL-DHCP-OUT
 permit udp any eq bootpc any eq bootps
ip access-list extended ACL-ESP
 permit esp any any
ip access-list extended ACL-GRE
 permit gre any any
ip access-list extended ACL-ICMP
 permit icmp any any
ip access-list extended ACL-IPSEC
 permit udp any any eq non500-isakmp
 permit udp any any eq isakmp
ip access-list extended ACL-PING-AND-TRACEROUTE
 permit icmp any any echo
 permit icmp any any echo-reply
 permit icmp any any ttl-exceeded
 permit icmp any any port-unreachable
 permit udp any any range 33434 33463 ttl eq 1
!
class-map type inspect match-any CLASS-OUTSIDE-TO-SELF-INSPECT
 match access-group name ACL-IPSEC
 match access-group name ACL-PING-AND-TRACEROUTE
```

```
class-map type inspect match-any CLASS-OUTSIDE-TO-SELF-PASS
 match access-group name ACL-ESP
 match access-group name ACL-DHCP-IN
 match access-group name ACL-GRE
class-map type inspect match-any CLASS-SELF-TO-OUTSIDE-INSPECT
 match access-group name ACL-IPSEC
 match access-group name ACL-ICMP
class-map type inspect match-any CLASS-SELF-TO-OUTSIDE-PASS
 match access-group name ACL-ESP
 match access-group name ACL-DHCP-OUT
!
policy-map type inspect POLICY-OUTSIDE-TO-SELF
 class type inspect CLASS-OUTSIDE-TO-SELF-INSPECT
  inspect
 class type inspect CLASS-OUTSIDE-TO-SELF-PASS
   pass
 class class-default
   drop
!
policy-map type inspect POLICY-SELF-TO-OUTSIDE
  class type inspect CLASS-SELF-TO-OUTSIDE-INSPECT
   inspect
  class type inspect CLASS-SELF-TO-OUTSIDE-PASS
   pass
  class class-default
  drop log
!
zone-pair security SELF-TO-OUTSIDE source self destination OUTSIDE
 service-policy type inspect POLICY-SELF-TO-OUTSIDE
!
zone-pair security OUTSIDE-TO-SELF source OUTSIDE destination self
  service-policy type inspect POLICY-OUTSIDE-TO-SELF
```

Verification of ZBFW for Guest Access

The verification of guest access can be seen on the ZBFW with the command **show policy-map type inspect zone-pair** *zone-pair-name* [**sessions**] which is demonstrated in Example 15-7. In the example, packets have been passed by the ZBFW for DNS, HTTP, and HTTPS and other traffic has been dropped.

Example 15-7 *Verification of ZBFW for Guest Access*

```
R41-Spoke# show policy-map type inspect zone-pair GUEST-TO-OUTSIDE sessions

policy exists on zp GUEST-TO-OUTSIDE
  Zone-pair: GUEST-TO-OUTSIDE

  Service-policy inspect : POLICY-GUEST-TO-OUTSIDE

    Class-map: CLASS-GUEST-TO-OUTSIDE (match-any)
      Match: protocol dns
        123 packets, 9630 bytes
        5 minute rate 1000 bps
      Match: protocol http
        2030 packets, 288079 bytes
        5 minute rate 5000 bps
      Match: protocol https
        172 packets, 15562 bytes
        5 minute rate 1000 bps

   Inspect

    Class-map: class-default (match-any)
      Match: any
      Drop
        84581 packets, 325696 bytes
```

Guest Access Quality of Service (QoS)

Regulating the amount of WAN bandwidth consumed by the guest network should be considered when developing DIA designs. The branch site location must have sufficient bandwidth for critical business operations (internal user applications) while providing the delta to the guest network. A common technique is to place all traffic received from the guest network in a Scavenger QoS class on the WAN egress interface. The Scavenger class is meant for applications of a lower priority than best effort; in essence any bandwidth not consumed by corporate users is available for the guest network.

The egress QoS policy may need adjustment to allow a Scavenger class to support guests under congestion, providing the guest users with appropriate performance to fit the business model. Some public-facing organizations (such as retail and hospitals) need to provide guest Internet access as part of their business model and need to guarantee that guest traffic is treated appropriately in comparison to internal users' application traffic. Based on the business model for guest user access, a dedicated queuing structure can be built to keep guest traffic separate from other traffic.

The following steps create a QoS policy map to mark all guest network traffic as part of the Scavenger class.

Step 1. **Configure QoS for control of guest user egress traffic.**

Enter policy map submode using the command **policy-map** *policy-name*.

Step 2. **Use a single queue for all guest user traffic.**

Enter default class submode using the command **class class-default**.

Step 3. **Assign all guest user traffic to a specific egress queue.**

Mark traffic to the appropriate queue using the command **set dscp** *dscp-value*.

Step 4. **Apply the QoS policy to the guest access interface.**

Enter interface configuration submode for the guest access network with the command **interface** *interface-id*. Apply the QoS policy with the command **service-policy input** *policy-name*.

In Example 15-8, all guest user traffic is assigned the DSCP value of CS1 by applying an input policy to the guest network interface. DSCP CS1 is a typical Scavenger class setting, where the traffic is considered less than best effort, based on other applications having preference for limited bandwidth.

Example 15-8 *R41 Guest in FVRF INET01 Quality of Service WAN Egress*

```
R41
policy-map POLICY-GUEST-TO-INTERNET
 description remark all GUEST Traffic Scavenger
 class class-default
  set dscp cs1
!
interface GigabitEthernet1/1
 description GUEST-USERS
 vrf forwarding INET01
 ip address 192.168.41.1 255.255.255.0
 service-policy input POLICY-GUEST-TO-INTERNET
```

This QoS marking policy addresses traffic flowing from the branch to the Internet, but the return traffic must be considered. A guest user could download a large file that congests inbound bandwidth from the Internet, affecting corporate users. It is recommended that a QoS traffic-shaping policy be deployed to rate-limit the amount of traffic consumed by guest users.

Because the guest network connects via a single inside interface, the use of an egress shaper toward the guest network is appropriate and is called *Remote Ingress Shaping*. This shaper is applied after the router has already received the traffic, so the router needs to limit network traffic below the rate that is allowed to the guest network. A good rule of thumb is to set the shaping rate to 95 percent of the offered bandwidth so that TCP windowing controls the maximum rate allowed. The use of fair queuing is configured to give multiple users a fair share of the bandwidth. Increasing the queue depth smooths out any drops but introduces the possibility of increasing delay.

The following steps create a Remote Ingress Shaper for the guest network traffic.

Step 1. **Define the guest access ingress QoS policy.**

Enter policy map submode using the command **policy-map** *policy-name*.

Step 2. **Define a single queue for all guest traffic.**

Enter default class submode using the command **class class-default**.

Step 3. **Rate-limit all guest access traffic.**

Guest access network traffic is shaped down to an acceptable bandwidth rate with the command **shape average** *kbps*.

Step 4. **Enable fair queuing.**

The fair queue is configured with the command **fair-queue**.

Step 5. **Apply the QoS policy to the guest access interface.**

Enter interface configuration submode for the guest access network with the command **interface** *interface-id*. Apply the QoS policy with the command **service-policy output** *policy-name*.

Example 15-9 demonstrates a sample configuration for assigning a QoS policy for traffic returning from the Internet to the guest access network.

Example 15-9 *R41 Guest in FVRF INET01 Quality of Service WAN Ingress*

```
policy-map INTERNET-TO-GUEST
 description shape all GUEST Traffic Allowed
 class class-default
! 95% of 2Mbps
  shape average 1900000
  fair-queue
!
interface GigabitEthernet1/1
 description GUEST-USERS
 vrf forwarding INET01
 ip address 192.168.41.1 255.255.255.0
service-policy output INTERNET-TO-GUEST
```

Note Now that the guest network is defined and operational, some changes to Per-Tunnel QoS should be reviewed to provide the needed headroom to prevent internal traffic from competing with the guest network. In this section, 2 Mbps was allocated for the guest network, so to protect internal traffic a reduction of the Per-Tunnel QoS shaper by 2 Mbps is required.

Guest Access Web-Based Acceptable Use Policy

Most organizations require that guest users accept an "acceptable use policy" via a web page. Depending on the level of security, users may be required to provide their consent, or authenticate with some form of password. This feature can be enabled by using the router's integrated consent feature with minimal configuration. The following sections cover both techniques.

Guest Network Consent

The consent feature allows the specification of an appropriate login banner and login HTML page specifying the required acceptable use policy. This feature requires that the HTTP server be enabled on the router. A standard numbered ACL (such as 99) denies all traffic to the HTTP server, so that the authentication proxy can establish an HTTP session for the specific client redirection.

The authentication proxy operates by defining an admission name and correlating an ACL for any traffic that requires consent. The ACL denies specific protocols that should be allowed such as DNS and DHCP. All other protocols are permitted and are then redirected and captured. The authentication proxy is then associated to an interface where it should listen for traffic that matches the ACL.

The authentication proxy feature requires an HTML file with the organization's acceptable use policy. Example 15-10 provides a reference file, consent.html, that will be copied onto the router.

Example 15-10 *Guest Access Consent File (consent.html)*

```
<html>
<head>
<title>Guest Network Access</title>
</head>
<body>
<center><h1>Guest Network Access</h1></center>
<p>By agreeing to this page, you take full responsibility for your actions.
<p>You agree that your data will be logged, documented, and handed over to
  authorities should it be requested.
<br>
</body>
</html>
```

The following steps configure the guest network consent function on a router:

Step 1. **Copy the consent banner to the router.**

An HTML file is created with acceptable company verbiage. This file is copied onto the router with the command **copy** {**ftp** | **http** | **https** | **tftp** | **scp**}:*//html-file.html* {**bootflash** | **disk0** | **flash** | **harddisk**}:*//html-file.html*.

Step 2. **Identify the consent banner file.**

The consent banner file is configured using the command **ip admission consent-banner file** {**bootflash** | **disk0** | **flash** | **harddisk**}:*//html-file.html*.

Step 3. **Define a simple text consent banner (optional).**

If only a simple banner is required, a simple text banner can be created with the command **ip admission consent-banner text** *delimiting-character consent-banner-text delimiting-character.*

Step 4. **Define an ACL for interesting traffic.**

Configure an ACL for traffic that will trigger the consent page. The consent ACL uses a **deny** statement for allowed traffic and a **permit** to capture traffic for IP admission to use.

Step 5. **Configure an admission name.**

An admission policy is created for placement on the ingress interface, using the command **ip admission name** *admission-name* **consent list** *access-list-name*.

Step 6. **Enable the HTTP server.**

Enable the HTTP server capability on the router with the command **ip http server**. A dedicated HTTP process is triggered for the IP admission process.

Configure a standard numbered ACL that restricts all guest network traffic to the server with the command **access-list 99 deny any**. Apply this access list to the HTTP process using the command **ip http server access-class 99**.

Step 7. **Apply the admission name to the ingress guest interface.**

Enter interface submode using the command **interface** *interface-id* and apply the admission name using the command **ip admission** *admission-name*.

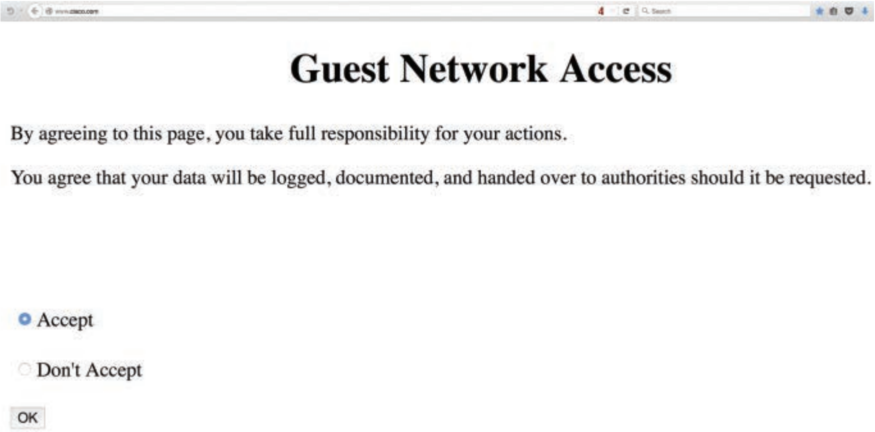
In Example 15-11, the ACL-GUEST is configured so that DNS and DHCP are denied, which then allows a client to receive an IP address, resolve an IP from an FQDN, and then establish an initial session to trigger the acceptable use policy web page for acceptance.

Example 15-11 *R41 Guest Consent Acceptance*

```
ip admission consent-banner file flash:consent.html
ip admission consent-banner text ^C Guest Network Consent ^C
ip admission watch-list enable
ip admission max-login-attempts 3
ip admission max-nodata-conns 20
ip admission auth-proxy-audit
ip admission name GUEST consent inactivity-time 60 list ACL-GUEST
!
interface GigabitEthernet1/1
 description GUEST-USERS
 vrf forwarding INET01
 ip address 192.168.41.1 255.255.255.0
 ip admission GUEST
!
ip http server
ip http access-class 99
!
ip access-list extended ACL-GUEST
 deny   udp any any eq domain
 deny   udp any eq bootpc any eq bootps
 permit ip any any
!
access-list 99 deny any
end
```

Note The command **ip admission auth-proxy-audit** logs new connections with the initiating IP address when the connection is first established. This information can be correlated to the DHCP lease logs for granular reporting.

In Figure 15-3, the web browser has been opened, and the initial web request to the browser's home page (www.cisco.com) has been redirected to the router's consent page. The guest user will choose "Accept" or "Don't Accept" for the agreement as specified in the **ip admission consent-banner file** *file-name* command. After the agreement has been accepted, by default the browser displays the initially requested web page.



Guest Network Access

By agreeing to this page, you take full responsibility for your actions.

You agree that your data will be logged, documented, and handed over to authorities should it be requested.

● Accept

○ Don't Accept

Figure 15-3 *Client Consent Web Page*

Example 15-12 demonstrates the syslog information that is provided when **ip admission auth-proxy-audit** is enabled on the router.

Example 15-12 *IP Admission Logs*

```
%AP-6-AUTH_PROXY_AUDIT_START: initiator (192.168.41.10) start
%AP-6-AUTH_PROXY_AUDIT_STOP: initiator (192.168.41.10) send 0 packets 0 bytes;
  duration time 00:18:10
```

A list of recent guest clients that have accepted the consent page can be seen with the command **show ip admission cache** as shown in Example 15-13.

Example 15-13 *R41 Verification of Consenting Clients*

```
R41-Spoke# show ip admission cache
Authentication Proxy Cache
 Client Name N/A, Client IP 192.168.41.10, Port 54727, timeout 60,
  Time Remaining 60, state ESTAB
```

Note When configuring the various **ip admission** commands, IOS automatically configures the equivalent legacy **ip auth-proxy** commands. In essence, it appears as if the configuration is duplicated, but the removal of either the **ip admission** or the **ip auth-proxy** command removes both instances.

Guest Authentication

Some organizations may feel that a consent page is insecure because anyone sitting in range of the wireless network can gain access by accepting the terms on the consent page. An additional level of security can be added by using a pre-shared *WiFi Protected Access (WPA)* password that can be stored on devices. Another method of securing the wireless network is to require users to authenticate via a web authentication portal. The router can use a radius, TACACS, or Cisco Identity Services Engine's (ISE) guest portal.

The same configuration structure as consent is used for guest authentication. The same consent.html file is used, and users are authenticated locally for this section. Users are required to provide a valid username and password to connect to the Internet. The process for configuring authentication is as follows:

Step 1. **Create local user accounts.**

The authentication mechanism requires the use of the *authentication, authorization, and accounting (AAA)* access model. An administrative user account (privilege level 15) should be created with the command **username** *username* [**privilege** *privilege-level*] [{**password** | **secret**} *password*].

Our example uses a local shared user *guest-account* account that will be used for authenticating into the portal. This user should be assigned a system-level privilege of 0.

Step 2. **Initialize the AAA model.**

Initialize the AAA model with the command **aaa new-model.**

Step 3. **Create a new AAA authentication mechanism.**

The AAA authentication mechanism identifies how a specific router's access method verifies the validity of the account. This does not tell the router the type of access that the user is allowed.

The command is **aaa authentication login** {**default** | *aaa-list-name*} [*method1*] [*method2*] [*method3*] . . . . A wide variety of methods are available, but the short list is:

- **enable:** The enable password.
- **local:** Local user database is used.
- **none:** No authentication.
- **group radius:** Authenticate off a defined radius server.
- **group tacacs:** Authenticate off a defined TACACS server.

This example uses local authentication.

Step 4. **Create a new AAA authorization for auth-proxy.**

The AAA authorization mechanism defines the level of access that a user is allowed. The command is **aaa authorization auth-proxy default** [**group** *server-group-name*] [**local**]. Either a group or local method must be configured. This example uses local.

Step 5. **Create an AAA attribute list for the guest user account.**

The user account has a default privilege level of 0 on the router but requires elevated privileges for the **auth-proxy protocol**. An AAA attribute list is created with the command **aaa attribute list** *aaa-attribute-list-name*.

Then the privilege level 15 is associated to the **auth-proxy protocol** with the command **attribute type priv-lvl 15 service auth-proxy protocol ip**.

Step 6. **Associate the auth-proxy AAA attribute to the guest user account.**

Now the AAA attribute list is associated with the local user account with the command **username** *username* **aaa attribute list** *aaa-attribute-list-name*.

Step 7. **Copy the authentication banner to the router.**

An HTML file is created with acceptable company verbiage. This file is copied onto the router with the command **copy** {**ftp** | **http** | **https** | **tftp** | **scp**}*://html-file.html* {**bootflash** | **disk0** | **flash** | **harddisk**}*://html-file.html*.

Step 8. **Identify the authentication banner file.**

The consent banner file is configured using the command **ip admission auth-proxy-banner file** {**bootflash** | **disk0** | **flash** | **harddisk**}*://html-file.html*.

Step 9. **Configure a simple text authentication banner (optional).**

If only a simple authentication banner is required, it can be made with the command **ip admission auth-proxy-banner text** *delimiting-character consent-banner-text delimiting-character*.

Step 10. **Define an ACL for interesting traffic.**

Configure an ACL for traffic that will trigger the authentication page. The authentication ACL uses a **deny** statement for allowed traffic and a **permit** to capture traffic for IP admission to authenticate.

Step 11. **Configure an admission policy.**

An admission policy is created for placement on the ingress interface using the command **ip admission name** *admission-name* **proxy http list** *access-list-name*.

Step 12. **Enable the HTTP server.**

Enable the HTTP server capability on the router with the command **ip http server**. A dedicated HTTP process is triggered for the IP admission process.

Configure a standard numbered ACL that restricts all guest network traffic to the server with the command **access-list 99 deny any**. Apply this access list to the HTTP process using the command **ip http server access-class 99**.

Step 13. **Apply the admission policy to the ingress guest interface.**

Enter interface submode using the command **interface** *interface-id* and apply the admission policy using the command **ip admission** *admission-name*.

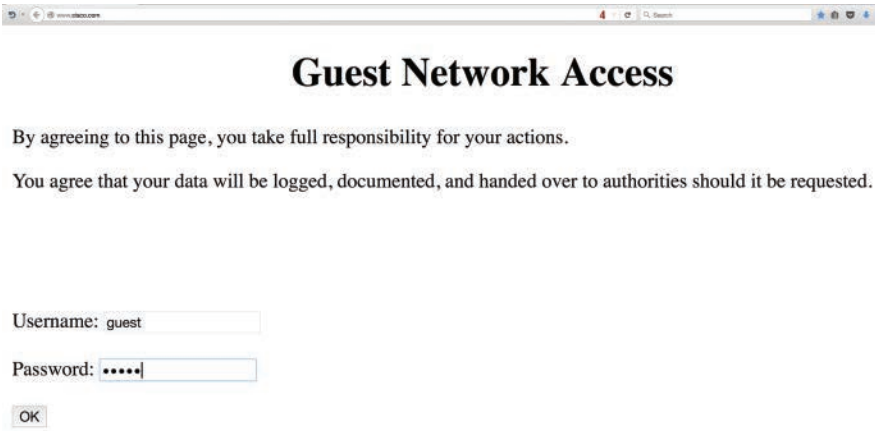
Example 15-14 demonstrates the configuration for guest authentication. Notice that the **ip admission consent** commands have been replaced with **ip admission auth-proxy** commands. Also, the keywords **proxy http** have been added to the IP admission name command.

Example 15-14 *R41 Guest Authentication*

```
username guest-account privilege 0 secret PASSWORD123
!
aaa new-model
aaa authentication login AAA-AUTH-PROXY local
aaa authorization auth-proxy default local
!
username guest-account aaa attribute list GUEST
aaa attribute list GUEST
 attribute type priv-lvl 15 service auth-proxy protocol ip
!
ip admission auth-proxy-banner file flash:consent.html
ip admission auth-proxy-banner http * Guest Network Authentication *
ip admission watch-list enable
ip admission watch-list expiry-time 5
ip admission max-nodata-conns 1000
ip admission init-state-timer 15
ip admission auth-proxy-audit
ip admission name GUEST-AUTH proxy http inactivity-time 60 list ACL-GUEST
!
interface GigabitEthernet1/1
 description GUEST-USERS
 vrf forwarding INET01
 ip address 192.168.41.1 255.255.255.0
 ip admission GUEST-AUTH
!
ip http server
ip http access-class 99
!
ip access-list extended ACL-GUEST
 deny udp any any eq domain
 deny   udp any eq bootpc any eq bootps
 permit ip any any
!
access-list 99 deny any
end
```

Note The same HTML file can be used for consent or authentication. IP admission provides either the radio buttons for acceptance or the username and password prompts for authentication.

In Figure 15-4, the web browser has been opened, and the initial web request to the browser's home page has been redirected to the router's auth-proxy-banner page. Here the user is prompted for username and password. Attempting to authenticate implies the acceptance of the specified agreement from the **ip admission auth-proxy-banner file**. After the user is authenticated, by default the browser displays the initially requested web page.



Guest Network Access

By agreeing to this page, you take full responsibility for your actions.

You agree that your data will be logged, documented, and handed over to authorities should it be requested.

Username: guest

Password: •••••

Figure 15-4 *Client Authentication Web Page*

Note In Figure 15-4, local user authentication was used, but it is possible to use most AAA methods available by changing the **aaa authorization auth-proxy default** to a different method.

Now that the basic web-based authentication has been demonstrated, additional customization can be done by providing a unique file for login, success, failure, or expiration. The commands for additional customization are **ip admission proxy http** {**login** | **success** | **failure** | **expired**} **page file** *url*. It is even possible to send an HTTP redirect to a different website URL with the command **ip admission proxy http success redirect** *url*. Example 15-15 demonstrates the configuration.

Example 15-15 *Guest Authentication Customized Web Pages*

```
ip admission proxy http login page file bootflash:login.htm
ip admission proxy http success page file bootflash:success.htm
ip admission proxy http failure page file bootflash:fail.htm
ip admission proxy http expired page file bootflash:expired.htm
ip admission proxy http success redirect www.company.com
```

Note The *Authentication Proxy Configuration Guide* listed at the end of this chapter includes more information on how to customize the web proxy authentication web pages.

To verify authenticated clients, just as with consent, the command **show ip admission cache** displays the users that are connected. The command **show ip admission watch-list** displays the IP addresses of those systems that are on the network but have not authenticated. Example 15-16 demonstrates the use of both commands. Only the 192.168.41.10 is present on the network and has authenticated with the router.

Example 15-16 *R41 Verification of Authenticating Clients*

```
R41-Spoke# show ip admission cache
Authentication Proxy Cache
 Client Name N/A, Client IP 192.168.41.10, Port 54727, timeout 60, Time
   Remaining 60, state ESTAB
```

