JunosE™ Software for E Series™ Broadband Services Routers

BGP and MPLS Configuration Guide

Release

15.1.x

Published: 2014-08-18

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

JunosE™ Software for E Series™ Broadband Services Routers BGP and MPLS Configuration Guide
Release 15.1.x

Revision History
August 2014—FRS JunosE 15.1.x

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

Abbreviated Table of Contents

Table of Contents

Part 2 Multiprotocol Layer Switching

List of Figures

List of Tables

About the Documentation

E Series and JunosE Documentation and Release Notes

For a list of related JunosE documentation, see http://www.juniper.net/techpubs/software/index.html.

If the information in the latest release notes differs from the information in the documentation, follow the *JunosE Release Notes.*

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at http://www.juniper.net/techpubs/.

Audience

This guide is intended for experienced system and network specialists working with Juniper Networks E Series Broadband Services Routers in an Internet access environment.

E Series and JunosE Text and Syntax Conventions

Table 1 on page xxxiv defines notice icons used in this documentation.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

Table 2 on page xxxiv defines text and syntax conventions that we use throughout the E Series and JunosE documentation.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents commands and keywords in text.	• Issue the **clock source** command. • Specify the keyword **exp-msg**.
Bold text like this	Represents text that the user must type.	**host1(config)#traffic class low-loss1**

`Fixed-width text like this`	Represents information as displayed on your terminal's screen.	**host1#show ip ospf 2**
		`Routing Process OSPF 2 with Router ID 5.5.0.250`
		`Router is an Area Border Router (ABR)`

Italic text like this	• Emphasizes words. • Identifies variables. • Identifies chapter, appendix, and book names.	• There are two levels of access: *user* and *privileged*. • *clusterId, ipAddress.* • *Appendix A, System Specifications*
Plus sign (+) linking key names	Indicates that you must press two or more keys simultaneously.	Press Ctrl + b.

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
Syntax Conventions in the Command Reference Guide		
Plain text like this	Represents keywords.	terminal length
Italic text like this	Represents variables.	*mask*, *accessListName*
| (pipe symbol)	Represents a choice to select one keyword or variable to the left or to the right of this symbol. (The keyword or variable can be either optional or required.)	diagnostic | line
[ ] (brackets)	Represent optional keywords or variables.	[internal | external]
[ ]* (brackets and asterisk)	Represent optional keywords or variables that can be entered more than once.	[level1 | level2 | l1]*
{ } (braces)	Represent required keywords or variables.	{ permit | deny } { in | out } { *clusterId* | *ipAddress* }

Obtaining Documentation

To obtain the most current version of all Juniper Networks technical documentation, see the Technical Documentation page on the Juniper Networks website at http://www.juniper.net/.

To download complete sets of technical documentation to create your own documentation CD-ROMs or DVD-ROMs, see the Portable Libraries page at

http://www.juniper.net/techpubs/resources/index.html

Copies of the Management Information Bases (MIBs) for a particular software release are available for download in the software image bundle from the Juniper Networks website at http://www.juniper.net/.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation to better meet your needs. Send your comments to **techpubs-comments@juniper.net**, or fill out the documentation feedback form at https://www.juniper.net/cgi-bin/docbugreport/. If you are using e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number
- Software release version

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: http://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://tools.juniper.net/SerialNumberEntitlementSearch/

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at http://www.juniper.net/cm/.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see http://www.juniper.net/support/requesting-support.html.

PART 1

Border Gateway Protocol

CHAPTER 1

Configuring BGP Routing

This chapter contains the following sections:

Overview

The Border Gateway Protocol (BGP) provides loop-free interdomain routing between autonomous systems (ASs). This section describes some of the main concepts of BGP.

Conventions in This Chapter

Certain terms used with BGP, such as the names of attributes and messages, are typically expressed in all uppercase letters in the RFCs. For improved readability, those terms are represented in lowercase in this chapter. Table 3 on page 4 lists the terms and their variant spellings.

Table 3: Conventions for BGP Terms

In This Chapter	In RFCs
aggregator	AGGREGATOR
AS-confed-set	AS_CONFED_SET
AS-path or AS path	AS_PATH
AS-sequence	AS_SEQUENCE
AS-set	AS_SET
atomic-aggregate	ATOMIC_AGGREGATE
cluster-list	CLUSTER_LIST
keepalive	KEEPALIVE
local-pref	LOCAL_PREF
multiexit discriminator or MED	MULTI_EXIT_DISC
new-as-path	NEW_AS_PATH
new-aggregator	NEW_AGGREGATOR
next-hop or next hop	NEXT_HOP
no-advertise	NO_ADVERTISE
no-export	NO_EXPORT
no-export-subconfed	NO_EXPORT_SUBCONFED
notification	NOTIFICATION
open	OPEN
origin	ORIGIN
originator-ID	ORIGINATOR_ID
route-refresh	ROUTE-REFRESH
update	UPDATE

Autonomous Systems

An autonomous system (AS) is a set of routers that use the same routing policy while running under a single technical administration. An AS runs interior gateway protocols (IGPs) such as RIP, OSPF, and IS-IS within its boundaries. ASs use exterior gateway protocols (EGPs) to exchange routing information with other ASs. BGP is an EGP.

The outside world views an AS as a single entity, even though it can be a collection of IGPs working together to provide routing within its interior.

Each AS has an identification number provided by an Internet registry or by an Internet service provider (ISP) that uniquely identifies it to the outside world.

BGP Speaker

A router that has been configured to run the BGP routing protocol is called a BGP speaker.

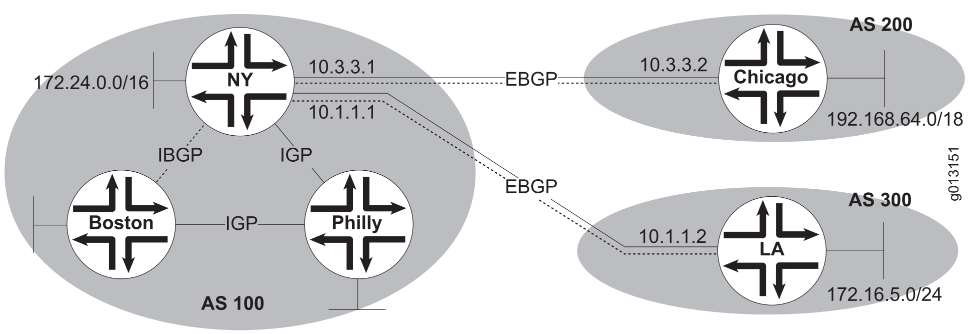
BGP Peers and Neighbors

Unlike some other routing protocols, BGP speakers do not automatically discover each other and begin exchanging information. Instead, each BGP speaker must be explicitly configured with a set of BGP peers with which it exchanges routing information. BGP peers do not have to be directly connected to each other in order to share a BGP session. Another term for BGP peer is BGP neighbor. A BGP *peer group* consists of two or more BGP peers that share a common set of update policies.

In Figure 1 on page 5, router NY and router Chicago are peers. Router NY and router LA are peers. Router NY and router Boston are peers. Router NY and router Philly are not peers. Router Chicago and router LA are not peers.

NOTE: The figures in this chapter indicate a BGP session with a dotted line. A physical link is represented by a solid line.

Figure 1: BGP Peers



BGP Session

When two BGP speakers have both been configured to be BGP peers of each other, they will establish a BGP session to exchange routing information. A BGP session is simply a

TCP connection over which routing information is exchanged according to the rules of the BGP protocol.

Because BGP relies on TCP to provide reliable and flow-controlled transmission of routing information, the BGP protocol itself is very simple. However it also implies that two routers can be BGP peers of each other only if they are reachable from each other in the sense that they can exchange IP packets.

In practice this means that either of the following must be true:

- The BGP peers must be connected to a common IP subnet.
- The BGP peers must be in the same AS, which runs an IGP enabling the BGP peers to reach each other.

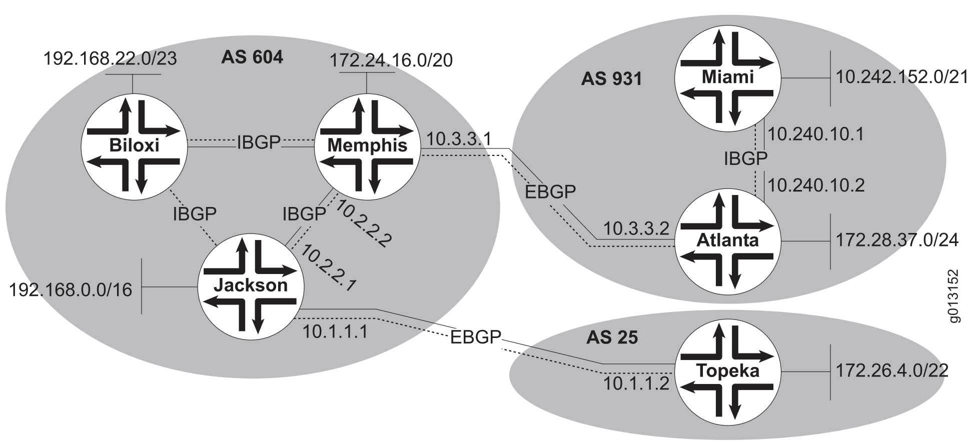
IBGP and EBGP

When two BGP speakers are in the same autonomous system, the BGP session is called an *internal* BGP session, or IBGP session. When two BGP speakers are in different autonomous systems, the BGP session is called an *external* BGP session, or EBGP session. BGP uses the same types of message on IBGP and EBGP sessions, but the rules for when to send which message and how to interpret each message differ slightly; for this reason some people refer to IBGP and EBGP as two separate protocols.

IBGP requires that BGP speakers within an autonomous system be fully meshed, meaning that there must be a BGP session between each pair of peers within the AS. IBGP does not require that all the peers be physically connected. EBGP does not require full meshing of BGP speakers. EBGP sessions typically exist between peers that are physically connected.

Figure 2 on page 6 shows an example of the exchange of information between routers running IBGP and EBGP across multiple ASs.

Figure 2: Internal and External BGP



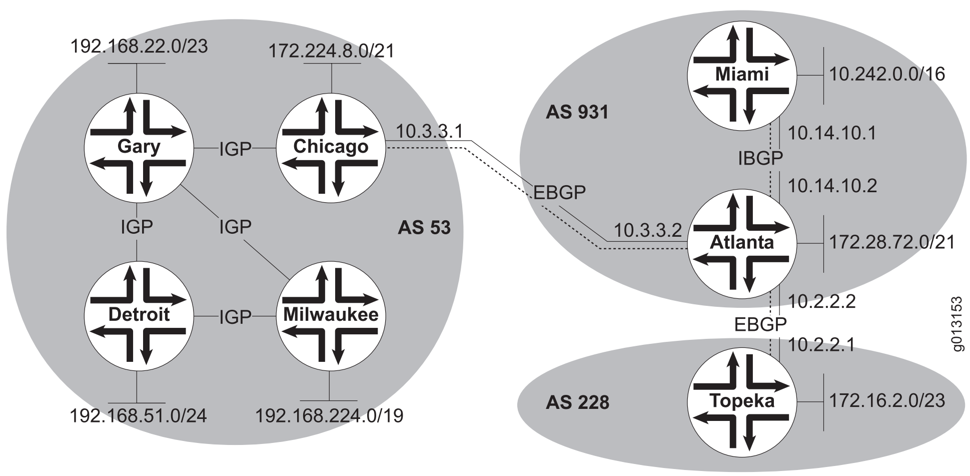
Interior Gateway Protocols

Not all the routers within an AS have to be BGP peers. For example, in some large enterprise networks, ASs generally have many more non-BGP routers. These routers communicate using an interior gateway protocol (IGP) such as the following:

- Intermediate System-to-Intermediate System (IS-IS)
- Open Shortest Path First (OSPF)
- Routing Information Protocol (RIP)

Figure 3 on page 7 shows that the routers in AS 53 all communicate with each other using an IGP. Routing information internal to AS 53 is redistributed from the IGP into BGP at router Chicago. Router Chicago redistributes into the IGP the routing information it receives from its external BGP peer, router Atlanta. Router Atlanta has an internal BGP link within its AS, and an external BGP link to router Topeka.

Figure 3: Interior Gateway Protocols



BGP Messages

BGP speakers exchange routing information with each other by exchanging BGP messages over a BGP session. BGP uses the following five message types:

- Open BGP messages—When two BGP speakers establish a BGP session with each other, the first message they exchange after the underlying TCP session has been established is an *open message*. This message contains various bits of information that enable the two BGP peers to determine whether they want to establish a BGP session with each other—for example, the AS number of the BGP speaker—and to negotiate certain parameters for the BGP session—for example, how often to send a keepalive message.
- Update messages—The update message is the most important message in the BGP protocol. A BGP speaker sends update messages to announce routes to prefixes that it can reach and to withdraw routes to prefixes that it can no longer reach.

- Keepalive messages—BGP speakers periodically exchange keepalive messages to determine whether the underlying TCP connection is still up.
- Notification messages—If a BGP speaker wishes to terminate a BGP session (either because it has been configured to do so or because it has detected some error condition), it will send a notification message to its peer specifying the reason for terminating the BGP session.

  If the session is being terminated for a nonfatal error, the notification messages includes the error code cease. Subcodes sent in the notification message can inform network operators about peering problems and help them better understand network events. Table 4 on page 8 lists the subcodes defined for BGP notification messages bearing the cease code.

Table 4: Cease Notification Message Subcodes

Subcode	Reason	Symbolic Name
1	The number of address prefixes received from the peer has exceeded the upper bound configured with the **neighbor maximum-prefix** command. The notification message can include address family and upper bound information in the data field.	Maximum Number of Prefixes Reached
2	The BGP speaker is administratively shutting down the session.	Administratively Shutdown
3	The BGP speaker is removing the peer configuration.	Peer Unconfigured
4	The BGP speaker is administratively resetting the session.	Administratively Reset
5	The BGP speaker is rejecting the connection (for example, because the peer is not configured locally on the speaker) after accepting a transport protocol connection.	Connection Rejected
6	The BGP speaker is administratively resetting the session for some other configuration.	Other Configuration Change

- Route-refresh messages—BGP speakers can send route-refresh messages to peers that advertise the route-refresh capability. The messages contain a request for the peer to resend its routes to the router. This feature enables the BGP speaker to apply modified or new policies to the routes when it receives them again.

BGP Route

A *BGP route* consists of two parts, a prefix and a set of path attributes. It is not uncommon to use the term *path* to refer to a BGP route, although that term technically refers to one of the path attributes of that route.

Routing Information Base

BGP routes are stored in a BGP speaker's routing information base (RIB), also known as its routing table, which conceptually consists of the following three parts:

- Adj-RIBs-In store unprocessed routes learned from update messages received by the BGP speaker.
- Loc-RIB contains local routes resulting from the BGP speaker applying its local policies to the routes contained in its Adj-RIBs-In.
- Adj-RIBs-Out store routes that the BGP speaker will advertise to its peers in the update messages it sends.

Prefixes and CIDR

A *prefix* describes a set of IP addresses that can be reached using the route. For example, the prefix 10.1.1.0/24 indicates all IP addresses whose first 24 bits contain the value 10.1.1. The term *network* is sometimes used instead of *prefix* to describe a set of addresses. To reduce confusion, this chapter restricts *network* to its more common usage, to refer to a physical structure of routers and links.

Prefixes are made possible by classless interdomain routing (CIDR). CIDR addresses have largely replaced the concept of classful addresses (such as Class A, Class B, and Class C) in the Internet. Classful addresses have an implicit, fixed-length mask corresponding to the predefined class boundaries. For example, 192.56.0.0 is a Class B address with an implicit (or natural) mask of 255.255.0.0.

CIDR uses network prefixes and explicit masks, represented by a prefix length, enabling network prefixes of arbitrary lengths. CIDR represents the sample address above as 192.56.0.0/16. The /16 indicates that the high-order 16 bits (the first 16 bits counting from left to right) in the address mask are all 1s.

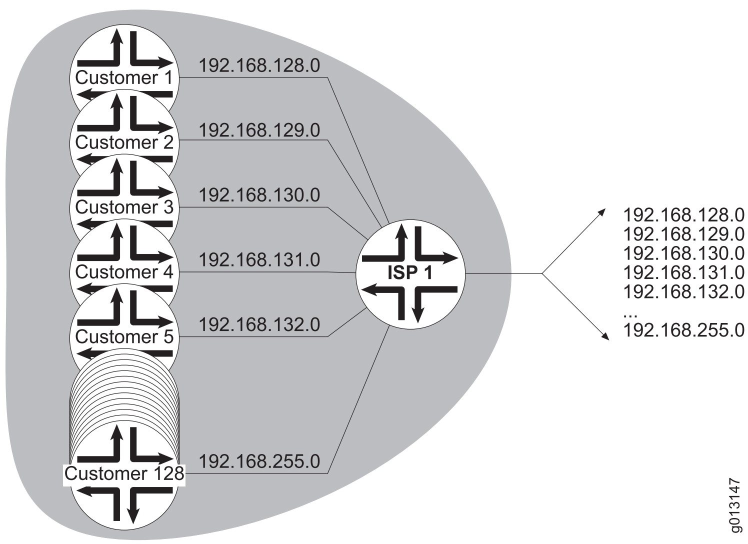
CIDR enables you to aggregate multiple classful addresses into a single classless advertisement, reducing the number of advertisements that must be made to provide full access to all the addresses. Suppose an ISP has customers with the following addresses:

192.168.128.0
192.168.129.0
192.168.130.0
192.168.131.0

192.168.132.0
192.168.133.0
...
192.168.255.0

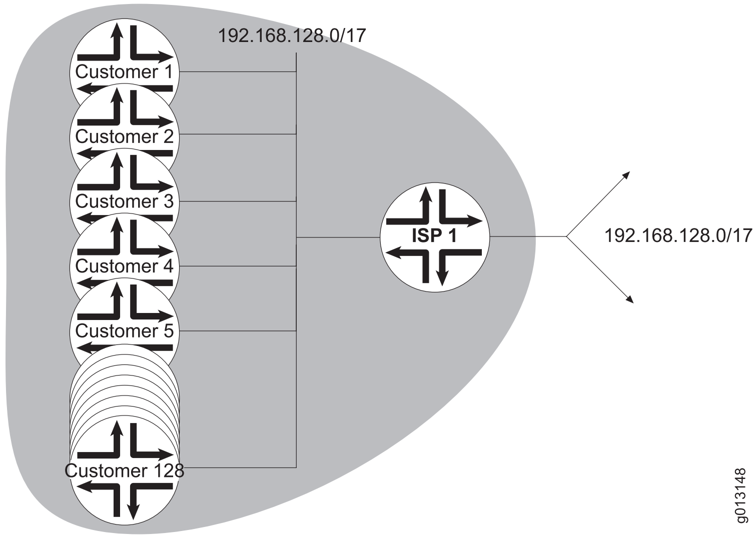
Without CIDR, the ISP has to advertise a route to each address, as shown in Figure 4 on page 10.

Figure 4: Routing Without CIDR



With CIDR, the ISP can aggregate the routes as 192.168.128.0/17 and advertise a single address to that prefix, as shown in Figure 5 on page 11.

Figure 5: Routing with CIDR



Path Attributes

A path attribute provides some additional information about a route. If a BGP speaker has more than one route to the same destination prefix, it selects one of those routes to use (the "best" route) based on the path attributes. BGP as implemented on the Juniper Networks E Series Broadband Services Router specifies detailed and complex criteria for picking the best route; this helps ensure that all routers will converge to the same routing table, a necessary behavior to avoid routing loops. See "Selecting the Best Path" on page 104 for more information.

The following are some of the most important path attributes:

- *AS-path* specifies the sequence of autonomous systems that must be crossed to reach a certain destination. This path attribute is used to avoid routing loops and to prefer shorter routes over longer routes.
- *Next-hop* specifies the IP address of the ingress router in the next autonomous system on the path to the destination.
- *Local-pref* and *multiexit discriminator* (MED) are metrics that administrators can tune to ensure that certain routes are more attractive over other routes. The local-pref attribute specifies a degree of preference that enables a router to select among multiple routes to the same prefix. The MED is used for ASs that have more than one connection to each other. The administrator of one AS sets the MED to express a degree of preference for one link versus another; the BGP peer in the other AS uses this MED to optimize traffic.
- *Originator-ID* specifies the IP address of the router that originates the route. The router ignores updates that have this attribute set to its own IP address.
- *Atomic-aggregate* and *aggregator* inform peers about actions taken by a BGP speaker regarding aggregation of routes. If a BGP speaker aggregates routes that have differing path attributes, it includes the atomic-aggregate attribute with the aggregated prefix to inform update recipients that they must not deaggregate the prefix. A BGP speaker

aggregating routes can include the aggregator attribute to indicate the router and AS where the aggregation was performed.

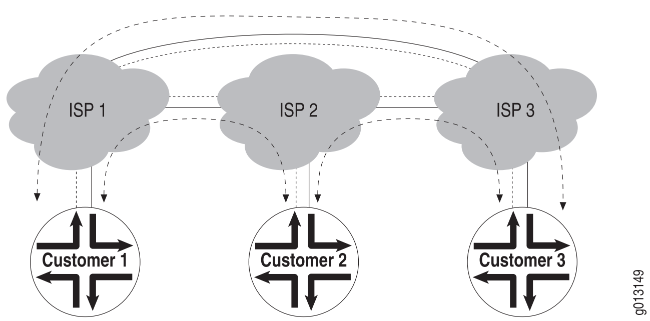
- *Community* and *extended community* identify prefixes as sharing some common attribute, providing a means of grouping prefixes and enacting routing policies on the group of prefixes. A prefix can belong to more than one community. You can specify a community name as a 32-bit string, a standards-defined well-known community, or an AS number combined with a 32-bit number to create a unique identifier. An extended community name consists of either an IP address or an AS number, combined with a 32-bit or 16-bit number to create a unique identifier.

Transit and Nontransit Service

While an ISP provides connectivity to its customers, it also provides connectivity to customers of other ISPs. In doing this, an ISP must be able to ensure the appropriate use of its resources.

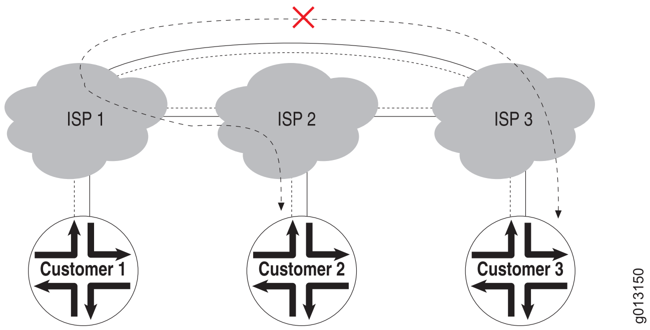
For example, Figure 6 on page 12 shows three ISPs and three customers. ISP 1, ISP 2, and ISP 3 are directly connected to one another through a physical link and a corresponding EBGP session (represented here by a single line). Customer 1 is connected to ISP 1 through a physical link and corresponding EBGP session. Customer 2 is similarly connected to ISP 2, and Customer 3 is similarly connected to ISP 3. Each ISP provides *transit* service to its own customers. Figure 6 on page 12 illustrates how the ISP permits traffic to transit across its backbone from its own customers or to its own customers.

Figure 6: Transit Service



Each ISP provides *nontransit* service to other ISPs. For example, Figure 7 on page 13 shows that ISP 1 does not permit traffic between ISP 2 and ISP 3 to cross its backbone. If ISP 1 permits such traffic, it squanders its own resources with no benefit to its customers or itself.

Figure 7: Nontransit Service



IPv6 BGP Support

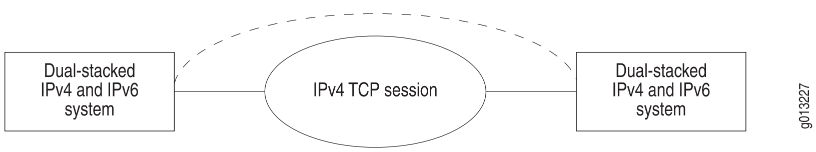
Most of the extensions and features available in BGP for IPv4 are also available for the IPv6 address family, such as policy-based routing, redistributing routes to and from other protocols, route aggregation, route flap dampening, and confederations. For a description of IPv6, see *Configuring IPv6* in *JunosE IP, IPv6, and IGP Configuration Guide*.

Multiprotocol Extensions for BGP-4 (MP-BGP) allow the exchange of IPv6 routing information over TCP IPv4 (Figure 8 on page 13) or TCP IPv6 transport (Figure 9 on page 14).

Exchange of IPv6 Routing Information over TCP IPv4

Figure 8 on page 13 illustrates the exchange of IPv6 routing information over a TCP IPv4 connection.

Figure 8: IPv6 Routing over TCP IPv4



The E Series router's MP-BGP implementation uses BGP update messages to announce the feasible routes to an associated IPv6 BGP next hop and also to announce the nonfeasible routes that need to be withdrawn from the peer. The E Series router announces only IPv6 global addresses as the BGP next-hop address; it does not use the optional link-local IPv6 address as the BGP next hop.

BGP determines the next-hop addresses to be announced by using the IPv4-compatible IPv6 address. For example, the following table shows the translation of an IPv4 address.

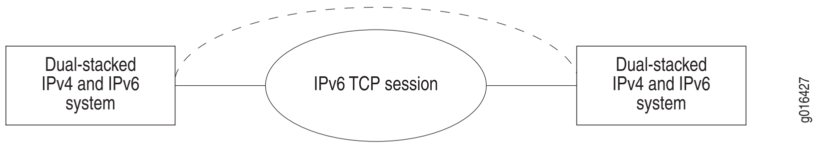
IPv4 address	IPv6 address
10.1.1.1	::10.1.1.1

When a BGP speaker receives a BGP update message carrying IPv6 feasible routes, the speaker resolves the announced IPv6 BGP next hop by performing a route lookup to the IPv6 address in the IPv6 route table.

Exchange of IPv6 Routing Information over TCP IPv6

Figure 9 on page 14 illustrates the exchange of IPv6 routing information over a TCP IPv6 connection.

Figure 9: IPv6 Routing over TCP IPv6



Link-Local Next Hops in MP-BGP Packets

When the router has an external directly connected (non-multihop) BGP peer, the router advertises two next hops. It advertises the global next hop and a next hop with a link-local address. The link-local next hop is advertised even when the router has been configured with the next-hop self feature. Advertising the link-local next hop enables the configuration of single-hop EBGP sessions for IPv6 next hops.

For all other types of peers, the router advertises only the global BGP IPv6 next hop.

You can overwrite the global and link-local IPv6 next-hop addresses by configuring and applying a route map that sets the addresses. The **set ipv6 next-hop** clause in the route map can specify a global address, a link-local address, or both for the next hop.

However, a neighbor outbound route map that adds a link-local IPv6 address for peers where the router should not advertise a link-local next hop is considered an invalid configuration.

The router accepts both global and link-local BGP IPv6 next-hop addresses received from its BGP IPv6 peers. As a consequence, when advertising a route to an internal peer, the router can modify the network address of the next-hop field by removing the link-local IPv6 address of the next hop.

For static BGP peers, the JunosE Software does not support the use of link-local addresses when you configure BGP peers. You cannot configure the local interface for a neighbor that has been configured with a link-local address. Although you can configure a neighbor with a link-local address, a BGP session to that peer over TCP IPv6 does not come up.

For dynamic BGP peers, an E Series router can accept incoming TCP sessions with the link-local address as the source address. However, the BGP peering does not come up for such a connection.

Platform Considerations

For information about modules that support BGP on the ERX7xx models, ERX14xx models, and the Juniper Networks ERX310 Broadband Services Router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.

- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support BGP.

For information about modules that support BGP on Juniper Networks E120 and E320 Broadband Services Routers:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support BGP.

References

For more information about the BGP protocol, consult the following resources:

- Address Prefix Based Outbound Route Filter for BGP-4—draft-chen-bgp-prefix-orf-07.txt (March 2004 expiration)
- BGP Extended Communities Attribute—draft-ietf-idr-bgp-ext-communities-07.txt (February 2004 expiration)
- Connecting IPv6 Islands across IPv4 Clouds with BGP—draft-ietf-ngtrans-bgp-tunnel-04.txt (July 2002 expiration)
- Cooperative Route Filtering Capability for BGP-4—draft-ietf-idr-route-filter-09.txt (February 2003 expiration)
- Dynamic Capability for BGP-4—draft-ietf-idr-dynamic-cap-04.txt (February 2004 expiration)
- *JunosE Release Notes*, *Appendix A*, *System Maximums*—Refer to the Release Notes corresponding to your software release for information about maximum values.
- RFC 1657—Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) using SMIv2 (July 1997)
- RFC 1745—BGP4/IDRP for IP—OSPF Interaction (December 1994)
- RFC 1772—Application of the Border Gateway Protocol in the Internet (March 1995)
- RFC 1773—Experience with the BGP-4 protocol (March 1995)
- RFC 1774—BGP-4 Protocol Analysis (March 1995)
- RFC 1863—A BGP/IDRP Route Server alternative to a full mesh routing (October 1995)
- RFC 1930—Guidelines for creation, selection, and registration of an Autonomous System (AS) (March 1996)
- RFC 3065—Autonomous System Confederations for BGP (February 2001)
- RFC 1966—BGP Route Reflection An alternative to full mesh IBGP (June 1996)
- RFC 1997—BGP Communities Attribute (August 1996)
- RFC 1998—An Application of the BGP Community Attribute in Multi-home Routing (August 1996)

- RFC 2270—Using a Dedicated AS for Sites Homed to a Single Provider (January 1998)
- RFC 2385—Protection of BGP Sessions via the TCP MD5 Signature Option (August 1998)
- RFC 2439—BGP Route Flap Damping (November 1998)
- RFC 2519—A Framework for Inter-Domain Route Aggregation (February 1999)
- RFC 2545—Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing (March 1999)
- RFC 2796—BGP Route Reflection—An Alternative to Full Mesh IBGP (April 2000)
- RFC 3392—Capabilities Advertisement with BGP-4 (November 2002)
- RFC 4760—Multiprotocol Extensions for BGP-4 (January 2007)
- RFC 2918—Route Refresh Capability for BGP-4 (September 2000)
- RFC 3032—MPLS Label Stack Encoding (January 2001)
- RFC 3065—Autonomous System Confederations for BGP (February 2001)
- RFC 3392—Capabilities Advertisement with BGP-4 (November 2002)
- RFC 4271—A Border Gateway Protocol (BGP-4) (January 2006)
- RFC 4364—BGP/MPLS IP Virtual Private Networks (VPNs) (February 2006)
- RFC 4721—Graceful Restart Mechanism for BGP (January 2007)
- RFC 4893—BGP Support for Four-octet AS Number Space (May 2007)
- Subcodes for BGP Cease Notification Message—draft-ietf-idr-cease-subcode-05.txt (March 2004 expiration)

NOTE: IETF drafts are valid for only 6 months from the date of issuance. They must be considered as works in progress. Please refer to the IETF Website at http://www.ietf.org for the latest drafts.

Features

Some of the more important BGP features supported by the E Series router are the following:

- Access lists
- Advertisement intervals
- Aggregation
- BGP/MPLS VPNs
- Communities
- Confederations
- EBGP multihop

- IBGP single hop
- Highly scalable BGP-4 architecture
- Multicast
- Next-hop self
- Peer groups
- Route dampening (also referred to as route damping)
- Route mapping and attribute manipulation
- Route origins
- Route redistribution
- Route reflectors
- Soft-reconfiguration inbound
- Synchronization enabling and disabling
- Update source

Before You Configure BGP

Before you attempt to configure BGP, ensure that you have TCP/IP reachability to the BGP peers with which you want your router to communicate. This may include tasks such as setting up interfaces and creating routes.

See the *JunosE Link Layer Configuration Guide* and *JunosE Physical Layer Configuration Guide* for information about how to configure appropriate interfaces. See *JunosE IP Services Configuration Guide,* for information about setting up routing information.

Configuration Tasks

BGP is a very flexible protocol, often providing more than one way to achieve a routing goal. The configuration tasks required therefore vary depending on your needs and decisions. Read all of the following sections to determine the best method for configuring BGP for your needs.

Basic Configuration

Two tasks are common to every BGP configuration: You must enable the BGP routing process, and you must configure BGP neighbors. All other basic configuration tasks are optional.

You can configure certain BGP attributes globally, for peer groups, or for individual peers. The most specific level of configuration takes precedence. For example, if you configure an attribute both globally and for a peer group, the peer group configuration takes precedence for that peer group, but does not affect other peer groups. If you configure an attribute both for a peer group and for a peer, the peer configuration takes precedence for that peer, but does not affect other members of that peer group.

Enabling BGP Routing

All BGP configurations require that you enable the BGP routing process on one or more routers.

router bgp

- Use to enable the BGP routing protocol and to specify the local AS—the AS to which this BGP speaker belongs.
- All subsequent BGP configuration commands are placed within the context of this router and AS; you can have only a single BGP instance per virtual router.
- Specify only one BGP AS per virtual router.
- This command takes effect immediately.
- Example

  ```
  host1(config)#router bgp 100
  ```
- Use the **no** version to remove the BGP process.
- See *router bgp*.

Understanding BGP Command Scope

BGP commands can be sorted into the following categories, each of which has a different scope; that is, each configures parameters within a different area of applicability. Individual command descriptions in this chapter and in "Configuring BGP-MPLS Applications" on page 389, provide more information about command behavior.

- The commands listed in Table 5 on page 19 configure parameters for the BGP process globally, regardless of address family.

Table 5: Commands Affecting BGP Globally

bgp advertise-inactive	bgp graceful-restart path-selection-defer-time
bgp advertise best-external-to-internal	bgp graceful-restart restart-time
bgp always-compare-med	bgp graceful-restart stalepaths-time
bgp bestpath med confed	bgp log-neighbor-changes
bgp bestpath missing-as-worst	bgp maxas-limit
bgp client-to-client reflection	bgp redistribute-internal
bgp cluster-id	bgp router-id
bgp confederation identifier	bgp shutdown
bgp confederation peers	ip bgp-community new-format
bgp default local-preference	overload shutdown
bgp default route-target filter	rib-out disable
bgp enforce-first-as	router bgp
bgp fast-external-fallover	timers bgp
bgp graceful-restart	

- The commands listed in Table 6 on page 19 configure parameters for all address families within the current VRF context.

Table 6: Commands Affecting All Address Families in a VRF

distance bgp	synchronization

- The commands listed in Table 7 on page 19 configure parameters only for the current address family context.

Table 7: Commands Affecting the Current Address Family

address family	disable-dynamic-redistribute
aggregate-address	external-paths
auto-summary	ip route-type

Table 7: Commands Affecting the Current Address Family *(continued)*

bgp dampening	maximum-paths
bgp wait-on-end-of-rib	network
check-vpn-next-hops	redistribute
default-information originate	table-map

- The commands listed in Table 8 on page 20 configure parameters for a peer or peer group, regardless of address family. If the peer or peer group is activated in more than one address family, the values are changed in all those address families. These commands are said to apply on a per-VRF basis. In the following example, EBGP multihop is configured for the session, but when you configure an address family, it is not available—that is, EBGP multihop is not configurable per address family:

```
host1(config-router)#neighbor 10.1.3.4 remote-as 1234
host1(config-router)#neighbor 10.2.3.4 ebgp-multihop 5
host1(config-router)#address-family ipv4 multicast
host1(config-router-af)#neighbor 10.2.3.4 ebgp-multihop ?
% Invalid input detected at '^' marker.
host1(config-router-af)#exit-address-family
```

Table 8: Commands Affecting All Address Families for the Specified Peer or Peer Group

neighbor advertisement-interval	neighbor maximum-update-size
neighbor allow	neighbor passive
neighbor bfd-liveness-detection	neighbor password
neighbor capability	neighbor peer-type
neighbor description	neighbor remote-as
neighbor ebgp-multihop	neighbor rib-out disable
neighbor graceful-restart	neighbor shutdown
neighbor graceful-restart restart-time	neighbor site-of-origin
neighbor graceful-restart stalepaths-time	neighbor timers
neighbor ibgp-singlehop	neighbor update-source
neighbor lenient	neighbor weight

- The commands listed in Table 9 on page 21 configure parameters separately for each address family exchanged over the BGP session. If you configure these parameters for a peer or peer group that is activated in more than one address family, the values are affected only for the current address family. The inbound route map is such a parameter; the following example demonstrates that a BGP session can have a different inbound route map for each address family.

```
host1(config-router)#neighbor 1.1.3.4 remote-as 1234
host1(config-router)#neighbor 1.2.3.4 route-map ucast-map in
host1(config-router)#address-family ipv4 multicast
host1(config-router-af)#neighbor 1.2.3.4 activate
host1(config-router-af)#neighbor 1.2.3.4 route-map mcast-map in
host1(config-router-af)#exit-address-family
```

Table 9: Commands Affecting Only the Current Address Family for the Specified Peer or Peer Group

neighbor activate	neighbor peer-group
neighbor advertise-map	neighbor prefix-list
neighbor allowas-in	neighbor prefix-tree
neighbor as-override	neighbor remote-private-as
neighbor default-originate	neighbor route-map
neighbor distribute-list	neighbor route-reflector-client
neighbor filter-list	neighbor send-community
neighbor local-as	neighbor send-label
neighbor maximum-prefix	neighbor soft-reconfiguration inbound
neighbor next-hop-self	neighbor unsuppress-map
neighbor next-hop-unchanged	

Inheritance of Configuration Values

Peer groups inherit all configuration values that are globally configured. However, attributes configured for a peer group override inherited global configuration values. Individual peers that are members of peer groups inherit all configuration values from the peer group. However, attributes configured on a peer override values inherited from the peer group of which it is a member.

The **neighbor** commands enable you to control features or set parameters for individual peers or for peer groups. These commands can be classified into the four categories shown in Table 10 on page 22, based on whether the command enables a feature or sets parameters, the levels at which it behaves, and how the **no** version of the command compares with the **default** version.

Table 10: Behavior of Neighbor Commands

Category A: Enable or disable a feature that can be configured for a peer or for a peer group	Category B: Enable or disable a feature that can be configured for a peer, for a peer group, or globally	Category C: Set parameters for a peer or for a peer group	Category D: Set parameters for a peer, for a peer group, or globally
• neighbor activate • neighbor advertise-map • neighbor as-override • neighbor bfd-liveness-detection • neighbor capability • neighbor ebgp-multihop • neighbor ibgp-singlehop • neighbor lenient • neighbor next-hop-self • neighbor next-hop-unchanged • neighbor passive • neighbor remove-private-as • neighbor route-reflector-client • neighbor send-community • neighbor soft-reconfiguration inbound	• neighbor default-originate • neighbor graceful-restart • neighbor rib-out disable • neighbor shutdown	• neighbor advertisement-interval • neighbor allow • neighbor allowas-in • neighbor description • neighbor distribute-list • neighbor filter-list • neighbor graceful-restart restart-time • neighbor graceful-restart stalepaths-time • neighbor local-as • neighbor maximum-orf-entries • neighbor maximum-prefix • neighbor maximum-update-size • neighbor password • neighbor peer-group • neighbor peer-type • neighbor prefix-list • neighbor prefix-tree • neighbor remote-as • neighbor route-map • neighbor send-label • neighbor site-of-origin • neighbor unsuppress-map • neighbor update-source • neighbor weight	• neighbor timers

Some of the commands in Table 10 on page 22 inherit global values set by other commands. Table 11 on page 22 describes the relationship between these commands.

Table 11: Inheritance from Other Commands

Category B Command	Inherits Global Values Set By
neighbor default-originate	default-information originate

Table 11: Inheritance from Other Commands *(continued)*

Category B Command	Inherits Global Values Set By
neighbor graceful-restart	bgp graceful-restart
neighbor rib-out disable	rib-out disable
neighbor shutdown	bgp shutdown
neighbor graceful-restart restart-time	bgp graceful-restart restart-time
neighbor graceful-restart stalepaths-time	bgp graceful-restart stalepaths-time

Example 1

For category A and B commands, the behavior of the **no** version of the command is different from the behavior of the **default** version of the command. The **no** version explicitly disables the feature:

- Applied to a peer, the **no** version disables the feature regardless of whether the feature is enabled for any peer group to which it belongs.
- Applied to a peer group, the **no** version disables the feature regardless of whether the feature is enabled for BGP globally or by default.

The **default** version simply unconfigures the feature for the peer or peer group.

- Applied to a peer, the **default** version causes the peer to inherit the state of the feature (enabled or disabled) from any peer group to which it belongs.
- Applied to a peer group, the **default** version causes the peer group to inherit the state of the feature (enabled or disabled) from the BGP global configuration.

The following example illustrates this difference and the inheritance concept with the **neighbor soft-reconfiguration inbound** command.

```
host1(config-router)#neighbor lisbon peer-group
host1(config-router)#neighbor 10.19.7.8 peer-group lisbon
```

Inbound soft-reconfiguration is disabled by default, hence it is currently disabled for both the lisbon peer group and peer 10.19.7.8.

```
host1(config-router)#neighbor lisbon soft-reconfiguration inbound
```

Inbound soft-reconfiguration is now enabled for the lisbon peer group. Because the peer inherits values from the peer group, inbound soft-reconfiguration is now also enabled for peer 10.19.7.8.

```
host1(config-router)#no neighbor 10.19.7.8 soft-reconfiguration inbound
```

The **no** command disables inbound soft-reconfiguration for peer 10.19.7.8, overriding the configuration of the peer group to which the peer 10.19.7.8 belongs. The configuration of an individual peer takes precedence over the configuration of the peer group to which the peer belongs.

host1(config-router)#**default neighbor 10.19.7.8 soft-reconfiguration inbound**

The **default** version returns the peer to inheriting the peer group configuration. Because inbound soft-reconfiguration is still enabled for lisbon, it is now also enabled for peer 10.19.7.8.

host1(config-router)#**default neighbor lisbon soft-reconfiguration inbound**

Finally, this last command returns the peer group configuration to the default value, disabling inbound soft-reconfiguration. The peer 10.19.7.8 inherits this value.

Example 2 For category C and D commands, the behavior of the **no** version of the command is the same as the behavior of the **default** version of the command. The following example illustrates this behavior and the inheritance concept for the **neighbor timers** command.

By default, the BGP global keepalive timer is 30 seconds and the global hold-time timer is 90 seconds.

host1(config-router)#**neighbor eastcoast peer-group**
host1(config-router)#**neighbor 10.10.21.23 peer-group eastcoast**

Peer group eastcoast and peer 10.10.21.23 both have the default timer values. The peer group inherits the global timer values; the peer is a member of eastcoast and inherits the timer values from the peer group.

host1(config-router)#**neighbor eastcoast timers 15 40**

Now peer group eastcoast has a keepalive timer of 15 seconds and a hold-time timer of 40 seconds. Peer 10.10.21.23 inherits these values from the peer group.

host1(config-router)#**no neighbor 10.10.21.23 timers**

Now peer 10.10.21.23 has its timers reset to the global values of 30 and 90 seconds. The configuration of an individual peer takes precedence over the configuration of the peer group to which the peer belongs, which in turn takes precedence over the global configuration.

host1(config-router)#**default neighbor 10.10.21.23 timers**

Nothing changes. For commands in categories C and D, the behavior of the **default** version is the same as the **no** version. Peer 10.10.21.23 still has the global timer values.

host1(config-router)#**neighbor eastcoast timers 20 20**

The eastcoast peer group now has timer values of 20 seconds. Peer 10.10.21.23 still has the global timer values.

Limitations on Inheritance

All BGP peers that are members of the same peer group must send essentially the same updates. Accordingly, all members of a peer group must be the same kind of peer; that is, all must be internal peers, all must be external peers, or all must be confederation peers.

Outbound policies configured for peer groups are still inherited by peer group members, but you cannot override this inherited outbound policy by configuring a different outbound policy on individual members of that peer group with the following commands:

Table 12: Commands That Do Not Override Inherited Outbound Policy

neighbor as-override	neighbor next-hop-unchanged	neighbor route-map out
neighbor default-originate	neighbor prefix-list out	neighbor route-reflector-client
neighbor distribute-list out	neighbor prefix-tree out	neighbor send-community
neighbor filter-list out	neighbor remove-private-as	neighbor unsuppress-map
neighbor next-hop-self		

NOTE: **This restriction does not apply to inbound policy, which you can still override per peer.**

The update messages can vary for members of a peer group as follows:

- The next hop can be different for each update sent to peer group members if the members are all external peers.
- The AS path can be different for each update sent to peer group members if the members are all external peers if you have enabled AS override with the **neighbor as-override** command.

Setting the BGP Identifier

By default, the router ID of the router is used as the BGP identifier. You can use the **bgp router-id** command to configure an IP address as the BGP identifier.

bgp router-id

- Use to configure an IP address as the BGP identifier.
- Example

  host1(config-router)#**bgp router-id 10.25.1.1**

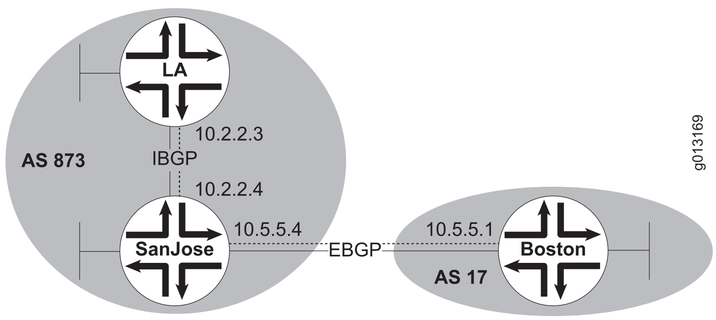
- The new BGP identifier is used in open messages sent after you issue the command. To use the new BGP identifier for sessions already in the established state, you must use the **clear ip bgp** command to perform a hard clear.
- Use the **no** version to restore the router ID as the BGP identifier.
- See *bgp router-id*

Configuring Neighbors

Use the **neighbor remote-as** command to create a BGP peering session with a given BGP peer—identified by its IP address—in a given AS. Note that the **neighbor remote-as** command must be issued on both routers on either side of a BGP session for the BGP session to become established.

Consider the simple network structure shown in Figure 10 on page 26. Routers LA and SanJose are IBGP peers within AS 873. Router SanJose has an EBGP peer, router Boston, in AS 17.

Figure 10: Configuring Neighbors



The following commands configure router Boston with router SanJose as a peer:

host1(config)#**router bgp 17**
host1(config-router)#**neighbor 10.5.5.4 remote-as 873**

The following commands configure router SanJose with router LA and router Boston as peers:

host2(config)#**router bgp 873**
host2(config-router)#**neighbor 10.2.2.3 remote-as 873**
host2(config-router)#**neighbor 10.5.5.1 remote-as 17**

The following commands configure router LA with router SanJose as a peer:

host3(config)#**router bgp 873**
host3(config-router)#**neighbor 10.2.2.4 remote-as 873**

neighbor remote-as

- Use to add an entry to the BGP neighbor table.
- Specifying a neighbor with an AS number that matches the AS number specified in the router bgp command identifies the neighbor as internal to the local AS. Otherwise, the neighbor is treated as an external neighbor.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer.
- This command takes effect immediately.

- Use the **no** version to remove an entry from the table.
- See *neighbor remote-as*

Configuring BGP Peer Groups

You will often want to apply the same policies to most or all of the peers of a particular BGP speaker. Update policies are usually defined by route maps, filter lists, and distribution lists. You can reduce the configuration effort by defining a peer group made up of these peers.

A peer group is defined relative to a particular BGP speaker. Figure 11 on page 28 shows two peer groups, *eastcoast* and *leftcoast*. Each of these peer groups is defined for router Chicago, the hub router. Routers Boston, NY, and Miami have no knowledge of being members of Router Chicago's *eastcoast* peer group. Similarly, routers SanFran, LA, and SanDiego have no knowledge of being members of router Chicago's *leftcoast* peer group.

The following commands configure the *eastcoast* peer group on router Chicago:

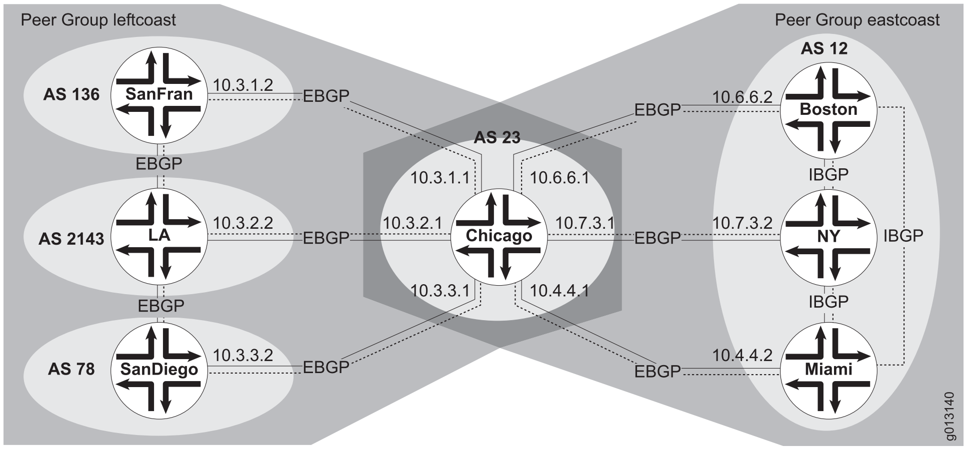
```
host1(config)#router bgp 23
host1(config)#route-map wtset permit 10
host1(config-route-map)#set weight 25
host1(config-route-map)#exit
host1(config-router)#neighbor eastcoast peer-group
host1(config-router)#neighbor eastcoast route-map wtset in
host1(config-router)#neighbor 10.6.6.2 remote-as 12
host1(config-router)#neighbor 10.6.6.2 peer-group eastcoast
host1(config-router)#neighbor 10.7.3.2 remote-as 12
host1(config-router)#neighbor 10.7.3.2 peer-group eastcoast
host1(config-router)#neighbor 10.4.4.2 remote-as 12
host1(config-router)#neighbor 10.4.4.2 peer-group eastcoast
```

The following commands configure the *leftcoast* peer group on router Chicago:

```
host1(config-router)#neighbor leftcoast peer-group
host1(config-router)#neighbor 10.3.3.2 remote-as 78
host1(config-router)#neighbor 10.3.3.2 peer-group leftcoast
host1(config-router)#neighbor 10.3.2.2 remote-as 2143
host1(config-router)#neighbor 10.3.2.2 peer-group leftcoast
host1(config-router)#neighbor 10.3.1.2 remote-as 136
host1(config-router)#neighbor 10.3.1.2 peer-group leftcoast
```

The multiprotocol extensions to BGP enable the exchange of information within different types of *address families*. By default, peers and peer groups exist in the unicast IPv4 address family and exchange unicast IPv4 addresses. For information on configuring and activating BGP peer groups within address families, see "Configuring the Address Family" on page 43.

Figure 11: BGP Peer Groups



neighbor peer-group

- Two versions of this command exist. Use to create a BGP peer group or to configure a BGP neighbor to be a member of a peer group.
- To create a BGP peer group, specify a *peerGroupName* for the new peer group. Use the **no** version to remove a peer group.
- To assign members to a peer group, specify an *ip-address* and a *peerGroupName* of a BGP neighbor that belongs to this group.
- This command takes effect immediately.
- Use the **no** version to remove a neighbor from a peer group.
- See *neighbor peer-group*

NOTE: You cannot mix IPv4 and IPv6 peer members in a peer group. Only one type peer is allowed, IPv4 or IPv6. For example, the following error is generated if an IPv6 peer group member is added to a peer group that already has IPv4 members; that is, where the peer-group type is IPv4:

```
host1(config-router)#neighbor 1::1 peer-group hamburg
% Unable to set 'peer-group' for address family ipv4:unicast for peer
1::1 in core (IPv6 peer cannot be member of a peer-group of type IPv4)
```

For information about the inheritance of configuration values by peer groups and peers, see "Inheritance of Configuration Values" on page 21.

Setting the Peer Type

Each peer group must have a peer type before any BGP sessions for members of that peer group are allowed to come up and before the Adj-RIBs-Out table of that peer group can be filled. You can use the **neighbor peer-type** command to explicitly configure a peer type for a peer group.

Alternatively, you can implicitly configure the peer type of a peer group by either of the following methods:

- Configure a remote AS for the peer group.
- Assign a peer with a configured remote AS as a member of the peer group.

In both of these implicit cases, the remote AS is combined with the local AS, the configured confederation ID, and the configured confederation peers to determine the peer type of the peer group.

neighbor peer-type

- Use to specify a peer type for a peer group.
- This command is supported only for peer groups; it is not available for individual peers.
- Use the **internal** keyword to specify that peers must be in the same AS or, if confederations are employed, in the same sub-AS in the same confederation.
- Use the **external** keyword to specify that peers must be in a different AS.
- Use the **confederation** keyword to specify that peers must be in a different sub-AS in the same confederation. Use this keyword only if confederations are employed.
- This command takes effect immediately. If the command changes the peer type of the peer group, all BGP sessions for members of that peer group are automatically bounced.
- All the members of the peer group inherit the characteristic configured with this command. It cannot be overridden for a specific peer, because the command applies only to peer groups.
- Example

  host1(config-router)#**neighbor promispeers peer-type internal**
- Use the **no** version to remove the configuration from the peer group.
- See *neighbor peer-type*

Assigning a Description

You can associate a description with a BGP neighbor or a peer group. This is a convenient way to store minimal pertinent information about the neighbor.

neighbor description

- Use to associate a textual description of up to 80 characters with a BGP neighbor or peer group.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer.
- This command takes effect immediately.
- Example

host1(config-router)#**neighbor 10.11.0.5 description bostonmetropeer**

- Use the **no** version to remove the description.
- See *neighbor description*

Logging Neighbor State Changes

You can force BGP to log a message whenever a peer enters or leaves the Established state.

bgp log-neighbor-changes

- Use to log a notice message to the bgpNeighborChanges log when a neighbor enters or leaves the Established state for any reason.
- The severity of the log message is notice by default.
- Issue the **log destination console severity notice** command to display the messages on the console.
- This command takes effect immediately.
- Example

```
host1:3(config)#bgp log destination console severity notice
host1:3(config)#router bgp 100
host1:3(config-router)#bgp log-neighbor-changes
NOTICE 04/30/2001 21:06:22 bgpNeighborChanges (3,4.4.4.4): peer 4.4.4.4 in core
  leaves established state
NOTICE 04/30/2001 21:06:22 bgpNeighborChanges (3,5.5.5.5): peer 5.5.5.5 in core
  leaves established state
NOTICE 04/30/2001 21:06:22 bgpNeighborChanges (3,6.6.6.6): peer 6.6.6.6 in core
  leaves established state
NOTICE 04/30/2001 21:06:22 bgpNeighborChanges (3,13.13.13.1): peer 13.13.13.1 in core
  leaves established state
NOTICE 04/30/2001 21:06:31 bgpNeighborChanges (3,4.4.4.4): peer 4.4.4.4 in core
  enters established state
NOTICE 04/30/2001 21:06:31 bgpNeighborChanges (3,5.5.5.5): peer 5.5.5.5 in core enters
  established state
NOTICE 04/30/2001 21:06:31 bgpNeighborChanges (3,6.6.6.6): peer 6.6.6.6 in core
  enters established state
NOTICE 04/30/2001 21:06:31 bgpNeighborChanges (3,13.13.13.1): peer 13.13.13.1 in core
  enters established state
```

- Use the **no** version to stop logging.
- See *bgp log-neighbor-changes*

Specifying a Source Address for a BGP Session

By default, BGP uses the IP address of the outgoing interface toward the peer as the source IP address for the TCP connection over which the BGP session runs. If the outgoing interface goes down, the BGP session is dropped because the IP source address is no longer valid. This is appropriate behavior for EBGP sessions because the EBGP peers typically can reach each other only by virtue of being connected to a common subnet.

For IBGP sessions, however, you typically want BGP sessions to be automatically rerouted around interfaces that are down. You can issue the **neighbor update-source** command to accomplish this. This command instructs BGP to use the IP address of a specified interface as the source address of the underlying TCP connection. Typically, a loopback interface is used because it is inherently stable.

For example, you can specify that BGP use loopback interface 2 as the source for messages that it sends to peer 192.50.30.1:

```
host1(config)#neighbor 192.50.30.1 update-source loopback 2
```

neighbor update-source

- Use to allow a BGP session to use the IP address of a specific operational interface as the source address for TCP connections.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer.
- This command takes effect immediately and automatically bounces the BGP session.
- If you specify an interface in this command and the interface is later removed, then this command is also removed from the router configuration.
- Use the **no** version to restore the interface assignment to the closest interface.
- See *neighbor update-source*

The source address that you specify with the **neighbor update-source** command is also used by BGP as the default value for the next hop address advertised for IPv4 or IPv6 prefixes.

The source addresses and next hop address that result from using the **neighbor update-source** command vary depending on the configuration of the command. Table 13 on page 31 lists the results for different configurations.

Table 13: Source Addresses and Default Next Hop Addresses for Various Configurations

Configured Neighbor Address	Configured Update Source Address	Source Address used for TCPv4 and TCPv6 Connection	Default Next Hop Value for IPv4 Prefixes	Default Next Hop Value for IPv6 Prefixes
IPv4 neighbor address	IPv4 source address	IPv4 source address	IPv4 source address	IPv4 source address mapped to an IPv6 address
IPv4 neighbor address	IPv6 source address	Not allowed	Not allowed	Not allowed
IPv4 neighbor address	Interface name	IPv4 address of the interface. If the interface does not have an IPv4 address, then the session does not come up.	IPv4 address of the interface	IPv6 address of the interface. If the interface does not have an IPv6 address, then the IPv4 address of the interface is mapped to an IPv6 address.

Table 13: Source Addresses and Default Next Hop Addresses for Various Configurations *(continued)*

Configured Neighbor Address	Configured Update Source Address	Source Address used for TCPv4 and TCPv6 Connection	Default Next Hop Value for IPv4 Prefixes	Default Next Hop Value for IPv6 Prefixes
IPv6 neighbor address	IPv6 source address	IPv6 source address	0.0.0.0	IPv6 source address
IPv6 neighbor address	IPv4 source address	Not allowed	Not allowed	Not allowed
IPv6 neighbor address	Interface name	IPv6 address of the interface. If the interface does not have an IPv6 address, then the session does not come up.	IPv4 address of the interface. If the interface does not have an IPv4 address, then 0.0.0.0.	IPv6 address of the interface

You can override a native IPv6 next-hop address with either the **neighbor update-source** command or an outbound route map.

When you specify an interface with the **neighbor update-source** command, the IPv4-mapped IPv6 address of the interface is used instead of the native IPv6 address for the next hop.

```
host1(config)#interface loopback 0
host1(config-if)#ip address 10.1.1.1/32
host1(config-if)#exit
host1(config)#router bgp 100
host1(config-router)#neighbor 2::2 update-source loopback 0
```

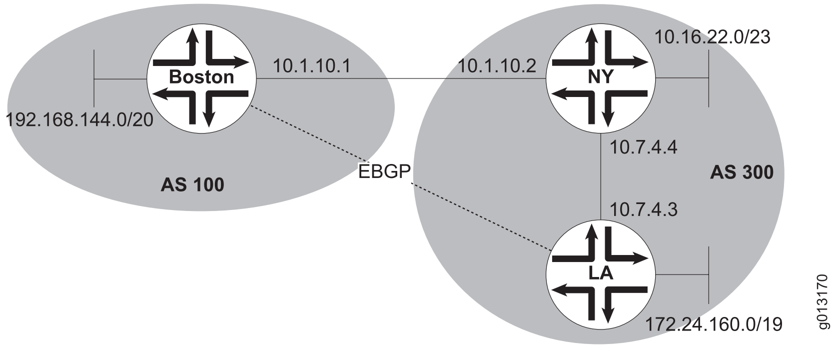
In this example, the IPv4-mapped IPv6 address of the loopback 0 interface is the next-hop address sent when IPv6 prefixes are advertised. However, if loopback 0 has an IPv6 address, then that address is used as the default next hop for advertising IPv6 prefixes.

Specifying Peers That Are Not Directly Connected

Normally, EBGP speakers are directly connected. When you cannot connect EBGP speakers directly, you can use the **neighbor ebgp-multihop** command to specify that the neighbor is more than one hop away. You generally need static routes to configure multihop connections. By default, the one-hop limitation per EBGP peers is enforced by the time-to-live attribute. You can override this default limit by using the *ttl* variable to specify the maximum number of hops to the peer.

In Figure 12 on page 33, router Boston and router LA are connected together through router NY, rather than by a direct connection. Routers Boston and LA are configured as external peers with the **neighbor ebgp-multihop** command because no direct connection exists between them. Because router NY is not a BGP speaker, static routes are configured on routers Boston and LA. The configuration for router NY is not shown, because it does not involve BGP.

Figure 12: Using EBGP-Multihop



The following commands achieve the BGP configuration.

To configure router Boston:

```
host1(config)#ip route 10.7.4.0 255.255.255.0 10.1.10.2
host1(config)#router bgp 100
host1(config-router)#neighbor 10.7.4.3 remote-as 300
host1(config-router)#neighbor 10.7.4.3 ebgp-multihop
```

To configure router LA:

```
host2(config)#ip route 10.1.10.0 255.255.255.0 10.7.4.4
host2(config)#router bgp 300
host2(config-router)#neighbor 10.1.10.1 remote-as 100
host2(config-router)#neighbor 10.1.10.1 ebgp-multihop
```

neighbor ebgp-multihop

- Use to configure BGP to accept route updates from external peers in networks that are not directly connected to the local peer.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer.
- Any external BGP peering that is not resolved by a connected route is treated as a multihop. Configurations with loopback-to-loopback external BGP peering require the **neighbor ebgp-multihop** command to work properly. In these configurations, the **neighbor remote-as** command is issued with the address of a loopback interface.
- This command takes effect immediately and automatically bounces the BGP session.
- Use the **no** version to return BGP to halt acceptance of such routers. Use the **default** version to remove the explicit configuration from the peer or peer group and reestablish inheritance of the feature configuration.
- See *neighbor ebgp-multihop*

Specifying a Single-Hop Connection for IBGP Peers

IBGP peers are multihop by default. However, you can use the **neighbor ibgp-single-hop** command to enable single-hop connections for IBGP peers.

neighbor ibgp-singlehop

- Use to specify an internal BGP peer as a single-hop peer for IBGP sessions.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer.
- If the neighbor session type is anything other than internal BGP, issuing this command generates an error message.
- This command takes effect immediately and automatically bounces the BGP session.
- Example

  host1(config-router)#**neighbor 192.168.32.15 ibgp-singlehop**
- Use the **no** version to restore the default behavior, wherein the internal peer cannot be a single-hop peer. Use the **default** version to remove the explicit configuration from the peer or peer group and reestablish inheritance of the feature configuration.
- See *neighbor ibgp-singlehop*

Controlling the Number of Prefixes

As the routing table increases in size, the processor and memory resources required to process routing information increases. Some peers send so much routing information that a BGP speaker can be overwhelmed by the updates. You can use the **neighbor maximum-prefix** command to limit how many prefixes can be received from a neighbor.

The router resets the BGP connection when the specified maximum is exceeded. You can use the **warning-only** keyword to log a warning rather than reset the connection. You can also configure the router so that a warning is logged when a specified percentage of the maximum is exceeded.

In the following example, the router is configured to reset the BGP connection when it receives more than 1,000 prefixes from its neighbor at 2.2.2.2:

host1(config)#**router bgp 100**
host1(config-router)#**neighbor 2.2.2.2 maximum-prefix 1000**

neighbor maximum-prefix

- Use to control how many prefixes can be received from a neighbor.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer.
- By default, BGP checks the maximum prefix limit only against accepted routes. You can specify the **strict** keyword to force BGP to check the maximum prefix against all received routes. The accepted and received routes will likely differ when you have configured inbound soft reconfiguration and route filters for incoming traffic.
- This command takes effect immediately. To prevent a peer from continually flapping, when it goes to state idle because the maximum number of prefixes has been reached, the peer stays in state idle until you use the **clear ip bgp** command to issue a hard clear.

- Use the **no** version to remove the maximum number of prefixes.
- See *neighbor maximum-prefix*

Removing Private AS Numbers from Updates

You might choose to conserve AS numbers by assigning private AS numbers to some customers. You can assign private AS numbers from the range 64,512 to 65,535. However, when BGP advertises prefixes to other ISPs, it is undesirable to include the private AS numbers in the path. Configure the external neighbors to drop the numbers with the **neighbor remove-private-as** command.

neighbor remove-private-as

- Use to remove private AS numbers only in updates sent to external peers.
- All private AS numbers are removed regardless of their position in the AS-path attribute and regardless of the presence of public AS numbers.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command. You cannot override the characteristic for a specific member of the peer group.
- Example

  host1(config-router)#**neighbor 10.10.128.52 remove-private-as**
- New policy values are applied to all routes that are sent (outbound policy) or received (inbound policy) after you issue the command.

  To apply the new policy to routes that are already present in the BGP routing table, you must use the **clear ip bgp** command to perform a soft clear or hard clear of the current BGP session.

  Behavior is different for outbound policies configured for peer groups for which you have enabled Adj-RIBs-Out. If you change the outbound policy for such a peer group and want to fill the Adj-RIBs-Out table for that peer group with the results of the new policy, you must use the **clear ip bgp peer-group** command to perform a hard clear or outbound soft clear of the peer group. You cannot merely perform a hard clear or outbound soft clear for individual peer group members because that causes BGP to resend only the contents of the Adj-RIBs-Out table.
- Use the **no** version to halt the removal of private AS numbers in updates sent to external peers. Use the **default** version to remove the explicit configuration from the peer or peer group and reestablish inheritance of the feature configuration.
- See *neighbor remove-private-as*

Checking AS-Path Length

You can use the **bgp maxas-limit** command to prevent the forwarding of routes having AS paths longer than a specified limit.

bgp maxas-limit

- Use to require BGP to check the AS path in all received update messages.
- If a received AS path is longer than the specified limit:
  - The route is stored in the BGP routing table and therefore is displayed by the **show ip bgp** commands.
  - The route is not a candidate for being selected as a best path, is not stored in the forwarding information base, and is not propagated to external or internal peers.
- Changes in the limit do not affect routes previously received. Clearing the BGP sessions (**clear ip bgp**) forces a resend of all routes; the new limits are then applied on receipt of the routes.
- Example

  host1(config-router)#**bgp maxas-limit 42**
- Causes BGP to check the AS path of all routes received after you issue the command.

  To apply the new behavior to routes that are already present in the BGP routing table, you must use the **clear ip bgp** command to perform a soft clear or hard clear of the current BGP session.
- Use the **no** version to halt checking of received AS-Path lengths.
- See *bgp maxas-limit*

If you use the **fields as-path** option with the **show ip bgp** command, the display indicates routes whose AS path exceeds the limit. The following display illustrates the result of setting the AS-Path length limit to 5:

```
host1:3# show ip bgp fields intro best peer loc-pref as-path
Local router ID 13.13.13.3, local AS 200
 10 paths, 5 distinct prefixes (520 bytes used)
 6 paths selected for route table installation
 14 path attribute entries (1943 bytes used)
Status codes: > best
 Prefix              Peer             LocPrf AS-path
  10.23.40.1/32      192.168.13.1     200    100 211 32 15 67 44 (too long)
> 10.23.40.1/32      172.123.23.2     100    100 211
> 10.23.40.2/32      192.168.13.1     200    100 211 32 15 67
  10.23.40.2/32      172.123.23.2     100    100 211 32
> 10.23.40.3/32      192.168.13.1            100 211 32 15
  10.23.40.3/32      172.123.23.2            100 211 32 15
  10.23.40.4/32      192.168.13.1     100    100 211 32
> 10.23.40.4/32      172.123.23.2     200    100 211 32 15 67
> 10.23.40.5/32      192.168.13.1     100    100 211
  10.23.40.5/32      172.123.23.2     200    100 211 32 15 67 44 (too long)
```

Enabling MD5 Authentication on a TCP Connection

You can use the **neighbor password** command to enable MD5 authentication on a TCP connection between two BGP peers. Enabling MD5 authentication causes each segment sent on the TCP connection between them to be verified.

You must configure MD5 authentication with the same password on both BGP peers; otherwise, the router does not make the connection between the BGP peers.

The MD5 authentication feature uses the MD5 algorithm. When you specify this command, the router generates and checks the MD5 digest on every segment sent on the TCP connection.

In the following example, the password is set to “opensesame”:

host1(config)#**router bgp 100**
host1(config-router)#**neighbor 2.2.2.2 password opensesame**

The **show ip bgp neighbors** command does not reveal the password, but does indicate whether MD5 authentication is configured for the session. The output of the **show configuration** command varies as follows:

- If you use the **8** keyword to specify that the password is encrypted, then the output of the **show configuration** command displays the text that you entered (the ciphertext password).
- If you do not use the **8** keyword (that is, you use the **0** keyword or no encryption keyword), and if the **service password-encryption** command has not been issued, then the output of the **show configuration** command displays the text that you entered (the plaintext password).
- If you do not use the **8** keyword (that is, you use the **0** keyword or no encryption keyword) but the **service password-encryption** command has been issued, then the output of the **show configuration** command displays an encrypted password that is equivalent to the cleartext password that you entered.

neighbor password

- Use to enable MD5 authentication on a TCP connection between two BGP peers.
- If you configure a password for a neighbor, an existing session is torn down and a new one established.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer.
- If a router has a password configured for a neighbor, but the neighbor router does not, a message indicating this condition appears on the console while the routers attempt to establish a BGP session between them.
- Similarly, if the two routers have different passwords configured, a message appears on the console indicating that this condition exists.
- Use the **8** keyword to indicate that the password is encrypted (entered in ciphertext). Use the **0** keyword to indicate that the password is unencrypted (entered in plaintext).
- This command takes effect immediately and automatically bounces the BGP session.
- Use the **no** version to disable MD5 authentication.
- See *neighbor password*

Setting the Maximum Size of Update Messages

You can use the **neighbor maximum-update-size** command to set the maximum size of update messages transmitted to a BGP peer.

For example, to set the maximum update size to 2,000 octets:

```
host1(config)#router bgp 100
host1(config-router)#neighbor 10.12.2.5 maximum-update-size 2000
```

neighbor maximum-update-size

- Use to set the maximum size for transmitted BGP update messages.
- Set the maximum-update-size to a range: 256–4096.
- The default is 1024 octets.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer.
- BGP always *accepts* updates of up to 4096 octets, regardless of the setting for transmitted updated messages.
- Applies to all update messages sent after you issue the command.
- Use the **no** version to restore the default value.
- See *neighbor maximum-update-size*.

Setting Automatic Fallover

You can use the **bgp fast-external-fallover** command to specify that in the event of the failure of a link to any adjacent external peer, the BGP session is immediately and automatically brought down rather than waiting for the TCP connection to fail or for the hold timer to expire.

bgp fast-external-fallover

- Use to immediately bring down a BGP session if the link to an adjacent external peer fails.
- If you do not issue this command, the BGP session is not brought down in the event of a link failure until the TCP connection fails or the hold timer expires.
- This command takes effect immediately.
- Use the **no** version to stop automatically bringing down the session in the event of link failure.
- See *bgp fast-external-fallover*.

Setting Timers

BGP uses a keepalive timer to control the interval at which keepalive messages are sent. A hold-time timer controls how long BGP waits for a keepalive message before declaring a peer not available.

BGP negotiates the hold time with each neighbor when establishing the BGP connection. The peers use the lower of the two configured hold times. BGP sets the keepalive timer based on this negotiated hold time and the configured keepalive time.

neighbor timers

- Use to set the keepalive and hold-time timers for the specified neighbor or peer group.
- Overrides timer values set with the **timers bgp** command.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer.
- If you set the keepalive timer to 0, BGP does not send any keepalive messages.
- If you do not expect the peer to send any keepalives, set the hold-time timer to 0.
- This command takes effect immediately and automatically bounces the session to force BGP to send a new open message to renegotiate the new timer values.
- Example

  host1(config-router)#**neighbor 192.168.21.5 timers 90 240**
- Use the **no** version to restore the default values on the specified neighbor or peer group—30 seconds for the keepalive timer and 90 seconds for the hold-time timer.
- See *neighbor timers*.

timers bgp

- Use to set the keepalive and hold-time timers for all neighbors.
- If you set the keepalive timer to 0, BGP does not send any keepalive messages.
- If you do not expect the peer to send any keepalives, set the hold-time timer to 0.
- Example

  host1(config-router)#**timers bgp 75 300**
- The new timer values are used by every session that comes up after you issue the command; timers configured specifically for the sessions take precedence over these values.

  To force sessions that are already established to use the new timer values, you must use the **clear ip bgp** command to perform a hard clear.
- Use the **no** version to restore the default values on all neighbors—30 seconds for the keepalive timer and 90 seconds for the hold-time timer.
- See *timers bgp*.

Automatic Summarization of Routes

By default, all routes redistributed into BGP from an IGP are automatically summarized to their natural network masks.

auto-summary

- Use to reenable automatic summarization of routes redistributed into BGP.
- Automatic summarization is enabled by default. However, creating an address family for a VRF automatically disables automatic summarization for that address family.
- This command takes effect immediately.
- Use the **no** version to disable automatic summarization of redistributed routes.
- See *auto-summary*.

Administrative Shutdown

You can administratively shut down particular BGP neighbors or peer groups without removing their configuration from BGP by using the **neighbor shutdown** command.