```
R41-Spoke# show ip admission watch-list
Authentication Proxy Watch-list is enabled
Watch-list expiry timeout is 5 minutes
No entries in the watch-list
```

If the design requires the ability to log user access and limit the amount of time the user can use the free service, an external system is necessary. This is where the Cisco ISE is required. Many options are available via ISE to support guest users with both self-registration and sponsored guest access. ISE offers web-based authentication with acceptable use policy, duration limits, one-time authentication, and logging. A key design component requires that the ISE subnet be reachable by both internal and guest users. More information on the use of ISE with guest access can be found in Cisco Validated Design, *IWAN Security for Remote Site Direct Internet Access and Guest Wireless*.

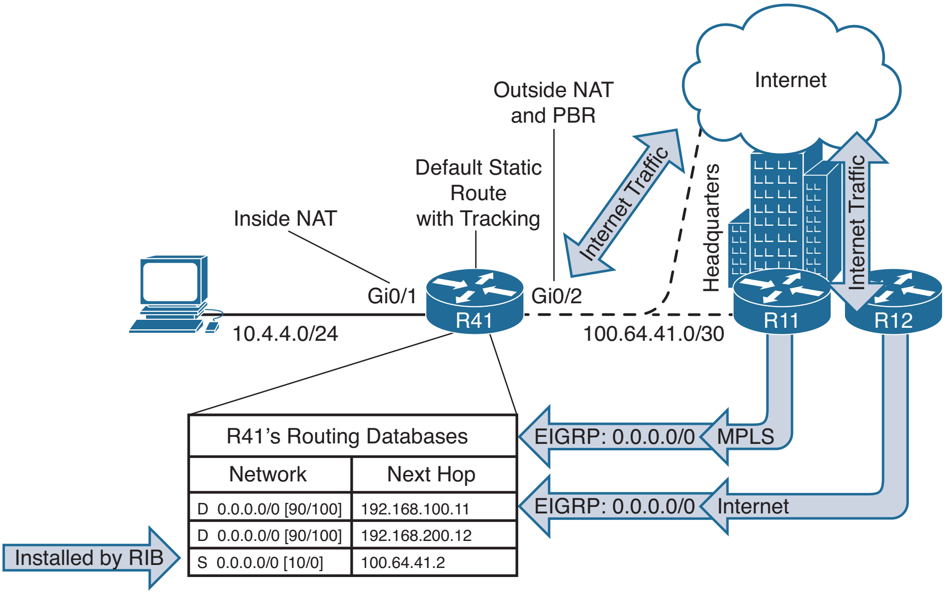
Internal User Access

Granting direct access to the Internet for internal users can increase productivity, offload bandwidth-consuming applications from the WAN, and decrease latency to critical cloud-based applications. In the centralized Internet model, all traffic to and from the Internet traverses a rather complex security architecture and minimizes the number of touchpoints to the Internet. As the application footprint has changed, the ability to maintain a centralized Internet model meeting bandwidth and low-latency requirements has become an issue. Moving to a distributed model is an easy choice for those working with the applications and the departments funding the circuits, but the security team needs to validate that the solution meets their requirements.

As with guest Internet access, the simplest solution for providing Internet access to internal branch users is by using a static route to the Internet SP. The Internet's interface is placed in a dedicated FVRF for segmentation, interface selection for transport independence, and so forth. It is possible to connect the INET01's routing table with the global routing table.

In this design, the FVRF's default route is used to establish DMVPN tunnels, and the global table's fully specified static route provides direct access to the Internet. The design includes the monitoring of the local Internet connection, and in the event of failure traffic can be routed to the Internet (via the DMVPN tunnel) in a centralized Internet access model.

Figure 15-5 illustrates all these concepts from R41's perspective. Notice that R41 has multiple default routes in the routing table but installs the static route into the RIB as long as the tracking provides successful verification to the Internet.



R41's Routing Databases

Network	Next Hop
D 0.0.0.0/0 [90/100]	192.168.100.11
D 0.0.0.0/0 [90/100]	192.168.200.12
S 0.0.0.0/0 [10/0]	100.64.41.2

Figure 15-5 *Direct Internet Access Components*

Several steps are involved in providing internal users with direct Internet access. The configuration looks similar to a typical split-tunneled VPN deployment that includes

- A default route via the Internet-facing interface
- NAT to use the provided service provider routable address
- ZBFW to statefully allow only requested responses and denial of unrequested traffic
- The return traffic from the Internet remains in the front-door VRF and requires restoration to the global routing table. This is accomplished using policy-based routing (PBR) to restore traffic after processing by NAT and ZBFW to the global routing table.

Note Because the static route installs over the default route advertised from the DMVPN hub routers, more explicit enterprise network prefixes must be advertised from the DMVPN hub routers.

Fully Specified Static Default Route

The first step for internal direct Internet access is to create a route in the global routing table that allows connectivity to the Internet FVRF physical interface, with an administrative distance that is preferred over a routing protocol. (Remember, the default route is advertised from the DMVPN hub routers too.) The default administrative distance for static routes is 1, which prevents us from installing an additional preferred route in the future, so our example uses an administrative distance of 10. The outbound interface must be specified for two essential reasons:

- If the interface loses link protocol (connectivity) to the Internet, the static route is removed from the routing table.
- It must be included to provide outbound connectivity from the global VRF to the FVRF.

The command **ip route 0.0.0.0 0.0.0.0** *internet-interface* {*next-hop-ip* | **dhcp**} [*administrative-distance*] configures the appropriate static route. Example 15-17 provides the sample configuration and verification of the default route.

Example 15-17 *R41 Internal Internet Access Default Route Configuration*

```
R41
ip route 0.0.0.0 0.0.0.0 GigabitEthernet0/2 100.64.41.2 10
```

```
R41-Spoke# show ip route 0.0.0.0
Routing entry for 0.0.0.0/0, supernet
  Known via "static", distance 10, metric 0, candidate default path
  Routing Descriptor Blocks:
  * 100.64.41.2, via GigabitEthernet0/2
      Route metric is 0, traffic share count is 1
```

Verification of Internet Connectivity

The installed route points to a dedicated interface, and the static default route is disabled when the interface goes down. A majority of broadband Internet circuits are provided via cable, DSL, or a fiber-optic service that is not Ethernet, requiring the use of an external modem that does not provide link propagation. This leaves the router's Ethernet interface up when the Internet service is down or degraded.

The IWAN architecture is built with reliable hub sites that can test connectivity by tracking the Internet DMVPN tunnel interface state, but the failure detection time is fairly long because detection time is based on NHRP timers. The same issue occurs with tracking routing protocol neighbors because of the increased timers. Detection time is enhanced with the use of Cisco IP SLA capability that is built into IOS. IP SLA can monitor DNS resolution, HTTP requests, or an ICMP echo to a reliable destination

Tracking one remote endpoint is not sufficient because of maintenance cycles. Is it sufficient for a reachability test to connect to only one specific website or IP address? *Enhanced object tracking (EOT)* provides intelligence by querying the state of multiple items on the Internet. Combined with IP SLA capability, it enhances the ability to determine if the Internet circuit is functioning properly. Sending a ping to the DMVPN hub routers at an acceptable rate provides a reliable measure for a source that is controlled by your organization.

Note Some people might ping the next-hop IP address of the ISP, but if the router is on DHCP, it may be hard to determine. Other people might ping Google's DNS servers, but what happens if they apply a security policy to deny ICMP packets? The DMVPN hub servers are controlled by your organization, and no outsiders can change the configuration or security policies, which can cause unexpected issues.

Combining these monitoring capabilities to increase the reliability of the tracked objects is of utmost importance. This is where using a tracked object with thresholds is required. EOT supports the tracking of other tracked objects (nested) so that multiple objects can be tracked and a percentage or weight value assigned for each state. The EOT tracking is then added to the static default route, which removes the route if the tracked status is *down*.

In a simple example, the status of the Internet can be determined by tracking

- The Internet DMVPN tunnel interface
- Routing over the DMVPN tunnel interface
- Availability of DNS query to both Google DNS servers for www.google.com

When 51 percent or less of IP SLAs are available, the router should identify the state as *down*, indicating that DIA connectivity is not available. But if 74 percent of the IP SLAs are available, then the state is *up* and DIA connectivity is available. This logic requires that three of the four tests be available for DIA connectivity to be viable. Having only two of four available—50 percent—is lower than the configured 51 percent so the DIA path is considered down.

Another option is to establish that one tracked item is more important than another tracked object. Multiple tracked objects are then nested into a single tracked object. The child tracked objects can be weighted to give one precedence over another. This logic is used in Example 15-11. It is also possible to use weight as shown in Example 15-18.

The following steps define how Internet connectivity can be monitored on the branch routers.

Step 1. **Configure an IP SLA.**

Cisco IOS IP SLAs generate traffic in a continuous, reliable, and predictable manner to measure network performance. The network traffic includes data about response time, one-way latency, jitter (inter-packet delay variance), packet loss, network resource availability, and server response time.

Every test is provided with its own instance. The IP SLA instance is created with the command **ip sla** *ip-sla-number*.

Step 2. **Define the type of IP SLA.**

IP SLA provides multiple functions, but in this book IP SLA is used for a basic IP connectivity test via ICMP and DNS query.

ICMP IP SLAs are configured with the command **icmp-echo** {*destination-ip-address* | *destination-hostname*} [**source-ip** *ip-address* | **source-interface** *interface-id*]. Specifying a **source-ip** or **source-interface** is optional.

DNS query IP SLAs are configured with the command **dns** {*destination-ip-address* | *destination-hostname*} [**source-ip** *ip-address* | **source-interface** *interface-id*]. Specifying a **source-ip** or **source-interface** is optional.

Step 3. **Define the type of IP SLA.**

If the IP SLA resides in a VRF, the VRF must be specified with the command **vrf** *vrf-name*.

Step 4. **Define the frequency of the IP SLA test.**

The frequency of the IP SLA test needs to be defined with the command **frequency** *seconds*. The book's example uses a value of 15.

Step 5. **Define the upper threshold value of the IP SLA test.**

The upper threshold value for calculating network-monitoring statistics created by an IP SLA is defined with the command **threshold** *milliseconds.*

Step 6. **Define the timeout of the IP SLA test.**

The amount of time that an IP SLA waits in response for a packet is defined with the command **timeout** *milliseconds*. The book's example uses a value of 3000.

Step 7. **Schedule the IP SLA to run.**

The IP SLA needs to have a start and stop time identified. The command **ip sla schedule** *ip-sla-number* **life forever start-time now** continuously runs the IP SLA.

Step 8. **Create child tracked objects.**

Every one of the individual IP SLAs needs to be configured as a child tracked object with the command **track** *track-number* **ip sla** *ip-sla-number*.

The state of a DMVPN tunnel can be tracked with the command **track** *track-number* **interface** *interface-id* **line-protocol**. DMVPN tunnel health monitoring must be enabled with the interface parameter command **if-state nhrp**.

Step 9. **Create a track list to monitor all child objects.**

The command **track** *track-number* **list threshold {percentage | weight}** creates the master tracking entity. Underneath the master track list, the previously tracked objects are added with the command **object** *track-number* [**weight** *weight*].

If all objects have an equal percentage or weight, **percentage** can be used. If some objects are more important than others, use of **weight** is recommended.

Step 10. **Define a threshold for when a track list is up or down.**

A threshold needs to be defined for when a track list is up or down. The command **threshold weight** [**up** *up-value*] [**down** *down-value*]. At least one of the **up** and **down** keywords must be specified with this command.

Step 11. **Update the static route to reflect the tracked route.**

The fully specified static route can now be included to include tracking with the command **ip route 0.0.0.0 0.0.0.0** *internet-interface* {*next-hop-ip* | **dhcp**} [*administrative-distance*] **track** *track-number*.

Note It is recommended to match the track number to the IP SLA number to simplify verification of threshold-based tracked objects.

Example 15-18 provides a sample configuration for weighted tracking of the Internet. Four IP SLAs are created. Two are ICMP echo for the Internet DMVPN hub transport IP addresses, and the other two IP SLAs are DNS queries for root DNS servers. There are multiple root DNS servers with the same FQDN. The master tracked list combines the four IP SLA entries and the Internet DMVPN tunnel status.

Example 15-18 *R41 Internet Monitoring*

```
! Internet Tunnel Interface Hub at Data Center 1
ip sla 201
icmp-echo 100.64.12.1 source-interface GigabitEthernet0/2
 vrf INET01
 threshold 1000
 frequency 15
ip sla schedule 201 life forever start-time now
! Internet Tunnel Interface Hub at Data Center 2
ip sla 202
 icmp-echo 100.64.22.1 source-interface GigabitEthernet0/2
 vrf INET01
 threshold 1000
 frequency 15
ip sla schedule 202 life forever start-time now
! D DNS Root Server which is Anycast
ip sla 211
 dns d.root-servers.net name-server 199.7.91.13
 vrf INET01
 threshold 1000
 timeout 3000
 frequency 15
ip sla schedule 211 life forever start-time now
! F DNS Root Server which is Anycast
ip sla 212
 dns f.root-servers.net name-server 192.5.5.241
 vrf INET01
 threshold 1000
 timeout 3000
 frequency 15
ip sla schedule 212 life forever start-time now
!
! Internet Tunnel Interface State
! "if-state nhrp" configuration required
track 20 interface Tunnel200 line-protocol
!
track 21 ip sla 201 reachability
track 22 ip sla 202 reachability
track 23 ip sla 211 reachability
track 24 ip sla 212 reachability
!
track 100 list threshold weight
 ! The default weight 10
 object 20
 object 21
 object 22
! The DNS request is twice as important
 object 23 weight 20
 object 24 weight 20
 threshold weight down 21 up 40
! Delay coming up for 30 seconds in case we are having transient issues
 delay up 30
!
ip route 0.0.0.0 0.0.0.0 GigabitEthernet0/2 100.64.41.2 10 track 100
```

One of the problems associated with using a DHCP learned next hop is that direct tracking is not an option as shown in Example 15-19. The solution is to install a static host route in the global table pointed to an IP address that does not require connectivity, because this host route will be in the routing table when the interface is up but the path is unusable.

Example 15-19 *Static Default Routes with DHCP Interfaces*

```
R41-Spoke(config)# ip route 0.0.0.0 0.0.0.0 GigabitEthernet0/2 dhcp ?
 <1-255>  Distance metric for this route
```

Example 15-20 demonstrates the workaround where the F root DNS servers (192.5.5.241) were selected as the host routes. The F root DNS server is used by recursive DNS servers but should never be queried directly by hosts. A second static route is directed to the host (F root DNS server) to recursively use this next hop and can add tracked object capability. Even when the interface is shut down, this default route is dependent on the tracked object. It works based on the tracked object changing state.

Example 15-20 *Static Default Routes with DHCP Interface Workaround*

```
ip route 192.5.5.241 255.255.255.255 GigabitEthernet0/2 dhcp 10
ip route 0.0.0.0 0.0.0.0 GigabitEthernet0/2 192.5.5.241 10 track 100
```

The following commands are used to verify the components of a router's Internet connectivity:

- **show ip route track-table** displays any routes that are dependent upon tracking and the tracked state.
- **show track** *track-number* displays the status of a tracked object. If the tracked object contains nested child objects, their status is shown as well.
- **show ip sla statistics** *ip-sla-number* displays the current state of the IP SLA, count of successes and failures, and the last operation start time.

Example 15-21 demonstrates the use of each of these commands and provides the output that is associated to each command.

Example 15-21 *R41 Verification of Tracked Default Route*

```
R41-Spoke# show ip route track
ip route 0.0.0.0 0.0.0.0 GigabitEtherne0/2 192.5.5.241 10 track 100 state is [up]
```

```
R41-Spoke# show track 100
Track 100
  List threshold weight
  Threshold Weight is Up (70/70)
    3 changes, last change 00:00:45
    object 20 Up (10/70)
    object 21 Up (10/70)
    object 22 Up (10/70)
    object 23 weight 20 Up (20/70)
    object 24 weight 20 Up (20/70)
  Threshold weight down 21 up 40
  Delay up 30 secs
R41-Spoke# show track 24
Track 24
  IP SLA 212 reachability
  Reachability is Up
    2 changes, last change 00:00:45
  Latest operation return code: OK
  Latest RTT (millisecs) 25
  Tracked by:
    Track List 100
R41-Spoke# show ip sla statistics 202
IPSLAs Latest Operation Statistics

IPSLA operation id: 202
        Latest RTT: 25 milliseconds
Latest operation start time: 23:11:08 EST Sun Jan 3 2016
Latest operation return code: OK
Number of successes: 45
Number of failures: 0
Operation time to live: Forever
```

Network Address Translation (NAT)

The global routing table now has a default route to the Internet and the router can verify connectivity. The next step is to provide the internal user IP addresses with NAT using the IP address assigned to the Internet interface. The logic must accommodate

IP addresses assigned dynamically from DHCP or statically assigned. Although some companies have publicly routable IP addresses, these addresses are reserved for hosting services out of their data center. NAT is considered a requirement at the branch in all cases.

The process for configuring NAT for internal user traffic follows:

Step 1. **Define the outside interface for NAT.**

Enter interface configuration submode for the guest access network with the command **interface** *interface-id*.

Identify the outside interface for NAT with the command **ip nat outside**.

Step 2. **Define the inside interface for NAT.**

Enter interface configuration submode for the guest access network with the command **interface** *interface-id*.

Identify the inside interface for NAT with the command **ip nat inside**.

Step 3. **Enable NAT.**

NAT is enabled on the branch router for internal users with the command **ip nat inside source** {**list** {*access-list-number* | *access-list-name*} | **route-map** *name*} **interface** *interface-id* [**overload**].

The *interface-id* is the outside interface attached to the Internet. The **overload** keyword enables many-to-one addressing through port address translation and is required in this scenario.

Although the route map is not necessary in the IWAN hybrid model, it is necessary in designs with multiple Internet providers. The book's example demonstrates the configuration that supports any of the designs.

Step 4. **Create an ACL to define the traffic to be translated.**

A standard or extended ACL can be configured to identify the traffic that should be translated. Because of the explicit definition of traffic, an extended ACL is used. The extended ACL is defined with the command **ip access-list extended** *access-list-name*. The ACE entries are defined with {**permit** | **deny**} *source-subnet source-wildcard-netmask* [*destination-subnet destination-wildcard-netmask* | **any**].

Step 5. **Create a route map for NAT.**

The route map should match on the destination interface and the destination network traffic defined in the ACL in Step 4. The route map is created with the command **route-map** *route-map-name* [*sequence-number*] *action*. In the route map sequence, the command **match ip address** *access-list-name* defines the ACL from Step 4. The command **match interface** *interface-id* defines the Internet-facing interface. Because the match statements are of different types, both are required for traffic to pass.

Example 15-22 provides a sample configuration for providing NAT to internal users.

Example 15-22 *R41 Internal Internet Access Network Address Translation*

```
R41
! The following two lines are all one command in the CLI
ip nat inside source route-map RM-INSIDE-TO-ETHERNET02 interface
  GigabitEthernet0/2 overload
!
route-map RM-INSIDE-TO-ETHERNET02 permit 10
 match ip address ACL-INSIDE
 match interface GigabitEthernet0/2
!
ip access-list extended ACL-INSIDE
 permit ip 10.4.4.0 0.0.0.255 any
!
interface GigabitEthernet0/2
 description Internet Link
 vrf forwarding INET01
! The DHCP server is assigning the IP address of 100.64.41.1 to this interface
 ip address dhcp ip nat outside
!
interface GigabitEthernet1/0
 description SITE-LAN
 ip address 10.4.4.1 255.255.255.0
 ip nat inside
end
```

Policy-Based Routing (PBR)

At this point of the configuration, traffic from the internal network is transmitted from the router (leaked through the FVRF), but the FVRF is not aware of any of the networks in the global routing table. *Policy-based routing (PBR)* is used to return the traffic from the FVRF to the global routing table. PBR functions after firewall and NAT processing, so the router matches on the internal IP addressing. To simplify the configuration (for templating purposes), the enterprise aggregate non-guest subnet (10.0.0.0/8) can be used across all branch locations.

Note ZBFW sends TCP resets for traffic that should not be established or has ended and timed out. ZBFW traffic is locally sourced and not processed by the incoming interface. A local PBR policy configuration is necessary to support this traffic.

The following process is used to configure PBR:

Step 1. **Create an extended ACL for the internal destination network.**

An extended ACL can be configured to identify the internal networks that are being translated by NAT so that the FVRF knows where to route the traffic.

The extended ACL is defined with the command **ip access-list extended** *access-list-name*. The ACE entries are defined with {**permit** | **deny**} *source-subnet source-wildcard-netmask* [*destination-subnet destination-wildcard-netmask* | **any**].

Step 2. **Create a route map for PBR.**

The route map is created with the command **route-map** *route-map-name* [*sequence-number*] *action*. In the route map sequence, the command **match ip address** *access-list-name* defines the ACL from Step 1.

The command **set global** forces packets to route through the global routing table.

Step 3. **Assign the PBR policy on the Internet interface.**

The route map is associated to the Internet interface so that traffic received from the Internet is policy-based routed. The command **ip policy route-map** *route-map-name* enables PBR with the route map.

Step 4. **Assign a system PBR policy for locally generated traffic.**

PBR works on transit network traffic by default. The policy must be enabled globally with the command **ip local policy route-map** *route-map-name* for traffic generated by the router. This is required to support ZBFW's locally generated TCP resets for traffic that should not be established or has ended and timed out.

Example 15-23 provides the PBR configuration to allow return traffic to be forwarded from the INET01 FVRF to the 10.4.4.0/24 network.

Example 15-23 *R41 Internal Internet Access Global Table Restoration*

```
R41
ip access-list extended ACL-INSIDE-RESTORATION
 permit ip any 10.4.4.0 0.0.0.255
!
route-map RM-RESTORE-GLOBAL permit 10
 match ip address ACL-INSIDE-RESTORATION
 set global
!
interface GigabitEthernet0/2
 description Internet Link
 vrf forwarding INET01
```

```
! The DHCP server is assigning the IP address of 100.64.41.1 to this interface
 ip address dhcp
 ip nat outside
 ip policy route-map RM-RESTORE-GLOBAL
!
ip local policy route-map RM-RESTORE-GLOBAL
end
```

Internal Access Zone-Based Firewall (ZBFW)

Now that connectivity has been established, the internal users need to be provided with security too. A stateful firewall should be deployed so that only return traffic is allowed. The ZBFW is already configured for guest access to the Internet, so a new policy needs to be created for internal users.

The Outside security zone is already defined and on the Internet-facing interface. An internal zone needs to be defined, along with the corresponding policy. The new internal zone needs to be defined on all internal interfaces, including the DMVPN tunnel interfaces. Intra-zone traffic is allowed by default, but inter-zone traffic must be explicitly defined.

As explained in Chapter 5, the current iteration of PfRv3 uses dynamic auto-tunnels that cannot be configured or assigned to a security zone. By using the ZBFW's *default zone* security zone for all the inside interfaces, security policies can be built as needed and allow PfRv3 to work properly. The default zone is not enabled by default and must be initialized in the configuration. The default zone is used on inside interfaces in order to allow ZBFW and PfRv3 to function in a dual-router deployment.

Note The default zone for ISR-G2s running IOS is released in 15.6(1)T. Configuring **zone security default** in IOS is accepted but does not function as described here.

The process for creating the ZBFW configuration for internal Internet access is as follows:

Step 1. **Define the security zones.**

Zones are configured using the command **zone security** *zone-name*. The default zone needs to be initialized. The Outside zone was created earlier.

Step 2. **Define the inspection class map.**

The class map for inspection defines a method for classification of traffic. The class map is configured using the command **class-map type inspect** [**match-all** | **match-any**] *class-name*. The **match-all** keyword requires that network traffic match all the conditions listed to qualify (Boolean AND), whereas the **match-any** keyword requires that network traffic match only one of the conditions listed to qualify (Boolean OR). If neither keyword is specified, the **match-all** function is selected.

Step 3. **Define the inspection policy map.**

The inspection policy map applies firewall policy actions to the class maps defined in the policy map. The policy map is then associated to a zone pair.

The inspection policy map is defined with the command **policy-map type inspect** *policy-name*. After the policy map is defined, the various class maps are defined with the command **class type inspect** *class-name*. Under the class map, the firewall action is defined with these commands:

- **drop** [**log**]: This is the default action, which silently discards packets that match the class map. The **log** keyword adds syslog information that includes source and destination information (IP address, port, and protocol).
- **pass** [**log**]: This action makes the router forward packets from the source zone to the destination zone. Packets are forwarded in only one direction. A policy must be applied for traffic to be forwarded in the opposite direction. The **pass** action is useful for protocols such as IPsec ESP and other inherently secure protocols with predictable behavior. The optional **log** keyword adds syslog information that includes the source and destination information.
- **inspect**: The **inspect** action offers state-based traffic control. The router maintains connection/session information and permits return traffic from the destination zone without the need to specify it in a second policy.

The inspection policy map has an implicit class default that uses a default **drop** action. This provides the same implicit "deny all" that is found in an ACL. Adding it to the configuration may simplify troubleshooting for junior network engineers.

Step 4. **Define the zone pairs.**

A policy map is now applied to a traffic flow source to a destination configured as **zone-pair security** *zone-pair-name* **source** *source-zone-name* **destination** *destination-zone-name*. The inspection policy map is then applied to the zone pair with the command **service-policy type inspect** *policy-name*.

Step 5. **Apply the security zones to the appropriate interfaces.**

An interface is assigned to the appropriate zone by entering interface configuration submode with the command **interface** *interface-id* and associating the interface to the correct zone with the command **zone-member security** *zone-name* as defined in Step 1.

The Outside security zone is associated to the Internet interface, and the default zone interfaces are associated automatically when the default zone is initialized.

Example 15-24 provides the ZBFW configuration for internal users to have Internet access.

Example 15-24 *Internal Zone-Based Firewall Configuration*

```
R41
zone security default
 description default zone used for Inside Network
zone security OUTSIDE
 description OUTSIDE Zone used for Internet Interface
!
class-map type inspect match-any CLASS-INSIDE-TO-OUTSIDE
 match protocol ftp
 match protocol tcp
 match protocol udp
 match protocol icmp
policy-map type inspect POLICY-INSIDE-TO-OUTSIDE-
 class type inspect CLASS-INSIDE-TO-OUTSIDE
  inspect
 class class-default
  drop
zone-pair security DEFAULT-TO-OUTSIDE source default destination OUTSIDE
 service-policy type inspect POLICY-INSIDE-TO-OUTSIDE
!
interface GigabitEthernet0/2
 description Internet Link
 vrf forwarding INET01
! The DHCP server is assigning the IP address of 100.64.41.1 to this interface
 ip address dhcp
 zone-member security OUTSIDE
```

Sometimes security teams still require the use of an external firewall for this type of connectivity, or the network team and security team need to provide *role-based access control (RBAC)* and monitoring to be separated. For an external security appliance, simply removing the NAT and firewall configuration on the router provides exactly what is needed. When NAT is used on the external appliance, the DMVPN sessions use *NAT traversal (NAT-T)* on UDP/4500. The router still most likely requires tracked object configuration, unless the external appliance can provide link propagation (shut down the inside interface when its own monitoring finds that the path is unavailable).

Cloud Web Security (CWS)

Limiting access to inappropriate sites and content from a distributed Internet model may seem daunting because ensuring that all sites provide a consistent policy seems difficult. Cisco *Cloud Web Security (CWS)* integration with IWAN addresses this concern. Cloud Web Security, as the name suggests, is a web proxy service that is delivered via a public cloud. Unlike most web proxies that require configuration on a client (through a PAC file, DNS, or other means), CWS uses a cloud connector that inspects and redirects HTTP and HTTPS to the CWS. As a cloud-delivered application, CWS provides security for the distributed branch network against web-initiated threats without the need to provide configuration information to end-user devices.

CWS uses centralized policies that can be configured based on the function of the user. For example, a policy can be associated with all internal users and a different policy applied for the guest network. These policies are then consistently applied to all sites based on the role of the person, not geographic location. The CWS cloud service is hosted in multiple data centers (referred to as *towers*) so that users proxy traffic to the tower that is closest to their location router.

Cisco CWS is not just a content control system that classifies websites into categories. CWS includes the capability to protect users from known and unknown threat vectors via a worldwide collection of threat intelligence and defense capabilities. CWS provides the following advantages:

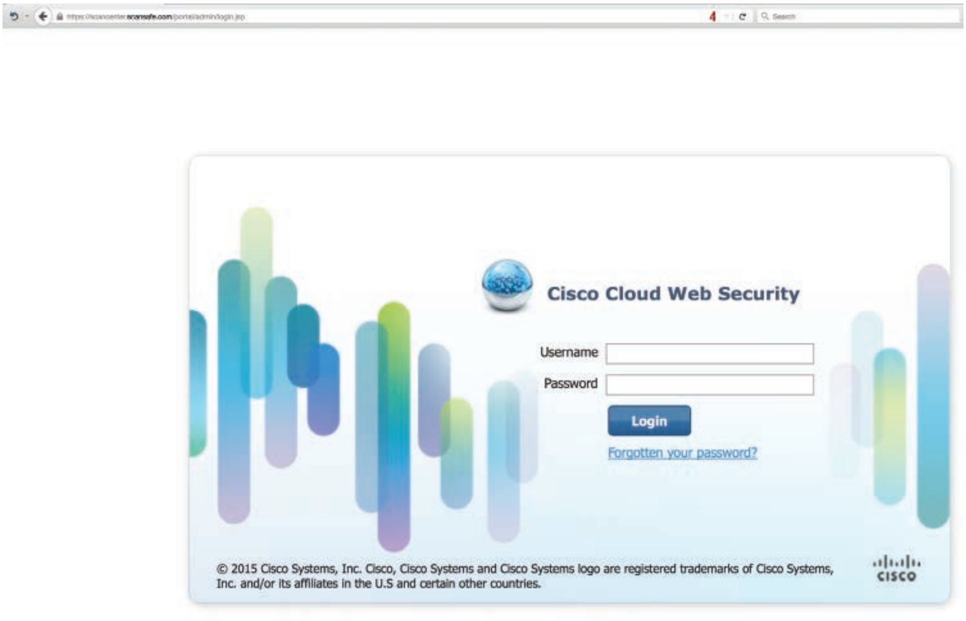
- It provides zero-day defense through multiple heuristics engines, signatures, and more through a single cloud-delivered service.
- It analyzes more than 100 TB of security intelligence and 13 billion web requests daily to detect and mitigate threats.
- It provides granular visibility and control for more than 150,000 application and micro-applications.
- It integrates with *Advanced Malware Protection (AMP)* and *Cognitive Threat Analytics (CTA)* which increase visibility and intelligence into malware and other breaches that could be present in a network.
- It migrates security infrastructure from a CapEx to an OpEx operating model.
- It provides extended granular control with a variety of methods of associating users to policies and reporting of activities.

There are a couple of possible options for deployment of CWS on a router. What if the security team wanted to continue to use a centralized security policy for traffic that is not web based? This is one of the first use cases for CWS. The branch routers use the default route advertised from the DMVPN hub routers to forward all traffic to the centralized site. However, the DMVPN tunnel interfaces have applied the outbound proxy capture settings on the DMVPN tunnel, and only the HTTP/HTTPS traffic is directed toward the CWS tower for inspection. Connectivity to the CWS tower is achieved via the Internet-facing FVRF interface.

This type of architecture maintains the existing security model for non-web traffic within the already deployed security stack at headquarters while offloading web traffic and applications directly at the branch site. It also establishes a single policy management, reporting, and enforcement location for all branch sites.

Baseline Configuration

The CWS centralized policy control is managed via ScanCenter at https://scancenter.scansafe.com, and the cloud connector requires minimal configuration on the router. Figure 15-6 displays the login page for CWS.



https://scancenter.scansafe.com/portal/admin/login.jsp

© 2015 Cisco Systems, Inc. Cisco, Cisco Systems and Cisco Systems logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S and certain other countries.

Figure 15-6 *CWS ScanCenter Login Screen*

Upon login to ScanCenter, a baseline configuration is created and the license key is entered for using CWS. Figure 15-7 displays the login page.

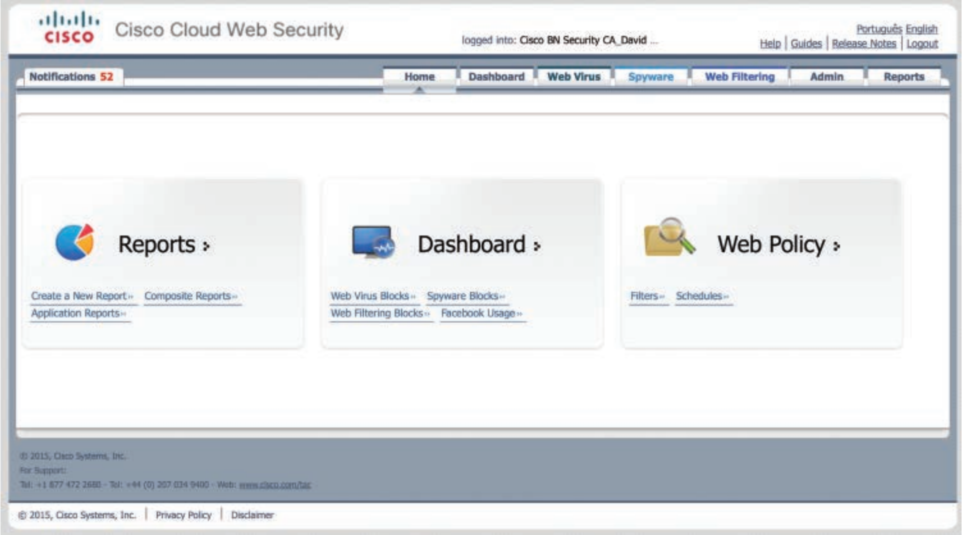
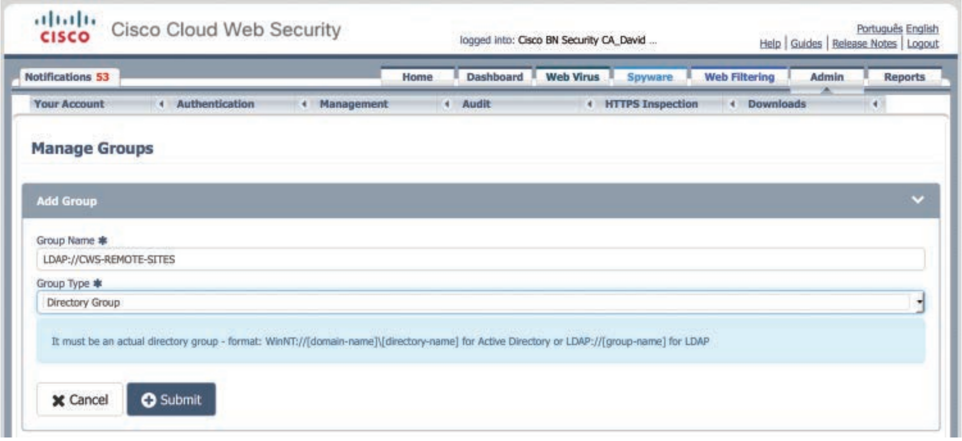


Figure 15-7 *CWS ScanCenter Home Page*

For the book's configuration, static group mappings are used. Although both LDAP and Active Directory can be used for authentication, static group mappings are a simple configuration option for both internal and guest user access. In the sample configuration, the group CWS-REMOTE-USERS is created for internal users and CWS-GUEST is created for guest users.

Configuring groups is accomplished by selecting **Admin** in the top toolbar, then **Management** in the pull-down, then **Group**. Within the **Group** page, select **Add Group** to configure a new group. These groups are configured as LDAP Directory Groups within the CWS configuration as shown in Figure 15-8.



Group Name *
LDAP://CWS-REMOTE-SITES

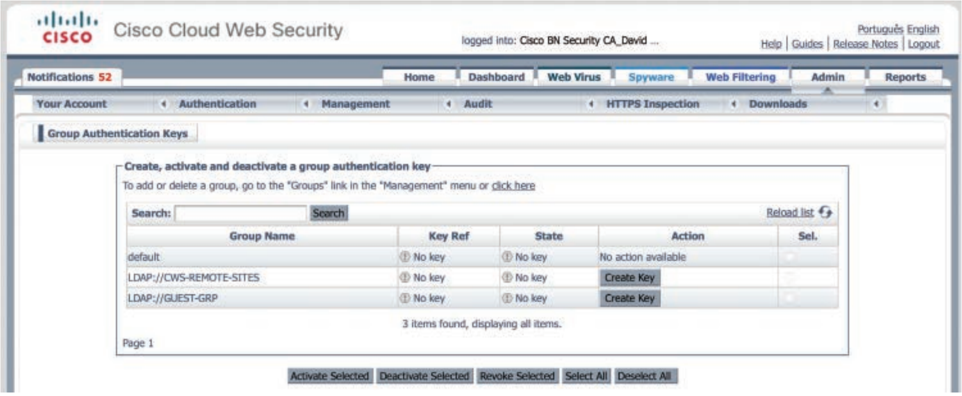
Group Type *
Directory Group

It must be an actual directory group - format: WinNT://[domain-name]\[directory-name] for Active Directory or LDAP://[group-name] for LDAP

Figure 15-8 *CWS ScanCenter Add Group*

After a group is defined, a license key is needed for authentication. This key is used on the routers to map the groups to your account. It is important that this key be configured properly on the router.

License keys are created by selecting **Admin** in the top toolbar, then **Authentication** in the pull-down, then **Group Keys**. Within the **Group Key** page, select **Create Key** as shown in Figure 15-9.



Create, activate and deactivate a group authentication key

To add or delete a group, go to the "Groups" link in the "Management" menu or click here

Search: Search Reload list

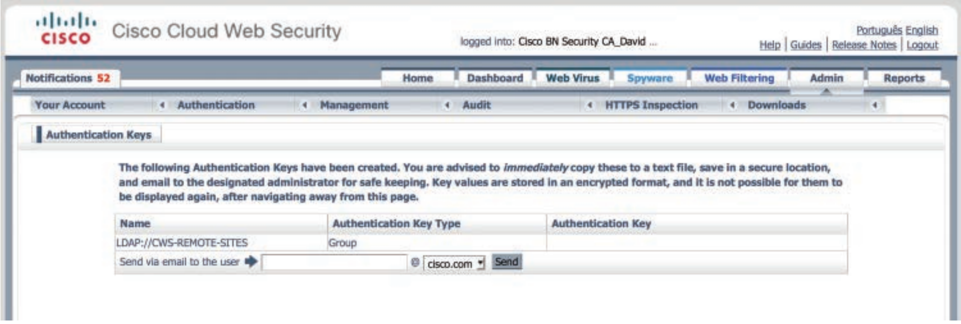
Group Name	Key Ref	State	Action	Sel.
default	No key	No key	No action available	
LDAP://CWS-REMOTE-SITES	No key	No key	Create Key	
LDAP://GUEST-GRP	No key	No key	Create Key	

3 items found, displaying all items.

Page 1

Figure 15-9 *CWS ScanCenter Admin > Authentication > Group Keys*

Now that the key is available, it needs to be configured on the router. It can either be copied from the screen or emailed to a user within the configured account domain. The best way to save the key is to have the key emailed as a method of record retention in case the configuration is unexpectedly delayed. Figure 15-10 displays the new license key.



The following Authentication Keys have been created. You are advised to *immediately* copy these to a text file, save in a secure location, and email to the designated administrator for safe keeping. Key values are stored in an encrypted format, and it is not possible for them to be displayed again, after navigating away from this page.

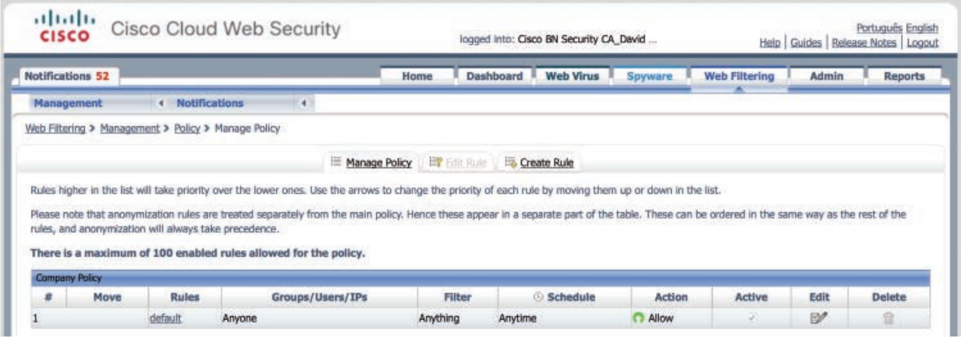
Name	Authentication Key Type	Authentication Key
LDAP://CWS-REMOTE-SITES	Group	

Send via email to the user @ cisco.com Send

Figure 15-10 *CWS ScanCenter Admin Group Keys Creation*

CWS policy management allows the use of a default policy, which is always the last policy applied if all other policies are not applied. Alternatively, it can be the only policy used. The book uses this as the only policy for internal users.

Figure 15-11 displays the Policy Management screen.



Web Filtering > Management > Policy > Manage Policy

Rules higher in the list will take priority over the lower ones. Use the arrows to change the priority of each rule by moving them up or down in the list.

Please note that anonymization rules are treated separately from the main policy. Hence these appear in a separate part of the table. These can be ordered in the same way as the rest of the rules, and anonymization will always take precedence.

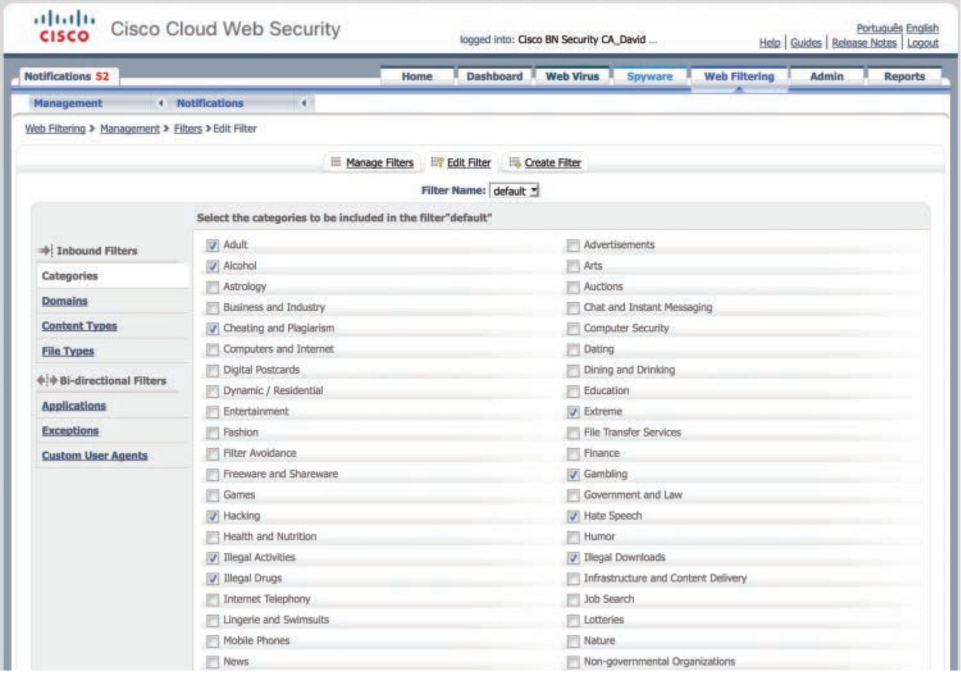
There is a maximum of 100 enabled rules allowed for the policy.

Company Policy

#	Move	Rules	Groups/Users/IPs	Filter	Schedule	Action	Active	Edit	Delete
1		default	Anyone	Anything	Anytime	Allow	✓		

Figure 15-11 *CWS ScanCenter Web Filtering > Management > Policy*

Clicking on **Edit** takes us to the Edit Filter window as shown in Figure 15-12. Here specific Categories, Domains, Content Types, and/or File Types can be set so that they "allow" or "deny" based on their application in the policy definition. For the default policy we have chosen categories to which we want to deny our users access, some of which can be seen in Figure 15-12.



Web Filtering > Management > Filters > Edit Filter

Filter Name: default

Select the categories to be included in the filter"default"

- ☑ Adult
- ☑ Alcohol
- □ Astrology
- □ Business and Industry
- ☑ Cheating and Plagiarism
- □ Computers and Internet
- □ Digital Postcards
- □ Dynamic / Residential
- □ Entertainment
- □ Fashion
- □ Filter Avoidance
- □ Freeware and Shareware
- □ Games
- ☑ Hacking
- □ Health and Nutrition
- ☑ Illegal Activities
- ☑ Illegal Drugs
- □ Internet Telephony
- □ Lingerie and Swimsuits
- □ Mobile Phones
- □ News
- □ Advertisements
- □ Arts
- □ Auctions
- □ Chat and Instant Messaging
- □ Computer Security
- □ Dating
- □ Dining and Drinking
- □ Education
- ☑ Extreme
- □ File Transfer Services
- □ Finance
- ☑ Gambling
- □ Government and Law
- ☑ Hate Speech
- □ Humor
- ☑ Illegal Downloads
- □ Infrastructure and Content Delivery
- □ Job Search
- □ Lotteries
- □ Nature
- □ Non-governmental Organizations

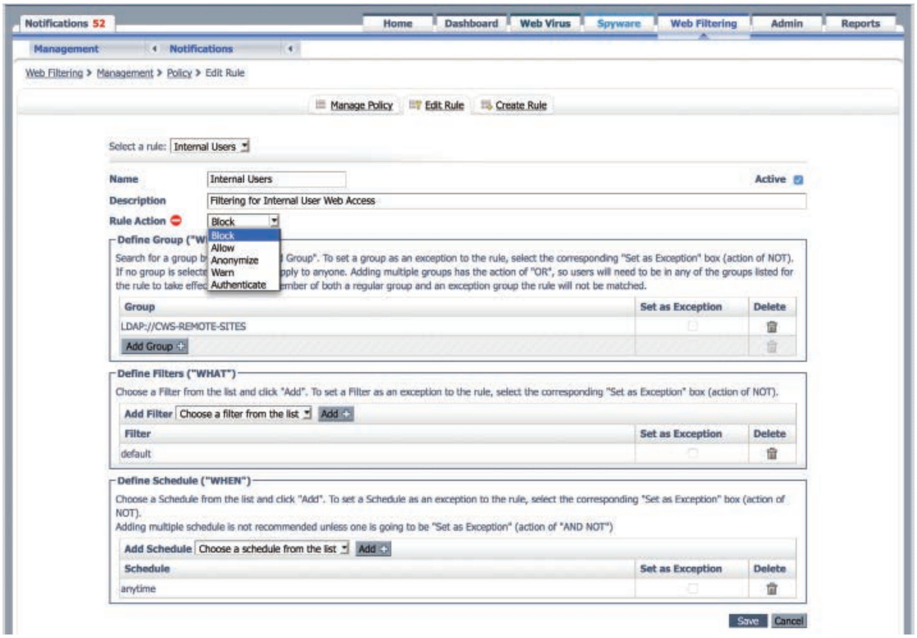
Figure 15-12 *CWS ScanCenter Web Filtering > Management > Filters*

Now a rule is created for the defined group. The rule can be applied with multiple options: **Allow**, **Block**, **Anonymize**, **Warn**, or **Authenticate**. A scenario for each of these is provided here:

- A typical deployment uses Block to specify categories that are not allowed when access is attempted. CWS specifies the site that was attempted and why it was blocked.
- A filter can be conditionally matched by the site's categorization, domain, content type, or a specific file type. The filtering action can consist of Warn or Authenticate. If warned, a user can continue to the site.
- Allow permits access to known sites that are required for business use cases.
- Authenticated requires that the user needs to authenticate in order to proceed. Authentication requires that LDAP or Active Directory authentication be configured within ScanCenter, a topic not covered in this book.
- Anonymize is equivalent to Allow, but the user tracking data is reported as undisclosed.

When applying the filters, by default they are treated as specified by the Policy Rule Action, or they can Set as Exception to use this filter as an exception to the Policy Rule Action. Setting as an exception in the case of Block would treat everything other than the exception as a block.

Figure 15-13 illustrates the Policy Create Rule function.



Select a rule: Internal Users

Name: Internal Users

Active ☑

Description: Filtering for Internal User Web Access

Rule Action: Block

Define Group ("W

Search for a group b Group". To set a group as an exception to the rule, select the corresponding "Set as Exception" box (action of NOT). If no group is selecte pply to anyone. Adding multiple groups has the action of "OR", so users will need to be in any of the groups listed for the rule to take effec ember of both a regular group and an exception group the rule will not be matched.

Group	Set as Exception	Delete
LDAP://CWS-REMOTE-SITES	□	🗑

Define Filters ("WHAT")

Choose a Filter from the list and click "Add". To set a Filter as an exception to the rule, select the corresponding "Set as Exception" box (action of NOT).

Add Filter: Choose a filter from the list

Filter	Set as Exception	Delete
default	□	🗑

Define Schedule ("WHEN")

Choose a Schedule from the list and click "Add". To set a Schedule as an exception to the rule, select the corresponding "Set as Exception" box (action of NOT).
Adding multiple schedule is not recommended unless one is going to be "Set as Exception" (action of "AND NOT")

Add Schedule: Choose a schedule from the list

Schedule	Set as Exception	Delete
anytime	□	🗑

Figure 15-13 *CWS ScanCenter Web Filtering > Policy > Create Rule*

Outbound Proxy

The next phase of configuring CWS is to place the cloud connector configuration on the router. The configuration consists of the following steps:

Step 1. **Define the CWS servers.**

The CWS servers are defined under the CWS parameter map. The CWS parameter map is entered with the command **parameter-map type cws global**.

The CWS tower IPs (primary and secondary) are defined with the command **server** {**primary** | **secondary**} **ipv4** *tower-ip-address* **port http** *http-port* **https** *https-port*. Port 8080 is typically used for HTTP and HTTPS.

Step 2. **Define the CWS license.**

The CWS license that was created and emailed earlier is then associated to the router with the command **license** *license-key*.

Step 3. **Define the source interface.**

The source interface that connects to the CWS towers must be defined with the command **source interface** *interface-id*. The Internet interface is the one used.

Step 4. **Define the user group for policy processing.**

A user group can be specified for default processing in the event that the user group cannot be identified. The command **user-group** *default-user-group* applies it to the CWS policy map.

Step 5. **Define the access policy for when connectivity to CWS fails.**

In the event that the CWS cloud cannot be reached, traffic can be allowed or blocked. The command **server on-failure** [**allow-all** | **block-all**] sets the policy.

Step 6. **Identify the interface that CWS should protect.**

The interface of the CWS clients needs to be defined. The interface parameter command **cws out** is placed on the interface facing the client devices.

Step 7. **Define a whitelist for internal network traffic.**

Not all traffic needs to be sent to CWS. Web-based applications or websites that are hosted out of the data center can be whitelisted so that they are sent across the WAN interface. CWS whitelisting is based on an extended ACL.

After creating the extended ACL, the global configuration command **cws whitelisting** is entered. Underneath that configuration mode, the whitelist ACL is defined with the command **cws whitelist** *acl-name*.

Step 8. **Add TCP port 8080 to the ZBFW Self zone.**

If the router uses ZBFW, the policy map must include a class map that permits TCP port 8080 to the towers so that traffic can be inspected and returned to the router.

In Example 15-25, the default route for internal users comes from the DMVPN tunnel interfaces. The configuration uses the towers for all web traffic with ScanCenter policy control, while leaving other Internet-bound applications to use a centralized Internet model. This deployment model uses DIA with CWS for web traffic, offloading the majority of Internet-bound traffic.

Example 15-25 *R41 Cloud Web Security Configuration*