You can also administratively shut down BGP globally by using the **bgp shutdown** command.

bgp shutdown

- Use to shut down BGP globally.
- This command takes effect immediately.
- Example

  host1(config-router)#**bgp shutdown**
- Use the **no** version to reenable BGP.
- See *bgp shutdown*.

neighbor shutdown

- Use to shut down a neighbor or peer group without removing their configuration.
- This command takes effect immediately.
- Use the **no** version to reenable a neighbor or peer group that was previously shut down. Use the **default** version to remove the explicit configuration from the peer or peer group and reestablish inheritance of the feature configuration.
- See *neighbor shutdown*.

Configuring BGP for Overload Conditions

You can specify how you want BGP to behave when it is running out of memory in an overload condition. You can have BGP either shut itself down or continue running; in the latter case, BGP performance might be altered because of the lack of resources.

overload shutdown

- Use to shut down BGP if it runs out of memory.
- The default behavior is for BGP to transition from the Up state to the Overload state and continue running.
- This command takes effect immediately.
- Example

  host1(config-router)#**overload shutdown**
- Use the **no** version to restore the default behavior.
- See *overload shutdown*.

The following partial outputs show how the BGP state is indicated by the **show ip bgp summary** command:

```
host1#show ip bgp summary
Local router ID 10.1.0.1, local AS 1
  Administrative state is Start
  Operational state is Overload
  Shutdown in overload state is disabled
  Default local preference is 100
...
host1#show ip bgp summary
Local router ID 10.1.0.1, local AS 1
  Administrative state is Start
  Operational state is Down due to transition from Overload state
  Shutdown in overload state is enabled
  Default local preference is 100
...
```

Enabling Route Storage in Adj-RIBs-Out Tables

By default, a BGP speaker does not store a copy of each route it sends to a BGP peer in the Adj-RIBs-Out table for that peer. However, you can force BGP to store a copy of routes in the Adj-RIBs-Out table for a particular peer or peer group by enabling that Adj-RIBs-Out table (“ enabling rib-out”) with the **no neighbor rib-out disable** command. Alternatively, you can use the **no rib-out disable** command to affect all BGP peers. The details of route storage vary between peers and peer groups.

For peers, BGP stores a single bit with each route in the table to indicate whether it has previously advertised the route to the peer, enabling the avoidance of spurious withdrawals. The full set of attributes for each route is not stored in the peer Adj-RIBs-Out table.

After enabling rib-out for a peer, you can issue the **show ip bgp neighbors advertised-routes** command to display the routes that have been advertised to the peer. The attributes displayed for the routes are those from the local routing table, not those that were advertised. In other words, BGP stores the attributes prior to the application of any outbound policy.

For peer groups, BGP stores the full set of attributes associated with the route after the application of any outbound policy; that is, it stores the attributes as they will be advertised. BGP does not store a bit to track whether a route was advertised to the peer group. Storing the full attribute set for each peer group route is memory intensive but acceptable for peer groups, because the number of peer groups is relatively small. An advantage of enabling rib-out for peer groups is that convergence is accelerated because the attributes for each route are already determined for all routes to be advertised to the peer group. BGP has to apply outbound policy only once for each route rather than once for each peer for each route.

After enabling rib-out for a peer group, you can issue the **show ip bgp advertised-routes** command to display the routes that will be advertised to the peer group and the attributes (after the application of any outbound policy) that will be advertised with the routes.

When you have enabled rib-out for individual peers or a peer group, before sending an advertisement or withdrawal the router compares the route it is about to send with the last route sent for the same prefix (and stored in the Adj-RIBs-Out table for the peer or peer group) and sends the update message only if the new information is different from the old.

The comparison prevents the sending of unnecessary withdrawals for both peers and peer groups, because the BGP speaker will not send a withdrawal if the table indicates it has not previously advertised that route to the peer. However, because the route attributes are no longer stored with the routes in peer Adj-RIBs-Out tables, BGP cannot compare them with the attributes in the new update message. Consequently, BGP cannot determine whether the update contains new attributes or the same attributes as those previously advertised, and might send superfluous advertisements to peers. This circumstance does not happen for peer groups, because their Adj-RIBs-Out tables store the full attribute set.

Effects of Changing Outbound Policies

After you change the outbound policy for a peer or peer group, the policy changes do not take effect until you issue either a hard clear or an outbound soft clear. (See "Resetting a BGP Connection" on page 96 for information about performing clears with the **clear ip bgp** command.) The clear action causes BGP to reapply the outbound policy of the peer or peer group to each route in the BGP routing table. BGP then stores the results in the Adj-RIBs-Out table for that peer or peer group. The BGP session with each peer or peer group member takes the routes from the appropriate Adj-RIBs-Out table and sends them in update messages to the peer or peer group member.

NOTE: **You cannot change outbound policy for an individual peer group member. You can change outbound policy only for a peer group as a whole or for peers that are not members of a peer group.**

neighbor rib-out disable

- Use to disable storage of routes (disable rib-out) in the specified neighbor's Adj-RIBs-Out table or in a single Adj-RIBs-Out table for the entire specified peer group.
- Route storage is disabled by default.

- If you enable storage for a peer, the peer's Adj-RIBs-Out table contains all routes actually sent to the peer. By contrast, if you enable storage for a peer group, the peer group's Adj-RIBs-Out table contains those routes that the BGP speaker intends to send to the peer group members; individual members might or might not have already received updates that advertise the routes.
- If you specify a BGP peer group by using the *peerGroupName* argument, a single Adj-RIBs-Out table is enabled for the entire peer group. You can override this configuration for a member of the peer group by issuing the command for that peer.
- Limit the number of Adj-RIBs-Out tables to no more than ten for peer groups to conserve memory resources. No limit applies to peers.
- This command takes effect immediately and automatically bounces the BGP session(s) if the command changes the current configuration.
- Example

  host1(config-router)#**no neighbor 10.15.24.5 rib-out disable**
- Use the **no** version to enable the route storage. Use the **default** version to remove the explicit configuration from the peer or peer group and reestablish inheritance of the feature configuration.
- See *neighbor rib-out disable*.

rib-out disable

- Use to disable storage of routes in the Adj-RIBs-Out tables (disable rib-out) for all BGP peers.
- Route storage is disabled by default.
- This command takes effect immediately and automatically bounces the BGP session if the command changes the current configuration.
- Example

  host1(config)#**rib-out disable**
- Use the **no** version to enable the route storage. Use the **default** version to remove the explicit global configuration from all peers and reestablish inheritance of the feature configuration.
- See *rib-out disable*.

Configuring the Address Family

The BGP multiprotocol extensions specify that BGP can exchange information within different types of *address families*. The JunosE BGP implementation defines the following different types of address families:

- Unicast IPv4—If you do not explicitly specify the address family, the router is configured to exchange unicast IPv4 addresses by default. You can also configure the router to exchange unicast IPv4 routes in a specified VRF.
- Multicast IPv4—If you specify the multicast IPv4 address family, you can use BGP to exchange routing information about how to reach a multicast source instead of a unicast destination. For information about BGP multicasting commands, see "Configuring BGP Routing" on page 3. For a general description of multicasting, see *JunosE Multicast Routing Configuration Guide*.
- VPN IPv4—If you specify the VPN-IPv4 (also known as VPNv4) address family, you can configure the router to provide IPv4 VPN services over an MPLS backbone. These VPNs are often referred to as BGP/MPLS VPNs. For detailed information, see "Configuring BGP-MPLS Applications" on page 389.
- Unicast IPv6—If you specify the IPv6 unicast address family, you can configure the router to exchange unicast IPv6 routes or unicast IPv6 routes in a specified VRF. For a description of IPv6, see *JunosE IP, IPv6, and IGP Configuration Guide*.
- Multicast IPv6—If you specify the multicast IPv6 address family, you can use BGP to exchange routing information about how to reach an IPv6 multicast source instead of an IPv6 unicast destination. For a general description of multicasting, see *JunosE Multicast Routing Configuration Guide*.
- VPN IPv6—If you specify the VPN-IPv6 address family, you can configure the router to provide IPv6 VPN services over an MPLS backbone. These VPNs are often referred to as BGP/MPLS VPNs.
- L2VPN—If you specify the L2VPN address family, you can configure the PE router for VPLS L2VPNs or VPWS L2VPNs to exchange layer 2 network layer reachability information (NLRI) for all VPLS or VPWS instances. Optionally, you can use the **signaling** keyword with the **address-family** command for the L2VPN address family to specify BGP signaling of L2VPN reachability information. Currently, you can omit the **signaling** keyword with no adverse effects. For a description of VPLS, see "Configuring VPLS" on page 613. For a description of VPWS, see "Configuring VPWS" on page 677.
- Route-target—If you specify the route-target address family, you can configure the router to exchange route-target membership information to limit the number of routes redistributed among members. For a description of route-target filtering, see "Configuring BGP-MPLS Applications" on page 389.
- VPLS—If you specify the VPLS address family, you can configure the router to exchange layer 2 NLRI for a specified VPLS instance. For a description of VPLS, see "Configuring VPLS" on page 613.
- VPWS—If you specify the VPWS address family, you can configure the PE router to exchange layer 2 NLRI for a specified VPWS instance. For a description of VPWS, see "Configuring VPWS" on page 677.

Any command issued outside the context of an address family applies to the unicast IPv4 address family by default.

To limit the exchange of routes to those from within the address family and to set other desired BGP parameters:

1. Access Router Configuration mode and create peers and peer groups. These peers and peer groups are in the default IPv4 address family.

   host1(config)#**router bgp 100**
   host1(config-router)#**neighbor 10.10.2.2 remote-as 100**
   host1(config-router)#**neighbor 10.10.3.3 remote-as 100**
   host1(config-router)#**neighbor ibgp peer-group**

2. In Router Configuration mode, create the address family within which the router exchanges addresses; this creation accesses Address Family Configuration mode.

   host1(config-router)#**address-family vpn4 unicast**

3. From within the address family, activate individual neighbors or peer groups to exchange routes from within the current address family. These peers or peer groups must first be created in the IPv4 address family.

   host1(config-router-af)#**neighbor ibgp activate**

4. If you have activated a peer group, from within the address family add peers as members of the peer group. These peers must first be created in the IPv4 address family.

   host1(config-router-af)#**neighbor 10.10.2.2 peer-group ibgp**
   host1(config-router-af)#**neighbor 10.10.3.3 peer-group ibgp**

5. From within the address family, configure BGP parameters for the address family.

6. Exit Address Family Configuration mode.

   host1:vr1(config-router-af)#**exit-address-family**

address-family

- Use to configure the router or VRF to exchange IPv4 or IPv6 addresses by creating the specified address family.
- IPv4 and IPv6 addresses can be exchanged in unicast, multicast, or VPN mode.
- The default setting is to exchange IPv4 addresses in unicast mode from the default router.
- Creating an address family for a VRF automatically disables both synchronization and automatic summarization for that VRF.
- This command takes effect immediately.
- Examples

  host1:vr1(config-router)#**address-family ipv4 multicast**
  host1:vr1(config-router)#**address-family ipv4 unicast**
  host1:vr1(config-router)#**address-family ipv4 unicast vrf vr2**
  host1:vr1(config-router)#**address-family vpn4 unicast**
  host1:vr1(config-router)#**address-family ipv6 unicast**

- Use the **no** version to disable the exchange of a type of prefix.
- See *address-family*.

bgp default ipv4-unicast

- Use to configure all neighbors to exchange addresses in the IPv4 unicast address family.
- All neighbors must be activated with the **neighbor activate** command in the IPv4 address family.
- Example

  host1:vr1(config-router)#**bgp default ipv4-unicast**
- Affects only neighbors created after you issue the command. To affect existing neighbors created before you issued the command, you must use the **neighbor activate** command in the context of the IPv4 unicast address family.
- Use the **no** version to disable the exchange of IPv4 addresses on all neighbors.
- See *bgp default ipv4-unicast*.

exit-address-family

- Use to exit Address Family Configuration mode and access Router Configuration mode.
- Example

  host1:vr1(config-router-af)#**exit-address-family**
- There is no **no** version.
- See *exit-address-family*.

neighbor activate

- Use to specify a peer or peer group with which routes of the current address family are exchanged.
- A peer or peer group can be activated in more than one address family. By default, a peer is activated only for the IPv4 unicast address family.
- The peer or peer group must be created in unicast IPv4 before you can activate it in another address family.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer.
- The address families that are actively exchanged over a BGP session are negotiated when the session is established.
- This command takes effect immediately. If dynamic capability negotiation was not negotiated with the peer, the session is automatically bounced so that the exchanged address families can be renegotiated in the open messages when the session comes back up.

  If dynamic capability negotiation was negotiated with the peer, BGP sends a capability message to the peer to advertise or withdraw the multiprotocol capability for the address family in which this command is issued. If a neighbor is activated, BGP also sends the full contents of the BGP routing table of the newly activated address family.
- Example

host1:vr1(config-router-af)#**neighbor 192.168.1.158 activate**

- Use the **no** version to indicate that routes of the current address family are not to be exchanged with the peer. Use the **default** version to remove the explicit configuration from the peer or peer group and reestablish inheritance of the feature configuration.
- See *neighbor activate*.

If you have configured some or all neighbors to be in the multicast or VPN-IPv4 address families, you can quickly configure all neighbors to be part of the IPv4 unicast address family by issuing the **bgp default ipv4-unicast** command.

Enabling Lenient Behavior

You can use the **neighbor lenient** command to enable the BGP speaker to attempt to recover from malformed packet errors and finite state machine errors generated by a peer. If BGP can recover from the error, it logs a warning message and attempts to maintain the session with the peer. The normal, nonlenient behavior is for the BGP speaker to send a notification message to the peer generating the error and to terminate the session. By default, lenient behavior is disabled.

neighbor lenient

- Use to enable a BGP speaker to be more tolerant of some errors generated by a peer, such as malformed BGP messages or finite state machine errors.
- The speaker attempts to recover from the errors and avoid bringing down the BGP session with the peer.
- Lenient behavior is disabled by default.
- Example

  host1(router-config)#**neighbor 10.12.45.23 lenient**
- Use the **no** version to restore the default condition, disabling lenient behavior.
- See *neighbor lenient*.

Configuring Promiscuous Peers and Dynamic Peering

You can use the **neighbor allow** command to enable a peer group to accept incoming BGP connections from any remote address that matches an access list. Such a peer group is known as a promiscuous peer group; the member peers are sometimes referred to as promiscuous peers.

Promiscuous peers are useful when the remote address of the peer is not known ahead of time. An example is in B-RAS applications, in which interfaces for subscribers are created dynamically and the remote address of the subscriber is assigned dynamically from a local pool or by using RADIUS or some other method.

BGP automatically creates a dynamic peer when a peer group member accepts the incoming BGP connection. Dynamic peers are passive, meaning that when they are not in the established state, they will accept inbound connections but they will not initiate outbound connections. You cannot configure any attributes for the dynamic peers. You cannot remove a dynamic peer with the **no neighbor ip-address** command.

When a dynamic peer goes from the established state to the idle state for any reason, BGP removes the dynamic peer only if it does not go back to the established state within 1 minute. This delay enables you to see the dynamic peer in **show** command output; for example, you might want to see the reason for the last reset or how many times the session flapped.

While a dynamic peer is not in the established state, the **show ip bgp neighbor** command displays the number of seconds remaining until the dynamic peer will be removed.

If you have configured the **neighbor allow** command for multiple peer groups, when an incoming BGP connection matches the access list of more than one of these peer groups, the dynamic peer is created only in the first peer group. (BGP orders peer groups alphabetically by name.)

When the BGP speaker receives an open message from a dynamic peer, the remote AS number must match one of the following criteria; the connection is closed if it does not:

- If the peer group has a configured remote AS number, then the received AS number must be the same as the configured remote AS number.
- If the peer group does not have a configured AS number, then the received AS number must be consistent with the peer type of the peer group. Use the **neighbor peer-type** command to configure the type of the peer-group.

If a peer group has been configured with a peer type but not a remote AS, then the remote AS for dynamic peers is not known until an open message has been received from the peer. Until then, **show** commands display the remote AS as "?" or "unknown."

Static peers that you configure with the **neighbor remote-as** or **neighbor peer-group** commands take precedence over the dynamic peers created as a result of the **neighbor allow** command. If the remote address of an incoming BGP connection matches both a static peer and the access list, the static peer is used and no dynamic peer is created. If you configure a new static peer while a dynamic peer for the same remote address already exists, BGP automatically removes the dynamic peer.

You can optionally specify the maximum number of dynamic peers that BGP can create for the peer group. There is no default maximum. In the absence of a specified maximum, the number of dynamic peers allowed is determined by the available memory and CPU. Dynamic peers consume about the same resources as static peers.

When the maximum number of dynamic peers has been created for a peer group, BGP rejects all subsequent connection attempts for that group. This behavior means that you can specify a maximum to help protect against denial-of-service attacks that attempt to create many dynamic peers to overwhelm your router resources.

BGP generates a log message whenever a dynamic peer is created, rejected because the maximum has been reached, or removed. BGP maintains counters for each peer group for the current number of dynamic peers, the highest number of concurrent dynamic peers ever reached, and the number of times a dynamic peer was rejected because the maximum was reached.

Because dynamic peers always fully inherit their configuration from a peer group, any features that are available for peers but not for peer group members are not supported for the dynamic peers. Currently, only ORFs are not supported for peer group members and therefore are not supported for dynamic peers.

clear bgp ipv6 dynamic-peers

clear ip bgp dynamic-peers

- Use to remove all dynamic peers in the specified scope.
- You can specify the IP address of a BGP neighbor or the name of a BGP peer group as the scope. For IPv4 only, you can also include a VRF in the scope.
- Use the asterisk (*) to remove all BGP dynamic peers.
- This command takes effect immediately.
- Example

  host1#**clear ip bgp 192.168.1.158 vrf boston5 dynamic-peers**
- There is no **no** version.
- See *clear bgp ipv6 dynamic-peers*.
- See *clear ip bgp dynamic-peers*.

neighbor allow

- Use to configure a peer group to accept incoming BGP connections from any remote address that matches the specified access list.
- When the BGP connection is accepted, a dynamic peer is automatically created.
- This command is supported only for peer groups; it is not available for individual peers. These dynamic peers are not displayed by the **show configuration** command and are not stored in NVS. However, the dynamic peers are displayed by **show** commands that display information about BGP peers, such as **show ip bgp neighbors**, **show ip bgp summary**, and so on.
- Incoming connections that match the specified access list are rejected if no peer type has been configured for the peer group.
- This command takes effect immediately. Any existing dynamic BGP sessions that are no longer allowed by the new configuration are removed automatically and immediately. Preexisting dynamic peers that are still allowed by the new configuration are not affected.
- All the members of the peer group inherit the characteristic configured with this command. It cannot be overridden for a specific peer, because the command applies only to peer groups.
- Example

  host1(config-router)#**neighbor promispeers allow remotelist1 max-peers 1023**

- Use the **no** version to remove the configuration from the peer group.
- See *neighbor allow*.

Configuring Passive Peers

You can configure BGP to be passive regarding specific peers, meaning that the BGP speaker will accept inbound BGP connections from the peers but will never initiate an outbound BGP connection to the peers. This passive status conserves CPU and TCP connection resources when the neighbor does not exist.

For example, suppose you preprovision a router before installation with a large number of customer circuits to minimize the configuration changes you might have to make to the router. Any peers that do not exist will consume resources as BGP repeatedly attempts to establish a session with them.

If instead you initially configure the router as passive for those peers, BGP will not attempt to establish sessions to those peers but will wait until these remote peers initiate a session, thus conserving CPU resources.

If you configure both sides of a BGP session as passive, then the session can never come up because neither side can initiate the connection.

neighbor passive

- Use to configure the BGP speaker to only accept inbound BGP connections from the specified peer and never initiate outbound connections to that peer.
- This command takes effect immediately. If the session is not yet established, BGP immediately stops initiating outbound connections to the peer. If the session is already established, it is not bounced regardless of which side initiated the connection.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer.
- Example

  host1(config-router)#**neighbor 10.12.3.5 passive**
- Use the **no** version to restore the default condition, permitting the initiation of outbound connections to the peer.
- See *neighbor passive*.

Advertising Routes

Each BGP speaker advertises to its peers the routes to prefixes that it can reach. These routes include:

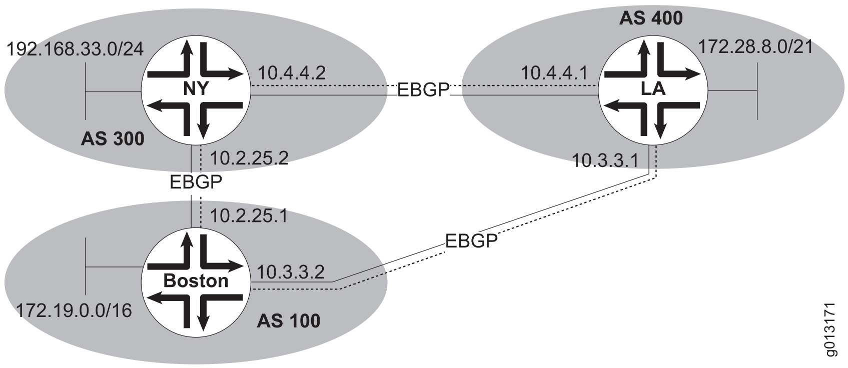
- Routes to prefixes originating within the speaker's AS
- Routes redistributed from another protocol, including static routes

By default, BGP does not advertise any route unless the router's IP routing table also contains the route.

Prefixes Originating in an AS

Use the **network** command to configure a router with the prefixes that originate within its AS. Thereafter the router advertises these configured prefixes with the origin attribute set to IGP. See "Understanding the Origin Attribute" on page 115 for more information about origins. Figure 13 on page 51 shows a network structure of three autonomous systems, each with a router that originates certain prefixes.

Figure 13: Prefixes Originating in an AS



The following commands configure router NY:

```
host1(config)#router bgp 300
host1(config-router)#neighbor 10.2.25.1 remote-as 100
host1(config-router)#neighbor 10.4.4.1 remote-as 400
host1(config-router)#network 192.168.33.0 mask 255.255.255.0
```

The following commands configure router Boston:

```
host2(config)#router bgp 100
host2(config-router)#neighbor 10.2.25.2 remote-as 300
host2(config-router)#neighbor 10.3.3.1 remote-as 400
host2(config-router)#network 172.19.0.0
```

Notice that a mask was not specified for the prefix originating with router Boston. The *natural* mask is assumed for networks without a mask.

The following commands configure router LA:

```
host3(config)#router bgp 400
host3(config-router)#neighbor 10.3.3.2 remote-as 100
host3(config-router)#neighbor 10.4.4.2 remote-as 300
host3(config-router)#network 172.28.8.0 mask 255.255.248.0
```

network

- Use to specify the prefixes in its AS that the BGP speaker advertises.
- BGP advertises the specified prefix only if a non-BGP route to the prefix exists in the IP forwarding table. If the non-BGP route does not exist when you issue the **network**

command, then BGP is notified as soon as the route becomes available in the IP routing table or IP tunnel routing table.

- For IPv4 addressing, specify a *network-number* and an optional *network-mask*. For IPv6 addressing, specify the IPv6 prefix.
- You can specify a route map to filter network routes or modify their path attributes.
- The default weight for network routes is 32768; use the **weight** keyword to modify the weight in the range 0–65535.
- Use the **backdoor** keyword to lower the preference of an EBGP route to the specified prefix by setting the administrative distance to that of an internal BGP route, 200. Use this option to favor an IGP backdoor route over an EBGP route to a specific network. BGP does not advertise the specified network. See "Configuring Backdoor Routes" on page 138 for more information.
- The next hop for the network is the next hop for the route contained in the routing table.
- This command takes effect immediately.
- Use the **no** version to remove the prefix.
- See *network*.

Advertising Best Routes

By default, BGP selects from its routing table one best route to each destination. If BGP learned that best route from an internal peer, then the BGP speaker does not advertise a route to that destination to the speaker's internal peers.

In earlier software releases, the default behavior was for BGP to select two best routes to any destination. The best route learned from external (including confederation) peers was advertised to the speaker's internal peers. The best route learned from all sources was advertised to the speaker's external peers.

You can issue the **bgp advertise-external-to-internal** command to cause BGP to revert to advertising two potentially different routes to its peers. See "Selecting the Best Path" on page 104 for information about the process BGP uses to determine best routes.

bgp advertise-best-external-to-internal

- Use to cause BGP to select two best routes to every destination as follows:
  - For external peers, BGP selects the best route from the complete set of routes known to BGP.
  - For internal peers, BGP selects the best route from the set of routes BGP has received from external and confederation peers.
- Changes apply automatically whenever BGP subsequently runs the best-path decision process for a destination prefix; that is, whenever a best route is picked for a given prefix.
- The behavior enabled by this command is the default behavior for the E Series router running software releases lower than 5.0.0.

- The command is disabled by default.
- Example

```
host2(config-router)#bgp advertise-best-external-to-internal
```

- Use the **no** version to restore the default condition, wherein BGP selects one best route for each destination from the complete set of routes; if the best route was received from an internal peer, no route to the destination is advertised to the internal peers.
- See *bgp advertise-best-external-to-internal*.

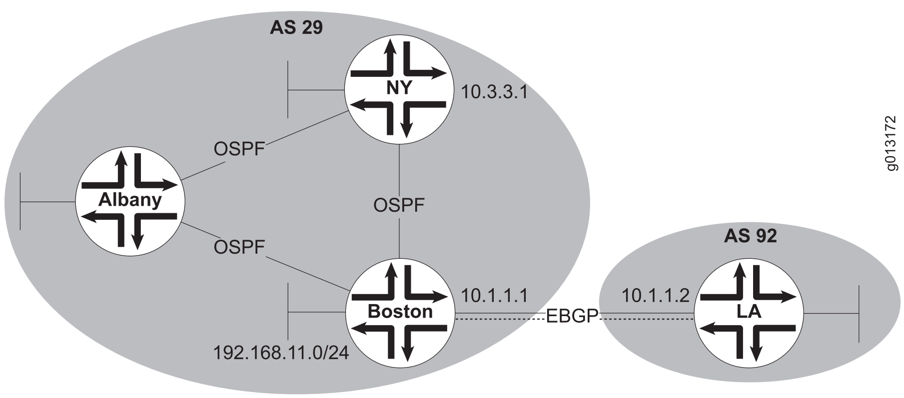
Redistributing Routes into BGP

BGP can learn about routes from sources other than BGP updates from peers. Routes known to other protocols can be *redistributed* into BGP. Similarly, routes manually configured on a router—static routes—can be redistributed into BGP. After the routes are redistributed, BGP advertises the routes. When you redistribute routes, BGP sets the origin attribute for the route to Incomplete. See "Understanding the Origin Attribute" on page 115 for more information about origins.

The following commands configure three static routes on router Boston and configure router Boston to redistribute the static routes and routes from OSPF into BGP for the network structure shown in Figure 14 on page 53:

```
host2(config)#ip route 172.30.0.0 255.255.0.0 192.168.10.12
host2(config)#ip route 172.16.8.0 255.255.248.0 10.211.5.7
host2(config)#ip route 192.168.4.0 255.255.254.0 10.14.147.2
host2(config)#router bgp 29
host2(config-router)#neighbor 10.1.1.2 remote-as 92
host2(config-router)#redistribute static
host2(config-router)#redistribute ospf
```

Figure 14: Redistributing Routes into BGP



clear bgp ipv6 redistribution

clear ip bgp redistribution

- Use to reapply policy to routes that have been redistributed into BGP.
- This command takes effect immediately.

- There is no **no** version.
- See *clear bgp ipv6 redistribution*.
- See *clear ip bgp redistribution*.

disable-dynamic-redistribute

- Use to halt the dynamic redistribution of routes that are initiated by changes to a route map.
- Dynamic redistribution is enabled by default.
- This command takes effect immediately.
- Example

  host1(config-router)#**disable-dynamic-redistribute**
- Use the **no** version to reenable dynamic redistribution.
- See *disable-dynamic-redistribute*.

redistribute

- Use to redistribute static routes and routes from other protocols into BGP.
- Specify the source protocol from which routes are being redistributed with one of the following keywords: **isis**, **ospf**, **static**, or **connected**. Use the **static** keyword to redistribute IP static routes. Use the **connected** keyword to redistribute routes that are established automatically by virtue of having enabled IP on an interface.
- You can specify a route map to filter the redistribution of routes from the source routing protocol into BGP. If you do not specify the **route-map** option, all routes are redistributed.
- Use the **metric** keyword to set the multiexit discriminator (MED) for routes redistributed into BGP. The default MED is the value of the IGP metric for the redistributed route.
- Use the **weight** keyword to set the weight for routes redistributed into BGP in the range 0–65535. The default weight is 32768.
- You can specify the type(s) of OSPF routes to redistribute into BGP: internal routes (**ospf match internal**), external routes of metric type 1 (**ospf match external 1**), or external routes of metric type 2 (**ospf match external 2**).
- This command takes effect immediately.
- Use the **no** version to end the redistribution of routes into BGP.
- See *redistribute*.

Redistributing Routes from BGP

If you have redistributed routes from BGP into an IGP, by default only EBGP routes are redistributed. You can issue the **bgp redistribute-internal** command followed by clearing all BGP sessions to permit the redistribution of IBGP routes in addition to EBGP routes.

NOTE: This default behavior does not apply to VPN routes. Redistribution of IBGP routes (routes received from an internal BGP peer) in a VRF is always enabled. You do not have to issue this command to enable redistribution of internal BGP routes in a VRF.

bgp redistribute-internal

- Use to enable the redistribution of IBGP routes in addition to EBGP routes into IGPs configured for BGP route redistribution.
- Redistribution of IBGP routes is disabled by default, except within a VRF where IBGP routes are always redistributed.
- You must clear all BGP sessions after issuing this command for it to take effect.
- Example

  ```
  host1(config-router)#bgp redistribute-internal
  host1(config-router)#exit
  host1(config)#exit
  host1(config)#clear ip bgp *
  ```

- All IBGP and EBGP routes subsequently placed in the IP routing table are redistributed to IGPs that have route redistribution enabled.

  To authorize redistribution of routes that are already present in the IP routing table, you must use the **clear ip bgp *** command (this command will bounce the BGP sessions) or the **clear ip routes *** command to reinstall BGP routes in the IP routing table.

- Use the **no** version to restore the default of permitting the redistribution only of EBGP routes.
- See *bgp redistribute-internal*.

Configuring a Default Route

Default routes can provide backup routes if primary connections fail or if the route information for a destination is unknown. A router uses the default route in its IP forwarding table to route traffic toward a destination for which no routing entry exists. The accepted BGP convention is to represent a default route by the network prefix 0.0.0.0/0.

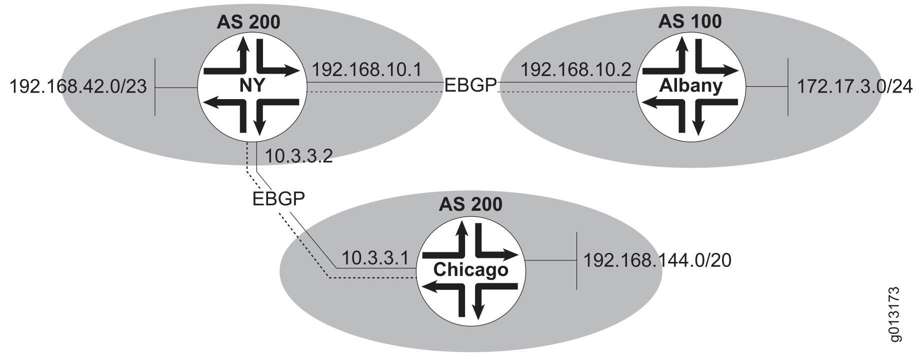
Advertising Default Routes

If you want a router to serve as a default destination for traffic from other routers that do not know where to forward traffic, you can configure the router to advertise a default route. Use the **neighbor default-originate** command to specify the neighbors to which this router will advertise the default route. Said another way, these neighbors will dynamically learn the default route from the router you configure.

If you issue the **neighbor default-originate** command, BGP sends the default route to that neighbor regardless of whether the default route exists in the IP forwarding table.

In Figure 15 on page 56, router NY originates the default route 0.0.0.0/0 to router Albany only. Router Chicago does not receive the default route.

Figure 15: Advertising a Default Route



To configure router NY:

```
host1(config)#router bgp 200
host1(config-router)#network 192.168.42.0 mask 255.255.254.0
host1(config-router)#neighbor 10.3.3.1 remote-as 300
host1(config-router)#neighbor 192.168.10.2 remote-as 100
host1(config-router)#neighbor 192.168.10.2 default-originate
```

You can also specify a route map to modify the attributes of the default route. If the default route does not match the route map, then the default route is not advertised.

Redistributing Default Routes

By default, the **redistribute** command does not permit a default route to be redistributed into BGP. You can use the **default-information originate** command to override this behavior and permit the redistribution of default routes into BGP.

default-information originate

- Use to enable the redistribution of default routes into BGP.
- Use the **route-map** keyword to specify outbound route maps to apply to the default route. The route map can modify the attributes of the default route.
- This command takes effect immediately. However, if the contents of the route map specified with this command change, the new route map may or may not take effect immediately. If the **disable-dynamic-redistribute** command has been configured, you must issue the **clear ip bgp redistribution** command to apply the changed route map.
- Outbound policy configured for the neighbor (using the **neighbor route-map out** command) is applied to default routes that are advertised because of the **default-information originate** command.
- Policy specified by a route map with the **default-information originate** command is applied at the same time as the policy for redistributed routes, before any outbound policy for peers.
- Example

  ```
  host1(config)#router bgp 100
  host1(config-router)#default-information originate
  ```


- Use the **no** version to restore the default, preventing the redistribution of default routes.
- See *default-information originate*.

Setting a Static Default Route

You might not want your routers to rely on dynamically learned default routes. Instead, you might prefer to specify a static default route that your routers use to forward traffic when they do not have a routing entry for a destination. Use the **ip route** command to configure a default route on a router. The static route can point to a network number, an IP address, or a physical interface. You can add a distance value to give preference to a specific static route when multiple entries exist for the same route.

Suppose that in Figure 16 on page 57, router KC has been configured to advertise a default route to router Chicago:

```
host1(config)#router bgp 62
host1(config-router)#network 172.17.24.0 mask 255.255.248.0
host1(config-router)#neighbor 10.8.3.1 remote-as 21
host1(config-router)#neighbor 10.8.3.1 default-originate
```

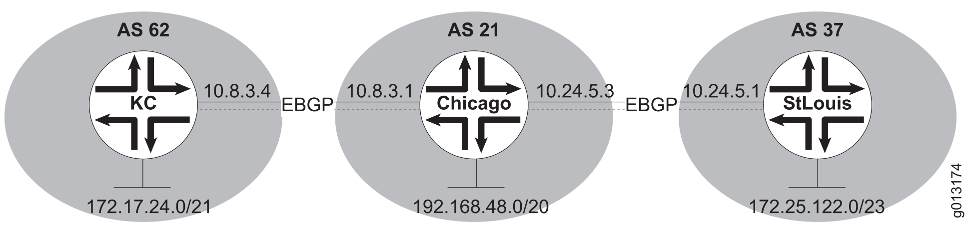
You prefer that router Chicago send traffic with unknown destinations to router StLouis, so you configure a static default route on router Chicago:

```
host2(config)#router bgp 21
host2(config-router)#network 192.168.48.0 mask 255.255.240.0
host2(config-router)#neighbor 10.8.3.4 remote-as 62
host2(config-router)#neighbor 10.24.5.1 remote-as 37
host2(config-router)#exit
host2(config)#ip route 0.0.0.0 0.0.0.0 172.25.122.0
```

Router StLouis is configured to advertise network 172.25.122.0/23 to router Chicago:

```
host3(config)#router bgp 37
host3(config-router)#network 172.25.122.0 mask 255.255.254.0
host3(config-router)#neighbor 10.24.5.3 remote-as 21
```

Figure 16: Setting a Static Default Route



ip route

- Use to establish static routes.
- Use the **no** version to remove static routes.
- See *ip route*.

neighbor default-originate

- Use to cause a BGP speaker (the local router) to send the default route 0.0.0.0/0 to a neighbor for use as a default route.
- Use the **route-map** keyword to specify outbound route maps to apply to the default route. The route map can modify the attributes of the default route.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command. You cannot override the characteristic for a specific member of the peer group.
- Outbound policy configured for the neighbor (using the **neighbor route-map out** command) is not applied to default routes that are advertised because of the **neighbor default-originate** command.
- This command takes effect immediately.
- Use the **no** version to prevent the default route from being advertised by BGP. Use the **default** version to remove the explicit configuration from the peer or peer group and reestablish inheritance of the feature configuration.
- See *neighbor default-originate*.

Setting the Minimum Interval Between Routing Updates

You can use the **neighbor advertisement-interval** command to set the minimum interval between the sending of BGP updates. Lower values for the advertisement interval cause route changes to be reported more quickly, but may cause routers to use more bandwidth and processor time.