```
parameter-map type cws global
 server primary ipv4 72.37.248.27 port http 8080 https 8080
 server secondary ipv4 69.174.58.187 port http 8080 https 8080
 license <license>
 source interface GigabitEthernet0/2
 user-group CWS-REMOTE-SITES
 server on-failure allow-all
!
! Add tcp/8080 to ZBFW for SELF to OUTSIDE
ip access-list extended ACL-RTR-OUT
 permit tcp any any eq 8080
!
ip access-list extended ACL-CWS-EXCLUDE
 permit ip any 10.0.0.0 0.255.255.255
!
cws whitelisting
 whitelist acl name ACL-CWS-EXCLUDE
!
interface Tunnel100
 cws out
!
interface Tunnel200
 cws out
end
```

For guest traffic filtering and logging, a new management group based on Lightweight Directory Access Protocol (LDAP) is created. In this case, the group LDAP://GUEST-GRP is used and associated to a dedicated policy for this group. The new group is then applied to the guest user ingress interface with the command **user-group default** *cws-user-group*. The CWS **proxy out** (capture) command is applied to the Internet FVRF interface. The validation of CWS can be done by accessing http://whoami.scansafe.net. Example 15-26 demonstrates the configuration.

Example 15-26 *R41 Cloud Web Security Proxy Capture Configuration*

```
interface GigabitEthernet1/1
 user-group default GUEST-GRP
!
interface GigabitEthernet0/2
 cws out
end
```

The status of the CWS session can be seen with the command **show cws** {**summary** | **session** [**active**] | **statistics**}. Example 15-27 displays the output of the various iterations of these commands.

Example 15-27 *R41 Cloud Web Security Verification*

```
R41-Spoke# show cws summary
Primary: 72.37.248.27 (Up)*
Secondary: 69.174.58.187 (Up)
Interfaces: Tunnel100 Tunnel200
```

```
R41-Spoke# show cws session active
Protocol      Source        Destination     Bytes             Time
```

```
R41-Spoke# show cws statistics
Current HTTP sessions: 0
Current HTTPS sessions: 0
Total HTTP sessions: 0
Total HTTPS sessions: 0
White-listed sessions: 0
Host Whitelisted session: 0
DNS Whitelisted session: 0
Time of last reset: never
-----------------------------------
Details:
Max Concurrent Active Sessions: 0
Connection Rate in last minute:
    Redirected
        HTTP: 0
        HTTPS: 0
    White-listed
        IP-Based: 0
        User/User-group: 0
        Header-Based: 0
Max Connection Rate per minute:
    Redirected
        HTTP: 0
        HTTPS: 0
    White-listed
        IP-Based: 0
        User/User-group: 0
        Header-Based: 0
```

WAAS and WCCP Redirect

Redirecting traffic to the WAAS appliance for DIA Internet traffic is not optimal because there is a significant delay in all flows while the WAAS appliance determines that it is not paired with another WAAS at the destination site. When WCCP is used to redirect traffic to WAAS, Internet-bound traffic needs to bypass WAAS redirection.

For example, R41 uses CWS on the tunnel interface for redirection of only web-based traffic, whereas other TCP traffic types use a centralized Internet access model. In scenarios like these, using WAAS for WAN acceleration of the non-web-based traffic is desired. Example 15-28 shows the enablement of redirection to WAAS for HTTP and HTTPS traffic destined for internal 10.0.0.0/8 address space, denial of HTTP and HTTPS traffic destined for the Internet, and redirection of any other TCP traffic to WAAS.

Example 15-28 *R41 WAAS Redirect Bypass Configuration for CWS-Only DIA*

```
R41
ip access-list extended ACL-WAAS-REDIRECT-LIST
 permit tcp any 10.0.0.0 0.255.255.255 eq www
 permit tcp any 10.0.0.0 0.255.255.255 eq 443
 deny tcp 10.0.0.0 0.255.255.255 any eq www
 deny tcp 10.0.0.0 0.255.255.255 any eq 443
 permit tcp any any
!
ip wccp 61 redirect-list ACL-WAAS-REDIRECT-LIST group-list WAVE
ip wccp 62 redirect-list ACL-WAAS-REDIRECT-LIST group-list WAVE
```

Prevention of Internal Traffic Leakage to the Internet

Now that DIA is functioning correctly, protecting internal data should be of concern. The branch routers have a default route to the Internet from their directly connected Internet interfaces, and they receive internal routing over the DMVPN tunnel interfaces.

What happens if the DMVPN tunnel interfaces cannot be established with the DMVPN hub routers, and the Internet path remains up? Internal network traffic could be sent out to the Internet. Although this traffic would be unusable for all practical purposes, a large-enough TCP window could expose some useful information if someone is watching. To ensure that internal network traffic is not sent externally (where it does not belong), a floating static route to the Null interface is used for internal prefixes.

Example 15-29 demonstrates the concept of the floating static null route for the 10.0.0.0/8 network that matches all the networks in this book's topology.

Example 15-29 *R41 Internal Traffic Denial When Only the Default Route Is Available*

```
ip route 10.0.0.0 255.0.0.0 null0 254
```

Summary

Direct Internet access is a key component of the IWAN solution. Solving everyday issues with application performance and bandwidth constraints is the key objective. Organizations are paying twice and adding delay for Internet bandwidth with a centralized Internet access model. DIA provides users with better network and application performance for software as a service (SaaS) offerings.

DIA is a great way to address tight IT budgets while giving branch users the bandwidth they need. As branch employees use more public cloud applications as a core part of their business activities, they get much better performance going directly to the Internet than if that traffic were backhauled over the corporate WAN.

Cisco Zone-Based Firewall (ZBFW) is an integrated stateful firewall in IOS-based operating systems. ZBFW is capable of examining Layers 4 through 7 of a network packet and verifying the state of transmission. ZBFW provides segmentation between interfaces and mitigates against DDoS threats.

IWAN's architecture uses Cisco Cloud Web Security (CWS) to provide a web proxy service that is delivered via a public cloud. CWS uses a cloud connector that inspects and redirects HTTP and HTTPS to the CWS and provides security against web-initiated threats in the distributed branch network without the need to provide configuration information to end-user devices.

Cisco continues to invest and introduce new security features in all its products. Future IWAN versions will include new DIA security technologies for applications outside of HTTP/HTTPS. Expect to see the following Cisco technologies in future IWAN versions:

- **Cisco Umbrella Branch:** Cisco Umbrella Branch is a cloud-delivered security service for the Cisco ISR. It provides the first layer of defense against threats at branch offices and offers easy-to-manage DNS-layer content filtering based on categories as well as reputation. Cisco Umbrella Branch prevents branch users and guests from accessing inappropriate content and known malicious sites that might contain malware and other security risks. It provides visibility and enforcement at the DNS layer, so requests to malicious domains and IPs are blocked before a connection is ever made.

- **Snort IPS:** This is an open-source network intrusion detection and prevention system capable of performing real-time traffic analysis and packet logging on IP networks based on a defined rule set. Cisco Snort IPS has two major components: a detection engine (Snort Engine) and a flexible rule language (Snort Rules) to describe traffic to be collected. The Snort Engine runs in the service container of the Cisco 4000 Series ISR. It provides lightweight threat defense for compliance with Payment Card Industry Data Security Standards (PCI DSS) and other regulatory compliance mandates.

- **Cisco Firepower Threat Defense for ISR:** This takes enterprise-level threat protection beyond Snort IPS functionality and combines full-stack traffic analysis by providing Intrusion Protection Systems (IPS), Application Visibility, URL Filtering, and Advanced Malware Protection (AMP) in one solution. Firepower Threat Defense maintains exceptional visibility into what is running on the network and provides intelligent security automation. It identifies and stops threats before they affect extended enterprise operations.

References in this Chapter

Cisco. *Authentication Proxy Configuration Guide*. www.cisco.com.

Cisco. “Cisco IOS Software Configuration Guides.” www.cisco.com.

Cisco. “Customizing Authentication Proxy Web Pages.” www.cisco.com.

Chapter 16

Deploying Cisco Intelligent WAN

This chapter covers the following topics:

- Pre-migration tasks
- Preparing the existing WAN
- Migrating point-to-point WAN technologies
- Migrating multipoint WAN technologies

The previous chapters explained the Cisco Intelligent WAN (IWAN) technologies and architecture. By now, the cost savings and benefits from Cisco IWAN architecture should be well understood. The last topic in this book is the process for migrating an existing WAN network to Cisco IWAN architecture in the most seamless way possible. Deploying IWAN, like any other network technology, requires proper planning to prevent suboptimal routing and minimize network downtime.

Cisco IWAN architecture is based on application optimization (WAAS and Akamai Connect), direct Internet access (ZBFW, CWS, and encryption), transport-independent overlay (DMVPN), and intelligent path control (PfRv3). Application optimization and direct Internet access (DIA) do not have any dependencies and can be deployed independently of the other technologies. Transport-independent overlay and intelligent path control are the foundational components of the IWAN architecture, and so this chapter will focus on migrating the typical WAN to a DMVPN overlay with PfRv3 for common migration scenarios.

Pre-Migration Tasks

Performing the following pre-migration tasks will help to avoid complications and headaches during the migration. Prep work involves a variety of tasks such as collecting inventory, reviewing the WAN design, and verifying configuration compliance. The following sections go into more detail.

Document the Existing WAN

Proper network design requires sufficient documentation of the existing WAN. Ideally the networking team has physical (Layer1), Layer 2, and logical drawings. The routing design should specify the objectives and requirements and illustrate a high-level design of the existing WAN.

A proper inventory of the routers targeted for DMVPN migration should be collected, including device types, software versions, licensing, and circuit speeds. This information should be reviewed with the features in the IWAN design. Sites should be classified to identify what the DMVPN template configuration will look like. Routers that do not meet performance guidelines should be replaced. Routers with outdated software should be upgraded before or during that site's migration window.

Network Traffic Analysis

WAN circuit link utilization should be collected, and application recognition should be enabled to understand the types of applications flowing across the network. The information collected may be a shock to some network engineers because they may not be aware of the various applications that are used on the network. All applications should be inventoried and categorized as business relevant, non-business relevant, or unknown at this time. Business-relevant applications should be given priority in the network, and non-business-relevant applications should be given the lowest priority.

The network traffic analysis and application classification should be used to help develop a proper QoS design as part of the IWAN deployment. The traffic flowing across the WAN backbone as a whole should be considered when creating the initial QoS and PfR policies. When designing QoS policies, pay special attention to how they are applied at the hub sites. This will impact the DMVPN hub deployment model that is selected.

Some organizations have accelerated the deployment of DIA after reviewing the number of Internet-based applications that are consuming their WAN bandwidth.

Proof of Concept

Up to this point, practicing and applying the concepts in this book are essential to forming a hands-on perspective. Deploying an IWAN *proof of concept (POC)* is encouraged for any customer preparing to deploy Cisco Intelligent WAN architecture. POCs typically do not have the stringent change control requirements of a production network. They can provide a method of verifying a solution's value before fully committing to a technology.

The first remote site for a POC should always be in a lab environment. Separate circuits (transports) should be brought in for testing where feasible to simulate an actual branch router. This allows all the network teams to learn the technology and provide a method to test monitoring and management tools such as Cisco Prime Infrastructure or APIC-EM.

Finalize the Design

The network's end state needs to be properly defined. Are all the branch sites being migrated to IWAN, or will a portion remain on the existing design? The answer will influence any post-migration cleanup activities. Has the documentation been updated to reflect the current IWAN routing design? Does the network have site-to-site multicast in it? The design should be finalized and validated before starting the migration.

Migration Overview

Most IWAN migrations do not occur overnight but are spread out over time and can take days, weeks, or months to complete. During the migration, devices on the DMVPN network can communicate with each other directly through spoke-to-spoke DMVPN tunnels. Devices in the legacy WAN communicate in the existing traffic patterns too. In order for devices on the IWAN network to communicate with devices on the legacy WAN, the traffic must flow through specific sites that are designated to provide transit connectivity. This chapter refers to such a network as a *migration network*.

Figure 16-1 illustrates the concept of the migration network that connects the DMVPN networks with the legacy WAN. In Site 1, R11 is the DMVPN hub router, and CE1 connects to the MPLS SP1 provider network. Site 3 communicates directly with Site 4 using a spoke-to-spoke tunnel in the DMVPN network. Network traffic from Site 3 flows through R11 to reach Site 1 where CE1 forwards the packets on to the legacy MPLS SP1 provider network to reach Site 6.

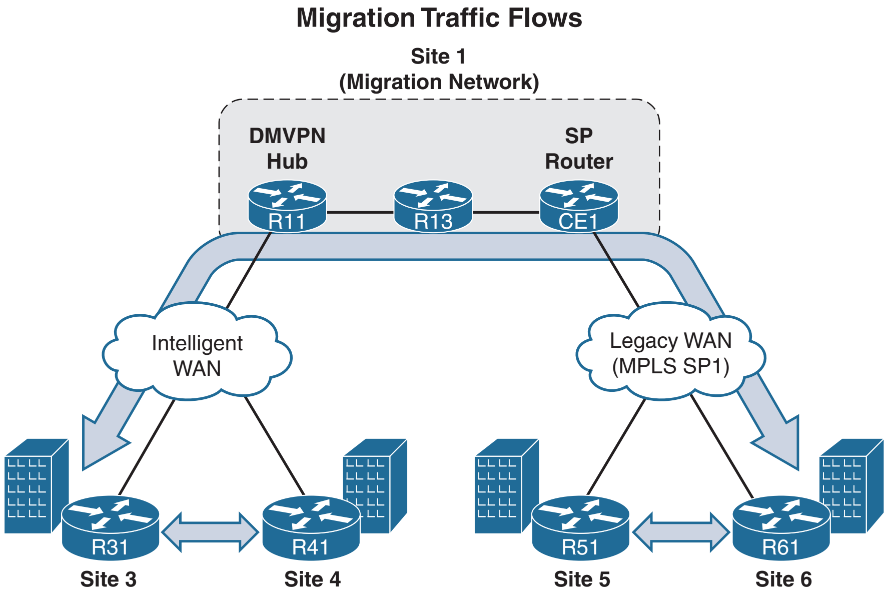


Figure 16-1 *Migration Network Flow*

Note R13 is a component of the network providing WAN aggregation (similar to a pair of WAN distribution switches) and helps visualize the traffic patterns between the legacy WAN and IWAN. R11 and CE1 could be directly connected.

IWAN Routing Design Review

In Chapter 4, "Intelligent WAN (IWAN) Routing," two DMVPN network routing designs were presented; here they are depicted in Figure 16-2. Both designs are expanded upon in the following section to demonstrate how the existing WAN integrates with the IWAN routing architecture during migration.

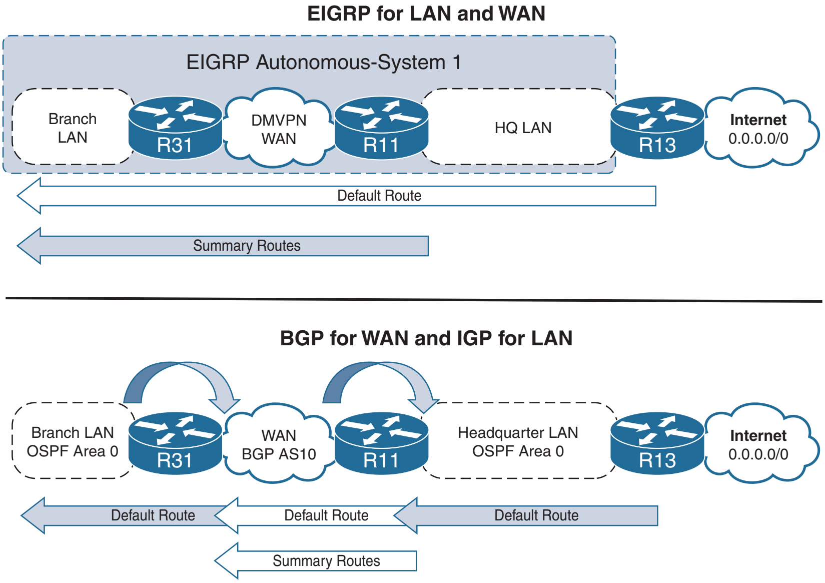


Figure 16-2 *IWAN Routing Design*

EIGRP for the IWAN and the LAN

The IWAN EIGRP design uses EIGRP for the IWAN routing protocol and all the LAN segments, which simplifies the topology because there is no route redistribution. Transit routing was eliminated with the use of the EIGRP stub site feature, and the DMVPN hub routers advertised summary prefixes to shrink the EIGRP table and query domain.

BGP for the IWAN and an IGP (OSPF) for the LAN

The IWAN BGP design uses BGP for the IWAN routing protocol, which redistributes into the IGP protocol for larger LAN networks. This book uses OSPF as the IGP, but it could be EIGRP or IS-IS too. This solution requires the use of route redistribution, which creates a major area of concern. With route redistribution the complete topology is not known from every router's perspective, and it introduces the potential for routing loops.

The branch sites redistribute the locally learned routes (branch LAN) into BGP that are then advertised to the DMVPN hub routers. The DMVPN hub routers summarize where needed, then redistribute the WAN routes into the headquarters LAN so that those devices know where to send the return network traffic. This design does not mutually redistribute between protocols, and it natively prevents routing loops.

The DMVPN hubs advertise a default route for Internet connectivity and summary routes for connectivity for all LAN/WAN networks toward the branch locations; the branch sites always send traffic toward the DVMPN hubs. The DMVPN hub router has all the routes learned from BGP (DMVPN network) or from the IGP (LAN networks).

Routing Design During Migration

Ideally, the routing design for the existing WAN uses the same logic as the IWAN routing design. The legacy WAN network connects to the same LAN (migration network) as the DMVPN hub routers. Realistically, as long as the current design prohibits routing loops, prevents branch transit routing, and injects all the routes into the migration network (the network that connects the DMVPN hub routers with the legacy WAN), no changes should be necessary in the existing network. The most essential portion of the design is that the migration network must contain an accurate routing table so that the routers in the migration network can forward packets accurately to the legacy WAN or IWAN networks. If summarization is used in the migration network, it is extremely important that there be no overlapping network ranges.

Figure 16-3 illustrates a complete routing design using the assumption that the existing legacy WAN is an MPLS L3VPN that uses BGP. R11, R13, and CE1 have the full routing table for the IWAN and legacy WAN networks.

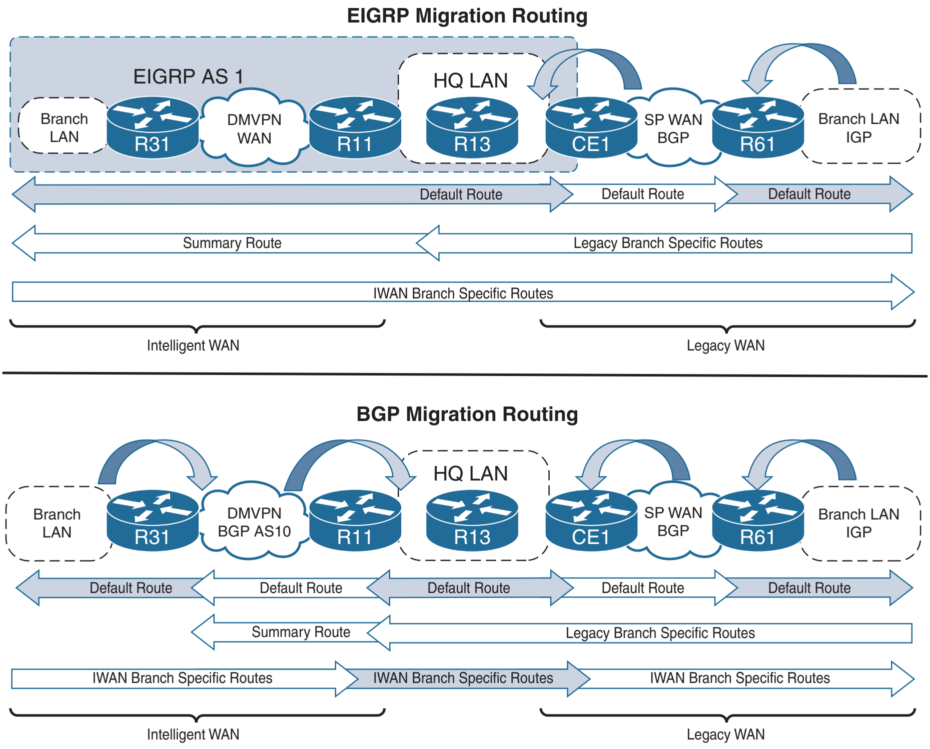


Figure 16-3 *Migration Network Flow and Routing*

Deploying DMVPN Hub Routers

Now that the existing WAN has been documentated, validated, and remediated, the environment can be prepared for deploying DMVPN. A critical component of any migration is the creation of an execution and backout plan. The first step for deploying an IWAN network is to establish the DMVPN hub routers into the existing network.

There are three models as illustrated in Figure 16-4:

- **Greenfield:** This model requires a new set of DMVPN hub routers and a new set of transport circuits. It is the simplest from an operational support perspective because the routing is straightforward and there are no constraints or dependencies on other aspects of the network. The new DMVPN hub routers connect to the existing LAN of the migration network.
- **Intermediate (IBlock):** This model requires a new set of DMVPN hub routers. A new link is required between the CE routers and the DMVPN router's FVRF interface. This model adds some complexity from an operational support perspective because network engineers must understand the traffic flow between the VRF and global interfaces.

  In addition, there are some dependencies on the existing network. For example, a failure of CE1 would impact the DMVPN tunnels that connect to R11.

- **Condensed:** This model assumes that the current CE routers are capable of hosting DMVPN- and IWAN-based services. In this model, the interface connecting to the SP network is placed in an FVRF. This model is the most complex because advanced routing protocol configuration is needed to exchange (leak) routes between the global routing table and the FVRF routing table. There are additional constraints with per-tunnel QoS policies not working on interfaces with hierarchical QoS policies on the encapsulating FVRF interface. Typically, a hierarchical QoS policy is used on most handoffs to the SP network. This book does not cover this model because of the additional complexities and depth of routing protocol configuration. More information and configurations can be found in the Cisco Live session "Migrating Your Existing WAN to Cisco's IWAN" mentioned in the "Further Reading" section of this chapter.

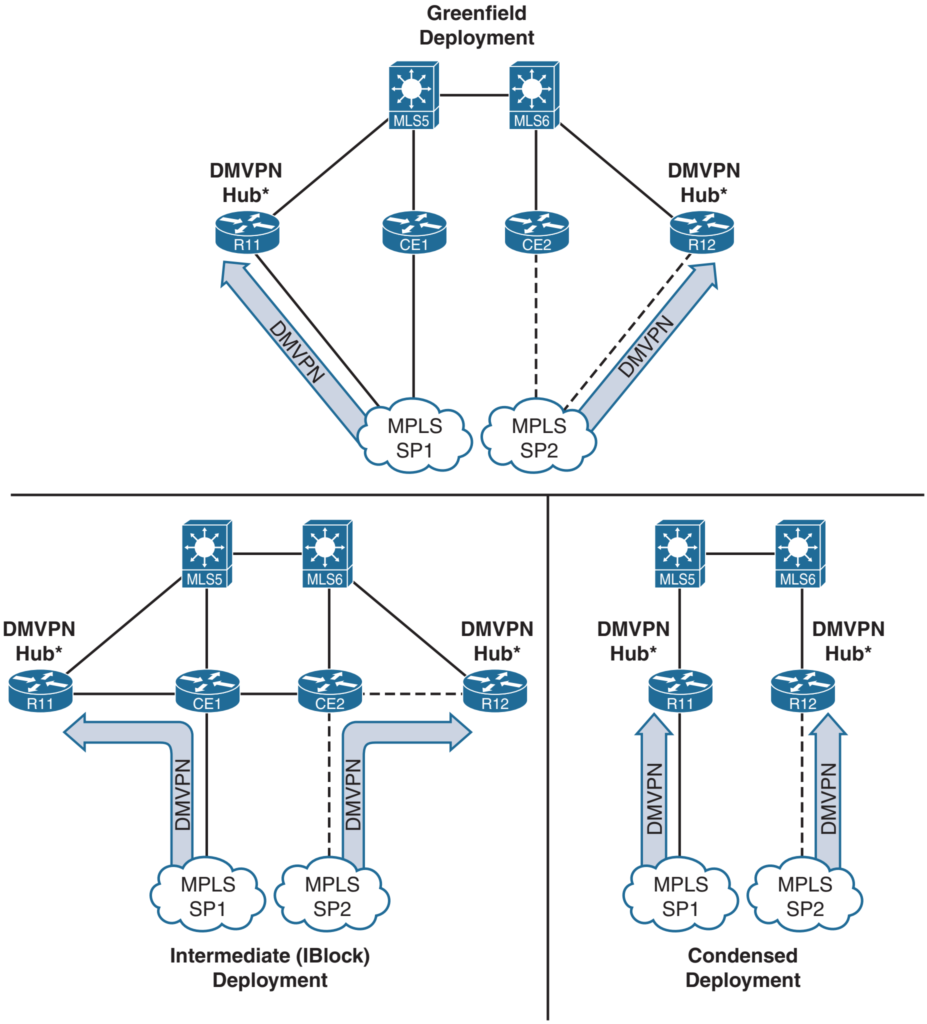


Figure 16-4 *IWAN Deployment Models*

Note This section provides a context for MPLS L3VPN for an MPLS transport but also applies to any multipoint transport such as MPLS L3VPN, VPLS, Metro Ethernet, and so on.

Deploying DMVPN hub routers in the IBlock model requires a link added between the CE device and the DMVPN hub router. The new network link that is added between R11 and CE1 (172.16.11.0/30) needs to be advertised into the SP network because R11's interface functions as the encapsulating interface for the DMVPN tunnel. This interface must be reachable by all the branches to terminate the DMVPN tunnel. CE1 accomplishes this task by advertising this network into BGP. R11's interface on the 172.16.11.0/30 network is associated to the FVRF and terminates the DMVPN tunnel. A similar link (172.32.12.0/30) is added between R12 and CE2 which is then advertised into BGP by CE2.

For organizations that are migrating from a dual MPLS topology to a hybrid topology (one MPLS transport and one Internet transport), only one DMVPN hub router needs to be connected to the MPLS and migration networks. The second DMVPN hub router connects to the Internet edge and the LAN. The interface terminating the DMVPN tunnel on the Internet is placed behind the Internet edge infrastructure that provides connectivity to the Internet.

Note Notice that multilayer switches (MLSs) are being used instead of a traditional router. The use of an MLS in the WAN aggregation layer can reduce the number of cables needed between devices through the use of 802.1Q VLAN tags and subinterfaces. Using either a router or an MLS is acceptable for the design as long as the device is sized appropriately for the routing table and traffic rates. This is fairly common and is part of the WAN distribution layer.

The figures do not display the redundant link connecting R11 or CE1 with multilayer switch MLS6 or the equivalent redundant links between R12 and CE2. Redundancy should be used where it does not affect the design or flow of the traffic.

Example 16-1 provides a sample configuration of CE1 and CE2 for the dual MPLS environment that houses the DMVPN hub routers. The configuration allows CE1 and CE2 to make all outbound connectivity routing decisions within BGP. Notice that in the configuration:

- There is no mutual redistribution between OSPF and BGP.
- BGP advertises the default route (originated by OSPF in the data center) toward the MPLS L3VPN PE routers.
- BGP advertises the new link between the CPE and DMVPN hub router.

- The *administrative distance (AD)* for BGP is not modified. All the BGP routes are redistributed into OSPF.
- The AD for all OSPF routes from the other CE router is set higher than IBGP's AD (200). The routes are matched on the OSPF router ID. This may affect any internal networks being advertised (loopback interfaces) in OSPF on the two CE routers, but the design is concerned with redistributed routes from the legacy WAN. This allows the BGP routing policy to influence outbound connectivity to the legacy network on both CE1 and CE2.

Example 16-1 *CPE Configuration for Dual MPLS and Dual MPLS DMVPN*

```
CE1
router ospf 1
 router-id 10.1.0.33
 redistribute bgp 100 subnets
 network 0.0.0.0 255.255.255.255 area 0
 distance 210 10.1.0.44 0.0.0.0
!
router bgp 100
 bgp router-id 10.1.0.33
 neighbor 10.1.34.14 remote-as 100
 neighbor 10.1.34.14 description CE2
 neighbor 172.16.13.2 remote-as 65000
 neighbor 172.16.13.2 description SP1 Router
 !
 address-family ipv4
  network 0.0.0.0
  network 172.16.11.0 mask 255.255.255.252
  network 172.16.13.0 mask 255.255.255.252
  neighbor 10.1.34.14 activate
  neighbor 172.16.13.2 activate
 exit-address-family
```

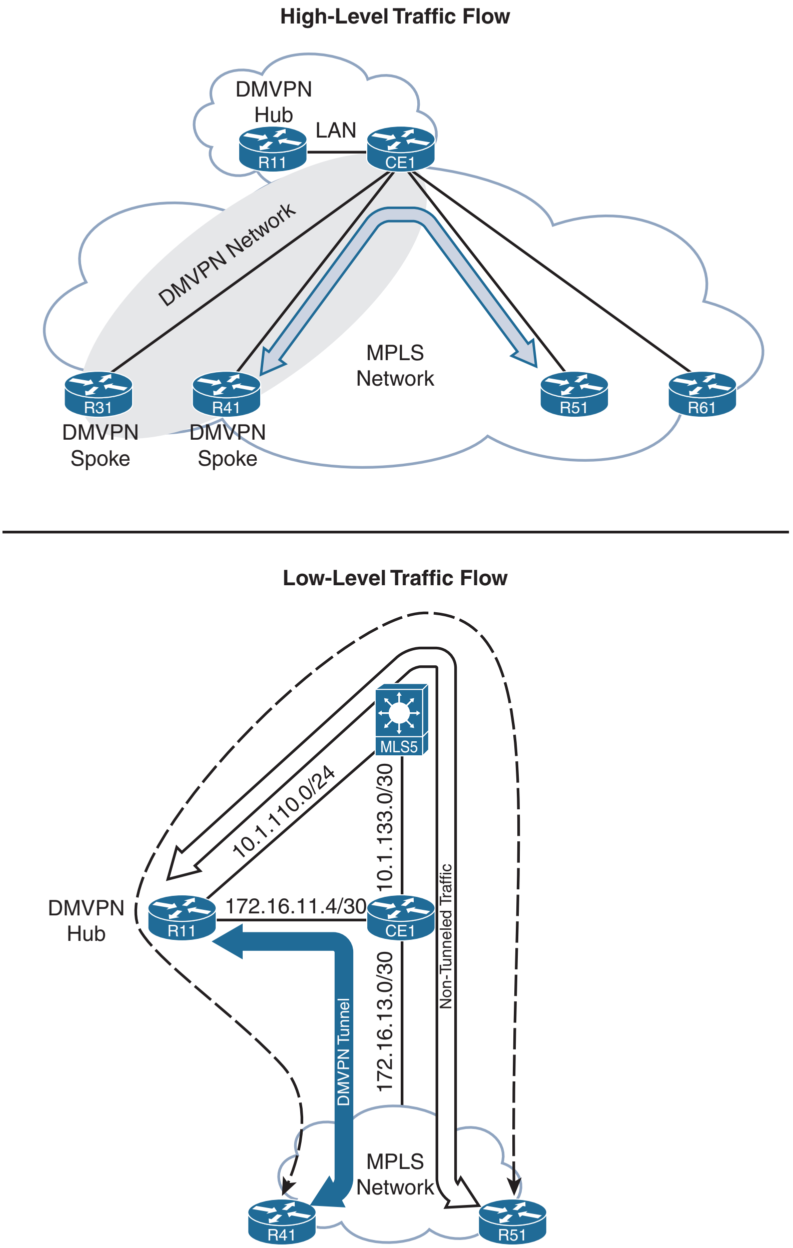
```
CE2
router ospf 1
 router-id 10.1.0.44
 redistribute bgp 100 subnets
 network 0.0.0.0 255.255.255.255 area 0
 distance 210 10.1.0.33 0.0.0.0
!
router bgp 100
 bgp router-id 10.1.0.14
 neighbor 10.1.34.13 remote-as 100
 neighbor 10.1.34.13 description CE1
 neighbor 100.64.14.2 remote-as 60000
 neighbor 100.64.14.2 description SP2 Router
 !
 address-family ipv4
  network 0.0.0.0
  network 100.64.12.0 mask 255.255.255.252
  network 100.64.14.0 mask 255.255.255.252
  neighbor 10.1.34.13 activate
  neighbor 100.64.14.2 activate
 exit-address-family
```

Figure 16-5 displays the low-level traffic patterns on an intermediate (IBlock) hub deployment for traffic flowing between the DMVPN network and the legacy WAN network. R41 transmits network traffic across the DMVPN network, which flows across the 172.16.13.0/30 and 172.16.11.0/30 networks in an encapsulated state. R11 decapsulates the packets and forwards the traffic out of its interface attached to the 10.1.110.0/24 network. MLS5 receives and forwards the packets on to CE1. CE1 then transmits the traffic (nonencapsulated) to the SP network where the traffic is forwarded on to R51.

Note As part of the migration planning, it is important to understand the traffic flows between the IWAN DMVPN and legacy networks. Additional bandwidth could be consumed at the migration sites for network traffic flowing between a migrated branch site connecting to a nonmigrated branch site. If a heavy-volume branch site that is accessed by other branch sites is migrated first, all the sites in the legacy WAN will route through the migration network. This may saturate those network links. Deployment of QoS policies or acquisition of additional bandwidth (burst model) may be suggested depending on the duration of the migration.

The bandwidth concern does not apply to intra-DMVPN (spoke-to-spoke) network traffic, or to traffic between remote sites on the legacy network.

If IPsec protection with PKI authentication is to be used, the deployment of the CA and CRL should be done at this stage, so that routers can request certificates as part of the migration strategy.



High-Level Traffic Flow

Low-Level Traffic Flow

Figure 16-5 *Traffic Patterns During Migration to DMVPN Networks*

Migrating the Branch Routers

Now that the DMVPN hub routers and PKI infrastructure (if required) have been deployed, the branch routers can be migrated onto the DMVPN network. The branch router's configuration needs to have the DMVPN tunnel configuration, routing protocol changes, and PfR configuration deployed as part of the migration.

In Chapter 3, "Dynamic Multipoint VPN," associating an FVRF to an interface was demonstrated, as well as the fact that the IP address was removed from that interface when the FVRF was associated. This causes a loss of connectivity during the change (the first stage of migration). Depending on the site's connectivity model, the migration might be executed without loss of service to the users at the branch.

It is extremely critical to back up the existing router configuration to the local router and to a centralized repository. Any changes to authentication should be made to allow access to the router in a timely manner (assuming that TACACS or radius servers cannot be reached). Also, the routers should accept remote console sessions from the workstations on which the migration will be performed, and any peer routers. Basically, these steps prevent locking yourself out of the router, so that changes can be made to continue or back out of the migration.

Establishing a baseline of network services is vital for a successful migration. Generating a pre-migration and post-migration test plan should help with the verification of the migration. For example, if an application does not work after the migration, how can the network engineering team be assured that the problem was caused by the migration? Performing the test before the migration establishes a baseline, so network engineering can verify that the problem was migration related. If an application fails a connectivity test both before and after the migration, the problem is not caused by the migration.

The migration of branch routers is performed remotely or on site. The migration team needs to decide if resources will be required on site depending on the team's comfort with the migration and the number of branch sites that have been previously migrated. Generally, migration of the first set of sites occurs with on-site resources, even if the remote procedure is used. This provides a method to overcome any automated techniques to keep the migration moving forward.

Migration of the branch routers involves the following simple steps:

- Creation and placement of the FVRF on the transport (WAN-facing) router interface.
- Re-association of the IP address on the transport interface because it was removed upon FVRF association.
- Configuration of DMVPN tunnel interfaces. The tunnels could be preconfigured, but connectivity cannot be established until the transport interface is placed in the FVRF.
- Reconfiguration of routing protocols.

- Verification of connectivity from remote networks to headquarters networks and vice versa.
- Verification of PfR (if configured).

Migration tasks can be performed from the command line or by using network management tools like APIC-EM or Prime Infrastructure that simplify many migration tasks.

Note If the DMVPN tunnels will authenticate via PKI, the PKI trustpoints should be configured and the certificates requested and installed on the routers before the migration of the branch routers. The relevant PKI trustpoint information such as the VRF or enrollment URL can be changed after the initial certificate is deployed on the router.

Migrating a Single-Router Site with One Transport

Branch sites without redundancy will encounter packet loss during the migration. However, they can be converted remotely without the need for console access. The following steps outline the process:

Step 1. **Preconfigure the DMVPN tunnel interface.**

The DMVPN tunnel interface should be configured as explained in Chapter 3. If the DMVPN tunnel is encrypted, the encryption configuration should be applied too. There are no interfaces associated to the FVRF, and the tunnel will remain in a line protocol state of *down*.

Step 2. **Configure the EEM applet on the router.**

The Embedded Event Manager executes multiple commands to complete the configuration. Preconfiguring the EEM script allows the commands to be verified before implementation and allows review to detect any typos or mistakes before it executes. EEM scripts execute faster than a user can manually type in the commands and executes locally (even if an SSH session is disconnected from the FVRF association).The template script provided in Example 16-2 can be used. The essential components of the EEM script are

- Application of FVRF to the transport interface
- Re-application of the IP address to the transport address
- Application of routing protocols

Depending upon the transport, additional tasks may need to be performed. For example, if the transport interface is a point-to-point IPsec tunnel, the crypto map may need to be removed from that interface in the EEM script.

Example 16-2 *EEM Applet for Migration*