In the following example, the minimum time between sending BGP routing updates is set to 5 seconds:

```
host1(config)#router bgp 100
host1(config-router)#neighbor 10.2.2.2 advertisement-interval 5
```

neighbor advertisement-interval

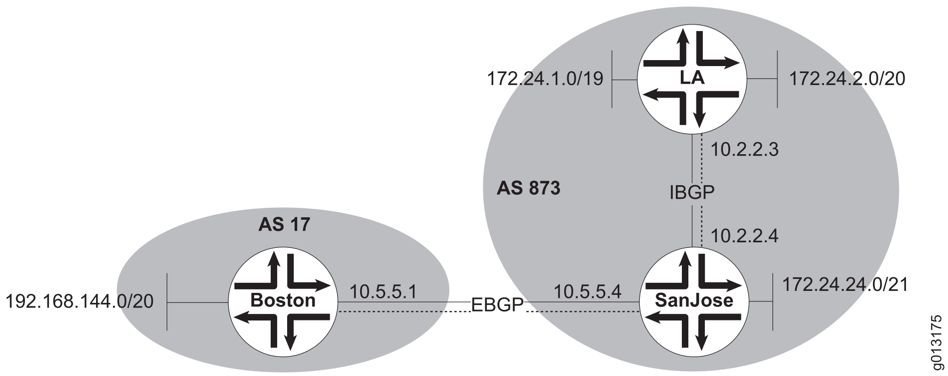
- Use to set the minimum interval between the sending of BGP updates for a given prefix.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer.
- This command takes effect immediately.
- Use the **no** version to restore the default, 30 seconds for external peers and 5 seconds for internal peers.
- See *neighbor advertisement-interval*.

Aggregating Routes

Aggregation applies only to routes that are present in the BGP routing table. BGP advertises an aggregate route only if the routing table contains at least one prefix that is more specific than the aggregate. You aggregate IPv4 routes by specifying the aggregate IP address, and IPv6 routes by specifying the aggregate IPv6 prefix.

Figure 17 on page 59 illustrates an IPv4 network structure where you might use aggregation. The following commands configure router LA and router SanJose so that router SanJose advertises an IPv4 aggregate route, 172.24.0.0/16, for the more specific prefixes 172.24.1.0/24, 172.24.2.0/24, and 172.24.24.0/21.

Figure 17: Configuring Aggregate Addresses



To configure router LA:

```
host1(config)#router bgp 873
host1(config-router)#neighbor 10.2.2.4 remote-as 873
host1(config-router)#network 172.24.1.0 mask 255.255.255.0
host1(config-router)#network 172.24.2.0 mask 255.255.255.0
```

To configure router SanJose:

```
host2(config)#router bgp 873
host2(config-router)#neighbor 10.2.2.3 remote-as 873
host2(config-router)#neighbor 10.5.5.1 remote-as 17
host2(config-router)#network 172.24.24.0 mask 255.255.248.0
host2(config-router)#aggregate-address 172.24.0.0 255.255.224.0
```

As configured above, router SanJose advertises the more specific routes as well as the aggregate route to router Boston. Alternatively, you can use the **summary-only** option to configure router SanJose to suppress the more specific routes and advertise only the aggregate route:

```
host2(config)#router bgp 873
host2(config-router)#neighbor 10.2.2.3 remote-as 873
host2(config-router)#neighbor 10.5.5.1 remote-as 17
host2(config-router)#network 172.24.24.0 mask 255.255.248.0
host2(config-router)#aggregate-address 172.24.0.0 255.255.224.0 summary-only
```

Each of these configurations sets the atomic-aggregate attribute in the aggregate route. This attribute informs recipients that the route *is* an aggregate and must not be deaggregated into more specific routes.

Aggregate routes discard the path information carried in the original routes. To preserve the paths, you must use the **as-set** option. This option creates an AS-Set that consists of all the AS numbers traversed by the summarized paths. The AS-Set is enclosed within curly brackets; for example, {3, 2}. Each AS number appears only once, even if it appears in more than one of the original paths. If you use the **as-set** option, the atomic-aggregate attribute is not set for the aggregated route. The following commands configure router SanJose to aggregate the routes while preserving the path information:

```
host2(config)#router bgp 873
host2(config-router)#neighbor 10.2.2.3 remote-as 873
host2(config-router)#neighbor 10.5.5.1 remote-as 17
host2(config-router)#network 172.24.24.0 mask 255.255.248.0
host2(config-router)#aggregate-address 172.24.0.0 255.255.224.0 summary-only as-set
```

If you do not want to aggregate all more specific routes, you can use a route map to limit aggregation. Consider Figure 17 on page 59 again. Suppose you do not want router SanJose to aggregate prefix 172.24.48.0/20. The following commands show how you can configure a route map on router SanJose to match this prefix, and how to invoke the route map with the **advertise-map** option:

```
host2(config)#router bgp 873
host2(config-router)#neighbor 10.2.2.3 remote-as 873
host2(config-router)#neighbor 10.5.5.1 remote-as 17
host2(config-router)#neighbor 10.2.2.3 route-map lmt_agg in
host2(config-router)#network 172.24.24.0 mask 255.255.248.0
host2(config-router)#aggregate-address 172.24.0.0 255.255.224.0 advertise-map lmt_agg
host2(config-router)#exit
host2(config)#route-map lmt_agg permit 10
host2(config-route-map)#match ip address 1
host2(config-route-map)#exit
host2(config)#access-list 1 permit 172.24.48.0 0.240.255.255
```

You can use the **attribute-map** option to configure attributes for the aggregated route. In Figure 17 on page 59, suppose that router LA has been configured to set the community attribute for route 172.24.160.0/19 to no-export. This attribute is passed along to router SanJose and preserved when the aggregate route is created. As a result, the aggregate route is not advertised outside the AS. The following commands demonstrate how to configure router SanJose to prevent the aggregate from not being advertised:

```
host2(config)#router bgp 873
host2(config-router)#neighbor 10.2.2.3 remote-as 873
host2(config-router)#neighbor 10.5.5.1 remote-as 17
host2(config-router)#network 172.24.24.0 mask 255.255.248.0
host2(config-router)#aggregate-address 172.24.0.0 255.255.224.0 attribute-map
  conf_agg_att
host2(config-router)#exit
host2(config)#route-map conf_agg_att permit 10
host2(config-route-map)#set community no-export
```

aggregate-address

- Use to create an aggregate entry in a BGP routing table that summarizes more specific routes.
- For IPv4 routes, you must specify an aggregate IP address (address) and aggregate IP mask (mask). For IPv6 routes, you must specify an aggregate IPv6 prefix (*ipv6Prefix*).
- The optional **as-set** keyword preserves path information by creating an AS-Set that contains all the AS numbers traversed by the aggregated routes.

NOTE: Do not use the **as-set** keyword when you have many paths to aggregate. If you do, the aggregated route is continually withdrawn and reupdated as AS-path reachability information changes for the summarized routes.

- The **summary-only** keyword advertises only the aggregate route; it suppresses the advertisement of all more specific routes. Contrast with the **suppress-map** keyword.
- The **suppress-map** keyword enables you to specify a route map to filter particular routes covered by the aggregate that will be suppressed. Contrast with the **summary-only** keyword.

NOTE: If you want to suppress advertisements only to certain neighbors, you can—with caution—use the **neighbor distribute-list** command. If a more specific route leaks out, all BGP speakers will prefer that route over the less specific aggregate you are generating (using longest-match routing).

- The **advertise-map** keyword enables you to specify the advertise-map-tag, a string of up to 32 characters that identifies the route map that sets the routes to create AS-Set origin communities.
- The **attribute-map** keyword enables you to specify the attribute-map-tag, a string of up to 32 characters that identifies the route map that sets the attributes of the aggregate route.
- This command takes effect immediately.
- Use the **no** version to remove the aggregate route entry from the routing table.
- See *aggregate-address*.

Advertising Inactive Routes

Under normal circumstances, routes that are not being used to forward traffic—*inactive* routes—are not advertised to peers unless synchronization is enabled. For example, suppose a BGP speaker receives a route to a particular prefix, determines that it is the best route to the prefix, and stores the route in the IP routing table (sometimes known as the forwarding information base, or FIB). This route might not be used for forwarding to that prefix; for example, if you have configured a static route to the same destination prefix. Because static routes have better administrative distances than BGP received routes, IP will use the static route rather than the BGP received route for forwarding traffic to that prefix. The BGP received route is inactive and is not advertised to peers. You can use the **bgp advertise-inactive** command to enable the advertisement of inactive received routes.

bgp advertise-inactive

- Use to enable the BGP speaker to advertise inactive routes—best routes in the IP forwarding table that are not being used to forward traffic. This feature is disabled by default.
- Issuing this command does not affect the BGP rules for best route selection, or how BGP populates the IP forwarding table.
- Example

  host1(config-router)#**bgp advertise-inactive**

- The new value is applied to all routes that are subsequently placed in the IP routing table.

  To apply the new value to routes that are already present in the IP routing table, you must use the **clear ip bgp *** command (this command will bounce the BGP sessions).

- Use the **no** version to prevent the advertising of received BGP routes unless one or both of the following are true:
  - The received route is in the BGP forwarding table and is being used to forward traffic (the route is active).
  - Synchronization is enabled.
- See *bgp advertise-inactive*.

Verifying an AS Path

You can use the **bgp enforce-first-as** command to cause BGP to compare the first AS in the AS path of a received route with the configured remote AS number of that EBGP peer. If the check fails, BGP returns a notification message to the peer.

bgp enforce-first-as

- Use to cause BGP to determine whether the first AS in the AS path of a route received from an EBGP peer matches the remote AS number of that peer.
- If the AS does not match, BGP sends a notification to the peer with the error code “update message error” and error subcode “malformed as-path”.
- This feature is disabled by default.
- Example

  host1(config-router)#**bgp enforce-first-as**

- Causes BGP to check the AS path of all routes received after you issue the command.

  To apply the new behavior to routes that are already present in the BGP routing table, you must use the **clear ip bgp** command to perform a soft clear or hard clear of the current BGP session.

- Use the **no** version to prevent the AS comparison from taking place.
- See *bgp enforce-first-as*.

Advertising IPv4 Routes Between IPv6 BGP Peers

When an IPv6 network connects two separate IPv4 networks, you can use IPv6 to advertise the IPv4 routes over the BGP session, using TCP IPv6 as the transport mechanism. Similarly, you can advertise IPv6 routes between two IPv4 peers over their BGP session.

Configure the peers by using IPv6 addresses within the IPv4 unicast address family. You can set the IPv4 next hop with a static route or by configuring an inbound or outbound route map. This action overrides the IPv4 next hop that is advertised to the peer for IPv4 routes over BGP IPv6 peers.

If you do not use the route map, then the advertised IPv4 next hop is set to the BGP router ID. That value generally makes the next hop unreachable by the other BGP IPv6 peer.

```
host1(config)#router bgp 100
host1(config-router)#neighbor 21:1 remote-as 200
host1(config-router)#route-map my-v4-nexthop
host1(config-router)#set ip next-hop 10.13.5.1
host1(config-router)#address-family ipv4 unicast
host1(config-router-af)#neighbor 21:1 activate
host1(config-router-af)#neighbor 21:1 route-map my-v4-nexthop out
```

Advertising Routes Conditionally

By default, a BGP speaker advertises the best routes in its routing table to its peers. However, in some circumstances, you might prefer that some routes be advertised to a peer or peer group only when another route is in the BGP routing table, or only when that route is not in the routing table. BGP conditional advertisement enables you to control route advertisement without having to rely on only the best routes.

For example, in a multi-homed network, you might want to advertise certain prefixes to one of the providers when a failure occurs in the peering session with a different provider, or when there is only partial reachability to that peer.

In other cases, the advertisement to a peer of certain routes might be useful only in the event that some other routes are present in the BGP routing table.

You can use the **neighbor advertise-map** command with route maps to configure conditional advertisement of BGP routes to a peer or peer group within an address family. BGP conditional advertisement does not create routes. The routes specified by the route map in the **neighbor advertise-map** command must already be present in the BGP routing table.

BGP conditional advertisement is supported in only the following address families:

- Unicast IPv4
- Unicast IPv6
- Multicast IPv4
- Multicast IPv6

- VPNv4 unicast
- VPNv6 unicast

NOTE: For VPNv4 unicast and VPNv6 unicast address families, we recommend that you include a **match extcommunity** clause to match a route with a specific route target. However, conditional advertisement in these address families can sometimes result in unintended behaviors: advertisement of or based on an incorrect VPN route or a non-VPN route.

BGP conditional advertisement is not supported in the following address families:

- L2VPN
- Route-target
- VPLS
- VPWS

Use the **exist-map** keyword when you want a route advertised only when another route is present. The determining route must match the specified route map. If the route map you specify with the **exist-map** keyword references multiple routes, only one of those routes needs to be in the routing table to trigger the conditional advertisement.

Use the **non-exist-map** keyword when you want a route advertised only when another route is absent. The determining route must match the specified route map. If the route map you specify with the **non-exist-map** keyword references multiple routes, all of those routes must be absent to trigger the conditional advertisement.

You can optionally specify a sequence number for the advertise route map that matches the determining route. The sequence number specifies the order in which the advertise route maps are processed. It indicates the position the specified advertise route map has in the list of all advertise route maps that are configured for a particular neighbor within the same address family.

If you do not specify a sequence number, the position of the advertise route map is considered to be the sum of the current largest sequence number plus five. An advertise route map with a lower sequence number has a higher priority and is processed before one with a higher sequence number.

If the route matches more than one advertise route map, only the first matching advertise route map, based on the sequence, controls the advertisement of a BGP route.

You can configure a maximum of 50 advertise maps for a given peer or peer group in an address family. However, the name and sequence number for the advertise route map must be unique for each entry. BGP applies any policy specified by the advertise map to the conditionally advertised routes before outbound policy specified for the neighbor is applied.

The route maps referenced by the **neighbor advertise-map** command must include a **match ip-address** clause. You can also include additional match clauses. All **match** commands supported by existing outbound policies are supported. The additional clauses are useful when you want to match only on a specific route with a specific set of attributes. Only the **permit** keyword is acted on in a match clause. The **deny** keyword is ignored. Only exact matching of a prefix referenced by exist maps or non-exist maps is supported. Consequently a range specified by the **ge** or **le** keyword in the prefix list referenced by these route maps is ignored.

Clauses in a route map that include **set** commands or the **match-set summary prefix-tree** command are ignored. To change the attributes of conditionally advertised routes, you must use outbound routing policy.

If the contents of a referenced route map are changed, the new route map takes effect automatically.

neighbor advertise-map

- Use to specify a peer or peer group within the current address family to which routes specified by a route map are advertised conditionally, depending on whether a second route map is matched by some other routes in the BGP routing table.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command. This characteristic cannot be overridden for individual members of the peer group.
- This command takes effect immediately.
- Example

  host1(config-router-af)#neighbor 192.168.2.2 advertise-map advertiseroutes exist-map matchroute sequence 10

- Use the **no** version to remove the conditions set for advertising to the peer or peer group the routes specified by the route map. Use the **default** version to remove the explicit configuration from the peer or peer group and reestablish inheritance of the feature configuration.
- See *neighbor advertise-map*.

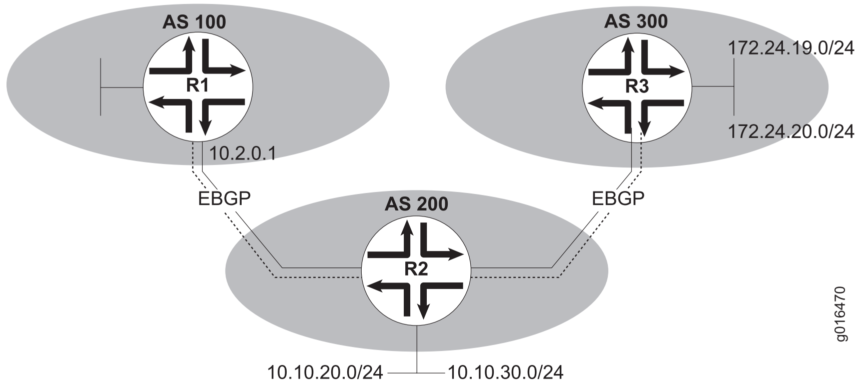
Advertising a Route Only When Another Route is Present

You can use the **exist-map** keyword with the **neighbor advertise-map** command to advertise a route only when the routing table contains some other particular route.

In the network shown in Figure 18 on page 66, router 2 (R2) has established BGP sessions with both router 1 (R1) and router 3 (R3). The plan is for router 2 to send router 1 an advertisement for the route to prefix 10.10.20.0/24 only if router 2 has received a route to prefix 172.24.19.0/24 from router 3.

Alternatively, if the route to prefix 172.24.20.0 has been installed in the BGP routing table on router 2, then router 2 advertises to router 1 the route to prefix 10.10.30.0. In this case, the route does not have to be learned from router 3.

Figure 18: Advertising a Route When Another Route is Present



The following commands represent a partial configuration of router R2:

```
host1(config)#router bgp 200
host1(config-router)#address-family ipv4 unicast
host1(config-router-af)#neighbor 10.2.0.1 remote-as 100
host1(config-router-af)#neighbor 10.2.0.1 advertise-map advertisetoR1 exist-map trigger1
   sequence 10
host1(config-router-af)#neighbor 10.2.0.1 advertise-map alternatetoR1 exist-map trigger2
host1(config-router-af)#exit
host1(config-router)#exit
!
!Configure route map to send one route to R1
!
host1(config)#access-list 77 permit 10.10.20.0 0.0.0.255
host1(config)#route-map advertisetoR1 permit 10
host1(config-route-map)#match ip address 77
host1(config-route-map)#exit
!
!Configure route map to match one trigger route from R3
!
host1(config)#ip as-path access-list 1 permit ^300
host1(config)#access-list 70 permit 172.24.19.0 0.0.0.255
host1(config)#route-map trigger1 permit 10
host1(config-route-map)#match ip address 70
host1(config-route-map)#match as-path 1
host1(config-route-map)#exit
!
!Configure route map to send alternate route to R1
!
host1(config)#access-list test permit 10.10.30.0 0.0.0.255
host1(config)#route-map alternatetoR1 permit 10
host1(config-route-map)#match ip address test
host1(config-route-map)#exit
!
!Configure route map to match alternate route from R3
!
host1(config)#access-list check permit 172.24.20.0 0.0.0.255
host1(config)#route-map trigger2 permit 10
host1(config-route-map)#match ip address check
host1(config-route-map)#exit
```


The **match as-path** clause in the route map referenced by the **exist-map** keyword ensures that router 2 sends router 1 the route to prefix 10.10.20.0 only if a route to 172.24.19.0/24 with an AS path of 300 is present in the BGP routing table. Similarly, you can impose additional restraints by including any other **match** clause that is supported by an existing outbound policy.

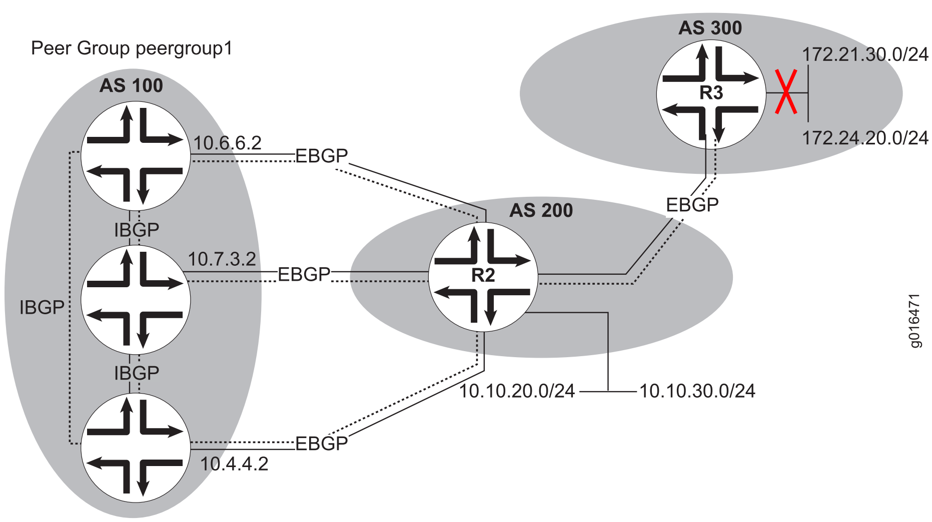
In this configuration, the condition1 route map has a sequence number of ten. Advertise route maps configured for this peer within the same address family and a lower sequence number are processed before the condition1 route map. The condition2 route map has no sequence number configured, thus giving the route map a sequence number of 15 and ensuring that condition2 is processed after the condition1 route map.

Advertising a Route Only When Another Route is Absent

You can use the **non-exist-map** keyword with the **neighbor advertise-map** command to advertise a route only when the BGP routing table does not contain some other particular route.

In the network shown in Figure 19 on page 67, router R2 has established BGP sessions with both router R1 and router R3. The plan is for router R2 to send peergroup1 an advertisement for the route to prefix 10.10.30.0/24 only if the route to prefix 172.24.20.0/24 is not present in the BGP routing table. Alternatively, if router R2 has not received a route to prefix 172.21.30.0 from router R3, then router R2 advertises to peergroup1 the route to prefix 10.10.20.0. In this sample network, router R3 advertises neither of the routes to router R2. Consequently, router R2 advertises both 10.10.20.0/24 and 10.10.30.0/24 to peergroup1.

Figure 19: Advertising a Route When Another Route is Absent



The following commands configure router R2:

```
host1(config)#router bgp 200
host1(config-router)#neighbor peergroup1 peer-group
host1(config-router)#neighbor peergroup1 remote-as 100
host1(config-router)#neighbor 10.6.6.2 peer-group peergroup1
host1(config-router)#neighbor 10.7.3.2 peer-group peergroup1
```



```
host1(config-router)#neighbor 10.4.4.2 peer-group peergroup1
host1(config-router)#neighbor peergroup1 advertise-map advertisetoPG1 non-exist-map
  condition1 sequence 5
host1(config-router)#neighbor peer-group1 advertise-map alternatetoPG1 non-exist-map
  condition2
host1(config-router)#exit
host1(config)#ip as-path access-list 1 permit ^300
!
!Configure route map to send one route to peergroup1
!
host1(config)#access-list 77 permit 10.10.30.0 0.0.0.255
host1(config)#route-map advertisetoPG1 permit 10
host1(config-route-map)#match ip address 77
host1(config-route-map)#exit
!
!Configure route map to match one trigger route
!
host1(config)#access-list 70 permit 172.24.20.0 0.0.0.255
host1(config)#route-map condition1 permit 10
host1(config-route-map)#match ip address 70
host1(config-route-map)#exit
!
!Configure route map to send alternate route to peergroup1
!
host1(config)#access-list allow permit 10.10.20.0 0.0.0.255
host1(config)#route-map alternatetoPG1 permit 10
host1(config-route-map)#match ip address allow
host1(config-route-map)#exit
!
!Configure route map to match an alternate trigger route
!
host1(config)#access-list test permit 172.21.30.0 0.0.0.255
host1(config)#route-map condition2 permit 10
host1(config-route-map)#match ip address test
host1(config-route-map)#match as-path 1
host1(config-route-map)#exit
```

In this configuration, the condition1 route map has a sequence number of five, placing it high in the list of all configured advertise route maps for this peer group within the same address family. The condition2 route map has no sequence number configured, thus placing it at the bottom of the route map list.

In this configuration, the condition1 route map has a sequence number of ten. Route maps configured for this peer group within the same address family and a lower sequence number are processed before the condition1 route map. The condition2 route map has no sequence number configured, thus giving the route map the sequence number of ten and ensuring that condition2 is processed after the condition1 route map.

Advertising a Default Route Only When Another Route Is Present

In some circumstances, you might want to control the advertisement of a default route based on the reachability of an IGP prefix. Because conditional advertisement tracks the BGP routing table rather than the IP routing table, the prefixes that govern the advertisement (the conditional prefixes) must be present in the BGP routing table. In order to use the IGP prefix as a condition, you must import the IGP prefixes into the BGP routing table. You must also configure the origination of the default route.

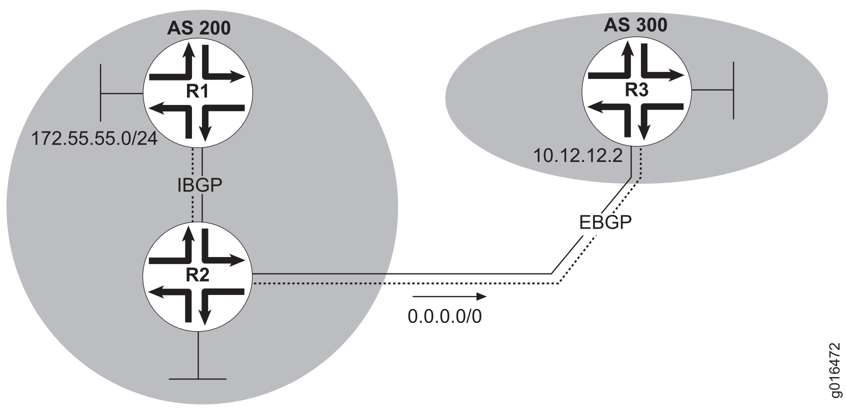
In the network shown in Figure 20 on page 69, router R2 has an EBGP session with router R3 and has an IGP session with router R1. Suppose you want to advertise the default route to router R3 based on the reachability of an IGP prefix, 172.55.55.0/24, on router R2.

On router R2, configure a conditional advertisement entry for the neighbor R3. The advertise map must match the default route and the route map referenced by the **exist-map** keyword must match the imported IGP prefix.

In case router R3 must not learn about the IGP prefix 172.55.55.0/24, you must configure an additional outbound route map to deny this prefix so that it is not advertised to router R3.

With this configuration, the default route is advertised to router R3 only when the IGP prefix 172.55.55.0/24 is reachable on router R2. The default route is withdrawn if this prefix becomes unreachable.

Figure 20: Advertising a Default Route When Another Route is Present



The following commands configure router R2:

```
host1(config)#ip prefix-list default permit 0.0.0.0/0
host1(config)#route-map default permit 10
host1(config-route-map)#match ip address prefix-list default
host1(config-route-map)#exit
host1(config)#ip prefix-list test-default permit 172.55.0.0/16
host1(config)#route-map test permit 10
host1(config-route-map)#match ip address prefix-list test-default
host1(config-route-map)#exit
host1(config)#route-map outbound deny 10
host1(config-route-map)#match ip address prefix-list test-default
host1(config-route-map)#exit
host1(config)#route-map outbound permit 20
host1(config-route-map)#exit
host1(config)#router bgp 200
host1(config-router)#neighbor 10.12.12.2 remote-as 300
host1(config-router)#network 172.55.55.0/24
host1(config-router)#aggregate-address 172.55.0.0/16 summary-only
```



```
host1(config-router)#neighbor 10.12.12.2 advertise-map default exist-map test
host1(config-router)#neighbor 10.12.12.2 default-originate
host1(config-router)#neighbor 10.12.12.2 route-map outbound out
host1(config-router)#exit
```

Configuring BGP Routing Policy

Routing policy determines how the router handles the routes it receives from and sends to BGP peers or other routing protocols. In many cases, routing policy consists of filtering routes, accepting certain routes, accepting and modifying other routes, and rejecting some routes. You can think of routing policy as a way to control the flow of routes into and out of the router.

You can use one or more of the following mechanisms to configure routing policy:

- Access lists
- Community lists
- Prefix lists
- Prefix trees
- Route maps

The remainder of this section provides detailed information about using these features with BGP. Before proceeding, please see *JunosE IP Services Configuration Guide*, for a thorough background on how these features work in general.

Types of BGP Route Maps

A route map consists of *match* clauses and *set* clauses. Match clauses, which consist of a **match** command, specify the attribute values that determine whether a route matches the route map. Set clauses, which consist of a **set** command, modify the specified attributes of routes that pass all match clauses in the route map.

BGP route maps can be applied to inbound routes, outbound routes, and redistributed routes. BGP route maps are of two types, those that support both **match** and **set** clauses, and those that support only **match** clauses.

The match-and-set route maps consist of the route maps configured with any of the commands listed in Table 14 on page 70.

Table 14: Commands That Create Match-and-Set Route Maps

aggregate-address attribute-map	global import map
bgp dampening route-map	neighbor route-map in
export map	neighbor route-map out
import map	redistribute route-map
global export map	table-map

BGP supports the clauses listed in Table 15 on page 71 for match-and-set route maps.

Table 15: Clauses Supported in BGP Match-and-Set Route Maps

match as-path	set as-path prepend
match community	set comm-list delete
match distance	set community
match extcommunity	set dampening
match ip address	set extcommunity
match ip next-hop	set ip next-hop
match level	set local-preference
match metric	set metric
match metric-type	set metric-type
match route-type	set origin
match tag	set tag
	set weight

The match-only route maps consist of the route maps configured with any of the commands listed in Table 16 on page 71. You can use any of the match clauses listed in Table 15 on page 71 in these route maps. Set clauses have no effect in these route maps.

Table 16: Commands That Create Match-Only Route Maps

aggregate advertise-map	aggregate support-map

BGP does not support the clauses listed in Table 17 on page 71. However, see "Applying Table Maps" on page 80 for exceptions for route maps applied with the **table-map** command.

Table 17: Clauses Not Supported in BGP Route Maps

set automatic-tag	set level
set distance	set route-type

match as-path

- Use to match an AS-path access list.
- The implemented weight is based on the first matched AS path.

- Example

  host1(config)#route-map nyc1 permit 10
  host1(config-route-map)#match as-path pathlist5

- Use the **no** version to delete the match clause from a route map or a specified value from the match clause.
- See *match as-path*.

match community

- Use to match a community list.
- Supported for inbound and outbound route maps.
- Example

  host1(config)#route-map nyc1 permit 10
  host1(config-route-map)#match community comm5

- Use the **no** version to delete the match clause from a route map or a specified value from the match clause.
- See *match community*

match distance

- Use to match any routes that have the specified administrative distance.
- Example

  host1(config)#route-map nyc1 permit 10
  host1(config-route-map)#match distance 25

- Use the **no** version to delete the match clause from a route map or a specified value from the match clause.
- See *match distance*.

match extcommunity

- Use to match an extended community list in a route map.
- You can specify one or more extended community list names in a match clause. If you specify more than one extended community list, the lists are logically joined by the OR operator.
- Example

  host1(config)#route-map nyc1 permit 10
  host1(config-route-map)#match extcommunity topeka10

- Use the **no** version to remove the match clause from a route map or a specified value from the match clause.
- See *match extcommunity*.

match ip address

- Use to match any route that has a destination network number that is permitted by an access list, a prefix list, or a prefix tree, or performs policy routing on packets.
- Example

  host1(config)#**route-map nyc1 permit 10**
  host1(config-route-map)#**match ip address prefix-tree boston**

- Use the **no** version to delete the match clause from a route map or a specified value from the match clause.
- See *match ip address*.

match ip next-hop

- Use to match any routes that have a next-hop router address passed by the specified access list, prefix list, or prefix tree.
- Example

  host1(config)#**route-map nyc1 permit 10**
  host1(config-route-map)#**match ip next-hop 5 192.54.24.1**

- Use the **no** version to delete the match clause from a route map or a specified value from the match clause.
- See *match ip next-hop*.

match level

- Use to match routes for the specified type.
- Example

  host1(config)#**route-map nyc1 permit 10**
  host1(config-route-map)#**match level level-1**

- Use the **no** version to delete the match clause from a route map or a specified value from the match clause.
- See *match level*.

match metric

- Use to match a route for the specified metric value.
- Example

  host1(config)#**route-map nyc1 permit 10**
  host1(config-route-map)#**match metric 10**

- Use the **no** version to delete the match clause from a route map or a specified value from the match clause.
- See *match metric*.

match metric-type

- Use to match a route for the specified metric type.
- Example

```
host1(config)#route-map nyc1 permit 10
host1(config-route-map)#match metric-type external
```

- Use the **no** version to delete the match clause from a route map.
- See *match metric-type*.

match route-type

- Use to match a route for the specified route type.
- Example

```
host1(config)#route-map nyc1 permit 10
host1(config-route-map)#match route-type level-1
```

- Use the **no** version to delete the match clause from a route map or a specified value from the match clause.
- See *match route-type*.

match tag

- Use to match the tag value of the destination routing protocol.
- Example

```
host1(config)#route-map 1
host1(config-route-map)#match tag 25
```

- Use the **no** version to delete the match clause from a route map or a specified value from the match clause.
- See *match tag*.

neighbor route-map

- Use to apply a route map to incoming or outgoing routes.
- If you specify an outbound route map, BGP advertises only routes that match at least one section of the route map. For routes that do not match, no further processing takes place with respect to this peer, and those routes are not advertised to this peer. The nonmatching route is still in the BGP RIB and can be sent to other peers depending on the outbound policy applied to those peers.
- If you specify an inbound route map, BGP processes only the received routes that match at least one section of the route map. The nonmatching routes are rejected from entering the local BGP RIB and no further processing takes place.
- A clause with multiple values matches a route having any of the values; that is, the multiple values are logically joined by the OR operator.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer. However, you cannot configure a member of a peer group to override the inherited peer group characteristic for outbound policy.
- New policy values are applied to all routes that are sent (outbound policy) or received (inbound policy) after you issue the command.

To apply the new policy to routes that are already present in the BGP routing table, you must use the **clear ip bgp** command to perform a soft clear or hard clear of the current BGP session.

Behavior is different for outbound policies configured for peer groups for which you have enabled Adj-RIBs-Out. If you change the outbound policy for such a peer group and want to fill the Adj-RIBs-Out table for that peer group with the results of the new policy, you must use the **clear ip bgp peer-group** command to perform a hard clear or outbound soft clear of the peer group. You cannot merely perform a hard clear or outbound soft clear for individual peer group members because that causes BGP to resend only the contents of the Adj-RIBs-Out table.

- Example

  host1(config)#**neighbor 192.168.5.34 route-map nyc1 in**

- Use the **no** version to remove the route map.
- See *neighbor route-map*.

route-map

- Use to define the conditions for redistributing routes from one routing protocol into another, and for filtering or modifying updates sent to or received from peers.
- Each **route-map** command has a list of match and set commands associated with it.
- The match commands specify the match criteria—the conditions under which redistribution is allowed for the current route map.
- The set commands specify the set actions—the redistribution actions to perform if the criteria enforced by the match commands are set.
- Use route maps when you wish to have detailed control over how routes are redistributed between routing processes.
- The destination routing protocol is the one you specify with the router command.
- The source routing protocol is the one you specify with the redistribute command.
- A clause with multiple values matches a route having any of the values; that is, the multiple values are logically joined by the OR operator.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer.
- Example

  host1(config)#**route-map nyc1 permit 10**

- Use the **no** version to delete the route map.
- See *route-map*.

set as-path prepend

- Use to modify an AS path for BGP routes by prepending one or more AS numbers or a list of AS numbers to the path list.
- The only global BGP metric available to influence the best-path selection is the AS-path length. By varying the length of the AS path, a BGP speaker can influence the best-path selection by a peer farther away.
- Example

  host1(config)#**route-map nyc1 permit 10**
  host1(config-route-map)#**set as-path prepend list list10**

- Use the **no** version to delete the set clause from a route map.
- See *set as-path prepend*.

set comm-list delete

- Use to remove communities specified by the community list from the community attribute of routes matching the route map.
- You can use this command to delete communities only if the community list was created with a single community per list entry as shown in the following sample configuration for router Test:

  host1(config)#**ip community-list 1 permit 231:10**
  host1(config)#**ip community-list 1 permit 231:20**
  host1(config)#**router bgp 45**
  host1(config-router)#**neighbor 10.6.2.5 remote-as 5**
  host1(config-router)#**neighbor 10.6.2.5 route-map indelete in**
  host1(config-router)#**route-map indelete permit 10**
  host1(config-route-map)#**set comm-list 1 delete**

  Router Test receives the same route from 10.6.2.5 and applies the indelete route map. BGP compares each list entry with the community attribute. A match is found for the list entry 231:10, and this community is deleted from the community attribute. Similarly, a match is found for the list entry of 231:20, and this community is deleted from the community attribute.

- Use the **no** version to delete the set clause from a route map.
- See *set comm-list delete*.

set community

- Use to set the community attribute in BGP updates.
- You can specify a community list number in the range 1–4294967295, or in the new community format of AA:NN, or one of the following well-known communities:
  - **local-as**—Prevents advertisement outside the local AS
  - **no-advertise**—Prevents advertisement to any peer
  - **no-export**—Prevents advertisement beyond the BGP confederation boundary
- Alternatively, you can use the **list** keyword to specify the name of a community list that you previously created with the **ip community-list** command.