```
Branch Router To Be Migrated
event manager applet MIGRATE
 event none
 action 010 cli command "enable"
 action 020 cli command "configure terminal"
! This section enables the MPLS FVRF and No Shuts the MPLS Tunnel
 action 030 cli command "interface GigabitEthernet0/1"
 action 040 cli command "vrf forwarding MPLS01"
 action 050 cli command "ip address 172.16.31.1 255.255.255.252"
 action 060 cli command "ip route vrf MPLS01 0.0.0.0 0.0.0.0 Giga0/1 172.16.31.2"
 action 070 cli command "interface Tunnel 100"
 action 080 cli command "no shut"
! This section enables the Internet FVRF and No Shuts the Internet Tunnel
 action 090 cli command "interface GigabitEthernet0/2"
 action 100 cli command "vrf forwarding INET01"
 action 110 cli command "ip address dhcp"
! The wait command allows the interface to obtain an IP address from DHCP
! before the Internet DMVPN tunnel is brought online
 action 120 wait 15
 action 130 cli command "interface Tunnel 200"
 action 140 cli command "no shut"
 action 150 syslog msg "Interface Configurations Performed "
! The last section is to remove the previous routing protocol configuration
! and then configure the routing protocols. Only a portion of this activity
! is shown, but this section should be completed based on your design.
 action 160 cli command "no router bgp 65000"
 action 170 cli command "no router ospf 1"
 action 180 cli command "router eigrp IWAN"
! Continue with rest of routing protocol configuration
 action 999 syslog msg "Migration Complete"
```

Note The EEM script action numbering is sorted alphanumerically. For example, if there are three actions, 1, 2, and 11, EEM places them in the following order: 1, 11, and 2. Keeping all the number lengths consistent means the order will be consistent with numeric sorting.

Step 3. **Save the current configuration.**

The router's configuration needs to be saved to nonvolatile memory with the command **copy running-config startup-config**.

Step 4. **Configure the router to reload in 15 minutes.**

In the event the DMVPN tunnel does not establish, the router should be reloaded to restore connectivity. The executive command **reload in 15** will count down to 15 minutes and initiate a reload of the router.

This step can be skipped if the migration is being performed locally at the branch site.

Step 5. **Execute the EEM script.**

The EEM script is executed with the command **event applet run** *applet-name*.

Step 6. **Restore connectivity to the router.**

Connectivity to the router will be lost if connected remotely. Connect back to the router using the tunnel IP address or the interface associated to the FVRF.

Step 7. **Cancel the router reload.**

Cancel the router reload with the command **reload cancel.**

Step 8. **Complete the migration.**

Once verification of all routing patterns and connectivity is completed, save the router's configuration and reenable any authentication or security policies that were changed for the migration.

Note Branch sites that contain backdoor network links should be migrated at the same time and considered as a dual-router IWAN site for the steps of migration. For example, a remote site has a high-speed MPLS VPN connection as a primary connection. It also maintains a backup dedicated T1 to another nearby branch that also uses a high-speed MPLS VPN connection as a primary connection. These two sites should be migrated during the same migration window.

Migrating a Single-Router Site with Multiple Transports

Branch sites that have a single router with multiple transports will encounter a small period of packet loss during the migration. The following steps outline the process:

Step 1. **Establish connectivity.**

Connect to the router via its loopback IP address.

Step 2. **Save the current configuration.**

The router's configuration needs to be saved to nonvolatile memory with the command **copy running-config startup-config.**

Step 3. **Configure the router to reload in 15 minutes.**

In the event that the migration causes a loss of connectivity, the router should be reloaded to restore connectivity. The executive command **reload in 15** counts down to 15 minutes and initiates a reload of the router.

Step 4. **Configure the FVRF.**

Create the FVRF on the router, and associate the FVRF to the secondary transport interface. Reapply the original IP address to the secondary transport interface.

If connectivity is lost, reestablish connectivity to the loopback interface.

Step 5. **Configure the DMVPN tunnel for the secondary transport.**

Configure the DMPVPN tunnel and crypto map and establish connectivity to the DMVPN hub router for the secondary transport.

Step 6. **Configure a static default route.**

Configure a static default route to provide connectivity to your workstation via the DMVPN cloud while the routing protocols are modified.

Step 7. **Modify the dynamic routing protocol configuration.**

Modify the routing protocol configuration so that the protocol peers with the DMVPN hub routers to exchange routes.

If connectivity is lost, reestablish connectivity to the DMVPN tunnel IP address.

Note If the BGP autonomous system number is changed, the BGP configuration must be completely removed. During this time, any routes advertised to the central site are rescinded and connectivity is lost until BGP can be reconfigured.

Step 8. **Configure the primary transport interface and DMVPN tunnel.**

Now that routing has established on the secondary DMVPN tunnel, the FVRF can be associated to the primary transport.

Step 9. **Configure the DMVPN tunnel.**

Configure the DMPVPN tunnel and crypto map and establish connectivity to the DMVPN hub router.

Step 10. **Modify the dynamic routing protocol configuration.**

Modify the routing protocol configuration so that the protocol peers with the DMVPN hub routers to exchange routes.

Step 11. **Cancel the router reload.**

Cancel the router reload with the command **reload cancel**.

Step 12. **Complete the migration.**

The last tasks need to be completed after the second transport has been migrated and connectivity from both DMVPN tunnels has been verified. The static default route from Step 6 needs to be backed out, the router's configuration saved, and authentication or security policies that were changed for the migration reenabled.

Migrating a Dual-Router Site with Multiple Transports

Sites that contain multiple routers with multiple transports can be migrated without any packet loss. The migration process can be executed with the following steps:

Step 1. **Establish connectivity.**

Connect to the router that is not being migrated in the branch site. This router acts as a jump box during the configuration of the first router. From this router an SSH session needs to be established to the router that is being migrated.

Step 2. **Save the current configuration.**

The router's configuration is saved to nonvolatile memory with the command **copy running-config startup-config**.

Step 3. **Configure the router to reload in 15 minutes.**

In the event that the migration causes a loss of connectivity, the router should be reloaded to restore connectivity. The executive command **reload in 15** counts down to 15 minutes and initiates a reload of the router.

Step 4. **Disable the secondary transport interface.**

Shut down the secondary transport interface. This terminates any routing protocol neighborships with the other routers and remove any routes learned from those neighborships.

Step 5. **Configure the routing protocols.**

Remove the routing protocol configuration and place the routing protocol configuration for the DMVPN topology.

Note Additional configuration may be needed to filter routes that are redistributed or advertised out toward the DMVPN network. It is important that routes learned from the active peer router (legacy WAN) not be advertised into the DMVPN network. This is done to prevent the site that is being migrated from being a transit site that could cause connectivity issues.

Step 6. **Configure the transport interface and DMVPN tunnel.**

Associate the FVRF to the transport interface, create the crypto map, create the DMVPN tunnel interface, and establish connectivity to the DMVPN cloud.

Step 7. **Cancel the router reload.**

Cancel the router reload with the command **reload cancel**.

Step 8. **Configure the other router.**

After the DMVPN tunnel has established and connectivity is verified across the DMVPN tunnel, repeat Steps 1 through 7 for the other router.

Step 9. **Complete the migration.**

The final tasks need to be completed after the second router has been migrated and connectivity from both DMVPN tunnels has been verified. Authentication and security policies should be reenabled and the router's configuration should be saved.

Post-Migration Tasks

Depending on the size of the environment, the migration may take days, weeks, or months. After all the branch routers have been completely migrated and the SP network is used only for transport between DMVPN routers, the migration has been completed. The last task is to clean up the environment.

Figure 16-6 illustrates the various stages of the DMVPN hub IBlock deployment at various stages. MLS5 is one of the WAN distribution switches, R11 is the DMVPN hub router, and CE1 is the customer edge router that connects to the SP network. The 10.1.133.0/30 link between the WAN distribution switch (MLS5) and CE1 router is no longer needed and can be removed. R11's GigabitEthernet0/1 interface on the 172.16.11.0/30 network belongs to the FVRF for that transport, and the GigabitEthernet0/3 interface on the 10.1.110.0/24 network faces the WAN distribution block.

Note If the final design does not require migrating all the sites, these steps should not be performed because they will remove connectivity between the IWAN and legacy WAN.

The second illustration in Figure 16-6 shows the topology with the 10.1.133.0/24 link removed. Removing the link has no impact because traffic should not be flowing between R10 and any other branch sites on the legacy WAN.

CE1 can be removed depending on the following factors:

- Who owns the device: your organization or the SP?
- What additional value does the device add to the design or operational perspective?

If deemed unnecessary, the device can be removed with one major caveat. How is connectivity established between the SP network and R11? If the cable on CE1 that connects to the MPLS L3VPN PE is moved directly to R11, R11 connectivity will break. R11's IP address is on the 172.16.11.0/30 network and the SP's PE router is on the 172.16.13.0/30 network. One of the devices will have to have a different IP address.

Changing R11's IP address from 172.16.11.1 to 172.16.13.1 maintains connectivity to the transport network, but unfortunately all the branch routers are configured to use the 172.16.11.1 IP address for their NBMA address of the DMVPN tunnel.

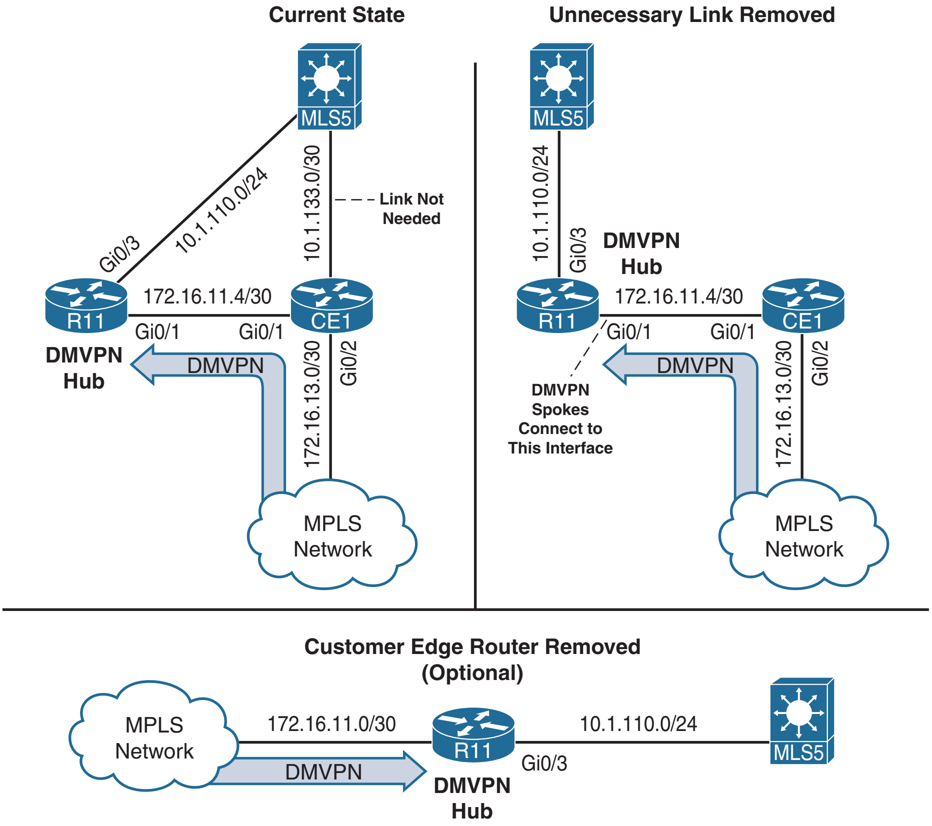


Figure 16-6 *Post-Migration Cleanup with CE Routers*

There are three solutions to this issue:

1. Configure a loopback interface on the hub router. Place the loopback into the FVRF and terminate the DMVPN sessions on the loopback interface. The loopback interface must be advertised and reachable in the transport network. Be aware of potential QoS problems with ECMP if there are multiple links in the FVRF to reach the SP network.

2. Coordinate the change with your SP. When the cable is moved from CE1 to R11, have the SP change the IP address on the PE router. Branch sites will still have connectivity to the other DMVPN hub for that tunnel. A change window should be identified in advance. All four hubs (two per transport and two transports) can be migrated over multiple maintenance windows.

3. Reconfigure the NHRP mappings on every branch site. During the time of reconfiguration, the spoke loses connectivity to one of the two hubs for that DMVPN cloud. This technique requires every branch site to be changed twice and is not very effective.

The first and second options provide the most straightforward approaches when removing a CE device. If there is a potential of the CE router being removed, change the design so that the DMVPN tunnel terminates on the hub router's loopback before any branch sites have been migrated. If branch sites have already been deployed, it may be easier just to perform the second option.

Note The CE1 router can be left in place. In essence, it has become a part of the transport network.

Migrating from a Dual MPLS to a Hybrid IWAN Model

The process for migrating from a legacy environment that uses two different MPLS SPs to a hybrid IWAN model is straightforward. A DMVPN hub router is deployed for the MPLS transport, and a DVMPN hub router is deployed for the Internet transport. Nothing needs to be done with the transport of the second MPLS SP. Figure 16-7 displays R11 (DMVPN hub for MPLS) connected to CE1 via the IBlock method to communicate with MPLS SP1, and R12 is connected directly to the Internet edge.

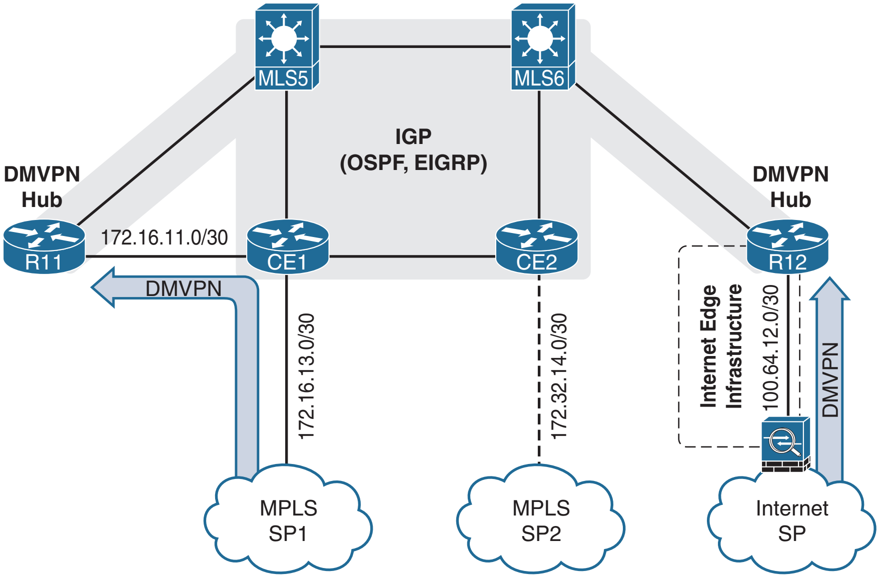


Figure 16-7 *Dual-MPLS-to-Hybrid-DMVPN Hub Topology*

The branch site migration consists of

- Ordering the new Internet circuits at the branch sites
- Configuring the MPLS SP1 and Internet FVRFs
- Configuring the transport interfaces: MPLS SP1 and Internet
- Configuring the DMVPN tunnels
- Configuring the routing protocols
- Removing the configuration for MPLS SP2 on the branch routers and canceling the circuit

The CE2 router remains in place until all branch sites have been migrated off of the legacy network. CE1 and CE2 are used to provide connectivity between the legacy sites with the data centers or IWAN branch sites. Once the migrations have been completed, the link between CE1 and R11 can process and the removal of CE2 and the data centers can occur.

Whether or not the CE1 router is maintained for the MPLS DMVPN hub is dependent upon the same variables as described in the previous section. Figure 16-8 displays the DMVPN hub topology at the data center after the migration is completed.

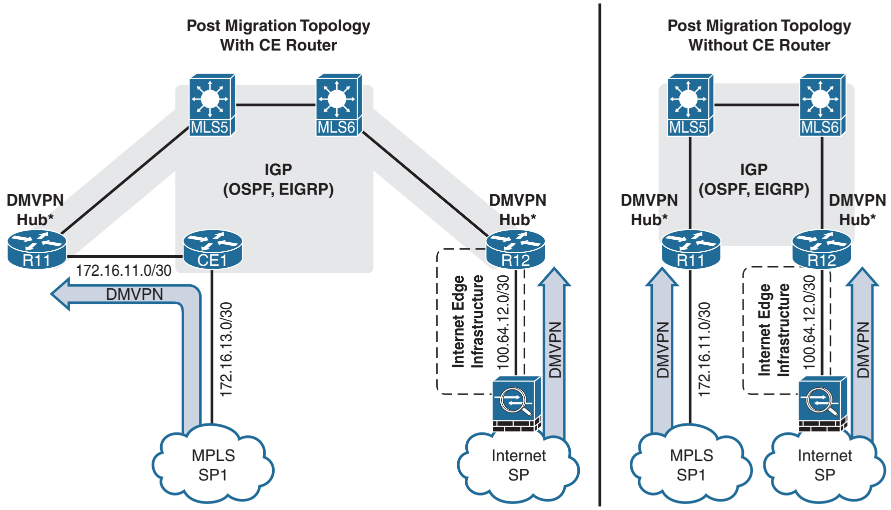


Figure 16-8 *Hybrid DMVPN Hub Topology After Migrations*

Migrating IPsec Tunnels

IPsec tunnels are a point-to-point technology. The migration process for IPsec tunnels involves the establishment of the DMVPN hub into the network. Figure 16-9 illustrates the migration process. In the initial IPsec topology:

- A point-to-point tunnel (192.168.13.0/30) exists between R1 and R3.
- A point-to-point tunnel (192.168.14.0/30) exists between R1 and R4.
- A point-to-point tunnel (192.168.45.0/30) exists between R4 and R5.

Step 1. **Deploy the DMVPN hub router.**

R2, the DMVPN hub router, is deployed and connected directly to R1. Traffic between the point-to-point topology and the DMVPN network should traverse R1 and R2.

Step 2. **Migrate non-transit sites first.**

R5 is identified as the first router to be migrated because it is not being used as a transit router. R5 is migrated using the steps in the section "Migrating the Branch Routers."

Step 3. **Shut down the unnecessary point-to-point link.**

The point-to-point tunnel between R4 and R5 is shut down to prevent transit routing or routing loops.

Step 4. **Migrate other sites.**

R4 is identified as the next router to be migrated. R4 is migrated using the steps listed in the section "Migrating the Branch Routers."

Step 5. **Shut down the unnecessary point-to-point link.**

The point-to-point tunnel between R1 and R4 is shut down to prevent transit routing or routing loops.

Step 6. **Continue the migration.**

Steps 4 and 5 are repeated as needed for other routers until all the point-to-point tunnels are migrated.

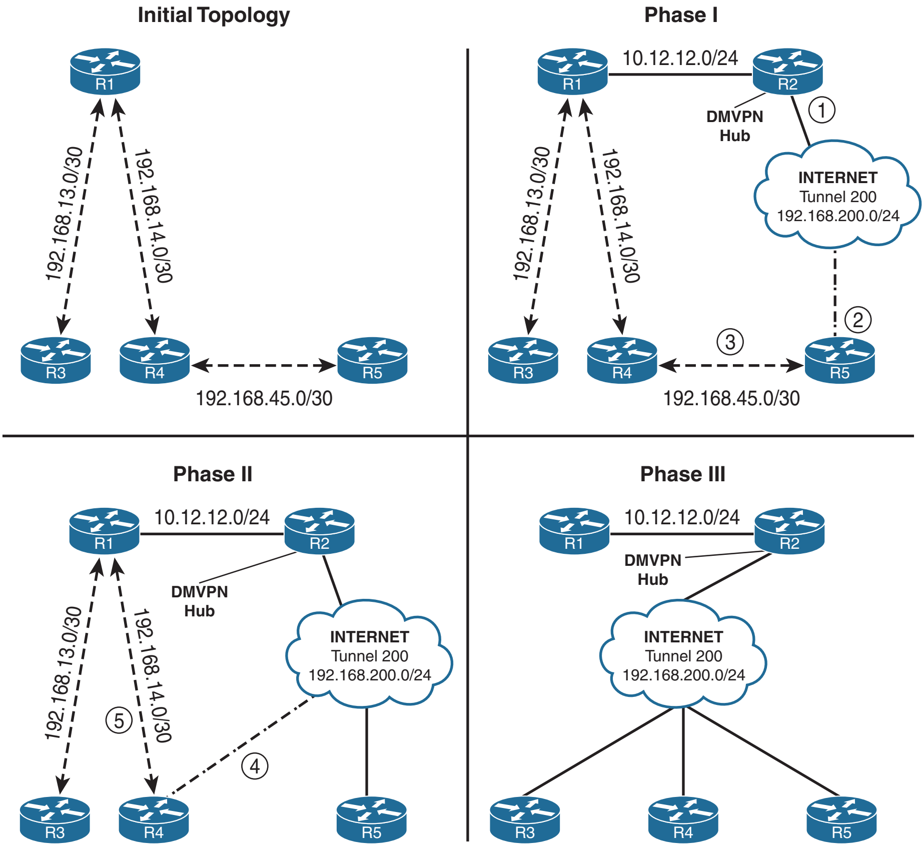


Figure 16-9 *Process to Migrate Point-to-Point Tunnels to DMVPN Tunnels*

The process for migrating an IPsec environment to DMVPN is based upon the concept that routers are terminating the IPsec connection. If the devices terminating the IPsec session are firewalls, the DMVPN routers need to be placed behind the firewalls. Traffic from the DMVPN routers needs to pass through the firewalls onto the transport network and not across the firewall's IPsec tunnels.

PfR Deployment

The PfR domain policy should be configured when the DMVPN hub routers are deployed. PfR is a relatively new technology that may take time to learn. Initial PfR policies should start with limited functions and increase over time as users become accustomed to working with the tool. As more and more sites migrate onto IWAN, and the network engineering team understands PfR better, additional logic can be added to the PfR domain policy.

In Chapter 8, "PfR Provisioning," the concepts of PfR site prefixes and PfR enterprise prefixes were introduced. They are essential components of the operation of PfR. The enterprise prefix list is configured on the Hub MC and defines the enterprise prefixes that PfR should control more granularly. Network traffic outside of the enterprise prefix list is considered a PfR Internet prefix which cannot check path characteristics such as latency or jitter. Only when both source and destination networks reside in the enterprise prefix list can path characteristics such as latency and jitter be monitored.

For example, if the enterprise prefix list includes only the 10.0.0.0/8 network range, traffic between 192.168.11.0/24 and 192.168.22.0/24 does not check a path's delay even if the network traffic matches the QoS EF DSCP that is defined in the policy that PfR should check for delayed EF DSCP traffic.

PfR includes the concept of a PfR site prefix that is a database containing the inside prefixes for every site. The PfR site prefix database is built from the egress performance monitor and is advertised to the Hub MC dynamically (branches only) or can be statically configured.

This section explains the internal PfR components during migration. Figure 16-10 displays a simple topology (assume that Site 2 does not exist), and Site 1 is the migration site (R11 and R12 are the only DMVPN hub routers). R31 and R41 are continuously communicating with each other with traffic that matches the appropriate PfR policy that monitors a path's delay.

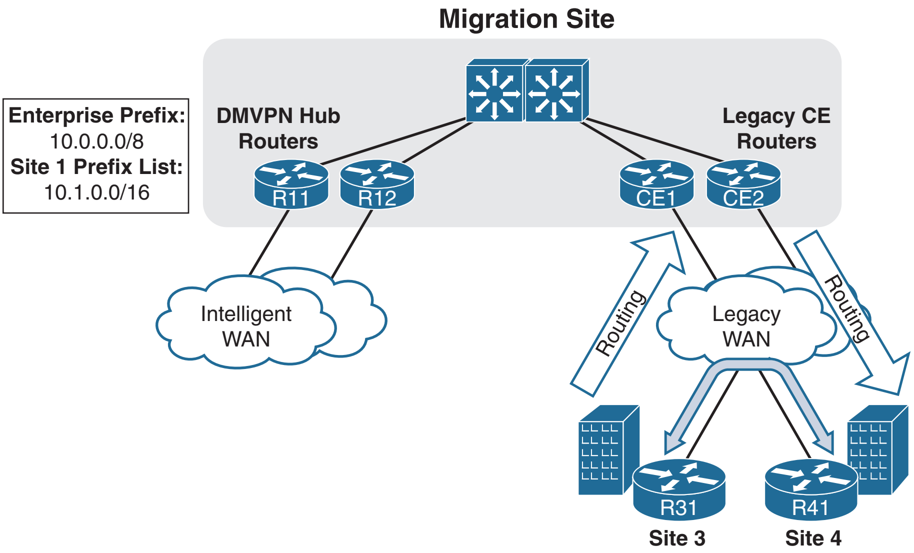


Figure 16-10 *PfR Interaction Between Legacy Site 3 and Site 4*

Example 16-3 displays the PfR configuration on R10. Notice that the enterprise prefix list contains the 10.0.0.0/8 network prefix and the 10.1.0.0/16 site prefix is programmed on R10.

Example 16-3 *R10's Initial PfR Configuration*

```
R10 (Hub MC)
domain IWAN
 vrf default
  master hub
   enterprise-prefix prefix-list ENTERPRISE_PREFIX
   site-prefixes prefix-list SITE_PREFIX
!
ip prefix-list ENTERPRISE_PREFIX seq 10 permit 10.0.0.0/8
ip prefix-list SITE_PREFIX seq 10 permit 10.1.0.0/16
```

At this time, R31 and R41 reside on the legacy networks. They are able to communicate directly with each other using the legacy SP transport network. Example 16-4 displays the PfR site prefix database which has only the static 10.1.0.0/16 network that was defined on R10. Notice that 10.0.0.0/8 site prefix is associated to all site IDs with the value of 255.255.255.255.

Example 16-4 *PfR Site Prefix Database—Initial Hubs*

```
R10-HUB-MC# show domain IWAN master site-prefix
  Prefix DB Origin: 10.1.0.10
  Prefix Flag: S-From SAF; L-Learned; T-Top Level; C-Configured; M-shared

Site-id              Site-prefix          Last Updated         DC Bitmap  Flag
--------------------------------------------------------------------------------
10.1.0.10             10.1.0.10/32         00:13:41 ago         0x1         L
10.1.0.10             10.1.0.0/16          00:13:41 ago         0x1         C,M
255.255.255.255      *10.0.0.0/8           00:13:41 ago         0x1         T
```

R31 has been migrated to the IWAN architecture as shown in Figure 16-11. Traffic between R31 and the rest of the network is on the IWAN network until it reaches the migration site. Traffic may terminate locally in Site 1 or be forwarded on to R41 which resides on the legacy WAN.

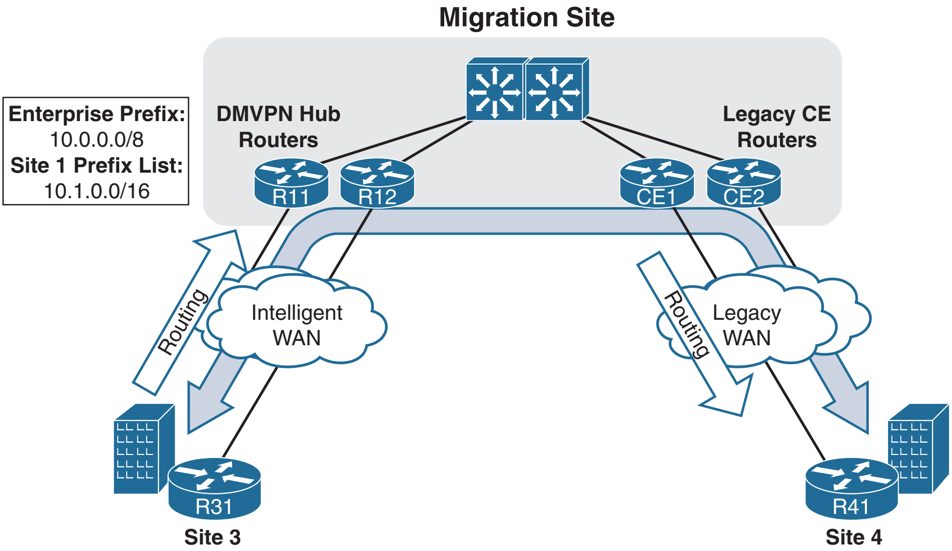


Figure 16-11 *PfR Interaction Between IWAN Sites and Legacy WAN Sites*

Example 16-5 displays the Site 3 prefixes (10.3.3.0/24 and 10.3.0.31/32) that have been dynamically discovered and reported to R10. Notice that the Site 4 prefix (10.4.4.0/24) has not been added even though traffic still flows between both sites. Remember that site prefixes must be statically defined on any hub or transit MCs. In the current state, PfR manages (verifies that the path's delay is within policy) network traffic between R31 (10.3.0.0/16) and Site 1 (10.1.0.0/16) and networks.

However, network traffic from R31 to R41 is not managed by PfR because Site 4's site prefixes have not been discovered in the PfR domain. Although the 10.4.0.0/16 network range may reside inside the 10.0.0.0/8 range displayed in Example 16-5, the 10.0.0.0/8 is not associated to a specific site. The site ID of 255.255.255.255 indicates that this entry is defined only in the enterprise prefix list. Traffic is not controlled by PfR because the source and destination have not been completely identified. PfR can only load-balance (if configured) the network traffic; it cannot monitor delay, jitter, or packet loss. In essence, R31 uses the routing table to reach the DMVPN hub routers, and then CE1 uses the routing table to reach R41.

Example 16-5 *R31 Is Migrated to IWAN While R41 Is on the Legacy WAN*

```
R10-HUB-MC# show domain IWAN master site-prefix
  Prefix DB Origin: 10.1.0.10
  Prefix Flag: S-From SAF; L-Learned; T-Top Level; C-Configured; M-shared

Site-id              Site-prefix          Last Updated         DC Bitmap  Flag
--------------------------------------------------------------------------------
10.1.0.10             10.1.0.10/32         00:20:51 ago         0x1         L
10.1.0.10             10.1.0.0/16          00:20:51 ago         0x1         C,M
10.3.0.31             10.3.0.31/32         00:01:19 ago         0x0         S
10.3.0.31             10.3.3.0/24          00:01:19 ago         0x0         S
255.255.255.255      *10.0.0.0/8           00:20:51 ago         0x1         T
```

It is possible to have PfR manage half of the path between the migrated branch site and legacy sites because all the traffic must flow through the DMVPN hubs located at Site 1. The 10.0.0.0/8 prefix can be added to Site 1's site prefix list and allows PfR to monitor the path between migrated sites and the DMVPN hubs.

Note The exact prefix added to the site prefix list should be advertised from the DMVPN hub for PfR to have a parent route. In essence, this is the enterprise summary route that was defined in Chapter 4 for the topology.

Example 16-6 demonstrates the addition of the 10.0.0.0/8 prefix to Site 1's prefix list.

Example 16-6 *R10's PfR Configuration with Enhanced Site Prefix List*