- Example

  host1(config)#route-map nyc1 permit 10
  host1(config-route-map)#set community no-advertise

- Use the **none** keyword to remove the community attribute from a route.
- Use the **no** version to delete the set clause from a route map.
- See *set community*.

set dampening

- Use to enable BGP route flap dampening only on routes that pass the match clauses of, and are redistributed by, a particular route map.
- BGP creates a dampening parameter block for each unique set of dampening parameters—such as suppress threshold and reuse threshold—used by BGP. For example, if you have a route map that sets the dampening parameters to one set of values for some routes and to another set of values for the remaining routes, BGP uses and stores two dampening parameter blocks, one for each set.
- Example

  host1(config)#route-map nyc1 permit 10
  host1(config-route-map)#set dampening 5 1000 1500 45 15

- Use the **no** version to delete the set clause from a route map.
- See *set dampening*.

set extcommunity

- Use to set the extended community attributes in a route map for BGP updates.
- You can specify a site-of-origin (**soo**) extended community and a route target (**rt**) extended community at the same time in a set clause without overwriting the other.
- Example

  host1(config)#route-map nyc1 permit 10
  host1(config-route-map)#set extcommunity rt 10.10.10.2:325

- Use the **no** version to delete the set clause from a route map.
- See *set extcommunity*.

set ip next-hop

- Use to set the next hop attribute of a route that matches a route map.
- This command is not supported for route maps used by the **table-map** command.
- You can specify an IP address or an interface as the next hop.
- Use the **peer-address** keyword to have the following effect:
  - On outbound route maps, disables the next hop calculation by setting the next hop to the IP address of the BGP speaker
  - On inbound route maps, overrides any third-party next-hop configuration by setting the next hop to the IP address of the peer

- Example

  host1(config)#route-map nyc1 permit 10
  host1(config-route-map)#set ip next-hop 192.56.32.1

- Use the **no** version to delete the set clause from a route map.
- See *set ip next-hop*.

set local-preference

- Use to specify a preference value for the AS path.
- Example

  host1(config)#route-map nyc1 permit 10
  host1(config-route-map)#set local-preference 200

- Use the **no** version to delete the set clause from a route map.
- See *set local-preference*.

set metric

- Use to set the metric value—for BGP, the MED—for a route.
- To establish an absolute metric, do not enter a plus or minus sign before the metric value.
- Example

  host1(config)#route-map nyc1 permit 10
  host1(config-route-map)#set metric 10

- To establish a relative metric, specify a plus or minus sign immediately preceding the metric value. The value is added to or subtracted from the metric of any routes matching the route map. The relative metric value can be in the range 0–4294967295.
- Example

  host1(config)#route-map nyc1 permit 10
  host1(config-route-map)#set metric -25

- You cannot use both an absolute metric and a relative metric within the same route map sequence. Setting either metric overrides any previously configured value.
- Use the **no** version to delete the set clause from a route map.
- See *set metric*.

set metric-type

- Use to set the metric type for a route.
- For BGP, affects all types of route maps. If the route map contains both a **set metric-type** and a **set metric** clause, the **set metric** clause takes precedence. Specifying the **internal** metric type in a BGP outbound route map, BGP sets the MED of the advertised routes to the IGP cost of the next hop of the advertised route. If the cost of the next hop changes, BGP is not forced to readvertise the route.
- For BGP, you can specify the following:

  - **external**—Reverts to the normal BGP rules for propagating the MED; this is the BGP default
  - **internal**—Sets the MED of a received route that is being propagated to an external peer equal to the IGP cost of the indirect next hop

- Example

  host1(config)#**route-map nyc1 permit 10**
  host1(config-route-map)#**set metric-type internal**

- Use the **no** version to delete the set clause from a route map.
- See *set metric-type*.

set origin

- Use to set the BGP origin of the advertised route.
- Example

  host1(config)#**route-map nyc1 permit 10**
  host1(config-route-map)#**set origin egp**

- Use the **no** version to delete the set clause from a route map.
- See *set origin*.

set tag

- Use to set the tag value of the destination routing protocol.
- Example

  host1(config)#**route-map nyc1 permit 10**
  host1(config-route-map)#**set tag 23**

- Use the **no** version to delete the set clause from a route map.
- See *set tag*.

set weight

- Use to specify the BGP weight for the routing table.
- The weights assigned with the set weight command in a route map override the weights assigned using the neighbor weight and neighbor filter-list weight commands.
- Example

  host1(config)#**route-map nyc1 permit 10**
  host1(config-route-map)#**set weight 200**

- Use the **no** version to delete the set clause from a route map.
- See *set weight*.

Applying Table Maps

You can use the **table-map** command on a per-address-family basis to apply a route map to modify IP attributes of BGP routes that are about to be added to the IP routing table. In these route maps, you can use only the set clauses in Table 18 on page 80.

Table 18: Set Clauses Supported in Route Maps Applied with the Table-Map Command

set distance	set metric-type
set level	set route-type
set metric	set tag

set distance

- Use to set the administrative distance attribute on routes being installed into the routing table that match the route map.
- Distance is used to establish preference between routes to the same prefix to identify the best route to that prefix. Setting distance in any other circumstance has no effect.
- Example

  host1(config-route-map)#**set distance 5**
- Use the **no** version to delete the set clause from a route map.
- See *set distance*.

set level

- Use to specify where to import routes when all of a route map's match criteria are met.
- Example

  host1(config-route-map)#**set level level-2**
- Use the **no** version to delete the set clause from a route map.
- See *set level*.

set route-type

- Use to set the routes of the specified type.
- Example

  host1(config-route-map)#**set route-type internal**
- Use the **no** version to delete the set clause from a route map.
- See *set route-type*.

table-map

- Use to apply a policy to BGP routes about to be added to the IP routing table.
- The route map can include any of the clauses listed in Table 18 on page 80.
- The new route map is applied to all routes that are subsequently placed in the IP routing table. To apply the new table map to routes that are already present in the IP routing table, you must refresh the IP routing table with the **clear ip routes *** command or the **clear ip bgp *** command (this command will bounce the BGP sessions).
- Example

  host1(config-router)#**table-map distmet1**
  host1(config-router)#**exit**
  host1(config)#**exit**
  host1#**clear ip routes ***

- Use the **no** version to halt application of the route map.
- See *table-map*.

For example, suppose you want to change the distance and metric attributes to particular values for routes advertised by a members of a particular community. The **show ip route bgp** command indicates that the routes currently in the table have a variety of values for these attributes:

```
host1#show ip route bgp
Protocol/Route type codes:
  I1- ISIS level 1, I2- ISIS level2,
  I- route type intra, IA- route type inter, E- route type external,
  i- metric type internal, e- metric type external,
  O- OSPF, E1- external type 1, E2- external type2,
  N1- NSSA external type1, N2- NSSA external type2

  Prefix/Length     Type      Next Hop        Dist/Met        Intf
------------------ ------- --------------- --------------- ------------
10.100.3.3/32        Bgp     10.12.12.1       20/0           ATM5/1.12
10.63.42.23/32      Bgp     10.45.2.31       12/5          ATM5/1.14
```

The following commands demonstrate how you can apply the policy to change these values:

host1(config)#**route-map distmet1 permit 5**
host1(config-route-map)#**match community boston42**
host1(config-route-map)#**set distance 33**
host1(config-route-map)#**set metric 44**
host1(config-route-map)#**exit**
host1(config)#**router bgp 100**
host1(config-router)#**table-map distmet1**
host1(config-router)#**exit**
host1(config)#**exit**
host1#**clear ip routes ***

The **show ip route bgp** command reveals the new values:

```
host1#show ip route bgp
Protocol/Route type codes:
  I1- ISIS level 1, I2- ISIS level2,
  I- route type intra, IA- route type inter, E- route type external,
  i- metric type internal, e- metric type external,
```



```
  O- OSPF, E1- external type 1, E2- external type2,
  N1- NSSA external type1, N2- NSSA external type2

  Prefix/Length     Type      Next Hop        Dist/Met        Intf
------------------ ------- --------------- -------------- ------------
10.100.3.3/32        Bgp     10.12.12.1       33/44          ATM5/1.12
10.63.42.23/32      Bgp     10.45.2.31      33/44          ATM5/1.14
```

Access Lists

An access list is a sequential collection of permit and deny conditions that you can use to filter inbound or outbound routes. You can use different kinds of access lists to filter routes based on either the prefix or the AS path.

Filtering Prefixes

To filter routes based on the prefix, you can do any of the following:

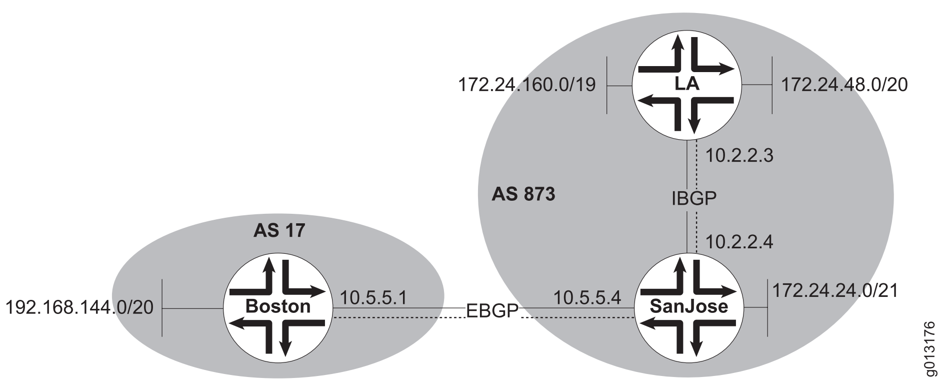
- Define an access list with the **access list** command and apply the list to routes received from or passed to a neighbor with the **neighbor distribute-list** command.
- Define a prefix list with the **ip prefix-list** command and apply the list to routes received from or passed to a neighbor with the **neighbor prefix-list** command.
- Define a prefix tree with the **ip prefix-tree** command and apply the list to routes received from or passed to a neighbor with the **neighbor prefix-tree** command.

The router compares each route's prefix against the conditions in the list or tree one by one. If the first match is for a permit condition, the route is accepted or passed. If the first match is for a deny condition, the route is rejected or blocked. The order of conditions is critical because testing stops with the first match. If no conditions match, the router rejects or blocks the address; that is, the last action of any list is an implicit deny condition for all routes. The implicit rule is displayed by **show access-list** and **show configuration** commands.

You cannot selectively place conditions in or remove conditions from an access list, prefix, list, or prefix tree. You can insert a new condition only at the end of a list or tree.

Consider the network structure in Figure 21 on page 82.

Figure 21: Filtering with Access Lists

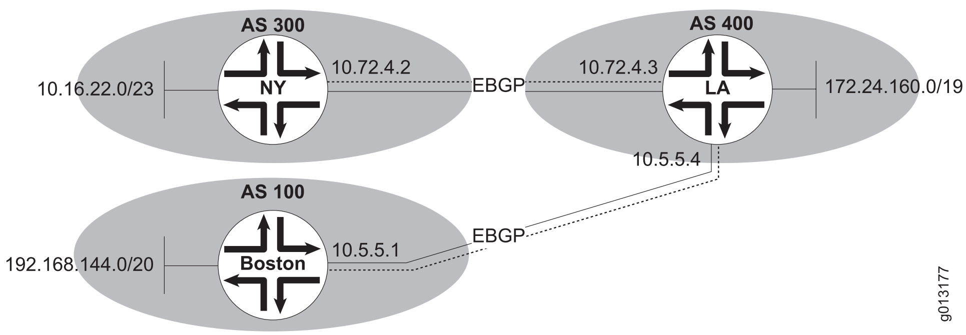


The following commands configure router Boston to apply access list reject1 to routes inbound from router SanJose. Access list reject1 rejects routes matching 172.24.160.0/19.

```
host3(config)#router bgp 17
host3(config-router)#neighbor 10.5.5.4 remote-as 873
host3(config-router)#neighbor 10.5.5.4 distribute-list reject1 in
host3(config-router)#exit
host3(config)#access-list reject1 permit 172.24.48.0 0.0.255
host3(config)#access-list reject1 deny 172.24.160.0 0.0.255
host3(config)#access-list reject1 permit 172.24.24.0 0.0.255
```

Consider the network shown in Figure 22 on page 83. Router NY originates network 10.16.22.0/23 and advertises it to router LA. Suppose you do not want router LA to advertise that network to router Boston. You can apply an access list to updates from router LA to router Boston that prevents router LA from propagating updates for network 10.16.22.0/23.

Figure 22: Filtering Routes with an Access List



The following commands configure router LA:

```
host2(config)#router bgp 400
host2(config-router)#network 172.24.160.0 mask 255.255.224.0
host2(config-router)#neighbor 10.72.4.2 remote-as 300
host2(config-router)#neighbor 10.5.5.1 remote-as 100
host2(config-router)#neighbor 10.5.5.1 distribute-list 1 out
host2(config-router)#exit
host2(config)#access-list 1 deny 10.16.22.0 0.254.255.255
```

access-list

- Use to define an IP access list to permit or deny routes based on the prefix.
- Each access list is a set of permit or deny conditions for routes based on matching a route's prefix.
- Use the **neighbor distribute-list** command to apply the access list to routes received from or forwarded to a neighbor.
- Use the **log** keyword to log an Info event in the ipAccessList log whenever an access-list rule is matched.
- Use the **no** version to delete an IP access list or the specified entry in the access list.
- See *access-list*.

clear access-list

- Use to clear IP access list counters.
- Each access list has a counter for its entries.
- Example

  host1#**clear access-list reject1**
- There is no **no** version.
- See *clear access-list*.

neighbor distribute-list

- Use to filter routes to selected prefixes as specified in an access list.
- Using distribute lists is one of three ways to filter BGP advertisements. The other ways are as follows:
  - Use AS-path filters with the **ip as-path access-list** and the **neighbor filter-list** commands.
  - If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer. However, you cannot configure a member of a peer group to override the inherited peer group characteristic for outbound policy.
  - Use filters with route maps with the **route-map** and the **neighbor route-map** commands.
- New policy values are applied to all routes that are sent (outbound policy) or received (inbound policy) after you issue the command.

  To apply the new policy to routes that are already present in the BGP routing table, you must use the **clear ip bgp** command to perform a soft clear or hard clear of the current BGP session.

  Behavior is different for outbound policies configured for peer groups for which you have enabled Adj-RIBs-Out. If you change the outbound policy for such a peer group and want to fill the Adj-RIBs-Out table for that peer group with the results of the new policy, you must use the **clear ip bgp peer-group** command to perform a hard clear or outbound soft clear of the peer group. You cannot merely perform a hard clear or outbound soft clear for individual peer group members because that causes BGP to resend only the contents of the Adj-RIBs-Out table.
- Use the **no** version to disassociate the access list from a neighbor.
- See *neighbor distribute-list*.

neighbor prefix-list

- Use to assign an inbound or outbound prefix list.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer. However, you cannot configure a member of a peer group to override the inherited peer group characteristic for outbound policy.

- Example

  host1(config-router)#neighbor 192.168.1.158 prefix-list seoul19 in

- New policy values are applied to all routes that are sent (outbound policy) or received (inbound policy) after you issue the command.

  To apply the new policy to routes that are already present in the BGP routing table, you must use the **clear ip bgp** command to perform a soft clear or hard clear of the current BGP session.

  Behavior is different for outbound policies configured for peer groups for which you have enabled Adj-RIBs-Out. If you change the outbound policy for such a peer group and want to fill the Adj-RIBs-Out table for that peer group with the results of the new policy, you must use the **clear ip bgp peer-group** command to perform a hard clear or outbound soft clear of the peer group. You cannot merely perform a hard clear or outbound soft clear for individual peer group members because that causes BGP to resend only the contents of the Adj-RIBs-Out table.

- Use the **no** version to remove the prefix list.
- See *neighbor prefix-list*.

neighbor prefix-tree

- Use to assign an inbound or outbound prefix tree.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer. However, you cannot configure a member of a peer group to override the inherited peer group characteristic for outbound policy.
- Example

  host1(config-router)#neighbor 192.168.1.158 prefix-tree newyork out

- New policy values are applied to all routes that are sent (outbound policy) or received (inbound policy) after you issue the command.

  To apply the new policy to routes that are already present in the BGP routing table, you must use the **clear ip bgp** command to perform a soft clear or hard clear of the current BGP session.

  Behavior is different for outbound policies configured for peer groups for which you have enabled Adj-RIBs-Out. If you change the outbound policy for such a peer group and want to fill the Adj-RIBs-Out table for that peer group with the results of the new policy, you must use the **clear ip bgp peer-group** command to perform a hard clear or outbound soft clear of the peer group. You cannot merely perform a hard clear or outbound soft clear for individual peer group members because that causes BGP to resend only the contents of the Adj-RIBs-Out table.

- IPv6 prefix trees are not supported, Therefore you can specify an IPv6 address with this command only within the IPv4 address family and when you want to advertise IPv4 routes to IPv6 peers.
- Use the **no** version to remove the prefix tree.
- See *neighbor prefix-tree*.

Filtering AS Paths with a Filter List

You can use a filter list to filter incoming and outgoing routes based on the value of the AS-path attribute. Whenever a BGP route passes through an AS, BGP prepends its AS number to the AS-path attribute. The AS-path attribute is the list of ASs that a route has passed through to reach a destination.

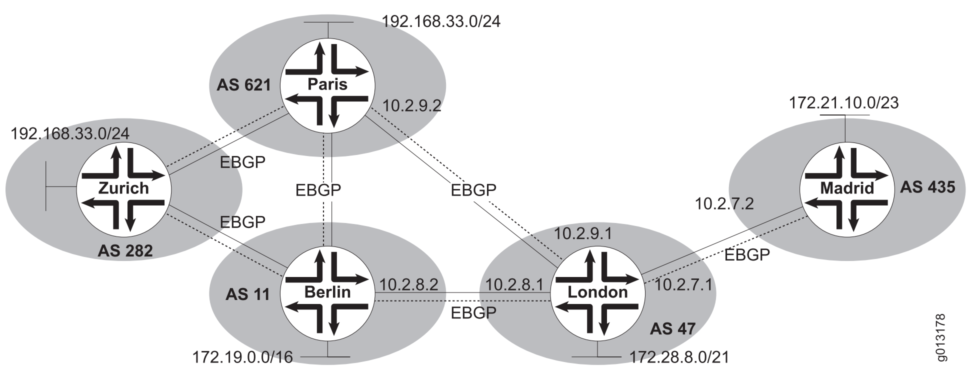
To filter routes based on the AS-path, define the access list with the **ip as-path access-list** command, and apply the list to routes received from or passed to a neighbor with the **neighbor filter-list** command. AS-path access lists use regular expressions to describe the AS path to be matched. A regular expression uses special characters—often referred to as metacharacters—to define a pattern that is compared with an input string. For a full discussion of regular expressions, with examples on how to use them, see *JunosE IP Services Configuration Guide*.

The router compares each route's AS path against the conditions in the access list one by one. If the first match is for a permit condition, the route is accepted or passed. If the first match is for a deny condition, the route is rejected or blocked. The order of conditions is critical because testing stops with the first match. If no conditions match, the router rejects or blocks the route; that is, the last action of any list is an implicit deny condition for all routes.

You cannot selectively place conditions in or remove conditions from an AS-path access list. You can insert a new condition only at the end of an AS-path access list.

Example 1 Consider the network structure in Figure 23 on page 86.

Figure 23: Filtering with AS-Path Access Lists



Suppose you want router London to behave in the following way:

- Accept routes originated in AS 621 only if they pass directly to router London
- Accept routes originated in AS 11 only if they pass directly to router London
- Forward routes from AS 282 to AS 435 only if they pass through either AS 621 or AS 11, but not both AS 621 and AS 11

The following commands configure router London to apply filters based on the AS path to routes received from router Berlin and router Paris and to routes forwarded to router Madrid.


```
host1(config)#router bgp 47
host1(config-router)#neighbor 10.2.9.2 remote-as 621
host1(config-router)#neighbor 10.2.9.2 filter-list 1 in
host1(config-router)#neighbor 10.2.8.2 remote-as 11
host1(config-router)#neighbor 10.2.8.2 filter-list 2 in
host1(config-router)#neighbor 10.2.7.2 remote-as 435
host1(config-router)#neighbor 10.2.7.2 filter-list 3 out
host1(config-router)#exit
host1(config)#ip as-path access-list 1 deny ^621_11$
host1(config)#ip as-path access-list 1 permit .*
host1(config)#ip as-path access-list 2 deny ^11_621$
host1(config)#ip as-path access-list 2 permit .*
host1(config)#ip as-path access-list 3 deny ^11_621_282
host1(config)#ip as-path access-list 3 deny ^621_11_282
host1(config)#ip as-path access-list 3 permit .*
```

AS-path access list 1 is applied to routes that router London receives from router Paris. Router London rejects routes with the AS path (621 11).

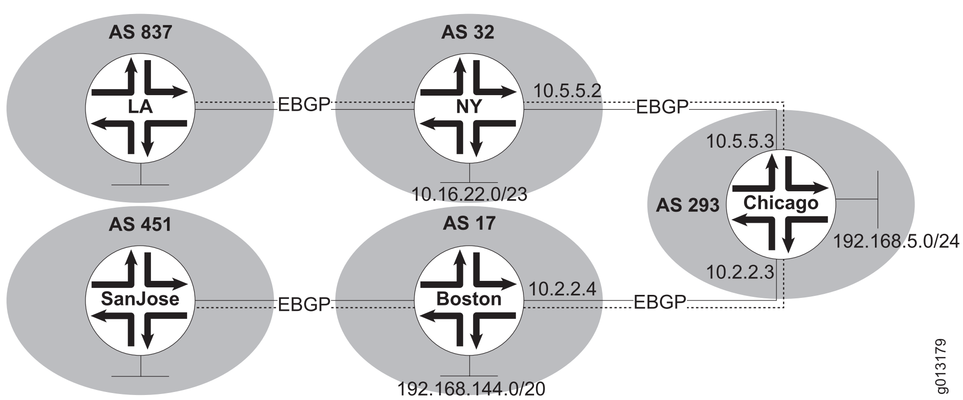
AS-path access list 2 is applied to routes that router London receives from router Berlin. Router London rejects routes with the AS path (11 621) or (621 282 11).

Router London accepts routes with the AS path (11 282), (621 282), (621 11 282), or (11 621 282). However, it applies AS-path access list 3 to routes it forwards to router Madrid, and filters out routes with the AS path (621 11 282) or (11 621 282).

Example 2 Consider the following commands used to configure router Chicago in Figure 24 on page 87:

```
host1(config)#router bgp 293
host1(config-router)#neighbor 10.5.5.2 remote-as 32
host1(config-router)#neighbor 10.5.5.2 filter-list 1 in
host1(config-router)#neighbor 10.2.2.4 remote-as 17
host1(config-router)#exit
host1(config)#ip as-path access-list 1 deny ^32$
```

Figure 24: Assigning a Filter List



Access list 1 denies routes that originate in AS 32—and therefore routes originated by router NY—because the AS-path attribute for these routes begins with (and indeed consists only of) the value 32.

Routes originating anywhere else—such as in AS 837, AS 17, or AS 451—are permitted, because their AS-path attributes do not begin with 32.

ip as-path access-list

- Use to define an AS-path access list to permit or deny routes based on the AS path.
- Each access list is a set of permit or deny conditions for routes based on matching a route's AS path with a regular expression. If the regular expression matches the representation of the AS path of the route as an ASCII string, then the permit or deny condition applies. The AS path does not contain the local AS number.
- Use the **neighbor filter-list** command to apply the AS-path access list. You can apply access list filters to inbound and outbound BGP routes. You can assign weights to routes matching the AS-path access list.
- Use the **no** version to remove a single access list entry if permit or deny and a path-expression are specified. Otherwise, the entire access list is removed.
- See *ip as-path access-list*.

neighbor filter-list

- Use to assign an AS-path access list to matching inbound or outbound routes with the **in** or **out** keywords.
- You can specify an optional weight value with the **weight** keyword to assign a relative importance to incoming routes matching the AS-path access list.
- The name of the access list is a string of up to 32 characters.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer. However, you cannot configure a member of a peer group to override the inherited peer group characteristic for outbound policy.
- New policy values are applied to all routes that are sent (outbound policy) or received (inbound policy) after you issue the command.

  To apply the new policy to routes that are already present in the BGP routing table, you must use the **clear ip bgp** command to perform a soft clear or hard clear of the current BGP session.

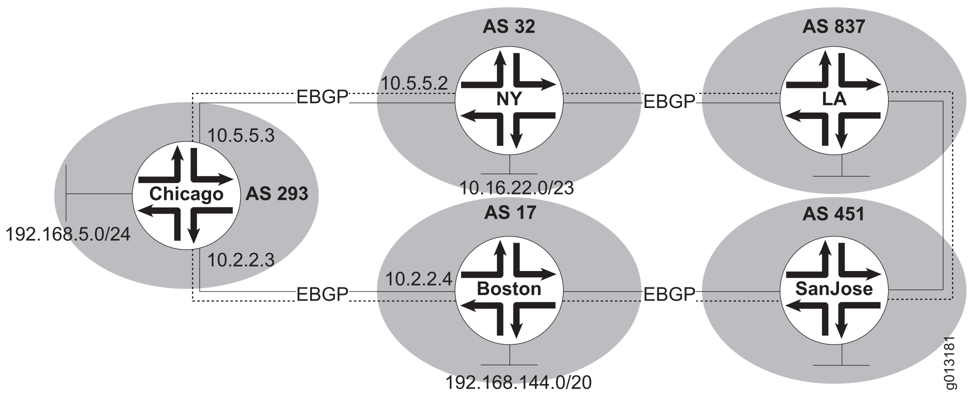
  Behavior is different for outbound policies configured for peer groups for which you have enabled Adj-RIBs-Out. If you change the outbound policy for such a peer group and want to fill the Adj-RIBs-Out table for that peer group with the results of the new policy, you must use the **clear ip bgp peer-group** command to perform a hard clear or outbound soft clear of the peer group. You cannot merely perform a hard clear or outbound soft clear for individual peer group members because that causes BGP to resend only the contents of the Adj-RIBs-Out table.
- Use the **no** version to disassociate the access list from a neighbor.
- See *neighbor filter-list*.

Filtering AS Paths with a Route Map

You can use a route map instead of the **neighbor filter-list** command to apply access lists for filtering routes. In Figure 25 on page 89, suppose router Chicago is configured as follows:

```
host1(config)#router bgp 293
host1(config-router)#network 192.168.5.0 mask 255.255.255.0
host1(config-router)#neighbor 10.2.2.4 remote-as 17
host1(config-router)#neighbor 10.2.2.4 weight 150
host1(config-router)#neighbor 10.5.5.2 remote-as 32
host1(config-router)#neighbor 10.5.5.2 weight 50
```

Figure 25: Route Map Filtering



Routes learned from router Boston have a weight of 150, whereas those learned from router NY have a weight of 50. Router Chicago therefore prefers all routes learned from router Boston to those learned from router NY. Based on this configuration, router Chicago prefers routes to prefixes originating in AS 837 or originating in AS 32 that pass through router Boston over routes to those same prefixes that pass through router NY.

This is a longer path than you might desire. You can avoid this result by configuring a route map to modify the weight of certain routes learned by router Chicago:

```
host1(config-router)#neighbor 10.5.5.2 route-map alpha in
host1(config-router)#exit
host1(config)#route-map alpha permit 10
host1(config-route-map)#match as-path dog1
host1(config-route-map)#set weight 175
host1(config-route-map)#exit
host1(config)#ip as-path access-list dog1 permit _32$
host1(config)#ip as-path access-list dog1 permit _837$
host1(config)#route-map alpha permit 20
host1(config-route-map)#match as-path dog2
host1(config-route-map)#exit
host1(config)#ip as-path access-list dog2 permit .*
```

BGP applies route map alpha to all routes learned from 10.5.5.2 (router NY). Instance 10 of route map alpha matches routes with access list dog1. This access list permits any route whose AS-path attribute ends in 32 or 837—that is, routes that originate in AS 32 or AS 837. It sets their weight to 175, overriding the neighbor weight (50) set for updates received from 10.5.5.2. Then, instance 20 of route map alpha permits all other routes with no modification.

The result of this improved configuration is the following:

- Router Chicago prefers routes learned from router Boston (weight 150) over routes learned from router NY (weight 50), except that
- Router Chicago prefers routes learned from router NY that originate in AS 837 or AS 32 (weight 175 as a result of route map alpha) over the same routes learned from router Boston (weight 150).

Refer to the commands and guidelines in the section "Types of BGP Route Maps" on page 70 for more information about configuring route maps.

Configuring the Community Attribute

A community is a logical group of prefixes that share some common attribute. Community members can be on different networks and in different autonomous systems. BGP allows you to define the community to which a prefix belongs. A prefix can belong to more than one community. The community attribute lists the communities to which a prefix belongs.

You can use communities to simplify routing policies by configuring which routing information a BGP speaker will accept, prefer, or distribute to other neighbors according to community membership. When a route is learned, advertised, or redistributed, a BGP speaker can set, append, or modify the community of a route. When routes are aggregated, the resulting BGP update contains a community attribute that contains all communities from all of the aggregated routes (if the aggregate is an AS-set aggregate).

Several well-known communities have been predefined. Table 19 on page 90 describes how a BGP speaker handles a route based on the setting of its community attribute.

Table 19: Action Based on Well-Known Community Membership

Well-Known Community	BGP Speaker Action
no-export	Does not advertise the route to any EBGP peers (does not advertise the route beyond the local AS)
no-advertise	Does not advertise the route to any peers, IBGP or EBGP
local-as (also known as no-export-subconfed)	Advertises the route only to peers within the local confederation
internet	Advertises this route to the Internet community; by default, all prefixes are members of the Internet community

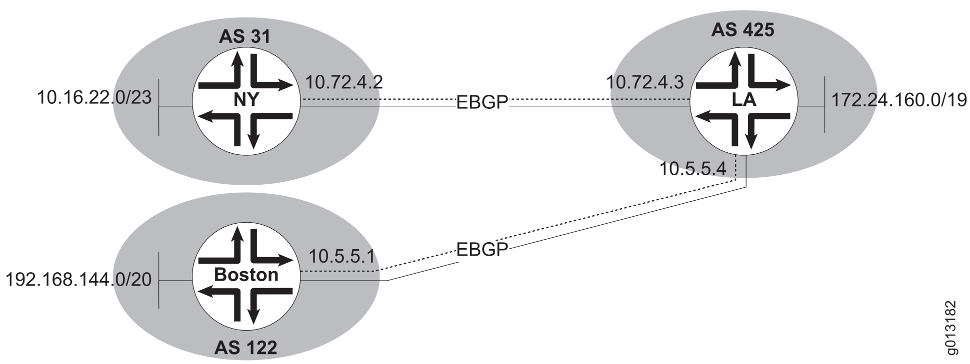
In addition to the well-known communities, you can define local-use communities, also known as private communities or general communities. These communities serve as a convenient way to categorize groups of routes to facilitate the use of routing policies. The community attribute consists of four octets, but it is common practice to designate communities in the *AA*:*NN* format. The autonomous system number (*AA*) comprises the higher two octets, and the community number (*NN*) comprises the lower two octets.

Both are expressed as decimal numbers. For example, if a prefix in AS 23 belongs to community 411, the attribute can be expressed as 23:411. Use the **ip bgp-community new-format** command to specify that the **show** commands display communities in this format.

Use the **set community** command in route maps to configure the community attributes. By default, the community attribute is not sent to BGP peers. To send the community attribute to a neighbor, use the **neighbor send-community** command.

Consider the network structure shown in Figure 26 on page 91. The following sample configurations illustrate some of the capabilities of using the community attribute.

Figure 26: Communities



The following commands configure router NY to apply route map *setcomm* to routes going out to 10.72.4.3. If the community attribute of such a route matches instance 10 of the route map, router NY sets the community attribute to 31:15. All locally originated routes will match this instance of the route map.

```
host1(config)#router bgp 31
host1(config-router)#network 10.16.22.0 mask 255.255.254.0
host1(config-router)#neighbor 10.72.4.3 remote-as 425
host1(config-router)#neighbor 10.72.4.3 send-community
host1(config-router)#neighbor 10.72.4.3 route-map setcomm out
host1(config-router)#exit
host1(config)#ip as-path access-list 1 permit ^$
host1(config)#route-map setcomm permit 10
host1(config-route-map)#match as-path 1
host1(config-route-map)#set community 31:15
```

The following commands configure router LA to apply route map *matchcomm* to routes coming in from 10.72.4.2. If the community attribute of such a route matches instance 10 of the route map, router LA sets the weight of the route to 25.

```
host2(config)#router bgp 425
host2(config-router)#network 172.24.160 mask 255.255.224.0
host2(config-router)#neighbor 10.72.4.2 remote-as 31
host2(config-router)#neighbor 10.72.4.2 send-community
host2(config-router)#neighbor 10.72.4.2 route-map matchcomm in
host2(config-router)#neighbor 10.5.5.1 remote-as 122
host2(config-router)#neighbor 10.5.5.1 send-community
host2(config-router)#exit
```



```
host2(config)#ip community-list 1 permit 31:15
host2(config)#route-map matchcomm permit 10
host2(config-route-map)#match community 1
host2(config-route-map)#set weight 25
```

The following commands configure router Boston to apply route map 5 to routes going out to 10.5.5.4. If the destination IP address of such a route matches instance 10 of the route map, router Boston sets the community attribute of the route to no-export.

```
host3(config)#router bgp 122
host3(config-router)#network 192.168.144.0 mask 255.255.240.0
host3(config-router)#neighbor 10.5.5.4 remote-as 425
host3(config-router)#neighbor 10.5.5.4 send-community
host3(config-router)#neighbor 10.5.5.4 route-map 5 out
host3(config-router)#exit
host3(config)#route-map 5 permit 10
host3(config-route-map)#match ip address access5
host3(config-route-map)#set community no-export
host3(config-route-map)#exit
host3(config)#access-list access5 permit 10.16.22.112
```

Suppose router Boston forwards a route destined for 10.16.22.112 through router LA. Route map 5 matches and sets the community attribute to no-export. As a consequence router LA does not export the route to router NY; the route does not reach its destination.

ip bgp-community new-format

- Use to specify that communities must be displayed in *AA*:*NN* format, where *AA* is a number that identifies the autonomous system and *NN* is a number that identifies the community within the autonomous system.
- Use the **no** version to restore the default display.
- See *ip bgp-community new-format*.

neighbor send-community

- Use to specify that a community attribute must be sent to a BGP neighbor.
- You can specify that only standard communities, only extended communities, or both be sent.
- When you create a neighbor in a VPNv4 address family, that neighbor automatically gets a **neighbor send-community extended** command; this command subsequently appears in a **show configuration** display because it is not the default.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command. You cannot override this inheritance for a peer group member.
- Example

  ```
  host1(config-router)#neighbor send-community westcoast extended
  ```
- New policy values are applied to all routes that are sent (outbound policy) or received (inbound policy) after you issue the command.

To apply the new policy to routes that are already present in the BGP routing table, you must use the **clear ip bgp** command to perform a soft clear or hard clear of the current BGP session.

Behavior is different for outbound policies configured for peer groups for which you have enabled Adj-RIBs-Out. If you change the outbound policy for such a peer group and want to fill the Adj-RIBs-Out table for that peer group with the results of the new policy, you must use the **clear ip bgp peer-group** command to perform a hard clear or outbound soft clear of the peer group. You cannot merely perform a hard clear or outbound soft clear for individual peer group members because that causes BGP to resend only the contents of the Adj-RIBs-Out table.

- Use the **no** version to send only standard communities to a BGP neighbor. Use the **default** version to remove the explicit configuration from the peer or peer group and reestablish inheritance of the feature configuration.
- See *neighbor send-community*.

set community

- Use to set the community attribute in BGP updates.
- You can specify a community list number in the range 1–4294967295, or in the new community format of *AA*:*NN*, or one of the following well-known communities:
  - **local-as**—Prevents advertisement outside the local AS
  - **no-advertise**—Prevents advertisement to any peer
  - **no-export**—Prevents advertisement beyond the BGP confederation boundary
- Alternatively, you can use the **list** keyword to specify the name of a community list that you previously created with the **ip community-list** command.
- Example

  ```
  host1(config)#route-map nyc1 permit 10
  host1(config-route-map)#set community no-advertise
  ```
- Use the **none** keyword to remove the community attribute from a route.
- Use the **no** version to delete the set clause from a route map.
- See *set community*.

Community Lists

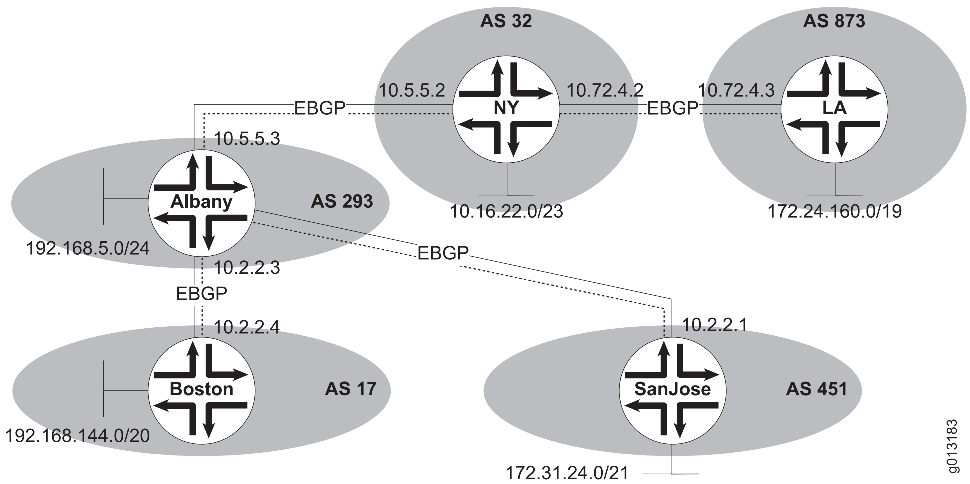
A community list is a sequential collection of permit and deny conditions. Each condition describes the community number to be matched. If you issued the **ip bgp-community new-format** command, the community number is in *AA*:*NN* format; otherwise it is in decimal format.

The router tests the community attribute of a route against the conditions in a community list one by one. The first match determines whether the router accepts (the route is permitted) or rejects (the route is denied) a route having the specified community.

Because the router stops testing conditions after the first match, the order of the conditions is critical. If no conditions match, the router rejects the route.

Consider the network structure shown in Figure 27 on page 94.

Figure 27: Community Lists



Suppose you want router Albany to set metrics for routes that it forwards to router Boston based on the communities to which the routes belong. You can create community lists and filter the routes with a route map that matches on the community list. The following commands demonstrate how to configure router Albany:

```
host1(config)#router bgp 293
host1(config-router)#neighbor 10.5.5.2 remote-as 32
host1(config-router)#neighbor 10.2.2.1 remote-as 451
host1(config-router)#neighbor 10.2.2.4 remote-as 17
host1(config-router)#neighbor 10.2.2.4 route-map commtrc out
host1(config-router)#exit
host1(config)#route-map commtrc permit 1
host1(config-route-map)#match community 1
host1(config-route-map)#set metric 20
host1(config-route-map)#exit
host1(config)#route-map commtrc permit 2
host1(config-route-map)#match community 2
host1(config-route-map)#set metric 75
host1(config-route-map)#exit
host1(config)#route-map commtrc permit 3
host1(config-route-map)#match community 3
host1(config-route-map)#set metric 85
host1(config-route-map)#exit
host1(config)#ip community-list 1 permit 25
host1(config)#ip community-list 2 permit 62
host1(config)#ip community-list 3 permit internet
```

Community list 1 comprises routes with a community of 25; their metric is set to 20. Community list 2 comprises routes with a community of 62; their metric is set to 75. Community 3 catches all remaining routes by matching the internet community; their metric is set to 85.

ip community-list

- Use to create a standard or a regular expression community list for BGP and controls access to it.
- A route can belong to any number of communities, so a community list can have many entries comprising many communities.
- You can specify one or more community values when you create a community list. A clause in a route map that includes a list having more than one value only matches a route having all of the values; that is, the multiple values are logically joined by the AND operator.
- Example

  host1(config)#**ip community-list 1 permit 100:2 100:3 100:4**
  host1(config)#**route-map marengo permit 10**
  host1(config-route-map)#**match community 1**

  A route matches this community list only if it belongs to at least all three communities in community list 1: Communities 100:2, 100:3, and 100:4.
- Use the **no** version to remove a single community list entry if permit or deny and a path-expression are specified. Otherwise, the entire community list is removed.
- See *ip community-list*.

The router supports the new BGP extended community attribute. This attribute enables the definition of a new type of IP extended community and extended community list unrelated to the community list that uses regular expressions. BGP speakers can use the new extended community attribute to control routes similarly to the way it uses the community attribute. The extended community attribute is currently defined in the Internet draft, BGP Extended Communities Attribute—draft-ietf-idr-bgp-ext-communities-07.txt (February 2004 expiration).

NOTE: IETF drafts are valid for only 6 months from the date of issuance. They must be considered as works in progress. Please refer to the IETF Website at http://www.ietf.org for the latest drafts.

ip extcommunity-list

- Use to create an extended community list for BGP and control access to it.
- A route can belong to any number of communities, so an extended community list can have many entries comprising many communities.
- You can specify one or more community values when you create an extended community list. A clause in a route map that includes a list having more than one value only matches a route having all of the values; that is, the multiple values are logically joined by the AND operator.
- Example

  host1(config)#**ip extcommunity-list boston1 permit 100:2 100:3 100:4**

```
host1(config)#route-map marengo permit 10
host1(config-route-map)#match extcommunity boston1
```

A route matches this community list only if it belongs to at least all three communities in extended community list boston1: Communities 100:2, 100:3, and 100:4.

- Use the **no** version to remove a single extended community list entry if permit or deny and a path-expression are specified. Otherwise, the entire community list is removed.
- See *ip extcommunity-list*.

Resetting a BGP Connection

When a routing policy changes, use the **clear ip bgp** and **clear bgp ipv6** commands to clear the current BGP session and implement the new policy.

Clearing a BGP session can create a major disruption in the network operation; this is known as a hard clear. For this reason, you can use the **soft in** and **soft out** options of the command (a soft clear) to activate policies without disrupting the BGP session.

The **soft in** option reapplies inbound policy to received routes; the **soft out** option resends routes to a neighbor after reapplying outbound policy. The **soft in prefix-list** option causes BGP to push any prefix list outbound route filters (ORFs) to the peer and then reapply inbound policy to received routes.

NOTE: Resetting the BGP connection is slightly different when you change outbound policies for peer groups for which you have enabled Adj-RIBs-Out. You cannot merely perform a hard clear or outbound soft clear for individual peer group members because that causes BGP to resend only the contents of the Adj-RIBs-Out table. If you change the outbound policy for such a peer group and want to fill the Adj-RIBs-Out table for that peer group with the results of the new policy, you must use the **clear ip bgp peer-group** command to perform a hard clear or outbound soft clear of the peer group.

clear bgp ipv6

clear ip bgp

- Use to clear the current BGP connection or to activate a new policy without clearing the BGP session.
- You can specify the IP address of a BGP neighbor, the name of a BGP peer group, or an address family to be cleared.
- Use the asterisk (*) to clear all BGP connections.
- If you do not use the **soft in** or **soft out** options, the clear is known as a hard clear and clears the current BGP connection.
- Use the **soft in** option to reapply inbound policy to all received routes without clearing the BGP session.

- Use the **soft in prefix-filter** option to push an ORF to the peer and reapply inbound policy to all received routes without clearing the BGP session.
- Use the **soft out** option to reapply outbound policy and resend routes without clearing the BGP session.
- This command takes effect immediately.
- There is no **no** version.
- See *clear bgp ipv6*.
- See *clear ip bgp*.

Changing Policies Without Disruption

Changing policies can cause major network disruptions when you bring down sessions to reapply the modified policies. You can use either of the methods in this section to minimize network disruptions.

Soft Reconfiguration

You can use *soft reconfiguration* to enable the nondisruptive reapplication of inbound policies. Issuing the command causes the router to store copies of the routes received from the specified peer or from all members of the specified peer group. The route copies are stored unmodified, before application of inbound policies.

If you then change your inbound policies, you can apply them to the stored routes without clearing your BGP sessions—and causing network disruptions—by issuing the **clear ip bgp soft in** command.

neighbor soft-reconfiguration inbound

- Use to initiate the storage of copies of routes received from the specified IP address or from all members of the specified peer group.
- Use with the **clear ip bgp soft in** command to reapply inbound policies to stored routes without clearing the BGP sessions.
- Example

  ```
  host1(config)#router bgp 37
  host1(config-router)#neighbor 192.168.1.1 remote-as 42
  host1(config-router)#neighbor 192.168.1.1 soft-reconfiguration inbound
  ```
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer.
- If the route-refresh capability was negotiated with the peer, BGP immediately sends a route-refresh message to the peer to populate the Adj-RIBs-In table.

  If the route-refresh capability was not negotiated with the peer, BGP automatically bounces the session. The Adj-RIBs-In table is repopulated when routes are received from the peer after the session comes back up.

- Use the **no** version to disable storage of the route copies. Use the **default** version to remove the explicit configuration from the peer or peer group and reestablish inheritance of the feature configuration.
- See *neighbor soft-reconfiguration inbound*.

Route-Refresh Capability

The route-refresh capability provides a lower-cost alternative to soft reconfiguration as a means to change policies without major disruptions. The router advertises the route-refresh capability when it establishes a BGP session with a peer to indicate that it is capable of exchanging BGP route-refresh messages. If inbound soft reconfiguration is disabled (the default) and you issue the **clear ip bgp soft in** command, the router sends route-refresh messages to its peers that have advertised this capability. The messages contain a request for the peer to resend its routes to the router. The new inbound policy is then applied to the routes as they are received.

Our implementation conforms to RFC 2918—Route Refresh Capability for BGP-4 (September 2000), but it also supports nonstandard implementations.

Cooperative Route Filtering

If a BGP speaker negotiates the cooperative route filtering capability with a peer, then the speaker can transfer inbound route filters to the peer. The peer then installs the filter as an outbound route filter (ORF) on the remote end. The ORF is applied by the peer after application of its configured outbound policies. This cooperative filtering has the advantage of both reducing the amount of processing required for inbound BGP updates and reducing the amount of BGP control traffic generated by BGP updates.

clear ip bgp

- Use to push an ORF to the peer and reapply inbound policy to all received routes without clearing the BGP session.
- You can specify the IP address of a BGP neighbor, the name of a BGP peer group, or an address family to be cleared.
- Use the asterisk (*) to clear all BGP connections.
- If the ORF capability is not configured or received on the peer, then the **prefix-filter** keyword is ignored and the router performs a normal inbound soft reconfiguration.
- This command takes effect immediately.
- There is no **no** version.
- See *clear ip bgp*.

neighbor capability

- Use to negotiate the exchange of inbound route filters and their installation as ORFs by specifying the **orf** keyword, an ORF type, and the direction of the capability.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer.
- You cannot configure the **receive** direction for the **orf** capability for a peer that is a member of a peer group or for a peer.
- When issued with the **orf** keyword, this command takes effect immediately and automatically bounces the BGP session.
- Example

  host1(config-router)#**neighbor 192.168.1.158 capability orf prefix-list both**
- Use the **no** version to prevent advertisement of the capability. Use the **default** version to restore the default, advertising the capability.
- See *neighbor capability*.

neighbor maximum-orf-entries

- Use to set the maximum number of ORF entries of any one type that will be accepted from the specified neighbor.
- Example

  host1(config-router)#**neighbor 192.168.1.158 maximum-orf-entries 125000**
- Use the **no** version to restore the default value of no limits.
- See *neighbor maximum-orf-entries*.

neighbor prefix-list

- Use to assign an inbound or outbound prefix list.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer. However, you cannot configure a member of a peer group to override the inherited peer group characteristic for outbound policy.
- Example

  host1(config-router)#**neighbor 192.168.1.158 prefix-list seoul19 in**
- New policy values are applied to all routes that are sent (outbound policy) or received (inbound policy) after you issue the command.

  To apply the new policy to routes that are already present in the BGP routing table, you must use the **clear ip bgp** command to perform a soft clear or hard clear of the current BGP session.

  Behavior is different for outbound policies configured for peer groups for which you have enabled Adj-RIBs-Out. If you change the outbound policy for such a peer group and want to fill the Adj-RIBs-Out table for that peer group with the results of the new policy, you must use the **clear ip bgp peer-group** command to perform a hard clear or outbound soft clear of the peer group. You cannot merely perform a hard clear or outbound soft clear for individual peer group members because that causes BGP to resend only the contents of the Adj-RIBs-Out table.

- Use the **no** version to remove the prefix list.
- See *neighbor prefix-list*.

Configuring Route Flap Dampening

Route flap dampening is a mechanism for minimizing instability caused by route flapping. *Route flapping* occurs when a link is having a problem and is constantly going up and down. Every time the link goes down, the upstream peer withdraws the routes from all its neighbors. When the link comes back up again, the peer advertises those routes globally. When the link problem appears again, the peer withdraws the routes again. This process continues until the underlying problem is fixed.

The router stores a penalty value with each route. Each time the route flaps, the router increases the penalty by 1000. If the penalty for a route reaches a configured *suppress* value, the router suppresses the route. That is, the router does not include the route as a forwarding entry and does not advertise the route to BGP peers.

The penalty decrements by 50 percent for each *half-life* interval that passes. The half-life interval resets when the route flaps and the penalty increments. The route remains suppressed until the penalty falls below the configured *reuse* threshold, at which point the router once again advertises the route. You can specify a *max-suppress-time* for route suppression; after this interval passes, the router once again advertises the route.

BGP creates a *dampening parameter block* for each unique set of dampening parameters—such as suppress threshold and reuse threshold—used by BGP. For example, if you have a route map that sets the dampening parameters to one set of values for some routes and to another set of values for the remaining routes, BGP uses and stores two dampening parameter blocks, one for each set.

Global Route Flap Dampening

Use the **bgp dampening** command if you want to enable route flap dampening with the same values on all BGP routes, or on all routes matching the specified route map. If you specify a route map, the router dampens only routes that are permitted by the route map. For example:

host1(config-router)#**bgp dampening 8 600 2500 30 route-map 1**

bgp dampening

- Use to enable BGP route flap dampening on all BGP routes or routes matching a specified route map.
- You can specify a complete set of values that determine how routes are dampened. If you choose to do so, you must specify the entire set:

  - *half-life*—When this period expires, the penalty assigned to a route is decreased by half
  - *reuse*—When the penalty for a flapping route falls below this limit, the route is unsuppressed (added back to the BGP table and used for forwarding)
  - *suppress*—When a route's penalty exceeds this limit, the route is suppressed
  - *max-suppress-time*—When the period a route has been suppressed exceeds this limit, the route becomes unsuppressed
- If you specify the preceding set of dampening values, you can optionally specify a *half-life-unreachable* period to apply to unreachable routes. If you do not specify this value, the same half-life period is used for both reachable and unreachable routes.
- Dampening applies only to routes learned by means of EBGP.
- The new dampening parameters are applied in future flaps. Changing the dampening parameters does not affect the Figure of Merit that has been calculated for routes using the old dampening parameters. To reset the Figure-of-Merit for all routes, you must issue the **clear ip bgp dampening** command.
- Use the **no** version to disable route flap dampening.
- See *bgp dampening*.

clear bgp ipv6 dampening

clear ip bgp dampening

- Use to clear the BGP route flap dampening information and unsuppress the suppressed routes.
- You can use the **flap-statistics** keyword as an alternative to the **dampening** keyword. Both achieve the same results.
- This command takes effect immediately.
- Examples

  To clear IPv6 dampening information for all routes in all routers:

  host1#**clear bgp ipv6 dampening**

  To clear IPv6 dampening information for a specific route:

  host1#**clear bgp ipv6 dampening 6000::/64**

  To clear IPv4 dampening information for all routes in all address families in all VRFs:

  host1#**clear ip bgp dampening**

  To clear IPv4 dampening information for all routes in VRF dogwood:

  host1#**clear ip bgp ipv4 dogwood dampening**

  To clear IPv4 dampening information for all non-VRF routes in the IPv4 unicast address family:

  host1#**clear ip bgp vrf unicast dampening**

  To clear IPv4 dampening information for a specific route:

host1#**clear ip bgp dampening 192.168.5.0 255.255.255.0**

To clear IPv4 dampening information for the most specific route matching an address:

host1#**clear ip bgp dampening 192.168.5.0**

- There is no **no** version.
- See *clear bgp ipv6 dampening*.
- See *clear ip bgp dampening*.

Policy-Based Route Flap Dampening

You can use policy-based route flap dampening to apply different dampening criteria to different routes. Establish one or more match clauses for an instance of a route map. Then use the **set dampening** command to specify the dampening values that apply to routes that pass all the match clauses for that route map. Consider the following example:

host1(config)#**route-map 21 permit 5**
host1(config-route-map)#**match as-path 1**
host1(config-route-map)#**set dampening 5 1000 1500 45 15**
host1(config-route-map)#**exit**
host1(config)#**ip as-path access-list 1 permit ^300_**

Access list 1 permits routes that originate in AS 300. Instance 5 of route map 21 permits routes that match access list 1 and applies the set of dampening criteria to only those routes; in this case, routes that originate in AS 300.

You can restore the advertisement of routes suppressed as a result of policy-based route flap dampening by issuing the **neighbor unsuppress-map** command. You can unsuppress routes from a specified neighbor or peer group. You must specify a route map; only those routes that match the route map are unsuppressed.

neighbor unsuppress-map

- Use to unsuppress routes that have been suppressed by a **set dampening** clause in a route map.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command. You cannot override this inheritance for a peer group member.
- Routes previously suppressed by a route map that are unsuppressed by this command are not automatically advertised; you must use the **clear ip bgp** command to perform a hard clear or outbound soft clear.
- Example

  host1(config-router)#**neighbor berlin5 unsuppress-map inmap3**

- Use the **no** version to restore the default values.
- See *neighbor unsuppress-map*.

set dampening

- Use to enable BGP route flap dampening only on routes that pass the match clauses of, and are redistributed by, a particular route map.
- BGP creates a dampening parameter block for each unique set of dampening parameters—such as suppress threshold and reuse threshold—used by BGP. For example, if you have a route map that sets the dampening parameters to one set of values for some routes and to another set of values for the remaining routes, BGP uses and stores two dampening parameter blocks, one for each set.
- Example

  host1(config)#**route-map nyc1 permit 10**
  host1(config-route-map)#**set dampening 5 1000 1500 45 15**
- Use the **no** version to delete the set clause from a route map.
- See *set dampening*.

Policy Testing

You can analyze and check your BGP routing policies on your network before you implement the policies. Use the **test ip bgp neighbor** and **test bgp ipv6 neighbor** commands to test the outcome of a BGP policy. The commands output displays the routes that are advertised or accepted if the specified policy is implemented.

NOTE: You can use the standard redirect operators to redirect the test output to network or local files. See the section *JunosE System Basics Configuration Guide*.

BGP routes must be present in the forwarding table for these commands to work properly. If you run the policy test on incoming routes, soft reconfiguration (configured with the **neighbor soft reconfiguration in** command) must be in effect.

NOTE: The output of these commands is always speculative. It does not reflect the current state of the router.

test bgp ipv6 neighbor

test ip bgp neighbor

- Use to test the effect of BGP policies on a router without implementing the policy.
- You can apply the test to routes advertised to peers or received from peers.
- You can test the following kinds of policies: distribute lists, filter lists, prefix lists, prefix trees, or route maps. If you do not specify a policy, then the test uses whatever policies are currently in effect on the router.

NOTE: If you test the current policies, the results might vary for routes learned before the current policies were activated if you did not clear the forwarding table when the policies changed.

- The following three items apply to the **test ip bgp neighbor** command only:
  - The *address-family identifier* for the route is the same as is used for identifying the neighbor.
  - If you do not specify a route, the test is performed for all routes associated with the *address-family identifier*.
  - Specifying only an address and mask without a route distinguisher causes all routes sharing the address and mask to be considered. Specifying only an address causes a best match to be performed for the route.
- If you completely specify a route with IP address, mask, and route distinguisher, the command displays detailed route information. Otherwise only summary information is shown. Use the **fields** option to select particular fields of interest.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer.
- You can set a weight value for inbound routes filtered with a filter list.
- Example

  host1#**test ip bgp neighbor 10.12.54.21 advertised-routes distribute-list boston5 fields**
- There is no **no** version.
- See *test bgp ipv6 neighbor*.
- See *test ip bgp neighbor*.

Selecting the Best Path

BGP selects only one route to a destination as the *best path*. When multiple routes to a given destination exist, BGP must determine which of these routes is the best. BGP puts the best path in its routing table and advertises that path to its BGP neighbors.

If only one route exists to a particular destination, BGP installs that route. If multiple routes exist for a destination, BGP uses tie-breaking rules to decide which one of the routes to install in the BGP routing table.

NOTE: During best path selection on the E series router, BGP takes into account BGP learned routes as well as the routes learned by other protocols to determine if a route should be marked as best and made active. The BGP decision to examine the routes learned by other protocols is driven by administrative distance. If the route learned is associated with a protocol that has a lower administrative distance than BGP, the route preempts the one learned by BGP. At that point, BGP marks its route as inactive and does not install or advertise it. You can use the ***bgp advertise-inactive*** command in the router configuration mode to change this default behavior so that such inactive routes are advertised by BGP. For more information about advertising inactive routes, see "Advertising Inactive Routes" on page 61.

BGP Path Decision Algorithm

BGP determines the best path to each destination for a BGP speaker by comparing path attributes according to the following selection sequence:

1. Select a path with a reachable *next hop*.
2. Select the path with the highest *weight*.
3. If path weights are the same, select the path with the highest *local preference* value.
4. Prefer locally originated routes (network routes, redistributed routes, or aggregated routes) over received routes.
5. Select the route with the shortest *AS-path* length.
6. If all paths have the same AS-path length, select the path based on *origin*: IGP is preferred over EGP; EGP is preferred over Incomplete.
7. If the origins are the same, select the path with lowest *MED* value.
8. If the paths have the same MED values, select the path learned by means of EBGP over one learned by means of IBGP.
9. Select the path with the lowest IGP cost to the next hop.
10. Select the path with the shortest route reflection cluster list. Routes without a cluster list are treated as having a cluster list of length 0.
11. Select the path received from the peer with the lowest BGP router ID.
12. Select the path that was learned from the neighbor with the lowest peer remote address.

The following sections discuss the attributes evaluated in the path decision process. Examples show how you might configure these attributes to influence routing decisions.

Configuring Next-Hop Processing

Routes sent by BGP speakers include the next-hop attribute. The next hop is the IP address of a node on the network that is closer to the advertised prefix. Routers that have traffic destined for the advertised prefix send the traffic to the next hop. The next hop can be the address of the BGP speaker sending the update or of a third-party node. The third-party node does not have to be a BGP speaker.

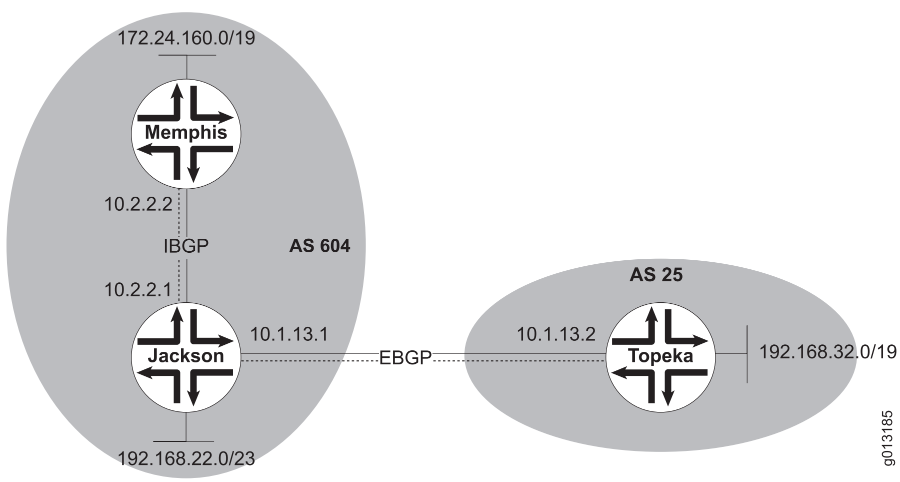
The next-hop attributes conform to the following rules:

- The next hop for EBGP sessions is the IP address of the peer that advertised the route.
- The next hop for IBGP sessions is one of the following:
  - If the route originated inside the AS, the next hop is the IP address of the peer that advertised the route.
  - If the route originated outside the AS—that is, it was injected into the AS by means of an EBGP session—the next hop is the IP address of the external BGP speaker that advertised the route.
- For routes advertised on multiaccess media—such as Frame Relay, ATM, or Ethernet—the next hop is the IP address of the originating router's interface that is connected to the medium.

Next Hops

If you use the **neighbor remote-as** command to configure the BGP neighbors, the next hop is passed according to the rules provided above when networks are advertised. Consider the network configuration shown in Figure 28 on page 106. Router Jackson advertises 192.168.22.0/23 internally to router Memphis with a next hop of 10.2.2.1. Router Jackson advertises the same network externally to router Topeka with a next hop of 10.1.13.1.

Figure 28: Configuring Next-Hop Processing



Router Memphis advertises 172.24.160/19 with a next hop of 10.2.2.2 to router Jackson. Router Jackson advertises this same network externally to router Topeka with a next hop of 10.1.13.1.

Router Topeka advertises network 192.168.32.0/19 with a next hop of 10.1.13.2 to router Jackson. Because this network originates outside AS 604, router Jackson then internally advertises this network (192.168.32.0/19) to router Memphis with the same next hop, 10.1.13.2 (the IP address of the external BGP speaker that advertised the route).

When router Memphis has traffic destined for 192.168.32.0/19, it must be able to reach the next hop by means of an IGP, because it has no direct connection to 10.1.13.2. Otherwise, router Memphis will drop packets destined for 192.168.32.0/19 because the next-hop address is not accessible. Router Memphis does a lookup in its IP routing table to determine how to reach 10.1.13.2:

Destination	Next Hop
10.1.13.0/24	10.2.2.1

The next hop is reachable through router Jackson, and the traffic can be forwarded.

The following commands configure the routers as shown in Figure 28 on page 106:

To configure router Jackson:

```
host1(config)#router bgp 604
host1(config-router)#neighbor 10.1.13.2 remote-as 25
host1(config-router)#neighbor 10.2.2.2 remote-as 604
host1(config-router)#network 192.168.22.0 mask 255.255.254.0
```

To configure router Memphis:

```
host2(config)#router bgp 604
host2(config-router)#neighbor 10.2.2.1 remote-as 604
host2(config-router)#network 172.24.160.0 mask 255.255.224.0
```

To configure router Topeka:

```
host3(config)#router bgp 25
host3(config-router)#neighbor 10.1.13.1 remote-as 604
host3(config-router)#network 172.31.64.0 mask 255.255.192.0
```

Additional configuration is required for routers Biloxi, Memphis, and Jackson; the details depend on the IGP running in AS 604.

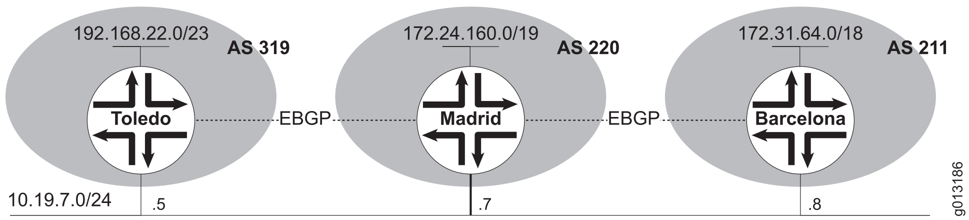
neighbor remote-as

- Use to add an entry to the BGP neighbor table.
- Specifying a neighbor with an AS number that matches the AS number specified in the router bgp command identifies the neighbor as internal to the local AS. Otherwise, the neighbor is treated as an external neighbor.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer.
- This command takes effect immediately.
- Use the **no** version to remove an entry from the table.
- See *neighbor remote-as*.

Next-Hop-Self

In some circumstances, using a third-party next hop causes routing problems. These configurations typically involve nonbroadcast multiaccess (NBMA) media. To better understand this situation, first consider a broadcast multiaccess (BMA) media network, as shown in Figure 29 on page 108.

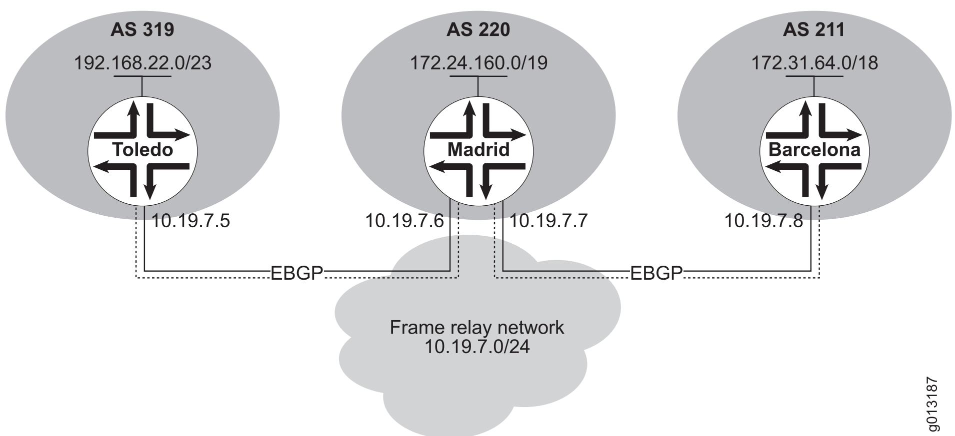
Figure 29: Next-Hop Behavior for Broadcast Multiaccess Media



Routers Toledo, Madrid, and Barcelona are all on the same Ethernet network, which has a prefix of 10.19.7.0/24. When router Toledo advertises prefix 192.168.22.0/23 to router Madrid, it sets the next-hop attribute to 10.19.7.5. Before router Madrid advertises this prefix to router Barcelona, it sees that its own IP address, 10.19.7.7, is on the same subnet as the next hop for the advertised prefix. If router Barcelona can reach router Madrid, then it should be able to reach router Toledo. Router Madrid therefore advertises 192.168.22.0/23 to router Barcelona with a next-hop attribute of 10.19.7.5.

Now consider Figure 30 on page 108, which shows the same routers on a Frame Relay—NBMA—network.

Figure 30: Next-Hop Behavior for Nonbroadcast Multiaccess Media



Routers Toledo and Madrid are EBGP peers, as are routers Madrid and Barcelona. When router Toledo advertises prefix 192.168.22.0/23 to router Madrid, router Madrid makes the same comparison as in the BMA example, and leaves the next-hop attribute intact when it advertises the prefix to router Barcelona. However, router Barcelona will not be able to forward traffic to 192.168.22.0/23, because it does not have a direct PVC connection to router Toledo and cannot reach the next hop of 10.19.7.5.

You can use the **neighbor next-hop-self** command to correct this routing problem. If you use this command to configure router Madrid, the third-party next hop advertised by router Toledo is not advertised to router Barcelona. Instead, router Madrid advertises 192.168.22.0/23 with the next-hop attribute set to its own IP address, 10.19.7.7. Router Barcelona now forwards traffic destined for 192.168.22.0/23 to the next hop, 10.19.7.7. Router Madrid then passes the traffic along to router Toledo.

To disable third-party next-hop processing, configure router Madrid as follows:

```
host1(config)#router bgp 319
host1(config-router)#neighbor 10.19.7.8 remote-as 211
host1(config-router)#neighbor 10.19.7.8 next-hop-self
```

neighbor next-hop-self

- Use to prevent third-party next hops from being used on NBMA media such as Frame Relay. This command is useful in nonmeshed networks such as Frame Relay or where BGP neighbors may not have direct access to the same IP subnet.
- Forces the BGP speaker to report itself as the next hop for an advertised route it learned from a neighbor.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command. You cannot override the characteristic for a specific member of the peer group.
- New policy values are applied to all routes that are sent (outbound policy) or received (inbound policy) after you issue the command.

  To apply the new policy to routes that are already present in the BGP routing table, you must use the **clear ip bgp** command to perform a soft clear or hard clear of the current BGP session.

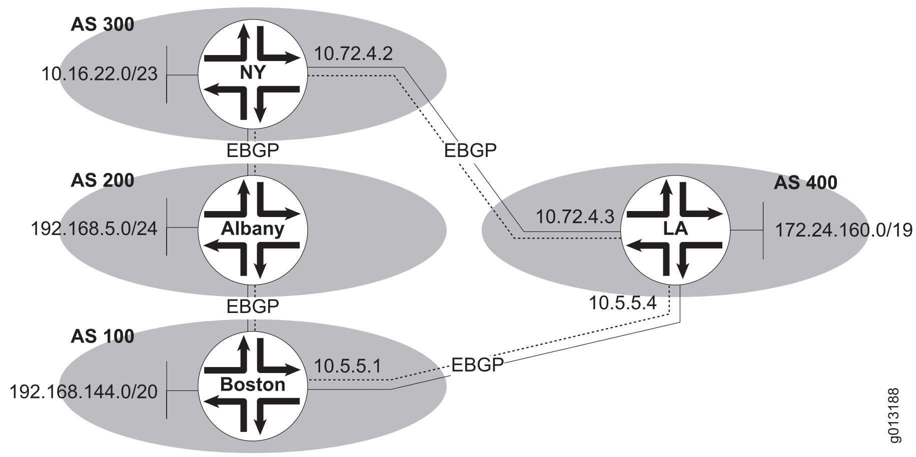
  Behavior is different for outbound policies configured for peer groups for which you have enabled Adj-RIBs-Out. If you change the outbound policy for such a peer group and want to fill the Adj-RIBs-Out table for that peer group with the results of the new policy, you must use the **clear ip bgp peer-group** command to perform a hard clear or outbound soft clear of the peer group. You cannot merely perform a hard clear or outbound soft clear for individual peer group members because that causes BGP to resend only the contents of the Adj-RIBs-Out table.
- Issuing this command automatically removes the **neighbor next-hop-unchanged** configuration (enabled or disabled) on the peer or peer group. Issuing the **no** or **default** version of this command has no effect on the **neighbor next-hop-unchanged** configuration.
- Use the **no** version to disable this feature (and therefore enable next-hop processing of BGP updates). Use the **default** version to remove the explicit configuration from the peer or peer group and reestablish inheritance of the feature configuration.
- See *neighbor next-hop-self*.

Assigning a Weight to a Route

You can assign a weight to a route when more than one route exists to the same destination. A weight indicates a preference for that particular route over the other routes to that destination. The higher the assigned weight, the more preferred the route. By default, the route weight is 32768 for paths originated by the router, and 0 for other paths.

In the configuration shown in Figure 31 on page 110, routers Boston and NY both learn about network 192.68.5.0/24 from AS 200. Routers Boston and NY both propagate the route to router LA. Router LA now has two routes for reaching 192.68.5.0/24 and must decide the appropriate route. If you prefer that router LA direct traffic through router Boston, you can configure router LA so that the weight of routes coming from router Boston are higher—more preferred—than the routes coming from router NY. Router LA subsequently prefers routes received from router Boston and therefore uses router Boston as the next hop to reach network 192.68.5.0/24.

Figure 31: Assigning a Weight to a Neighbor Connection



You can use any of the following three ways to set the weights in routes coming in from router Boston:

- The **neighbor weight** command
- The **set weight** command in route maps
- An AS-path access list

Using the neighbor weight Command

The following commands assign a weight of 1000 to all routes router LA receives from AS 100 and assign a weight of 500 to all routes router LA receives from AS 300:

```
host1(config)#router bgp 400
host1(config-router)#neighbor 10.5.5.1 remote-as 100
host1(config-router)#neighbor 10.5.5.1 weight 1000
host1(config-router)#neighbor 10.72.4.2 remote-as 300
host1(config-router)#neighbor 10.72.4.2 weight 500
```

Router LA sends traffic through router Boston in preference to router NY.

Using a Route Map

A route map instance is a set of conditions with an assigned number. The number after the **permit** keyword designates an instance of a route map. For example, instance 10 of route map 10 begins with the following:

```
host1(config)#route-map 10 permit 10
```

In the following commands to configure router LA, instance 10 of route map 10 assigns a weight of 1000 to any routes from AS 100. Instance 20 assigns a weight of 500 to routes from any other AS.

```
host1(config)#router bgp 400
host1(config-router)#neighbor 10.5.5.1 remote-as 100
host1(config-router)#neighbor 10.5.5.1 route-map 10 in
host1(config-router)#neighbor 10.72.4.2 remote-as 300
host1(config-router)#neighbor 10.72.4.2 route-map 20 in
host1(config-router)#exit
host1(config)#route-map 10
host1(config-route-map)#set weight 1000
host1(config-route-map)#route-map 20
host1(config-route-map)#set weight 500
```

See *JunosE IP Services Configuration Guide* for more information about using route maps.

Using an AS-Path Access List

The following commands assign weights to routes filtered by AS-path access lists on router LA:

```
host1(config)#router bgp 400
host1(config-router)#neighbor 10.5.5.1 remote-as 100
host1(config-router)#neighbor 10.5.5.1 filter-list 1 weight 1000
host1(config-router)#neighbor 10.72.4.2 remote-as 300
host1(config-router)#neighbor 10.72.4.2 filter-list 2 weight 500
host1(config-router)#exit
host1(config)#ip as-path access-list 1 permit ^100_
host1(config)#ip as-path access-list 2 permit ^300_
```

Access list 1 permits any route whose AS-path attribute begins with 100 (specified by ^). This permits routes that pass through router Boston, whether they originate in AS 100 (AS path = 100) or AS 200 (AS path = 100 200) or AS 300 (AS path = 100 200 300). Access list 2 permits any route whose AS-path attribute begins with 300. This permits routes that pass through router NY, whether they originate in AS 300 (AS path = 300) or AS 200 (AS path = 300 200) or AS 100 (AS path = 300 200 100).