```
R10 (Hub MC)
domain IWAN
 vrf default
  master hub
   enterprise-prefix prefix-list ENTERPRISE_PREFIX
   site-prefixes prefix-list SITE_PREFIX
!
ip prefix-list ENTERPRISE_PREFIX seq 10 permit 10.0.0.0/8
ip prefix-list SITE_PREFIX seq 10 permit 10.1.0.0/16
ip prefix-list SITE_PREFIX seq 20 permit 10.0.0.0/8
```

Figure 16-12 displays that R31 is now able to monitor path attributes such as delay, jitter, and packet loss between it and the DMVPN hub routers. After traffic reaches the DMVPN hub routers, they use the routing table to reach the legacy CE routers, which then forward the traffic on to the legacy WAN.

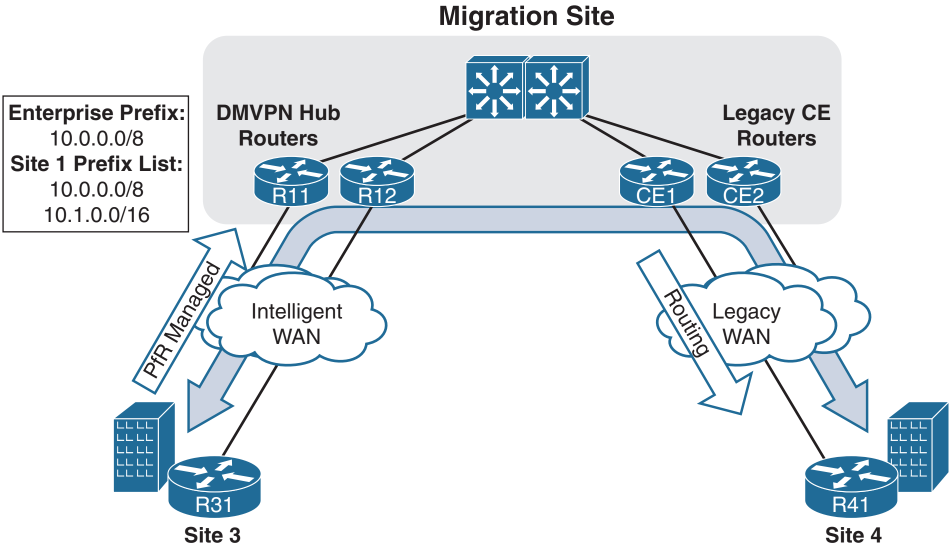


Figure 16-12 *PfR Managed Traffic for IWAN Segment Between Site 3 and Site 4*

Example 16-7 displays the site prefixes after the change is made. Notice that the 10.0.0.0/8 prefix is associated to Site 1 (10.1.0.10). PfR now monitors the path between R31 and R41 for delay in both directions between R31 and the DMVPN hub routers.

Example 16-7 *Added Site Prefix 10.0.0.0/8 on Hub MC (Transit Site)*

```
R10-HUB-MC# show domain IWAN master site-prefix
  Prefix DB Origin: 10.1.0.10
  Prefix Flag: S-From SAF; L-Learned; T-Top Level; C-Configured; M-shared

Site-id              Site-prefix          Last Updated         DC Bitmap  Flag
--------------------------------------------------------------------------------
10.1.0.10             10.1.0.10/32         00:39:39 ago         0x1         L
10.1.0.10             10.1.0.0/16          00:39:39 ago         0x1         C,M
10.3.0.31             10.3.0.31/32         00:02:46 ago         0x0         S
10.3.0.31             10.3.0.0/16          00:02:46 ago         0x0         S
10.1.0.10            *10.0.0.0/8           00:00:05 ago         0x1         T,C,M
```

Figure 16-13 displays that R41 has been migrated onto the IWAN architecture. Communication between R31 and R41 can occur directly after the spoke-to-spoke DMVPN tunnel is established. R31 is now able to monitor path attributes such as delay, jitter, and packet loss between it and R41.

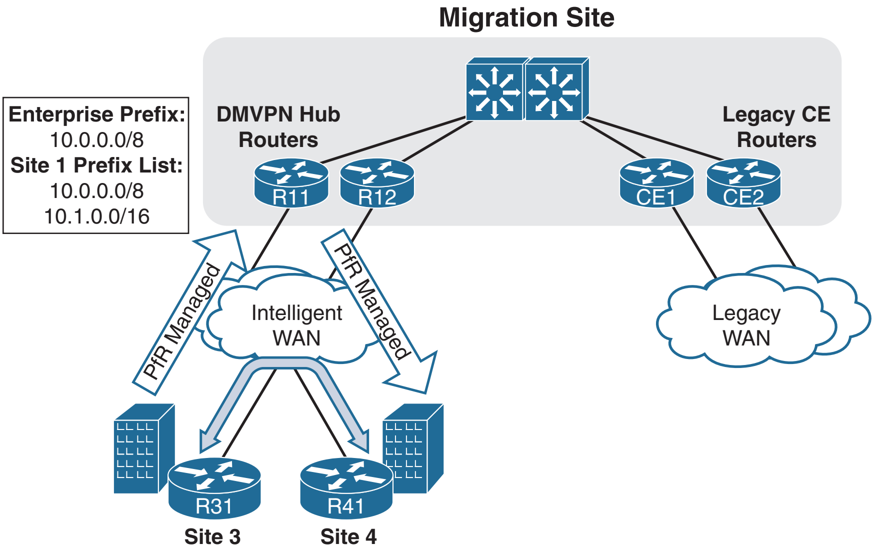


Figure 16-13 *PfR Managed Traffic for IWAN Site-to-Site Traffic*

The 10.4.4.0/24 and 10.4.0.41/32 site prefixes have been dynamically added to the site prefix list as shown in Example 16-8. Just like what happened with the routing table, PfR uses the most explicit match for identifying a network to a site ID. PfR can now monitor the end-to-end path between R31 and R41.

> **Note** Now that both R31 and R41 have been migrated to the IWAN architecture, a spoke-to-spoke tunnel has formed, providing direct connectivity between the two.

Example 16-8 *Site Prefix List After R41's Migration*

```
R10-HUB-MC# show domain IWAN master site-prefix
  Prefix DB Origin: 10.1.0.10
  Prefix Flag: S-From SAF; L-Learned; T-Top Level; C-Configured; M-shared

Site-id                  Site-prefix           Last Updated          DC Bitmap   Flag
--------------------------------------------------------------------------------
10.1.0.10                 10.1.0.10/32          00:56:20 ago          0x1          L
10.1.0.10                 10.1.0.0/16           00:56:20 ago          0x1          C,M
10.3.0.31                 10.3.0.31/32          00:19:27 ago          0x0          S
```

```
10.3.0.31             10.3.0.0/16          00:19:27 ago         0x0         S
10.4.0.41             10.4.0.41/32         00:06:09 ago         0x0         S
10.4.0.41             10.4.4.0/24          00:06:09 ago         0x0         S
10.1.0.10            *10.0.0.0/8           00:16:46 ago         0x1         T,C,M
```

Note Adding the entire enterprise prefix list to the migration site's prefix list allows PfR to monitor a portion of the path between the IWAN architecture and the legacy network. PfR cannot manage true end-to-end latency. This step is optional and does not have to be configured, but most customers find it beneficial.

Testing the Migration Plan

Any successful deployment of a new technology includes a solid design with a detailed migration plan. Testing the migration plan in advance of the actual implementation provides an opportunity to identify overlooked details that could cause issues.

Initial testing can be performed on Cisco VIRL (Virtual Internet Routing Lab) which provides a scalable, extensible network design and simulation environment. VIRL has been used by many customers for a variety of testing prior to deployment in a production network. It includes several Cisco Network Operating System virtual machines (IOSv, IOS XRv, CSR1000v, NX-OSv, IOSvL2, and ASAv) and has the ability to integrate with third-party-vendor virtual machines. It includes many unique capabilities such as "live visualization" that provide the ability to create protocol diagrams in real time from a running simulation. More information about VIRL can be found at http://virl.cisco.com.

Summary

This chapter covered the processes needed to successfully migrate an existing WAN to Cisco IWAN. A successful migration includes the following:

- Documenting the existing network
- Finalizing the design
- Deploying a proof of concept or production pilot of the network
- Creating a high-level migration plan
- Testing the execution plans in a lab environment and modifying the plan accordingly
- Deploying DMVPN hub routers
- Deploying PKI infrastructure (if necessary)

- Migrating branch routers
- Post-migration cleanup tasks

This chapter provided a high-level overview of the tasks to deploy the transport independence and intelligent path control components of the IWAN architecture. Testing the high-level and low-level execution plan is vital and allows the plans to be modified if necessary. Application optimizations and direct Internet access can be deployed before, after, or during the deployment of transport independence.

Reach out to the local Cisco partners or account team with any other questions you may have about migrations.

Further Reading

Cisco. “Cisco IOS Software Configuration Guides.” www.cisco.com.

Edgeworth, Brad, and Mani Ganesan. “Migrating Your Existing WAN to Cisco’s IWAN.” Presented at Cisco Live, Berlin, 2016.

This page intentionally left blank

Chapter 17

Conclusion and Looking Forward

Cisco Intelligent WAN (IWAN) architecture delivers a new paradigm for the capabilities of a WAN. Instead of having to rely on routing protocols to find the best path, engineers can monitor application performance to greatly improve the user experience and reduce operational support interaction. New capabilities and features will continue to be infused in future versions of IWAN.

Intelligent WAN Today

IWAN provides intelligence via a secure overlay network that

- Uses any transport.
- Provides bandwidth efficiency by placing application flows on the correct transport via policy or load distribution, thereby maximizing expensive WAN circuits.
- Provides application SLAs to protect applications end to end across the WAN.
- Reduces bandwidth consumption while improving application experience with application optimizations.
- Provides optimized access to Internet resources with direct Internet access. A centralized management and monitoring interface for real-time threat defense is provided with Cisco Cloud Web Security services.

All these functions are integrated into a complete *software-defined WAN* (SD-WAN) solution with the use of Cisco *Prime Infrastructure (PI)* and Cisco *Application Policy Infrastructure Controller—Enterprise Module (APIC-EM)*, which provide centralized operation and management automation in a simple and secure method.

Intelligent WAN Architecture

The IWAN solution, like many other emerging technologies, is constantly evolving. Over time, enhancements will be developed and integrated. IWAN as an architecture will evolve as new features are developed specifically for IWAN and existing features are integrated into IWAN solution testing. The currently available version, IWAN 2.1, has integrated several enhancements over the prior release of IWAN 2.0, which brought us the integration of DMVPN Phase 3 with the first release of PfRv3. IWAN 2.1 brings us support for multiple next hops and multiple data centers, providing additional redundancy and SLA monitoring.

The IWAN architecture today is a very prescriptive design, allowing complete validation of everything within the solution. This minimizes the number of components that need to be tested, encompassing interaction among all pillars of the solution. IWAN architecture testing will be expanded both to encompass new features and integrate existing ones. Cisco provides incremental improvements to the IWAN architecture while delivering new use cases based on customer feedback.

The available features within IOS provide the flexibility for a variety of design requirements within a network deployment. The ability to integrate additional features above and beyond the IWAN architecture is the reason IWAN is such a strong solution. Even as new versions of the IWAN architecture are developed by integrating new features, the IWAN architecture testing will never be able to integrate every single feature available in IOS-XE. The IWAN design as a baseline guarantees capability, timely deployment, and confidence in the solution. Using this foundational logic and ability to handle any deviations on a network-by-network basis is powerful. Being able to integrate additional features with minimal additional test cycles decreases the time needed for validation and deployment while producing a solid solution for critical applications.

Today IWAN best practices are available within *Cisco Validated Design (CVD)* documents. Deployment using the IWAN application in APIC-EM or IWAN workflow in Cisco PI speeds deployment using CVD-based templates. The APIC-EM IWAN application follows the very strict prescriptive model, whereas the IWAN PI workflow allows customization of templates to meet specific deployment requirements, granting the ability to meet any customer's requirements.

Intelligent WAN Tomorrow

Today's IWAN architecture can make real-time decisions about application performance; future versions of Cisco IWAN architecture hold so many possibilities.

Network functions virtualization (NFV) and vBranch in its simplest definition allows for the virtualization of traditional network devices such as routers, firewalls, IPS, or identity and authentication services. NFV allows for the deployment of new features and functions where network engineers were previously prohibited from deploying them because of cost or time to deploy these new services.

Cloud-based providers (IaaS, SaaS, or PaaS) continue to gain popularity. Cloud-based providers can provide more flexibility and functions, reduce costs, provide redundancy, or provide services that in-house employees cannot provide. Integrating the IWAN architecture with cloud-based services offers the best of both worlds by ensuring connectivity, integration, and cloud intelligence in the IWAN architecture. A cloud-integrated IWAN architecture would be composed of essential global policies, cloud points of presence for virtual private cloud infrastructure, application optimization, and cloud security.

Software-defined networking (SDN) continues to evolve. The future of SDN and Cisco IWAN opens the door for the network to learn about the WAN transports and make decisions based on current experiences. Predictive decisions will be based on prior knowledge (gained from an implementation's past experiences) and managed to preempt network anomalies. Integrating the software-defined WAN with enterprise SDN provides end-to-end orchestration of applications and services across the campus, WAN, and data center. These technologies are seen with the digital network architecture, providing rapid deployment of required features, allowing for faster innovation, and reducing cost and complexity with lower risk.

Ultimately, Cisco IWAN uses multiple WAN transports with high reliability and SLAs for business-critical applications while dramatically lowering WAN costs without compromising the network's integrity. The future of the intelligent network starts now.

This page intentionally left blank

Appendix A

Dynamic Multipoint VPN Redundancy Models

DMVPN provides two of forms of NHS redundancy. Although the IWAN prescriptive model supports only the NHRP clusterless model, the authors wanted to demonstrate both models to provide a better understanding.

NHRP Clusterless Model

The NHRP clusterless model places all NHS routers in the same cluster, assigns preference by priority, and restricts the maximum number of active connections to an NHS. This model presents a possible configuration where an NHS hub is not present at a specific site.

For example, assume that six hub routers are distributed equally across three data centers. The design states that an NHC maintains an active connection to only three NHS routers at a time, and that a connection is not made to two NHSs in the same data center. Increasing the priority for the second router at each data center is not enough.

Table A-1 provides the ideal state where there is a connection to R11, R21, and R31, so an active connection to an NHS in each data center is maintained. If R21 fails, it is possible for R12 to become the third active NHS router. But R12 is in the same data center as R11, which violates the design.

Table A-1 *NHRP Clusterless Scenario*

NHS Cluster Maximum Connections = 3

Router	*Location*	*NHS Priority*	*Cluster*	*Ideal State*	*Affected State*
R11	DC 1	10	0	Up	Up
R12	DC 1	20	0	Down (backup)	Up
R21	DC 2	10	0	Up	Down (probe)
R22	DC 2	20	0	Down (backup)	Down (backup)
R31	DC 3	10	0	Up	Up
R32	DC 3	20	0	Down (backup)	Down (backup)

NHRP Clustered Model

The NHRP clustered model provides additional granularity by placing each geographic grouping of NHS routers in the same cluster, thus preventing scenarios from the clusterless model from happening. The number of NHS cluster connections is reduced for all three clusters appropriately.

Table A-2 demonstrates the same scenario as before but deployed in a clustered model. When R21 fails, R22 changes to the *up* state. R12 and R32 are prohibited from becoming active because they are still restricted to one active session for their appropriate cluster group.

Table A-2 *NHRP Clustered Scenario*

NHS Cluster Maximum Connections = 1

Router	*Location*	*NHS Priority*	*Cluster*	*Ideal State*	*Affected State*
R11	DC 1	10	1	Up	Up
R12	DC 1	20	1	Down (backup)	Down (backup)
R21	DC 2	10	2	Up	Down (probe)
R22	DC 2	20	2	Down (backup)	Up
R31	DC 3	10	3	Up	Up
R32	DC 3	20	3	Down (backup)	Down (backup)

NHRP Clustered Model Configuration

Now that the concept has been explained, let's visit a simple four-router topology as shown in Figure 3-7. Routers R11 and R12 belong to the same data center (Data Center 1) and are placed in cluster 1. Routers R21 and R22 belong to the same data center

(Data Center 2) and are placed in cluster 2. R11 and R21 are assigned a priority of 1, and R12 and R22 are assigned a priority of 2.

Example A-1 provides the configuration for R31. Notice the additional keywords that were added to the NHS mapping entry.

Example A-1 *Configuration for Clustered DMVPN Model*

```
R31-Spoke
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.31 255.255.255.0
 no ip redirects
 ip mtu 1400
 ip nhrp network-id 100
 ip nhrp holdtime 600
 ip nhrp nhs 192.168.100.11 nbma 172.16.11.1 multicast priority 1 cluster 1
 ip nhrp nhs 192.168.100.12 nbma 172.16.12.1 multicast priority 2 cluster 1
 ip nhrp nhs 192.168.100.21 nbma 172.16.21.1 multicast priority 1 cluster 2
 ip nhrp nhs 192.168.100.22 nbma 172.16.22.1 multicast priority 2 cluster 2
 ip nhrp nhs cluster 1 max-connections 1
 ip nhrp nhs cluster 2 max-connections 1
 ip nhrp shortcut
 ip tcp adjust-mss 1360
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint
```

Example A-2 displays the current state of the DMVPN tunnels. Notice that the entries for R12 and R22 are no-socket entries in NHRP state and R11 and R22 are in an *up* state.

Example A-2 *Clustered DMVPN Output*

```
R31-Spoke# show dmvpn detail
! Output omitted for brevity
IPv4 NHS:
192.168.100.11  RE NBMA Address: 172.16.11.1 priority = 1 cluster = 1
192.168.100.12   W NBMA Address: 172.16.12.1 priority = 2 cluster = 1
192.168.100.21  RE NBMA Address: 172.16.21.1 priority = 1 cluster = 2
192.168.100.22   W NBMA Address: 172.16.22.1 priority = 2 cluster = 2
Type:Spoke, Total NBMA Peers (v4/v6): 4

# Ent  Peer NBMA  Peer Tunnel
         Addr       Add         State  UpDn  Attrb  Target Network
----- ---------- -------------- ----- ------ ----- -----------------
1 172.16.11.1    192.168.100.11    UP 6d07h     S  192.168.100.11/32
```

```
1 172.16.12.1    192.168.100.12  NHRP 6d07h    SX  192.168.100.12/32
1 172.16.21.1    192.168.100.21    UP 6d07h     S  192.168.100.21/32
1 172.16.22.1    192.168.100.22  NHRP 6d07h    SX  192.168.100.22/32
```

Example A-3 displays the NHS redundancy status. Notice that R31 is not *expecting* any responses from R12 and R22 but shows the queue in a *waiting* state instead. The number of maximum connections per NHS cluster is shown for each cluster group at the bottom of the output.

Example A-3 *Clustered NHRP NHS Status*

```
R31-Spoke# show ip nhrp nhs redundancy
Legend: E=Expecting replies, R=Responding, W=Waiting
No. Interface Clus       NHS    Prty  Cur-State  Cur-Queue Prev-State Prev-Queue
  1 Tunnel100  1  192.168.100.11   1         RE    Running          E    Running
  2 Tunnel100  1  192.168.100.12   2          W    Waiting         RE    Running
  3 Tunnel100  2  192.168.100.21   1         RE    Running          E    Running
  4 Tunnel100  2  192.168.100.22   2          W    Waiting         RE    Running

No. Interface Clus Status Max-Con Totl-NHS Register/UP Expecting Waiting Fallbk
  1 Tunnel100      1 Enable      1       2           1          0       1      0
  2 Tunnel100      2 Enable      1       2           1          0       1      0
```

From the output of Example A-4, one can conclude that R31 has established an EIGRP neighborship with two of the four hubs, R11 and R21.

Example A-4 *Routing Table for the Clustered Model*

```
R31-Spoke# show ip route eigrp
! Output omitted for brevity
      10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
D        10.4.4.0/24 [90/52992000] via 192.168.100.11, 00:01:58, Tunnel100
                     [90/52992000] via 192.168.100.21, 00:01:58, Tunnel100
```

Further Reading

Cisco. *Cisco IOS Software Configuration Guides*. www.cisco.com.

Cisco. “DMVPN Tunnel Health Monitoring and Recovery.” www.cisco.com.

Appendix B

IPv6 Dynamic Multipoint VPN

DMVPN uses GRE tunnels and is capable of tunneling multiple protocols. Enhancements to NHRP added support for IPv6 so that multipoint GRE tunnels can find the appropriate IPv6 address. This means that DMVPN supports the use of IPv4 and IPv6 as the tunnel protocol or the transport protocol in the combination required.

All the concepts and commands explained in Chapter 3, "Dynamic Multipoint VPN," have an equivalent command to support IPv6. Table B-1 provides a list of the tunneled protocol commands for IPv4 and the equivalent for IPv6.

Table B-1 *Correlation of IPv4 to IPv6 Tunneled Protocol Commands*

IPv4 Command	IPv6 Command
ip mtu *mtu*	**ipv6 mtu** *mtu*
ip tcp adjust-mss *mss-size*	**ipv6 tcp adjust-mss** *mss-size*
ip nhrp network-id *1-4294967295*	**ipv6 nhrp network-id** *1-4294967295*
ip nhrp nhs *nhs-address* **nbma** *nbma-address* [**multicast**] [**priority** *0-255*] [**cluster** *0-10*]	**ipv6 nhrp nhs** *nhs-address* **nbma** *nbma-address* [**multicast**] [**priority** *0-255*] [**cluster** *0-10*]
ip nrhp redirect	**ipv6 nhrp redirect**
ip nhrp shortcut	**ipv6 nhrp shortcut**
ip nhrp authentication *password*	**ipv6 nhrp authentication** *password*
ip nhrp registration no-unique	**ipv6 nhrp registration no-unique**
ip nhrp holdtime *1-65535*	**ipv6 nhrp holdtime** *1-65535*
ip nhrp registration timeout *1-65535*	**ipv6 nhrp registration timeout** *1-65535*

(*Continued*)

Table B-1 *Continued*

IPv4 Command	IPv6 Command
ip nhrp nhs cluster *cluster-number* **max-connections** *0-255*	**ipv6 nhrp nhs cluster** *cluster-number* **max-connections** *0-255*
ip nhrp server-only	**ipv6 nhrp server-only**
ip route vrf *vrf-name* 0.0.0.0 0.0.0.0 *next-hop-ip*	**ipv6 route vrf** *vrf-name* 0.0.0.0 0.0.0.0 *next-hop-ip*

Table B-2 provides a list of the configuration commands that are needed to support an IPv6 transport network. Any tunnel commands not listed in Table B-2 are transport agnostic and are used regardless of the transport IP protocol version.

Table B-2 *Correlation of IPv4 to IPv6 Transport Protocol Commands*

IPv4 Command	IPv6 Command
tunnel mode gre multipoint	**tunnel mode gre multipoint ipv6**
ip route vrf *vrf-name* 0.0.0.0 0.0.0.0 *next-hop-ip*	**ipv6 route vrf** *vrf-name* 0.0.0.0 0.0.0.0 *next-hop-ip*

IPv6 over DMVPN can be interpreted differently depending upon perspective. There are three possible interpretations:

- **IPv4 over IPv6:** IPv4 is the tunneled protocol over an IPv6 transport network.
- **IPv6 over IPv6:** IPv6 is the tunneled protocol over an IPv6 transport network.
- **IPv6 over IPv4:** IPv6 is the tunneled protocol over an IPv4 transport network.

Regardless of your interpretation, DMVPN supports the IPv4 or IPv6 protocol as the tunneled protocol or the transport, but choosing the correct set of command groups is vital and depends upon the tunneling technique selected. Table B-3 provides a matrix so that you can select the appropriate commands from Table B-1 and Table B-2. It is important to note that the *nhs-address* or *NBMA-address* in Table B-1 can be an IPv4 or IPv6 address.

Table B-3 *Matrix of DMVPN Tunnel Technique to Configuration Commands*

Tunnel Mode	Tunnel Protocol Commands	Transport Commands
IPv4 over IPv4	IPv4	IPv4
IPv4 over IPv6	IPv4	IPv6
IPv6 over IPv4	IPv6	IPv4
IPv6 over IPv6	IPv6	IPv6

Note It is vital that a unique IPv6 link-local IP address be assigned to the tunnel interface when the tunneling protocol is IPv6. IPv6 routing protocols use link-local addresses to discover each other and install into the routing table.

Table B-4 provides a list of IPv4 display commands correlated to the IPv6 equivalents.

Table B-4 *Display Commands for IPv6 DMVPN*

IPv4 Command	IPv6 Command
show ip nhrp [brief|detail]	**show ipv6 nhrp [brief|detail]**
show dmvpn [ipv4][detail]	**show dmvpn [ipv6][detail]**
show ip nhrp traffic	**show ipv6 nhrp traffic**
show ip nhrp nhs [detail]	**show ipv6 nhrp nhs [detail]**

IPv6-over-IPv6 Sample Configuration

To fully understand an IPv6 DMVPN configuration, this book provides a sample configuration using the topology from Figure 3-8 in Chapter 3 for the IPv6-over-IPv6 topology. To simplify the IPv6 addressing scheme, the book's IPv6 addresses' first two hextets use 2001:db8 (the RFC-defined address space for IPv6 documentation). After the first two hextets, an IPv4 octet number is copied into an IPv6 hextet so the IPv6 addresses should look familiar. Table B-5 provides an example of how the book converts existing IPv4 addresses and networks to an IPv6 format.

Table B-5 *IPv6 Addressing Scheme*

IPv4 Address	IPv4 Network	IPv6 Address	IPv6 Network
10.1.1.11	10.1.1.0/24	2001:db8:10:1:1::11	2001:db8:10:1:1::/80
172.16.11.1	172.16.11.0/30	2001:db8:172:16:11::1	2001:db8:172:16:11::/126
10.1.0.11	10.1.0.11/32	2001:db8:10:1:0::11	2001:db8:10.1::11/128

Example B-1 provides the IPv6-over-IPv6 DMVPN configuration for hub routers R11 and R12. The configuration would look almost identical for R21 and R22. The VRF definition uses the **address-family ipv6** command, and the GRE tunnel is defined with the command **tunnel mode gre multipoint ipv6**. Notice that the tunnel interface has a regular IPv6 address configured and a link-local IPv6 address. The tunnel number is integrated into the link-local IP addressing.

Example B-1 *IPv6 DMVPN Hub Configuration on R11 and R12*