The **neighbor filter-list** commands assign a weight attribute of 1000 to routes passing through router Boston and a weight attribute of 500 to routes passing through router NY. Regardless of the origin of the route, routes learned through router Boston are preferred.

ip as-path access-list

- Use to define a BGP access list; use the **neighbor filter-list** command to apply a specific access list.
- You can apply access list filters on inbound or outbound BGP routes, or both.
- You can **permit** or **deny** access for a route matching the condition(s) specified by the regular expression.
- If the regular expression matches the representation of the AS path of the route as an ASCII string, then the *permit* or *deny* condition applies.
- The AS path allows substring matching. For example, the regular expression *20* matches AS path = *20* and AS path = *100 200 300*, because *20* is a substring of each path. To disable substring matching and constrain matching to only the specified attribute string, place the underscore (_) metacharacter on both sides of the string, for example *_20_*.
- The AS path does not contain the local AS number.
- Use the **no** version to remove a single access list entry if permit or deny and a path-expression are specified. Otherwise, the entire access list is removed.
- See *ip as-path access-list*.

neighbor filter-list

- Use to apply an AS-path access list to advertisements inbound from or outbound to the specified neighbor, or to assign a weight to incoming routes that match the AS-path access list.
- You can specify an optional weight value with the **weight** keyword to assign a relative importance to incoming routes matching the AS-path access list.
- The name of the access list is a string of up to 32 characters.
- You can apply the filter to incoming or outgoing advertisements with the **in** or **out** keywords.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer. However, you cannot configure a member of a peer group to override the inherited peer group characteristic for outbound policy.
- New policy values are applied to all routes that are sent (outbound policy) or received (inbound policy) after you issue the command.

  To apply the new policy to routes that are already present in the BGP routing table, you must use the **clear ip bgp** command to perform a soft clear or hard clear of the current BGP session.

  Behavior is different for outbound policies configured for peer groups for which you have enabled Adj-RIBs-Out. If you change the outbound policy for such a peer group and want to fill the Adj-RIBs-Out table for that peer group with the results of the new policy, you must use the **clear ip bgp peer-group** command to perform a hard clear or outbound soft clear of the peer group. You cannot merely perform a hard clear or outbound soft clear for individual peer group members because that causes BGP to resend only the contents of the Adj-RIBs-Out table.

- Use the **no** version to remove the filter list.
- See *neighbor filter-list*.

neighbor weight

- Use to assign a weight to a neighbor connection.
- All routes learned from this neighbor will have the assigned weight initially.
- The route with the highest weight will be chosen as the preferred route when multiple routes are available to a particular network.
- The weights assigned with the set weight commands in a route map override the weights assigned with the neighbor weight and neighbor filter-list commands.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer.
- New policy values are applied to all routes that are sent (outbound policy) or received (inbound policy) after you issue the command.

  To apply the new policy to routes that are already present in the BGP routing table, you must use the **clear ip bgp** command to perform a soft clear or hard clear of the current BGP session.

  Behavior is different for outbound policies configured for peer groups for which you have enabled Adj-RIBs-Out. If you change the outbound policy for such a peer group and want to fill the Adj-RIBs-Out table for that peer group with the results of the new policy, you must use the **clear ip bgp peer-group** command to perform a hard clear or outbound soft clear of the peer group. You cannot merely perform a hard clear or outbound soft clear for individual peer group members because that causes BGP to resend only the contents of the Adj-RIBs-Out table.
- Use the **no** version to remove the weight assignment.
- See *neighbor weight*.

See “Access Lists” on page 82 for more information about using access lists.

Configuring the Local-Pref Attribute

The local-pref attribute specifies the preferred path among multiple paths to the same destination. The preferred path is the one with the higher preference value. Local preference is used only within an AS, to select an exit point.

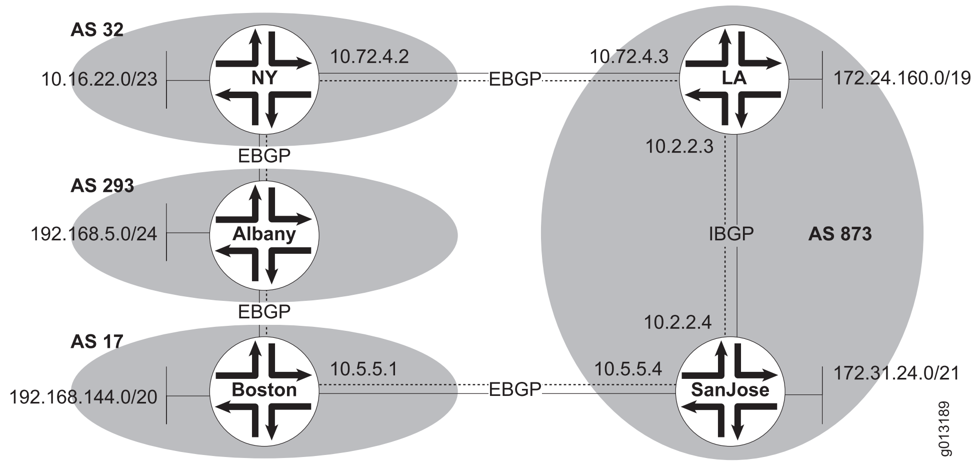
To configure the local preference of a BGP path, you can do one of the following:

- Use the **bgp default local-preference** command to set the local-preference attribute.
- Use a route map to set the local-pref attribute.

Using the bgp default local-preference Command

In Figure 32 on page 114, AS 873 receives updates for network 192.168.5.0/24 from AS 32 and AS 17.

Figure 32: Configuring the Local-Preference Attribute



The following commands configure router LA:

```
host1(config-router)#router bgp 873
host1(config-router)#neighbor 10.72.4.2 remote-as 32
host1(config-router)#neighbor 10.2.2.4 remote-as 873
host1(config-router)#bgp default local-preference 125
```

The following commands configure router SanJose:

```
host2(config-router)#router bgp 873
host2(config-router)#neighbor 10.5.5.1 remote-as 17
host2(config-router)#neighbor 10.2.2.3 remote-as 873
host2(config-router)#bgp default local-preference 200
```

Router LA sets the local preference for all updates from AS 32 to 125. Router SanJose sets the local preference for all updates from AS 17 to 200. Because router LA and router SanJose exchange local preference information within AS 873, they both recognize that routes to network 192.168.5.0/24 in AS 293 have a higher local preference when they come to AS 873 from AS 17 than when they come from AS 32. As a result, both router LA and router SanJose prefer to reach this network through router Boston in AS 17.

bgp default local-preference

- Use to change the default local preference value.
- Changes apply automatically whenever BGP subsequently runs the best-path decision process for a destination prefix; that is, whenever a best route is picked for a given prefix.

  To force BGP to run the decision process on routes already received, you must use the **clear ip bgp** command to perform an inbound soft clear or hard clear of the current BGP session.

- Use the **no** version to restore the default value, 100.
- See *bgp default local-preference*.

Using a Route Map to Set the Local Preference

When you use a route map to set the local preference you have more flexibility in selecting routes for which you can set a local preference based on many criteria, including AS. In the previous section, all updates received by router SanJose were set to a local preference of 200.

Using a route map, you can specifically assign a local preference for routes from AS 17 that pass through AS 293.

The following commands configure router SanJose.

```
host2(config-router)#router bgp 873
host2(config-router)#neighbor 10.2.2.3 remote-as 873
host2(config-router)#neighbor 10.5.5.1 remote-as 17
host2(config-router)#neighbor 10.5.5.1 route-map 10 in
host2(config-router)#exit
host2(config)#ip as-path access-list 1 permit ^17 293$
host2(config)#route-map 10 permit 10
host2(config-route-map)#match as-path 1
host2(config-route-map)#set local-preference 200
host2(config-route-map)#exit
host2(config)#route-map 10 permit 20
```

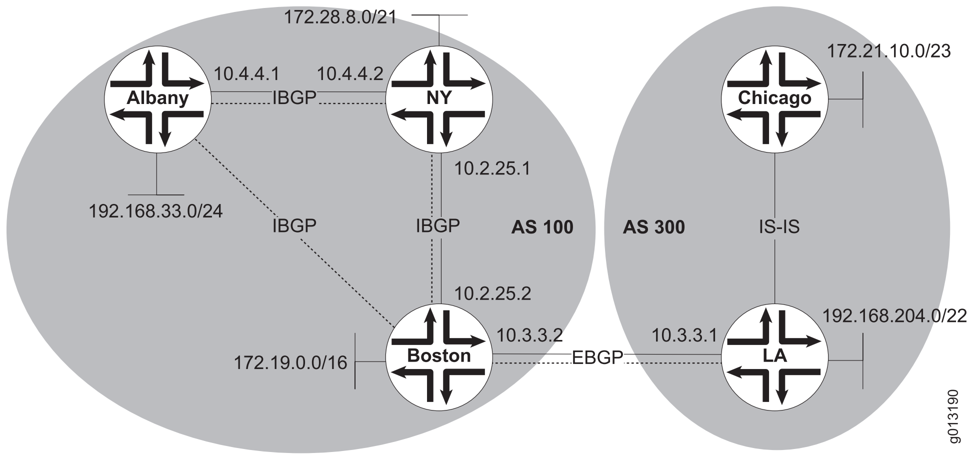
Router SanJose sets the local-pref attributes to 200 for routes originating in AS 293 and passing last through AS 17. All other routes are accepted (as defined in instance 20 of the route map 10), but their local preference remains at the default value of 100, indicating a less-preferred path.

Understanding the Origin Attribute

BGP uses the origin attribute to describe how a route was learned at the origin—the point where the route was injected into BGP. The origin of the route can be one of three values:

- IGP—Indicates that the route was learned by means of an IGP and, therefore, is internal to the originating AS. All routes advertised by the **network** command have an origin of IGP.
- EGP—Indicates that the route was learned by means of an EGP.
- Incomplete—Indicates that the origin of the route is unknown—that is, learned from something other than IGP or EGP. All routes advertised by the **redistribute** command—such as static routes—have an origin of Incomplete. An origin of Incomplete occurs when a route is redistributed into BGP.

Figure 33: The Origin Attribute



Consider the sample topology shown in Figure 33 on page 116. Because routers Albany and Boston are not directly connected, they learn the path to each other by means of an IGP (not illustrated).

The following commands configure router Boston:

```
host1(config)#ip route 172.31.125.100 255.255.255.252
host1(config)#router bgp 100
host1(config-router)#neighbor 10.2.25.1 remote-as 100
host1(config-router)#neighbor 10.4.4.1 remote-as 100
host1(config-router)#neighbor 10.3.3.1 remote-as 300
host1(config-router)#network 172.19.0.0
host1(config-router)#redistribute static
```

The following commands configure router NY:

```
host2(config)#router bgp 100
host2(config-router)#neighbor 10.4.4.1 remote-as 100
host2(config-router)#neighbor 10.2.25.2 remote-as 100
host2(config-router)#network 172.28.8.0 mask 255.255.248.0
```

The following commands configure router Albany:

```
host3(config)#router bgp 100
host3(config-router)#neighbor 10.4.4.2 remote-as 100
host3(config-router)#neighbor 10.2.25.2 remote-as 100
host3(config-router)#network 192.168.33.0 mask 255.255.255.0
```

The following commands configure router LA:

```
host4(config)#router bgp 300
host4(config-router)#neighbor 10.3.3.2 remote-as 100
host4(config-router)#network 192.168.204.0 mask 255.255.252.0
host4(config-router)#redistribute isis
```

Consider how route 172.21.10.0/23 is passed along to the routers in Figure 33 on page 116:

1. IS-IS injects route 172.21.10.0/23 from router Chicago into BGP on router LA. BGP sets the origin attribute to Incomplete (because it is a redistributed route) to indicate how BGP originally became aware of the route.
2. Router Boston learns about route 172.21.10.0/23 by means of EBGP from router LA.
3. Router NY learns about route 172.21.10.0/23 by means of IBGP from router Boston.

The value of the origin attribute for a given route remains the same, regardless of where you examine it. Table 20 on page 117 shows this for all the routes known to routers NY and LA.

Table 20: Origin and AS Path for Routes Viewed on Different Routers

Route	Router	Origin	AS Path
192.168.204.0/22	Albany	IGP	300
192.168.204.0/22	Boston	IGP	300
192.168.204.0/22	NY	IGP	300
192.168.204.0/22	LA	IGP	empty
172.21.10.0/23	Albany	Incomplete	300
172.21.10.0/23	Boston	Incomplete	300
172.21.10.0/23	NY	Incomplete	300
172.21.10.0/23	LA	Incomplete	empty
172.28.8.0/21	Albany	IGP	empty
172.28.8.0/21	Boston	IGP	empty
172.28.8.0/21	NY	IGP	empty
172.28.8.0/21	LA	IGP	100
172.31.125.100	Albany	Incomplete	empty
172.31.125.100	Boston	Incomplete	empty
172.31.125.100	NY	Incomplete	empty
172.31.125.100	LA	Incomplete	100
172.19.0.0/16	Albany	IGP	empty
172.19.0.0/16	Boston	IGP	empty

Table 20: Origin and AS Path for Routes Viewed on Different Routers *(continued)*

Route	Router	Origin	AS Path
172.19.0.0/16	NY	IGP	empty
172.19.0.0/16	LA	IGP	100
192.168.330/24	Albany	IGP	empty
192.168.330/24	Boston	IGP	empty
192.168.330/24	NY	IGP	empty
192.168.330/24	LA	IGP	100

As a matter of routing policy, you can specify an origin for a route with a **set origin** clause in a redistribution route map. Changing the origin enables you to influence which of several routes for the same destination prefix is selected as the best route. In practice, changing the origin is rarely done.

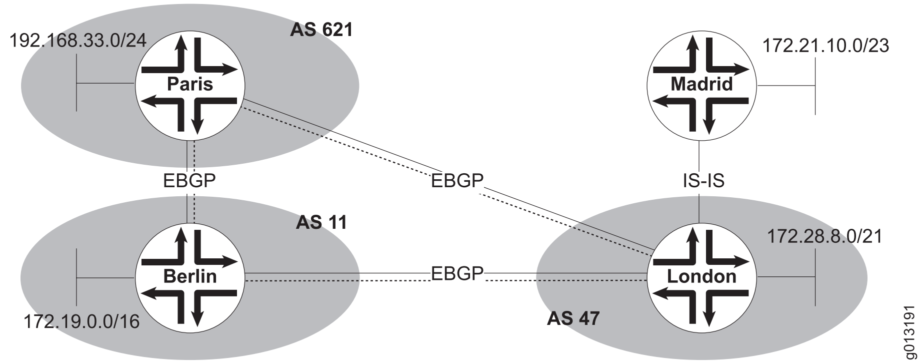
Understanding the AS-Path Attribute

The AS-path attribute is a list of the ASs through which a route has passed. Whenever a route enters an AS, BGP prepends the AS number to the AS-path attribute. This feature enables network operators to track routes, but it also enables the detection and prevention of routing loops.

Consider the following sequence of events for the routers shown in Figure 34 on page 119:

1. Route 172.21.10.0/23 is injected into BGP by means of router London in AS 47.
2. Suppose router London advertises that route to router Paris in AS 621. As received by router Paris, the AS-path attribute for route 172.21.10.0/23 is 47.
3. Router Paris advertises the route to router Berlin in AS 11. As received by router Berlin, the AS-path attribute for route 172.21.10.0/23 is 621 47.
4. Router Berlin advertises the route to router London in AS 47. As received by router London, the AS-path attribute for route 172.21.10.0/23 is 11 621 47.

Figure 34: AS-Path Attributes



A routing loop exists if router London accepts the route from router Berlin. Router London can choose not to accept the route from router Berlin because it recognizes from the AS-path attribute (11 621 47) that the route originated in its own AS 47.

As a matter of routing policy, you can prepend additional AS numbers to the AS-path attribute for a route with a **set as-path prepend** clause in an outbound route map. Changing the AS path enables you to influence which of several routes for the same destination prefix is selected as the best route.

Configuring a Local AS

You can change the local AS of a BGP peer or peer group within the current address family with the **neighbor local-as** command. By using different local AS numbers for different peers, you can avoid or postpone AS renumbering in the event the ASs are merged.

neighbor local-as

- Use to assign a local AS to the given BGP peer or peer group.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer.
- This command takes effect immediately and automatically bounces the BGP session.
- Use the **no** version for an individual peer to restore the value set for the peer group, if present, or set globally for BGP with the **router bgp** command. Use the **no** version for a peer group to restore the value set globally for BGP.
- See *neighbor local-as*.

The following example commands change the local AS number for peer 104.4.2 from the global local AS of 100 to 32:

```
host1(config)#router bgp 100
host1(config-router)#address-family ipv4 unicast vrf boston
host1(config-router)#neighbor 10.4.4.2 remote-as 645
host1(config-router)#neighbor 10.4.4.2 local-as 32
```


Configuring the MED Attribute

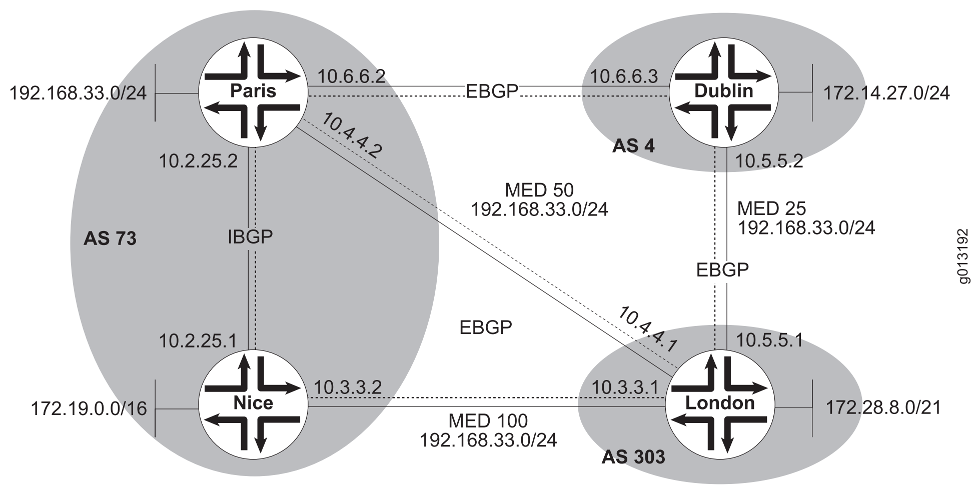
If two ASs connect to each other in more than one place, one link or path might be a better choice to reach a particular prefix within or behind one of the ASs. The MED value is a metric expressing a degree of preference for a particular path. Lower MED values are preferred.

Whereas the Local Preference attribute is used only within an AS (to select an exit point), the MED attribute is exchanged between ASs. A router in one AS sends the MED to inform a router in another AS which path the second router should use to reach particular destinations. If you are the administrator of the second AS, you must therefore trust that the router in the first AS is providing information that is truly beneficial to your AS.

You configure the MED on the sending router by using the **set metric** command in an outbound route map. Unless configured otherwise, a receiving router compares MED attributes only for paths from external neighbors that are members of the same AS. If you want MED attributes from neighbors in different ASs to be compared, you must issue the **bgp always-compare-med** command.

In Figure 35 on page 120, router London in AS 303 can reach 192.168.33.0/24 in AS 73 through router Paris or through router Nice to router Paris.

Figure 35: Configuring the MED



The following commands configure router London:

```
host1(config)#router bgp 303
host1(config-router)#neighbor 10.4.4.2 remote-as 73
host1(config-router)#neighbor 10.3.3.2 remote-as 73
host1(config-router)#neighbor 10.5.5.2 remote-as 4
host1(config-router)#network 122.28.8.0 mask 255.255.248.0
```

The following commands configure router Paris:

```
host2(config)#router bgp 73
host2(config-router)#neighbor 10.4.4.1 remote-as 303
host2(config-router)#neighbor 10.4.4.1 route-map 10 out
```

```
host2(config-router)#neighbor 10.2.25.1 remote-as 73
host2(config-router)#neighbor 10.6.6.1 remote-as 4
host2(config-router)#neighbor 10.6.6.1 route-map 10 out
host2(config-router)#network 192.168.33.0 mask 255.255.255.0
host2(config-router)#exit
host2(config)#route-map 10 permit 10
host2(config-route-map)#set metric 50
```

The following commands configure router Nice:

```
host3(config)#router bgp 73
host3(config-router)#neighbor 10.3.3.1 remote-as 303
host3(config-router)#neighbor 10.3.3.1 route-map 10 out
host3(config-router)#neighbor 10.2.25.2 remote-as 73
host3(config-router)#network 172.19.0.0
host3(config-router)#exit
host3(config)#route-map 10 permit 10
host3(config-route-map)#set metric 100
```

The following commands configure router Dublin:

```
host4(config)#router bgp 4
host4(config-router)#neighbor 10.5.5.1 remote-as 303
host4(config-router)#neighbor 10.5.5.1 route-map 10 out
host4(config-router)#neighbor 10.6.6.2 remote-as 73
host4(config-router)#network 172.14.27.0 mask 255.255.255.0
host4(config-router)#exit
host4(config)#route-map 10 permit 10
host4(config-route-map)#set metric 25
```

Router London receives updates regarding route 192.168.33.0/24 from both router Nice and router Paris. Router London compares the MED values received from the two routers: Router Nice advertises a MED of 100 for the route, whereas router Paris advertises a MED of 50. On this basis, router London prefers the path through router Paris.

Because BGP by default compares only MED attributes of routes coming from the same AS, router London can compare only the MED attributes for route 192.168.33.0/24 that it received from routers Paris and Nice. It cannot compare the MED received from router Dublin, because router Dublin is in a different AS than routers Paris and Nice.

However, you can use the **bgp always-compare-med** command to configure router London to take into account the MED attribute from router Dublin as follows:

```
host1(config)#router bgp 303
host1(config-router)#neighbor 10.4.4.2 remote-as 73
host1(config-router)#neighbor 10.3.3.2 remote-as 73
host1(config-router)#neighbor 10.5.5.2 remote-as 4
host1(config-router)#network 122.28.8.0 mask 255.255.248.0
host1(config-router)#bgp always-compare-med
```

Router Dublin advertises a MED of 25 for route 192.168.33.0/24, which is lower—more preferred—than the MED advertised by router Paris or router Nice. However, the AS path for the route through router Dublin is longer than that through router Paris. The AS path is the same length for router Paris and router Nice, but the MED advertised by router Paris is lower than that advertised by router Nice. Consequently, router London prefers the path through router Paris.

Suppose, however that router Dublin was not configured to set the MED for route 192.168.33.0/24 in its outbound route map 10. Would router London receive a MED of 50 passed along by router Paris through router Dublin? No, because the MED attribute is nontransitive. Router Dublin does not transmit any MED that it receives. A MED is only of value to a direct peer.

bgp always-compare-med

- Use to enable the comparison of the MED for paths from neighbors in different ASs.
- Unless you specify the **bgp always-compare-med** command, the router compares MED attributes only for paths from external neighbors that are in the same AS.
- The BGP path decision algorithm selects a lower MED value over a higher one.
- Unlike local preferences, the MED attribute is exchanged between ASs, but does not leave the AS.
- The value is used for decision making within the AS only.
- When BGP propagates a route received from outside the AS to another AS, it removes the MED.
- Example

  host1(config-router)#**bgp always-compare-med**
- Changes apply automatically whenever BGP subsequently runs the best-path decision process for a destination prefix; that is, whenever a best route is picked for a given prefix.

  To force BGP to run the decision process on routes already received, you must use the **clear ip bgp** command to perform an inbound soft clear or hard clear of the current BGP session.
- Use the **no** version to disable the feature.
- See *bgp always-compare-med*.

set metric

- Use to set the metric value—for BGP, the MED—for a route.
- Sets an absolute metric. You cannot use both an absolute metric and a relative metric within the same route map sequence. Setting either metric overrides any previously configured value.
- Example

  host1(config)#**route-map nyc1 permit 10**
  host1(config-route-map)#**set metric 10**
- Use the **no** version to delete the set clause from a route map.
- See *set metric*.

Missing MED Values

By default, a route that arrives with no MED value is treated as if it had a MED of 0, the most preferred value. You can use the **bgp bestpath missing-as-worst** command to specify that a route with any MED value is always preferred to a route that is missing the MED value.

bgp bestpath missing-as-worst

- Use to set a missing MED value to infinity, the least preferred value.
- After issuing this command, a route missing the MED is always preferred less than any route that has a MED configured.
- Example

  host1(config-router)#**bgp bestpath missing-as-worst**

- Changes apply automatically whenever BGP subsequently runs the best-path decision process for a destination prefix; that is, whenever a best route is picked for a given prefix.

  To force BGP to run the decision process on routes already received, you must use the **clear ip bgp** command to perform an inbound soft clear or hard clear of the current BGP session.

- Use the **no** version to restore the default condition, where a missing MED value is set to 0, the most preferred value.
- See *bgp bestpath missing-as-worst*.

Comparing MED Values Within a Confederation

A BGP speaker within a confederation of sub-ASs might need to compare routes to determine the best path to a destination. By default, BGP does not use the MED value when comparing routes originated in different sub-ASs within the confederation to which the BGP speaker belongs. (Within the confederation, routes learned from different sub-ASs are treated as having originated in different places.) You can use the **bgp bestpath med confed** command to force MED values to be taken into account within a confederation.

bgp bestpath med confed

- Use to specify that BGP take into account the MED when comparing routes originated in different sub-ASs within the confederation to which the BGP speaker belongs.
- This command does not affect the comparison of routes that are originated in other ASs and does not affect the comparison of routes that are originated in other confederations.
- Example

  host1(config-router)#**bgp bestpath med confed**

- Changes apply automatically whenever BGP subsequently runs the best-path decision process for a destination prefix; that is, whenever a best route is picked for a given prefix.

  To force BGP to run the decision process on routes already received, you must use the **clear ip bgp** command to perform an inbound soft clear or hard clear of the current BGP session.

- Use the **no** version to restore the default, where the MED is not taken into account.

Suppose a BGP speaker has three routes to prefix 10.10.0.0/16:

- Route 1 is originated by sub-AS 1 inside the confederation.
- Route 2 is originated by sub-AS 2 inside the confederation.
- Route 3 is originated by AS 3 outside the confederation.
- See *bgp bestpath med confed*.

BGP compares these routes to each other to determine the best path to the prefix. If you have issued the **bgp bestpath med confed** command, BGP takes into account the MED when comparing Route 1 with Route 2. However, BGP does not take into account the MED when comparing Route 3 with either Route 1 or Route 2, because Route 3 originates outside the confederation.

Capability Negotiation

The router accepts connections from peers that perform capability negotiation. Capabilities are negotiated by means of the open messages that are exchanged when the session is established. The router supports the following capabilities:

- Cisco-proprietary route refresh—Capability code 128
- Cooperative route filtering—Capability code 3
- Deprecated dynamic capability negotiation—Capability code 66
- Dynamic capability negotiation—Capability code 67
- Four-octet AS numbers—Capability code 65
- Graceful restart—Capability code 64
- Multiprotocol extensions—Capability code 1
  - address family IPv4 unicast—AFI 1 SAFI 1
  - address family IPv4 multicast—AFI 1 SAFI 2
  - address family IPv4 unicast and multicast—AFI 1 SAFI 3
  - address family VPN-IPv4 unicast—AFI 1 SAFI 128
- Route refresh—Capability code 2

The router advertises these capabilities—except for the cooperative route filtering capability—by default. You can prevent the advertisement of specific capabilities with the **no neighbor capability** command. You can also use this command to prevent all capability negotiation with the specified peer.

NOTE: The graceful restart capability is controlled with the **bgp graceful-restart** and **neighbor graceful-restart** commands rather than the **neighbor capability** command. However, the **no neighbor capability** command will prevent negotiation of the graceful restart capability.

Cooperative Route Filtering

The cooperative route filtering capability—also referred to as outbound route filtering (ORF)—enables a BGP speaker to send an inbound route filter to a peer and have the peer install it as an outbound filter on the remote end of the session.

You must specify both the type of inbound filter (ORF type) and the direction of ORF capability. The router currently supports prefix-lists as the inbound filter sent by the BGP speaker. The inbound filter sent by the BGP speaker can be a prefix list or a Cisco proprietary prefix list. The BGP speaker must indicate whether it will send inbound filters to peers, accept inbound filters from peers, or both. The router supports both standard and Cisco-proprietary orf messages.

Dynamic Capability Negotiation

If both peers acknowledge support of dynamic capability negotiation, then at any subsequent point after the session is established, either peer can send a capabilities message to the other indicating a desire to negotiate another capability or to remove a previously negotiated capability.

The data field of the capability message contains a list of all the capabilities that can be dynamically negotiated. In earlier versions, now deprecated, the data field did not carry this information. Use the **dynamic-capability-negotiation** keyword to include the list. Use the **deprecated-dynamic-capability-negotiation** keyword to exclude sending the list.

Nondynamic capability negotiation is supported for the cooperative route filtering, four-octet AS numbers, deprecated dynamic capability negotiation, and dynamic capability negotiation capabilities. Dynamic capability negotiation of these capabilities is not supported.

If both sides of the connection advertise support for the new dynamic capability negotiation capability, then the peers negotiate which capabilities are dynamic and which are not.

If both sides of the connection advertise support only for the deprecated dynamic capability negotiation, then the BGP speaker uses dynamic capability negotiation for all capabilities that allow it without attempting to negotiate this with the peer.

Four-Octet AS Numbers

BGP speakers that support four-octet AS and sub-AS numbers are sometimes referred to as “new” speakers. The four-octet AS numbers are employed by the AS-path and aggregator attributes. “Old” speakers are those that do not support the four-octet numbers.

Two new transitional optional attributes, new-as-path and new-aggregator, are used to carry the four-octet numbers across the old speakers. A new speaker communicating with an old speaker will send the new attributes with the four-octet numbers for locally-originated and propagated routes. The old speaker propagates the new attributes for received routes. The new speaker also sends the AS-path and aggregator attributes with two-octet numbers; any AS number greater than 65535 is replaced with a reserved AS number, 23456.

Graceful Restarts

When BGP restarts on a router, all of the router’s BGP peers detect that the BGP session transitioned from up to down. The transition causes a routing flap throughout the network as the peers recalculate their best routes in light of the loss of routes from that peering session.

The BGP graceful restart capability reduces the network disruption that normally results from a peer session going down. If the session is with a peer that had previously advertised the graceful restart capability, the receiving BGP speaker marks all routes from that peer in the BGP routing table as stale. BGP keeps these stale routes for a limited time and continues to use these routes to forward traffic. Any existing stale routes from that peer are deleted to account for consecutive restarts.

When the restarting peer reestablishes the session, the receiving BGP speaker replaces the stale routes with the fresh routes it receives from the peer. The restarted peer sends an End-of-RIB marker to signal when it has finished sending all its routes to the BGP speaker. Until this point, BGP has still been using the stale routes to forward traffic. Upon receipt of the End-of-RIB marker, the BGP speaker flushes any remaining stale routes from the restarted peer.

The End-of-RIB marker is an update message that contains no advertised or withdrawn prefixes; it is sent only to BGP speakers that have previously advertised the graceful restart capability.

The receiving speaker also sends its own routes to the restarted speaker, and sends an End-of-RIB marker when it completes the update. The restarted peer defers reinitiating the BGP best-path selection process until it has received this marker from all peers with which it had a session in the established state and from which it had received an End-of-RIB marker before it restarted.

After running the selection process to pick the best route to all prefixes using the fresh routes, BGP then installs the best routes in the IP routing table on the restarted peer. Any of these that are best overall routes to a prefix are then pushed by the router to the forwarding tables on the line modules.

By waiting for all restarted peers to send the End-of-RIB marker, BGP risks delaying the initiation of the best path decision process indefinitely due to a single very slow peer. For a specific peer, you can avoid this delay by hard clearing the peer or issuing the **clear ip bgp wait-end-of-rib** command. Either method removes that peer from the set of peers for which BGP is awaiting an End-of-RIB marker. Alternatively, you can minimize this effect by using the **bgp graceful-restart path-selection-defer-time-limit** command to specify a maximum period that the restarted peer waits for the marker from its peers.

Note that the receiving peer does not defer its best-path selection process while waiting for a restarted peer to reestablish a session. The receiving peer continues to use the stale routes from the restarted peer in the decision process. When it flushes stale routes, the receiving peer then uses the freshly updated routes.

A restarting peer must bring the session back up and refresh its routes within a limited period, or BGP on the receiving peer will flush all the stale routes. When a BGP speaker advertises the graceful restart capability, it also advertises how long it expects to take to reestablish a session if it restarts. If the session is not reestablished within this restart period, the speaker's peers flush the stale routes from the speaker. You can use the **bgp graceful-restart restart-time** command to modify the restart period advertised to all peers; the **neighbor graceful-restart restart-time** command modifies the restart period advertised to specific peers or peer groups. A receiving peer starts the timer as soon as it recognizes that the session with the restarting peer has transitioned to down.

The receiving peer also has a configurable timer that starts when it recognizes that the session with the restarting peer has gone down. The **bgp graceful-restart stalepaths-time** command determines how long a receiving peer is willing to use stale paths from any restarted peer; the **neighbor graceful-restart stalepaths-time** command does the same for a specified restarted peer or peer group. If the receiving peer does not receive an End-of-RIB marker from the restarted peer before the stalepaths timer expires, the receiving peer flushes all stale routes from the peer.

In this release, BGP supports the graceful restart capability to inform peers that the forwarding state for IPv6 address families, namely unicast, multicast, VPN unicast, and unicast labeled subsequent address family identifiers (SAFIs), can be preserved during a stateful SRP switchover. MPLS also provides high availability support for IPv6 by preserving the MPLS state for IPv6 interfaces during a stateful SRP switchover. This capability of MPLS enables BGP to support graceful restart for IPv6 labeled address families. During a restart, BGP acts as a restarting speaker for the IPv6 unlabeled and labeled address-families. The function of BGP as a graceful restart helper for both IPv4 and IPv6 address families had been available in lower-numbered releases, and there is no change to this functionality in this release.

NOTE: **The function of BGP as a graceful restart helper for both IPv4 and IPv6 address families had been available in lower-numbered releases, and there is no change to this functionality in this release.**

bgp graceful-restart

- Use to enable the BGP graceful restart capability.
- Advertisement of the graceful restart capability is disabled by default.
- The **no neighbor capability negotiation** command prevents the advertisement of all BGP capabilities, including graceful restart, to the specified peers.
- This command takes effect immediately and automatically bounces the session.

- Example

  host1(config-router)#**bgp graceful-restart**

- Use the **no** version to disable advertisement of the graceful restart capability. Use the **default** version to restore the default condition, of not advertising this capability.
- See *bgp graceful-restart*.

bgp graceful-restart path-selection-defer-time-limit

- Use to set the maximum time a restarted BGP speaker defers reinitiating the best-path selection process.
- This command takes effect immediately and automatically bounces the session.
- Example

  host1(config-router)#**bgp graceful-restart path-selection-defer-time-limit 180**

- Use the **no** version to restore the default value, 120 seconds.
- See *bgp graceful-restart path-selection-defer-time-limit*.

bgp graceful-restart restart-time

- Use to set the time BGP advertises to all peers within which it expects to reestablish a session after restarting. Peers flush stale routes from the speaker if the session is not restarted within this period.
- Specify an interval shorter than the stalepaths time.
- This command takes effect immediately and automatically bounces the session.
- Example

  host1(config-router)#**bgp graceful-restart restart-time 240**

- Use the **no** version to restore the default value, 120 seconds.
- See *bgp graceful-restart restart-time*.

bgp graceful-restart stalepaths-time

- Use to set the maximum time BGP waits to receive an End-of-RIB marker from any restarted peer before flushing all remaining stale routes from that peer. The timer begins when BGP recognizes that the peer session has gone down.
- This command prevents an excessive delay in BGP reconvergence due to a peer that brings a session back up but is slow to send fresh routes.
- Specify an interval longer than the restart time.
- This command takes effect immediately and automatically bounces the session.
- Example

  host1(config-router)#**bgp graceful-restart stalepaths-time 480**

- Use the **no** version to restore the default value, 360 seconds.
- See *bgp graceful-restart stalepaths-time*.

clear ip bgp wait-end-of-rib

- Use to clear a peer or peer group from the set of peers for which BGP is waiting to receive an End-of-RIB marker after a peer restart.
- Alternatively, performing a hard clear of a peer without this keyword has the same effect.
- This command takes effect immediately.
- Example

  host1#**clear ip bgp 192.168.1.158 wait-end-of-rib**
- There is no **no** version.
- See *clear ip bgp wait-end-of-rib*.

neighbor graceful-restart

- Use to control the advertisement of the BGP graceful restart capability for specified peers or peer groups.
- Advertisement of the graceful restart capability is disabled by default.
- The **no neighbor capability negotiation** command prevents the advertisement of all BGP capabilities, including graceful restart, to the specified peers, but does not affect global advertisement of the graceful restart capability.
- This command takes effect immediately and automatically bounces the session.
- Example

  host1(config-router)#**no neighbor 10.21.3.5 graceful-restart**
- Use the **no** version to disable advertisement of the graceful restart capability for specified peers or peer groups. Use the **default** version to remove the explicit configuration from the peer or peer group and reestablish inheritance of the capability configuration.
- See *neighbor graceful-restart*.

neighbor graceful-restart restart-time

- Use to set the time BGP advertises to specified peers or peer groups within which it expects to reestablish a session after restarting. Peers flush stale routes from the speaker if the session is not restarted within this period.
- Specify an interval shorter than the stalepaths time.
- This command takes effect immediately and automatically bounces the session.
- Example

  host1(config-router)#**neighbor graceful-restart restart-time 240**
- Use the **no** version to restore the default value, 120 seconds.
- See *neighbor graceful-restart restart-time*.

neighbor graceful-restart stalepaths-time

- Use to set the maximum time BGP waits to receive an End-of-RIB marker from the specified restarted peer or peer group before flushing all remaining stale routes from that peer. The timer begins when BGP recognizes that the peer session has gone down.
- This command prevents an excessive delay in BGP reconvergence due to a peer that brings a session back up but is slow to send fresh routes.
- Specify an interval longer than the restart time.
- This command takes effect immediately and automatically bounces the session.
- Example

  host1(config-router)#**neighbor graceful-restart stalepaths-time 480**
- Use the **no** version to restore the default value, 360 seconds.
- See *neighbor graceful-restart stalepaths-time*.