```
R11-Hub
vrf definition MPLS01
 address-family ipv6
 exit-address-family
!
interface Tunnel100
 description DMVPN-MPLS
 bandwidth 4000
 ipv6 tcp adjust-mss 1360
 ipv6 address FE80:100::11 link-local
 ipv6 address 2001:DB8:192:168:100::11/80
 ipv6 mtu 1380
 ipv6 nhrp authentication CISCO
 ipv6 nhrp map multicast dynamic
 ipv6 nhrp network-id 100
 ipv6 nhrp holdtime 600
 ipv6 nhrp redirect
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint ipv6
 tunnel key 100
 tunnel vrf MPLS01
!
interface GigabitEthernet0/1
 description MPLS01-TRANSPORT
 vrf forwarding MPLS01
 ipv6 address 2001:DB8:172:16:11::1/126
interface GigabitEthernet0/3
 description R12
 ipv6 address 2001:DB8:10:1:12::11/80
interface GigabitEthernet1/0
 description R10
 ipv6 address 2001:DB8:10:1:111::11/80
!
ipv6 route vrf MPLS01 ::/0 GigabitEthernet0/1 2001:DB8:172:16:11::2
```

```
R12-DC1-Hub
vrf definition INET01
 address-family ipv6
 exit-address-family
!
interface Tunnel200
 description DMVPN-Internet
 bandwidth 4000
 ipv6 tcp adjust-mss 1360
 ipv6 address FE80:200::12 link-local
 ipv6 address 2001:DB8:192:168:200::12/80
 ipv6 mtu 1380
 ipv6 nhrp authentication CISCO2
 ipv6 nhrp map multicast dynamic
 ipv6 nhrp network-id 200
 ipv6 nhrp holdtime 600
 ipv6 nhrp redirect
 tunnel source GigabitEthernet0/2
 tunnel mode gre multipoint ipv6
 tunnel key 200
 tunnel vrf INET01
!
interface GigabitEthernet0/2
 description INET01-TRANSPORT
 vrf forwarding INET01
 ipv6 address 2001:DB8:100:64:12::1/126
interface GigabitEthernet0/3
 description R11
 ipv6 address 2001:DB8:10:1:12::12/80
interface GigabitEthernet1/0
 description R10
 ipv6 address 2001:DB8:10:1:112::12/80
!
ipv6 route vrf INET01 ::/0 GigabitEthernet0/2 2001:DB8:100:64:12::2
```

Example B-2 provides the IPv6 DMVPN configuration for spoke routers R31 and R41.

Example B-2 *IPv6 DMVPN Configuration for R31 and R41*

```
R31-Spoke
vrf definition INET01
 address-family ipv6
 exit-address-family
vrf definition MPLS01
 address-family ipv6
 exit-address-family
!
interface Tunnel100
 description DMVPN-MPLS
 bandwidth 4000
 ipv6 tcp adjust-mss 1360
 ipv6 address FE80:100::31 link-local
 ipv6 address 2001:DB8:192:168:100::31/80
 ipv6 mtu 1380
 ipv6 nhrp authentication CISCO
 ipv6 nhrp map multicast dynamic
 ipv6 nhrp network-id 100
 ipv6 nhrp holdtime 600
 ipv6 nhrp nhs 2001:DB8:192:168:100::11 nbma 2001:DB8:172:16:11::1 multicast
 ipv6 nhrp nhs 2001:DB8:192:168:100::21 nbma 2001:DB8:172:16:21::1 multicast
 ipv6 nhrp shortcut
 if-state nhrp
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint ipv6
 tunnel key 100
 tunnel vrf MPLS01
!
interface Tunnel200
 description DMVPN-INET
 bandwidth 4000
 ipv6 tcp adjust-mss 1360
 ipv6 address FE80:200::31 link-local
 ipv6 address 2001:DB8:192:168:200::31/80
 ipv6 mtu 1400
 ipv6 nhrp authentication CISCO2
 ipv6 nhrp map multicast dynamic
 ipv6 nhrp network-id 200
 ipv6 nhrp holdtime 600
 ipv6 nhrp nhs 2001:DB8:192:168:200::12 nbma 2001:DB8:100:64:12::1 multicast
 ipv6 nhrp nhs 2001:DB8:192:168:200::22 nbma 2001:DB8:100:64:22::1 multicast
 ipv6 nhrp shortcut
 no nhrp route-watch
 if-state nhrp
 tunnel source GigabitEthernet0/2
 tunnel mode gre multipoint ipv6
 tunnel key 200
 tunnel vrf INET01
!
interface GigabitEthernet0/1
 description MPLS01-TRANSPORT
 vrf forwarding MPLS01
 ipv6 address 2001:DB8:172:16:31::1/126
interface GigabitEthernet0/2
 description INET01-TRANSPORT
 vrf forwarding INET01
 ipv6 address 2001:DB8:100:64:31::1/126
```

```
interface GigabitEthernet1/0
 description SiteB-Local-LAN
 ipv6 address 2001:DB8:10:3:3::31/80
!
ipv6 route vrf MPLS01 ::/0 GigabitEthernet0/1 2001:DB8:172:16:31::2
ipv6 route vrf INET01 ::/0 GigabitEthernet0/2 2001:DB8:100:64:31::2
```

R41-Spoke

```
vrf definition INET01
 address-family ipv6
 exit-address-family
vrf definition MPLS01
address-family ipv6
 exit-address-family
!
interface Tunnel100
 description DMVPN-MPLS
 bandwidth 4000
 ipv6 tcp adjust-mss 1360
 ipv6 address FE80:100::41 link-local
 ipv6 address 2001:DB8:192:168:100::41/80
 ipv6 mtu 1380
 ipv6 nhrp authentication CISCO
 ipv6 nhrp map multicast dynamic
 ipv6 nhrp network-id 100
 ipv6 nhrp holdtime 600
 ipv6 nhrp nhs 2001:DB8:192:168:100::11 nbma 2001:DB8:172:16:11::1 multicast
 ipv6 nhrp nhs 2001:DB8:192:168:100::21 nbma 2001:DB8:172:16:21::1 multicast
 ipv6 nhrp shortcut
 if-state nhrp
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint ipv6
 tunnel key 100
 tunnel vrf MPLS01
!
interface Tunnel200
 description DMVPN-INET
 bandwidth 4000
 ipv6 tcp adjust-mss 1360
 ipv6 address FE80:200::41 link-local
 ipv6 address 2001:DB8:192:168:200::41/80
 ipv6 mtu 1380
 ipv6 nhrp authentication CISCO2
 ipv6 nhrp map multicast dynamic
 ipv6 nhrp network-id 200
 ipv6 nhrp holdtime 600
 ipv6 nhrp nhs 2001:DB8:192:168:200::12 nbma 2001:DB8:100:64:12::1 multicast
 ipv6 nhrp nhs 2001:DB8:192:168:200::22 nbma 2001:DB8:100:64:22::1 multicast
 ipv6 nhrp shortcut
 ipv6 nhrp redirect
 no nhrp route-watch
 if-state nhrp
 tunnel source GigabitEthernet0/2
 tunnel mode gre multipoint ipv6
 tunnel key 200
 tunnel vrf INET01
!
interface GigabitEthernet0/1
 description MPLS01-TRANSPORT
 vrf forwarding MPLS01
 ipv6 address 2001:DB8:172:16:41::1/126
interface GigabitEthernet0/2
 description INET01-TRANSPORT
 vrf forwarding INET01
 ipv6 address 2001:DB8:100:64:41::1/126
interface GigabitEthernet1/0
 description Site4-Local-LAN
 ipv6 address 2001:DB8:10:4:4::41/80
!
ipv6 route vrf MPLS01 ::/0 GigabitEthernet0/1 2001:DB8:172:16:41::2
ipv6 route vrf INET01 ::/0 GigabitEthernet0/2 2001:DB8:100:64:41::2
```

IPv6 DMVPN Verification

The **show dmvpn** [**detail**] command can be used for viewing any DMVPN tunnel regardless of the tunnel or transport protocol. The data is structured slightly differently because of the IPv6 address format, but it still provides the same information as before.

Example B-3 displays the DMVPN tunnel state from R31 after it has established its static tunnels to the DMVPN hubs. Notice that the protocol transport now shows IPv6 and the NHS devices are using IPv6 addresses.

Example B-3 *Verification of IPv6 DMVPN*

```
R31-Spoke# show dmvpn detail
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        T1 - Route Installed, T2 - Nexthop-override
        C - CTS Capable
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
        UpDn Time --> Up or Down Time for a Tunnel
==========================================================================

Interface Tunnel100 is up/up, Addr. is 2001:DB8:192:168:100::31, VRF ""
   Tunnel Src./Dest. addr: 2001:DB8:172:16:31::1/MGRE, Tunnel VRF "MPLS01"
   Protocol/Transport: "multi-GRE/IPv6", Protect ""
   Interface State Control: Enabled
   nhrp event-publisher : Disabled

IPv6 NHS:
2001:DB8:192:168:100::11  RE NBMA Address: 2001:DB8:172:16:11::1 priority = 0
  cluster = 0
2001:DB8:192:168:100::21  RE NBMA Address: 2001:DB8:172:16:21::1 priority = 0
  cluster = 0
Type:Spoke, Total NBMA Peers (v4/v6): 2
    1.Peer NBMA Address: 2001:DB8:172:16:11::1
        Tunnel IPv6 Address: 2001:DB8:192:168:100::11
        IPv6 Target Network: 2001:DB8:192:168:100::11/128
        # Ent: 2, Status: UP, UpDn Time: 00:00:53, Cache Attrib: S
! Following entry is shown in the detailed view and uses link-local addresses
    2.Peer NBMA Address: 2001:DB8:172:16:11::1
        Tunnel IPv6 Address: FE80:100::11
        IPv6 Target Network: FE80:100::11/128
        # Ent: 0, Status: NHRP, UpDn Time: never, Cache Attrib: SC
    3.Peer NBMA Address: 2001:DB8:172:16:21::1
        Tunnel IPv6 Address: 2001:DB8:192:168:100::21
        IPv6 Target Network: 2001:DB8:192:168:100::21/128
        # Ent: 2, Status: UP, UpDn Time: 00:00:53, Cache Attrib: S
! Following entry is shown in the detailed view and uses link-local addresses
    4.Peer NBMA Address: 2001:DB8:172:16:21::1
        Tunnel IPv6 Address: FE80:100::21
        IPv6 Target Network: FE80:100::21/128
        # Ent: 0, Status: NHRP, UpDn Time: never, Cache Attrib: SC

Interface Tunnel200 is up/up, Addr. is 2001:DB8:192:168:200::31, VRF ""
   Tunnel Src./Dest. addr: 2001:DB8:100:64:31::1/MGRE, Tunnel VRF "INET01"
   Protocol/Transport: "multi-GRE/IPv6", Protect ""
   Interface State Control: Enabled
   nhrp event-publisher : Disabled

IPv6 NHS:
2001:DB8:192:168:200::12  RE NBMA Address: 2001:DB8:100:64:12::1 priority = 0
  cluster = 0
2001:DB8:192:168:200::22  RE NBMA Address: 2001:DB8:100:64:22::1 priority = 0
  cluster = 0
Type:Spoke, Total NBMA Peers (v4/v6): 2
    1.Peer NBMA Address: 2001:DB8:100:64:12::1
        Tunnel IPv6 Address: 2001:DB8:192:168:200::12
        IPv6 Target Network: 2001:DB8:192:168:200::12/128
        # Ent: 2, Status: UP, UpDn Time: 00:00:52, Cache Attrib: S
! Following entry is shown in the detailed view and uses link-local addresses
    2.Peer NBMA Address: 2001:DB8:100:64:12::1
        Tunnel IPv6 Address: FE80:200::12
        IPv6 Target Network: FE80:200::12/128
        # Ent: 0, Status: NHRP, UpDn Time: never, Cache Attrib: SC
    3.Peer NBMA Address: 2001:DB8:100:64:22::1
        Tunnel IPv6 Address: 2001:DB8:192:168:200::22
        IPv6 Target Network: 2001:DB8:192:168:200::22/128
        # Ent: 2, Status: UP, UpDn Time: 00:00:52, Cache Attrib: S
! Following entry is shown in the detailed view and uses link-local addresses
    4.Peer NBMA Address: 2001:DB8:100:64:22::1
        Tunnel IPv6 Address: FE80:200::22
        IPv6 Target Network: FE80:200::22/128
        # Ent: 0, Status: NHRP, UpDn Time: never, Cache Attrib: SC
```

Example B-4 demonstrates the IPv6 NHRP information. Notice that all the NHRP message flags are consistent between IPv4 and IPv6.

Example B-4 *DMVPN Configuration for R51 and R52 (Dual Routers at Site)*

```
R31-Spoke# show ipv6 nhrp  brief
****************************************************************************
    NOTE: Link-Local, No-socket and Incomplete entries are not displayed
****************************************************************************
Legend: Type --> S - Static, D - Dynamic
        Flags --> u - unique, r - registered, e - temporary, c - claimed
        a - authoritative, t - route
============================================================================

Intf     NextHop Address                                    NBMA Address
         Target Network                              T/Flag
-------- ------------------------------------------- ------ ----------------
```

```
Tu100    2001:DB8:192:168:100::11                                2001:DB8:172:16:11::1
         2001:DB8:192:168:100::11/128                   S/
Tu100    2001:DB8:192:168:100::21                                2001:DB8:172:16:21::1
         2001:DB8:192:168:100::21/128                   S/
Tu200    2001:DB8:192:168:200::12                                2001:DB8:100:64:12::1
         2001:DB8:192:168:200::12/128                   S/
Tu200    2001:DB8:192:168:200::22                                2001:DB8:100:64:22::1
         2001:DB8:192:168:200::22/128                   S/
```

```
R31-Spoke# show ipv6 nhrp
2001:DB8:192:168:100::11/128 via 2001:DB8:192:168:100::11
   Tunnel100 created 00:02:55, never expire
   Type: static, Flags: used
   NBMA address: 2001:DB8:172:16:11::1
2001:DB8:192:168:100::21/128 via 2001:DB8:192:168:100::21
   Tunnel100 created 00:02:55, never expire
   Type: static, Flags: used
   NBMA address: 2001:DB8:172:16:21::1
FE80:100::11/128 via FE80:100::11
   Tunnel100 created 00:02:55, never expire
   Type: static, Flags: used nhs-ll
   NBMA address: 2001:DB8:172:16:11::1
FE80:100::21/128 via FE80:100::21
   Tunnel100 created 00:02:55, never expire
   Type: static, Flags: used nhs-ll
   NBMA address: 2001:DB8:172:16:21::1
2001:DB8:192:168:200::12/128 via 2001:DB8:192:168:200::12
   Tunnel200 created 00:02:55, never expire
   Type: static, Flags: used
   NBMA address: 2001:DB8:100:64:12::1
2001:DB8:192:168:200::22/128 via 2001:DB8:192:168:200::22
   Tunnel200 created 00:02:55, never expire
   Type: static, Flags: used
   NBMA address: 2001:DB8:100:64:22::1
FE80:200::12/128 via FE80:200::12
   Tunnel200 created 00:02:54, never expire
   Type: static, Flags: used nhs-ll
   NBMA address: 2001:DB8:100:64:12::1
FE80:200::22/128 via FE80:200::22
   Tunnel200 created 00:02:54, never expire
   Type: static, Flags: used nhs-ll
   NBMA address: 2001:DB8:100:64:22::1
```

Example B-5 demonstrates the connectivity between R31 and R41 before and after the spoke-to-spoke DMVPN tunnel is established.

Example B-5 *IPv6 Connectivity Between R31 and R41*

```
! Initial packet flow
R31-Spoke# traceroute 2001:db8:10:4:4::41
Tracing the route to 2001:DB8:10:4:4::41
  1 2001:DB8:192:168:100::11 2 msec
    2001:DB8:192:168:100::21 4 msec
    2001:DB8:192:168:100::11 4 msec
  2 2001:DB8:192:168:100::41 5 msec 4 msec 5 msec
```

```
! Packet flow after spoke-to-spoke tunnel is established
R31-Spoke# traceroute 2001:db8:10:4:4::41
Tracing the route to 2001:DB8:10:4:4::41
  1 2001:DB8:192:168:100::41 4 msec 1 msec 4 msec
```

IPv4 over IPv6 Sample Configuration

To conclude this topic, a sample configuration for IPv4 over IPv6 has been provided in Example B-6 for the DMVPN hub routers.

Example B-6 *IPv6 DMVPN Hub Configuration on R11 and R12*

```
R11-Hub
vrf definition MPLS01
 address-family ipv6
 exit-address-family
!
interface Tunnel100
 description DMVPN-MPLS
 bandwidth 4000
 ip address 192.168.100.11 255.255.255.0
 ip mtu 1400
 ip nhrp authentication CISCO
 ip nhrp map multicast dynamic
 ip nhrp network-id 100
 ip nhrp holdtime 600
 ip nhrp redirect
 ip tcp adjust-mss 1360
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint ipv6
 tunnel key 100
 tunnel vrf MPLS01
```

```
!
interface GigabitEthernet0/1
 description MPLS01-TRANSPORT
 vrf forwarding MPLS01
 ipv6 address 2001:DB8:172:16:11::1/126
interface GigabitEthernet0/3
 description R12
 ip address 10.1.12.11 255.255.255.0
interface GigabitEthernet1/0
 description R10
 ip address 10.1.111.11
!
ip route vrf MPLS01 0.0.0.0 0.0.0.0 172.16.11.2
```

R12-DC1-Hub

```
vrf definition INET01
 address-family ipv6
 exit-address-family
!
interface Tunnel200
 description DMVPN-Internet
 bandwidth 4000
 ip mtu 1400
 ip nhrp authentication CISCO2
 ip nhrp map multicast dynamic
 ip nhrp network-id 200
 ip nhrp holdtime 600
 ip nhrp redirect
 ip tcp adjust-mss 1360
 tunnel source GigabitEthernet0/2
 tunnel mode gre multipoint ipv6
 tunnel key 200
 tunnel vrf INET01
!
interface GigabitEthernet0/2
 description INET01-TRANSPORT
 vrf forwarding INET01
 ipv6 address 2001:DB8:100:64:12::1/126
interface GigabitEthernet0/3
 description R11
 ip address 10.1.12.12 255.255.255.0
interface GigabitEthernet1/0
 description R10
 ip address 10.1.112.12 255.255.255.0
!
ip route vrf INET01 0.0.0.0 0.0.0.0 100.64.12.2
```

Example B-7 provides the IPv4-over-IPv6 DMVPN configuration for spoke router R31.

Example B-7 *IPv6 DMVPN Configuration for R31 and R41*

```
R31-Spoke
vrf definition INET01
 address-family ipv6
 exit-address-family
vrf definition MPLS01
 address-family ipv6
 exit-address-family
!
interface Tunnel100
 description DMVPN-MPLS
 bandwidth 4000
ip address 192.168.100.31 255.255.255.0
 ip mtu 1400
 ip nhrp authentication CISCO
 ip nhrp network-id 100
 ip nhrp holdtime 600
 ip nhrp nhs 192.168.100.11 nbma 2001:DB8:172:16:11::1 multicast
 ip nhrp nhs 192.168.100.21 nbma 2001:DB8:172:16:21::1 multicast
 ip nhrp shortcut
 ip tcp adjust-mss 1360
 if-state nhrp
 tunnel source GigabitEthernet0/1
 tunnel mode gre multipoint ipv6
 tunnel key 100
 tunnel vrf MPLS01
!
interface Tunnel200
 description DMVPN-INET
 bandwidth 4000
 ip address 192.168.200.31 255.255.255.0
 ip mtu 1400
 ip nhrp network-id 200
 ip nhrp holdtime 600
 ip nhrp nhs 192.168.200.12 nbma 2001:DB8:100:64:12::1 multicast
 ip nhrp nhs 192.168.200.22 nbma 2001:DB8:100:64:22::1 multicast
 ip nhrp registration no-unique
 ip nhrp shortcut
 ip tcp adjust-mss 1360
 load-interval 30
 if-state nhrp
 tunnel source GigabitEthernet0/2
 tunnel mode gre multipoint ipv6
 tunnel key 200
 tunnel vrf INET01
!
interface GigabitEthernet0/1
 description MPLS01-TRANSPORT
 vrf forwarding MPLS01
 ipv6 address 2001:DB8:172:16:31::1/126
interface GigabitEthernet0/2
 description INET01-TRANSPORT
 vrf forwarding INET01
 ipv6 address 2001:DB8:100:64:31::1/126
interface GigabitEthernet1/0
 description SiteB-Local-LAN
 ip address 10.3.3.31 255.255.255.0
!
ipv6 route vrf MPLS01 ::/0 Ethernet0/1 2001:DB8:172:16:31::2
ipv6 route vrf INET01 ::/0 Ethernet0/2 2001:DB8:100:64:31::2
```

IPv4-over-IPv6 Verification

Example B-8 displays the DMVPN tunnel state from R31 after it has established its static tunnels to the DMVPN hubs. Notice that the protocol transport now shows IPv4 addresses and the transport uses IPv6 addresses.

Example B-8 *Verification of IPv6 DMVPN*

```
R31-Spoke# show dmvpn detail
Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        T1 - Route Installed, T2 - Nexthop-override
        C - CTS Capable
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
        UpDn Time --> Up or Down Time for a Tunnel
==========================================================================
Interface Tunnel100 is up/up, Addr. is 192.168.100.31, VRF ""
   Tunnel Src./Dest. addr: 2001:DB8:172:16:31::1/MGRE, Tunnel VRF "MPLS01"
   Protocol/Transport: "multi-GRE/IPv6", Protect ""
   Interface State Control: Enabled
   nhrp event-publisher : Disabled

IPv4 NHS:
192.168.100.11  RE NBMA Address: 2001:DB8:172:16:11::1 priority = 0 cluster = 0
192.168.100.21  RE NBMA Address: 2001:DB8:172:16:21::1 priority = 0 cluster = 0
Type:Spoke, Total NBMA Peers (v4/v6): 2
```

```
# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1 2001:DB8:172:16:11::1
                       192.168.100.11    UP 00:00:08     S  192.168.100.11/32
    1 2001:DB8:172:16:21::1
                       192.168.100.21    UP 00:00:08     S  192.168.100.21/32

Interface Tunnel200 is up/up, Addr. is 192.168.200.31, VRF ""
   Tunnel Src./Dest. addr: 2001:DB8:100:64:31::1/MGRE, Tunnel VRF "INET01"
   Protocol/Transport: "multi-GRE/IPv6", Protect ""
   Interface State Control: Enabled
   nhrp event-publisher : Disabled

IPv4 NHS:
192.168.200.12  RE NBMA Address: 2001:DB8:100:64:12::1 priority = 0 cluster = 0
192.168.200.22  RE NBMA Address: 2001:DB8:100:64:22::1 priority = 0 cluster = 0
Type:Spoke, Total NBMA Peers (v4/v6): 2

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb    Target Network
----- --------------- --------------- ----- -------- ----- -----------------
    1 2001:DB8:100:64:12::1
                       192.168.200.12    UP 00:00:08     S  192.168.200.12/32
    1 2001:DB8:100:64:22::1
                       192.168.200.22    UP 00:00:08     S  192.168.200.22/32
```

Further Reading

Cisco. *Cisco IOS Software Configuration Guides*. www.cisco.com.

Cisco. “IPv6 over DMVPN.” www.cisco.com.

Index