Configuring Hold Timers for Successful Graceful Restart in Scaled Scenarios

In a scaled environment, we recommend that you increase the hold timers for the following protocols to appropriate values, based on the level of complexity of the network and scaling settings, so as to enable graceful restart to be completed successfully.

- BGP
- IS-IS
- LDP
- OSPF
- RSVP

Consider a scenario in which a provider edge router, PE1, at one side of the service provider core is connected to a provider core router, P, which is a label-switched router (LSR) that carries traffic for the VPN tunnel. The core router, P, is connected to another provider edge router, PE2, which provides egress from the VPN. Both PE1 and PE2 routers communicate with customer sites through a direct connection to a customer edge (CE) device that sits at the edge of the customer site.

PE1 is configured for graceful restart and PE2 functions as the helper node, and each PE router is configured with 1500 VRFs and 1500 adjacencies. In such an environment, you need to perform the following steps:

1. On PE1, which is the restarting router, use the **hello hold-time** command in LDP Profile Configuration mode to modify the period for which an LSR maintains link hello records before another link hello is sent as 90 seconds.

   host1(config)#**mpls ldp interface profile ldp1**
   host1(config-ldp)#**hello hold-time 90**
2. On the interface that connects PE1 to the core router, P, use the **isis hello-interval** command in Interface Configuration mode to set the frequency at which the router sends hello packets on the specified interface as 30 seconds.

   host1(config-if)#**isis hello-interval 30**

3. On PE2, which is the helper router, use the **bgp graceful-restart stalepaths-time** command in Router Configuration mode to set the maximum time BGP waits to receive an End-of-RIB marker from any restarted peer before flushing all remaining stale routes from that peer as 3600 seconds.

   host1(config-router)#**bgp graceful-restart stalepaths-time 3600**

This condition can occur even in environments that are not scaled to the maximum limits and contain minimal subscriber connections or attribute definitions.

We recommend that you perform IS-IS graceful restart only with point-to-point adjacencies because of certain limitations that exist with graceful restart support for LAN interfaces. IS-IS graceful restart (nonstop forwarding) does not work on the broadcast interface when the restarting router is the designated intermediate system (DIS). Graceful restart works properly when the restarting router is not the DIS.

Route Refresh

If the router detects that a peer supports both Cisco-proprietary and standard route refresh messages, it uses the standard route refresh messages.

neighbor capability

- Use to control the advertisement of BGP capabilities to peers. Capability negotiation and advertisement of all capabilities are enabled by default.
- You can specify the **deprecated-dynamic-capability-negotiation**, **dynamic-capability-negotiation**, **four-octet-as-numbers**, **orf**, **route-refresh**, and **route-refresh-cisco** capabilities. The graceful restart capability is controlled by specific **graceful-restart** commands.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer.
- You cannot configure the **receive** direction for the **orf** capability for a peer that is a member of a peer group or for a peer.
- If you issue the **route-refresh** or **route-refresh-cisco** keywords, the command takes effect immediately. If dynamic capability negotiation was negotiated for the session, a capability message is sent to inform the peer of the new capability configuration. If dynamic capability negotiation was not negotiated for the session, the session is bounced automatically.
- If you issue the **deprecated-dynamic-capability-negotiation**, **dynamic-capability-negotiation**, **four-octet-as-numbers**, **negotiation**, or **orf** keywords, the command takes effect immediately and bounces the session.
- If the BGP speaker receives a capability message for a capability that BGP did not previously advertise in the dynamic capability negotiation capability, BGP sends a notification to the peer with the error code “capability message error” and error subcode “unsupported capability code”.

- IPv6 ORF prefix lists are not supported. Therefore you can specify an IPv6 address with the **orf** keyword only within the IPv4 address family and when you want to advertise IPv4 routes to IPv6 peers.
- Example

  host1(config-router)#**neighbor 10.6.2.5 capability orf prefix-list both**
- Use the **no** version to prevent advertisement of the specified capability or use the **negotiation** keyword with the **no** version to prevent all capability negotiation with the specified peer. Use the **default** version to restore the default, advertising the capability.
- See *neighbor capability*.

Interactions Between BGP and IGPs

Interactions between BGP and an interior gateway protocol are more likely to occur in an enterprise topology than in a service provider topology. You can also encounter interactions when configuring small test topologies. The main interaction factors are the following:

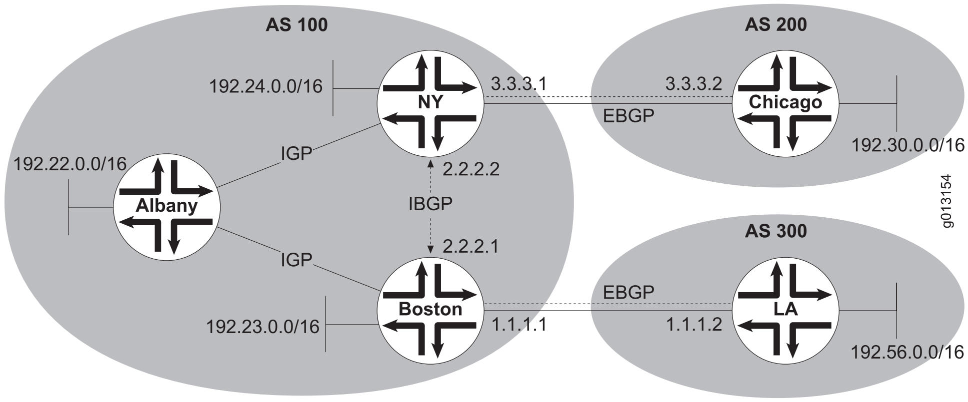
- Synchronization between BGP and IGPs
- Administrative distances for routes learned from various sources

Synchronizing BGP with IGPs

In Figure 36 on page 133, AS 100 provides transit service but does not run BGP on all of the routers in the AS. In this situation, you must redistribute BGP into the IGP so that the non-BGP routers—for example, router Albany—learn how to forward traffic to customer prefixes. If BGP converges faster than the IGP, a prefix might be advertised to other ASs before that prefix can be forwarded.

For example, suppose router LA advertises a route to router Boston using EBGP, and router Boston propagates that route to router NY using IBGP. If router NY propagates the route to router Chicago before the IGP within AS 100 has converged—that is, before router Albany learns the route—then router Chicago might start sending traffic for that route before router Albany can forward that traffic.

Figure 36: Synchronization



Synchronization solves this problem by preventing a BGP speaker from advertising a route over an EBGP session until all routers within the speaker's AS have learned about the route. If the AS contains routers connected by means of an IGP, the BGP speaker cannot propagate a BGP route that it learned from a peer until an IGP route to the prefix has been installed in the BGP speaker's IP routing table. The BGP speaker advertises the BGP route externally even if the IGP route is better than the BGP route. By contrast, if synchronization is disabled, a BGP speaker propagates a BGP route learned from a peer only if it is the best route to the prefix in the IP routing table.

Synchronization is enabled by default. However, you must configure redistribution of external routes into the IGP, or the routing tables will not receive the IGP routes.

NOTE: **When you create an address family for a VRF, synchronization is automatically disabled for that address family.**

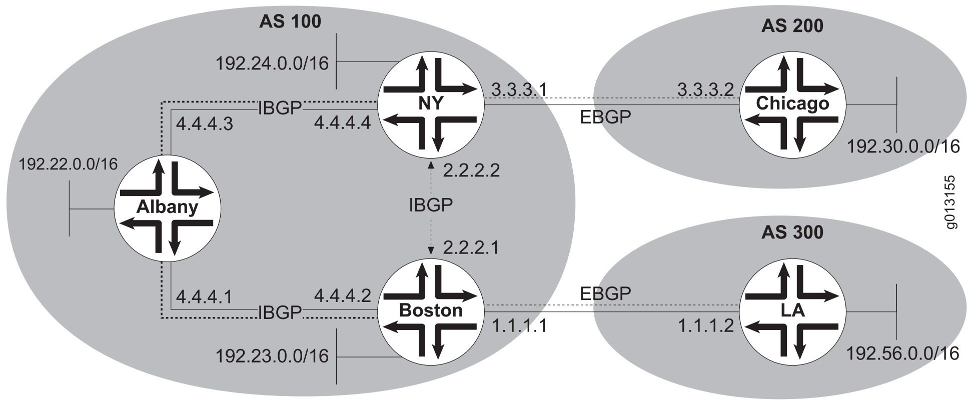
If synchronization is enabled and if redistribution is configured for the networks in Figure 36 on page 133, router NY checks its IGP routing table for a route to 192.56.0.0/16 when it learns about the prefix from the IBGP session with router Boston. If the route is not present, the prefix is not reachable through router Albany, so router NY does not advertise it as available. Router NY keeps checking its IGP routing table; if the route appears, router NY knows that it can pass traffic to the prefix and advertises the route by means of EBGP to router Chicago.

In practice, service providers rarely redistribute BGP into an IGP because existing IGPs cannot handle the full Internet routing table (about 100,00 routes). Instead, all routers in an AS typically run BGP; in these cases it is advisable to turn synchronization off everywhere.

Disabling Synchronization

Because the routes learned by means of EBGP are extensive, redistributing those routes into your IGP consumes processor and memory resources. You can disable synchronization if your AS does not pass traffic from one AS to another or if all the transit routers in your AS run BGP. Figure 37 on page 134 shows the same configuration as in the previous example, except that all the routers in AS 100 now run IBGP. As a result, all the routers receive updates learned by the area border routers from external BGP speakers.

Figure 37: Disabling Synchronization



If synchronization is disabled, a BGP speaker propagates a BGP route learned from a peer only if it is the best route to the prefix in the IP routing table. However, the speaker does advertise the routes that it originates.

The following commands show how to configure routers Boston, NY, and Chicago (shown in Figure 37 on page 134) with synchronization disabled for routers NY and Boston. The **no synchronization** command enables router NY to put the route to 192.56.0.0/16 in its IP routing table and advertise it to router Chicago without learning about 192.56.00/16 from router Albany. The command also enables router Boston to put the route to 192.30.0.0/16 in its IP routing table and advertise it to router LA without learning about 192.30.00/16 from router Albany.

To configure router Boston:

```
host1((config)#router bgp 100
host1(config-router)#neighbor 2.2.2.2 remote-as 100
host1(config-router)#neighbor 4.4.4.1 remote-as 100
host1(config-router)#neighbor 1.1.1.2 remote-as 300
host1(config-router)#no synchronization
```

To configure router NY:

```
host2(config)#router bgp 100
host2(config-router)#neighbor 3.3.3.2 remote-as 200
host2(config-router)#neighbor 2.2.2.1 remote-as 100
host2(config-router)#neighbor 4.4.4.3 remote-as 100
host2(config-router)#no synchronization
```

To configure router Albany:

```
host3(config)#router bgp 100
host3(config-router)#neighbor 4.4.4.4 remote-as 100
host3(config-router)#neighbor 4.4.4.2 remote-as 100
host3(config-router)#no synchronization
```

To configure router Chicago:

```
host4(config)#router bgp 200
host4(config-router)#neighbor 3.3.3.1 remote-as 100
```

To configure router LA:

```
host5(config)#router bgp 300
host5(config-router)#neighbor 1.1.1.1 remote-as 100
host5(config-router)#network 192.56.0.0
```

synchronization

- Use to enable and disable synchronization between BGP and an IGP.
- Synchronization is enabled by default. However, creating an address family for a VRF automatically disables synchronization for that address family.
- This command takes effect immediately.
- Use the **no** version to advertise a route without waiting for the IGP to learn a route to the prefix.
- See *synchronization*.

Setting the Administrative Distance for a Route

The administrative distance is an integer in the range 0–255 that is associated with each route known to a router. The distance represents how reliable the source of the route is considered to be. A lower value is preferred over a higher value. An administrative distance of 255 indicates no confidence in the source; routes with this distance are not installed in the routing table. As shown in Table 21 on page 135, default distances are provided for each type of source from which a route can be learned.

Table 21: Default Administrative Distances for Route Sources

Route Source	Default Distance
Connected interface	0
Static route	1
External BGP	20
OSPF	110
IS-IS	115
RIP	120
Internal BGP	200
Unknown	255

If the IP routing table contains several routes to the same prefix—for example, an OSPF route and an IBGP route—the route with the lowest administrative distance is used for forwarding.

By default, BGP propagates received BGP routes to EBGP routes only if the BGP route is used for forwarding traffic—that is, if it is the route with the lowest administrative distance in the IP forwarding table. However, you can modify this behavior by using the **bgp advertise-inactive** command. See "Advertising Inactive Routes" on page 61 for more information.

You can use the **distance bgp** command to configure the administrative distance associated with routes. If you choose to set an administrative distance, you must specify a value for all three of the following types of routes:

- external—Administrative distance for BGP external routes. External routes are routes for which the best path is learned from a BGP peer external to the AS. Acceptable values are from 1 to 255. The default value is 20.
- internal—Administrative distance for BGP internal routes. Internal routes are those routes that are learned from a BGP peer within the same AS. Acceptable values are from 1 to 255. The default value is 200.
- local—Administrative distance for BGP local routes. Local routes are those routes locally originated by BGP. BGP can locally originate routes if you issue the **network** command, if you configure redistribution into BGP, or by means of a non-AS-set aggregate route. Acceptable values are from 1 to 255. The default value is 200.

CAUTION: **Changing the administrative distance of BGP internal routes is considered dangerous and is not recommended. One problem that can arise is the accumulation of routing table inconsistencies, which can break routing.**

You can use the **distance bgp** command to configure these preferences. The following commands leave the internal distance at 200, set the external distance to 150, and set the local distance to 80:

```
host1(config)#router bgp 100
host1(config-router)#network 172.28.0.0
host1(config-router)#neighbor 156.128.5.5 remote-as 310
host1(config-router)#neighbor 142.132.1.1 remote-as 50
host1(config-router)#distance bgp 150 200 80
```

distance bgp

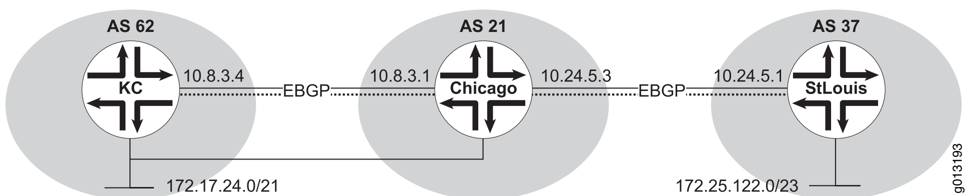
- Use to set the administrative distance for all BGP routes.
- You must specify the following:
  - *external-distance*—Administrative distance for routes external to the AS in the range 1–255. The default is 20.
  - *internal-distance*—Administrative distance for routes internal to the AS in the range 1–255. The default is 200.

- *local-distance*—Administrative distance for local routes in the range 1–255. The default is 200.

- The default value is 20 for external routes, 200 for internal route, and 200 for local routes.
- The new distance is applied to all routes that are subsequently placed in the IP routing table. To apply the new distance to routes that are already present in the IP routing table, you must use the **clear ip routes *** command to reinstall BGP routes in the IP routing table.
- Use the **no** version to return the distances to their default values, 20, 200, and 200.
- See *distance bgp*.

Example 1 Routes learned from other sources can be preferred to routes learned by means of BGP. Consider the network structure shown in Figure 38 on page 137.

Figure 38: Administrative Distances



Suppose router KC originates 172.17.24.0/21 and advertises the route to router Chicago by means of EBGP. Both router KC and router Chicago are directly connected to the network represented by 172.17.24.0/21. If you issue the **show ip route** command on router Chicago, the BGP route does not appear. Instead, only the connected route is displayed.

Both routes are in the IP routing table, but the **show ip route** command displays only the *best* route. (Use the **show ip route all** command to display all best routes; in this case the BGP route and the connected route.) Connected routes have a default distance of 0. Routes learned by means of EBGP have a default value of 20. The connected route is a better route than the EBGP route and appears in the command display.

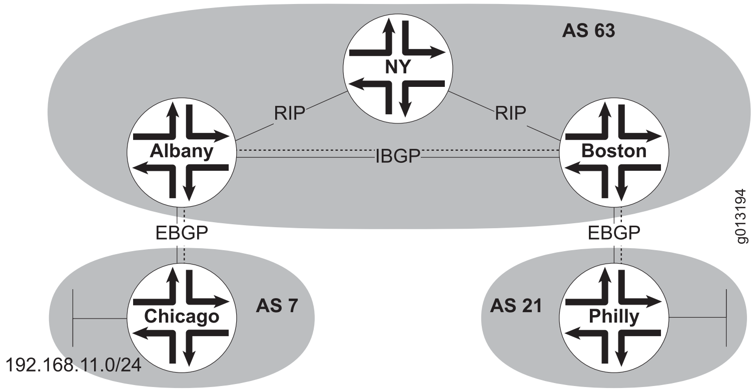
In practice, if two BGP peers are connected to the same network, both peers should originate the route.

Example 2 Consider the network structure shown in Figure 39 on page 138. Router Chicago originates prefix 192.168.11.0/24 and advertises it by means of EBGP to router Albany. Router Albany advertises the route to router Boston by means of IBGP.

Router Albany also redistributes the route into the interior gateway protocol RIP, which informs router NY of the route. Router NY propagates the route to router Boston by means of RIP, from which it is injected into BGP.

In this example, both router Albany and router Boston have synchronization turned on. When synchronization is on, BGP propagates a received route to EBGP peers, even if the IP forwarding table contains a non-BGP route with a better administrative distance than the BGP route. This example demonstrates why synchronization is needed.

Figure 39: Administrative Distance and Synchronization



Router Boston does *not* advertise the route externally to router Philly. At first, this is because router Boston has not yet heard about the prefix from router NY, and therefore the IGP route does not appear in router Boston's IP routing table.

BGP routes are not propagated until a route to the prefix by means of any IGP appears in the IP routing table. In other words, routers connected by means of an IGP must have a route to the prefix before a BGP speaker can advertise the route it learned from a peer.

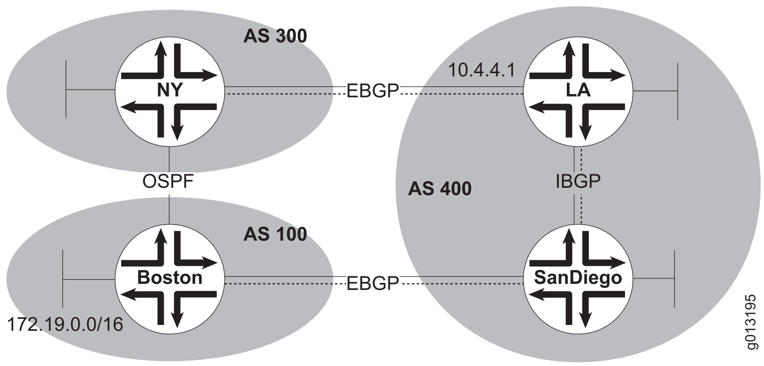
When the RIP route appears on router Boston, the router has both an IBGP route and a RIP route to the same prefix. Even though the RIP route has a better administrative distance, the IBGP route is propagated to router Philly because synchronization is turned on.

Configuring Backdoor Routes

In certain network topologies, a BGP speaker might learn routes to the same prefix from an external BGP peer and by means of an IGP protocol. Consider the network structure shown in Figure 40 on page 139.

A company has established an OSPF link between routers NY and Boston. This private link between the two routers is known as a *backdoor* link. Router NY learns two routes to prefix 172.19.0.0/16; one by means of OSPF from router Boston, and one by means of EBGP from router LA through router SanDiego. As was shown in Table 21 on page 135, EBGP routes have an administrative distance of 20 and are preferred over IGP routes, which have much higher administrative distances. In this example, the longer path by means of EBGP is preferred over the OSPF backdoor path with its distance of 110.

Figure 40: Backdoor Route



You can modify this behavior by issuing the **network backdoor** command on router NY:

```
host1(config)#router bgp 300
host1(config-router)#neighbor 10.4.4.1 remote-as 400
host1(config-router)#network 172.19.0.0 backdoor
```

Unlike the typical **network** command, **network backdoor** does not cause the BGP speaker to advertise the specified prefix. Instead, it sets the administrative distance for the EBGP path to that prefix to the same value as a route learned by means of IBGP. That is, the EBGP administrative distance is changed from the highly preferred value of 20 to the highly unpreferred value of 200. In Figure 40 on page 139, this change in value results in the backdoor OSPF being more preferred as a way to reach prefix 172.19.0.0/16.

network backdoor

- Use to cause a backdoor IGP route to be preferred over an EBGP route to the same prefix by setting the administrative distance of the EBGP route to that of an IBGP route, 200.
- Issuing this command does not cause the BGP speaker to advertise the specified route.
- This command takes effect immediately.
- Example

  ```
  host1(config-router)#network 10.53.42.0 backdoor
  ```
- Use the **no** version to restore the default distance to the EBGP route, 20.
- See *network*.

Setting the Maximum Number of Equal-Cost Multipaths

You can use the **maximum-paths** command to specify the number of equal-cost paths to the same destination that BGP can submit to the IP routing table.

If you set the value to 1, the router installs the single best route in the IP routing table. If you set the value greater than 1, the router installs that number of parallel routes.

By default, for IBGP routes, unequal-cost multipaths are installed in the routing table. You must use the **equal-cost** keyword with the **maximum-paths ibgp** command to enable equal-cost multipaths to be used in the routing table for traffic forwarding. IGP can use an asymmetric set of paths for a given destination. This functionality is called unequal-cost load balancing. Unequal-cost load balancing enables traffic to be distributed among multiple unequal-cost paths to ensure higher overall throughput and reliability.

maximum-paths

- Use to set the maximum number of equal-cost multipaths.
- Specify a value in the range 1–16; the default value is 1.
- If you do not specify a keyword, the maximum number applies only to routes learned from external peers. If you specify the **ibgp** keyword, the maximum number applies only to routes received from internal peers. If you specify the **eibgp** keyword (valid only for VRF IPv4 unicast or IPv6 unicast address families), the maximum number applies to routes received from internal peers and external peers.
- To prevent the possibility of routing loops from occurring, the IGP metric to the BGP next hop must be equal to the best-path IGP metric, unless you explicitly configure the router for unequal-cost IBGP multipath, in which case you need to manually configure settings to avoid the occurrence of routing loops. For example, a sufficient, but not necessary, condition to ensure that there is no routing loop for a given route, is that all of the IGP next hop routers have metrics that are less than or equal to the lowest local metric. The load-sharing algorithms based on hashing determines the packets that are looped. When you configure unequal-cost paths for IBGP using the **maximum-paths ibgp** command without specifying the **equal-cost** keyword, multiple IBGP routes to the same destination prefix might be installed in the routing table, although they all do not have equal IGP costs to the BGP next hop. This method of operation might cause routing loops.
- This command takes effect immediately; it does not bounce the session.
- You can specify the maximum number of equal-cost multipaths in the context of the virtual router, an IPv4 unicast or IPv6 unicast address family, or a VRF specified in the context of an IPv4 unicast or IPv6 unicast address family.
- This command is not supported for VPNv4 or VPNv6 address families.
- Example 1

  host1(config-router)#**maximum-paths 3**
- Example 2

  host1:vr1(config-router-af)#**maximum-paths ibgp 5**
- Use the **no** version to restore the default value, 1.
- See *maximum-paths*.

Detecting Peer Reachability with BFD

You can configure a Bidirectional Forwarding Detection (BFD) session with a BGP neighbor or peer group to determine relatively quickly whether the neighbor or peer group is reachable. For more information on BFD, see *JunosE IP Services Configuration Guide*. BFD is supported only for single-hop IBGP and EBGP sessions with either IPv4 or IPv6 neighbors in the core or within a VRF. BFD is not supported for multi-hop BGP sessions (IBGP multi-hop or EBGP multi-hop). BFD behavior is identical for IBGP and EBGP single-hop sessions, and for IPv4 and IPv6 neighbors.

When you configure BFD for a BGP session, the normal BGP keepalive mechanism is not disabled. Unless you configure BGP not to do so, BGP still sends keepalive messages and brings the BGP session down if the holdtimer expires.

When you configure this feature, BGP requests BFD to start a BFD protocol session as soon as the BGP session enters the established state. BGP allows the BFD protocol session to come up only when the source address of received BFD packets matches the destination address of the BGP neighbor. When the BFD protocol session comes up, BGP logs this event and reports the session in subsequent **show** commands. If the BFD protocol session goes down, BGP immediately brings down the BGP session and takes all associated actions.

Whenever a BGP session leaves the established state, BGP requests BFD to stop the BFD protocol session. BGP also requests BFD to bring the BFD protocol session down and inform BGP if the local interface goes down.

To enable a BGP session to come up even if the remote peer does not support BFD or has not been configured to use BFD, the following behavior applies:

- The BGP session can come up when the BFD protocol session is not yet up.
- The BGP session can stay up even when the BFD protocol session never comes up.

You can specify a desired rate for receiving BFD packets from the peer, transmitting them to the peer, or both, by setting a desired time interval between the packets. The actual timer values can be different as a result of other applications requesting BFD protocol sessions on the same interface with different timer values, as a result of timer value negotiation between the local and remote BFD speakers, or both.

In the following example, the router is configured to send BFD packets to peer 10.25.43.1 with a minimum interval of 450 milliseconds between the packets, and to accept BFD packets from the peer only with the same minimum interval:

```
host1(config)#router bgp 100
host1(config-router)#neighbor 10.25.43.1 bfd-liveness-detection minimum-interval 450
```

neighbor bfd-liveness-detection

- Use to enable BGP to detect whether a neighbor is unreachable by means of a BFD protocol session to the neighbor.
- The peers in a BGP adjacency use the configured values to negotiate the actual transmit intervals for BFD packets.
  - You can use the **minimum-transmit-interval** keyword to specify the interval at which the local peer proposes to transmit BFD control packets to the remote peer. The default value is 300 milliseconds.

  - You can use the **minimum-receive-interval** keyword to specify the minimum interval at which the local peer must receive BFD control packets from the remote peer. The default value is 300 milliseconds.
  - You can use the **minimum-interval** keyword to specify the same value for both of those intervals. Configuring a minimum interval has the same effect as configuring the minimum receive interval and the minimum transmit interval to the same value. The default value is 300 milliseconds.

- You can use the **multiplier** keyword to specify the detection multiplier value. The calculated BFD liveness detection interval can be different on each peer. The multiplier value is roughly equivalent to the number of packets that can be missed before the BFD session is declared to be down. The default value is 3.
- For details on liveness detection negotiation, see *JunosE IP Services Configuration Guide*.
- You can change the BFD liveness detection parameters at any time without stopping or restarting the existing session; BFD automatically adjusts to the new parameter value. However, no changes to BFD parameters take place until the values resynchronize with each peer.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer.
- This command takes effect immediately.
- The BGP session does not flap when you enable BFD for a session that is already up or change the BFD timer values for an established session.
- If you remove the BFD configuration while the BGP sessions and the BFD protocol session are up, BFD moves to the Admin Down state and communicates the change to the peer to enable the client protocols to handle this in a seamless manner without going down. For the Admin Down state to work, the peer, which receives the Admin Down state notification, must have the capability to distinguish between administratively down state and real link down.

> NOTE: The BFD Admin Down state is used to bring down a BFD session administratively, to protect client applications from BFD configuration removal, license issues, and clearing of BFD sessions.

- Use the **no** version to disable BFD liveness detection for the neighbor. Use the **default** version to remove the explicit configuration from the peer or peer group and reestablish inheritance of the feature configuration.
- See *neighbor bfd-liveness-detection*.

BFD and BGP Graceful Restart

So that BFD can maintain its BFD protocol sessions across a BGP graceful restart, BGP requests that BFD set the C bit to 1 in transmitted BFD packets. When the C bit is set to

1, BFD can maintain its session in the forwarding plane in spite of disruptions in the control plane. Setting the bit to 1 gives BGP neighbors acting as a graceful restart helper the most accurate information about whether the forwarding plane is up.

When BGP is acting as a graceful restart helper and the BFD session to the BGP peer is lost, one of the following actions takes place:

- If the C bit received in the BFD packets was 1, BGP immediately flushes all routes, determining that the forwarding plane on the BGP peer has gone down.
- If the C bit received in the BFD packets was 0, BGP marks all routes as stale but does not flush them because the forwarding plane on the BGP peer might be working and only the control plane has gone down.

Managing a Large-Scale AS

BGP requires that IBGP peers be fully meshed, creating significant routing overhead as the number of peers increases. The number of IBGP sessions increases rapidly with the number of routers:

For example, in an AS with 9 BGP peers, the peers can conduct 36 sessions:

$$\text{IBGP sessions} = \frac{(\text{number of BGP peers in the AS})^2 - (\text{number of BGP peers in the AS})}{2}$$

$$\text{IBGP sessions} = \frac{9^2 - 9}{2} = 36$$

BGP provides the following two alternative configuration strategies to reduce the number of fully meshed peers:

- Configure confederations.
- Configure route reflectors.

Both of these strategies are complex and can create their own problems. Neither strategy is typically used unless the mesh of IBGP peers approaches 100 sessions per peer.

Configuring a Confederation

IBGP requires that BGP speakers within an AS be fully meshed. You can reduce the IBGP mesh inside an AS by subdividing the AS into a confederation of sub-ASs. Each sub-AS must be fully meshed internally, but the sub-ASs do not have to be fully meshed with each other. Confederations are most useful when the number of IBGP speakers within an AS increases to the point that each router has about 100 peering sessions.

Figure 41 on page 144 shows a simpler topology. AS 29 consists of 10 fully meshed IBGP peers (for clarity, only the BGP sessions are shown). Border router Salem has an EBGP session with a neighbor in AS 325. Border router Boston has an EBGP session with a neighbor in AS 413.

Figure 41: A Fully Meshed Autonomous System

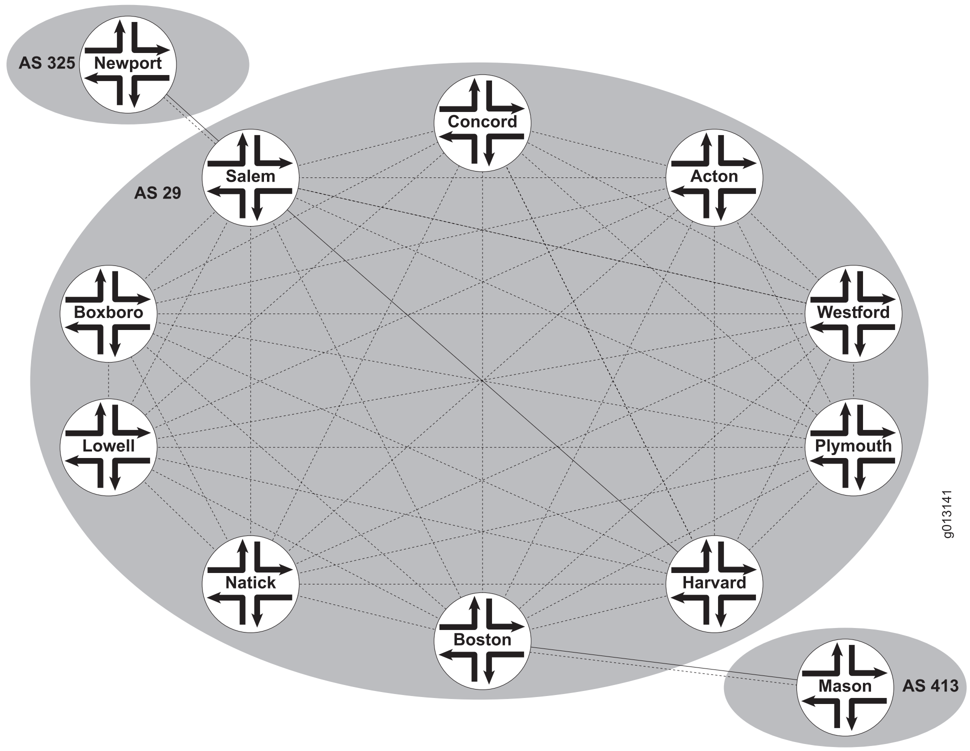
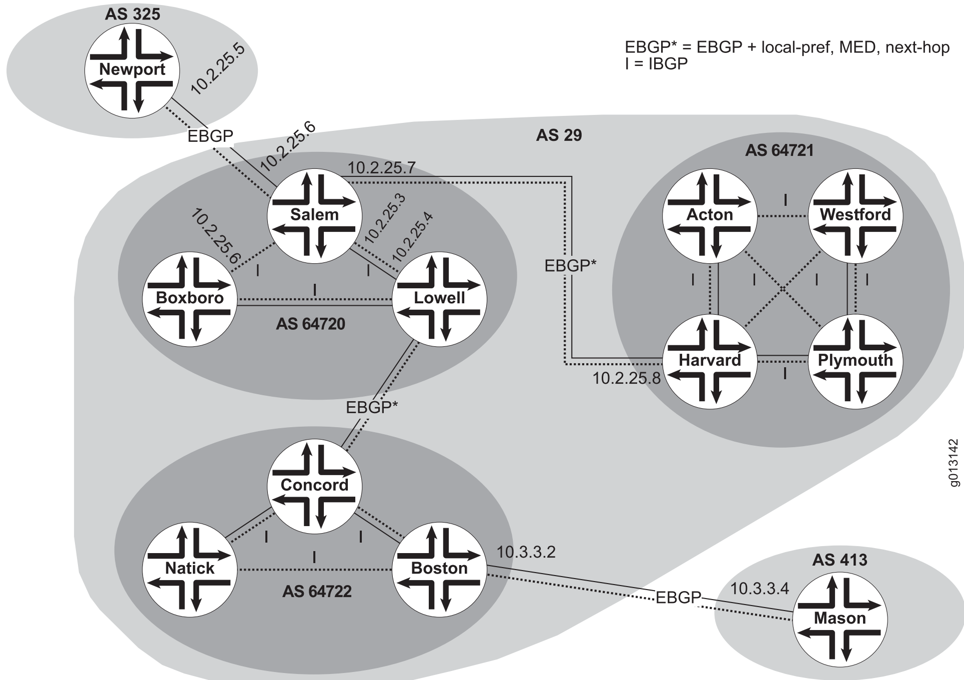


Figure 42 on page 145 illustrates how you can create three sub-ASs within AS 29 to greatly reduce the number of peering sessions. According to common practice, use a number from the private range of AS numbers—from 64512 to 65535—to identify each sub-AS. AS 29 is now a confederation of three sub-ASs: AS 64720, AS 64721, and AS 64722. Each sub-AS consists of fully meshed IBGP peers. A slightly modified version of EBGP runs between the sub-ASs: It acts like IBGP within an AS because the local-pref, MED, and next-hop attributes are preserved across the sub-AS boundaries. To the external neighbors, AS 29 appears the same as it ever was.

Figure 42: A Confederation of Subautonomous Systems



The following commands partially configure router Salem:

```
host1(config)#router bgp 64720
host1(config-router)#bgp confederation identifier 29
host1(config-router)#bgp confederation peers 64721 64722
host1(config-router)#neighbor 10.2.25.4 remote-as 64720
host1(config-router)#neighbor 10.2.25.8 remote-as 64721
host1(config-router)#neighbor 10.2.25.2 remote-as 325
```

The **bgp confederation identifier** command establishes router Salem as a member of Confederation 29. The **bgp confederation peers** command specifies that sub-AS 64721 and sub-AS 64722 are members of the same confederation as the sub-AS that includes router Salem. The **neighbor remote-as** commands specify the IBGP connection with a neighbor in sub-AS 64720 and the EBGP connections with neighbors in sub-AS 64721 and outside the confederation in AS 325.

Similarly, the following commands partially configure router Harvard:

```
host2(config)#router bgp 64721
host2(config-router)#bgp confederation identifier 29
host2(config-router)#bgp confederation peers 64720 64722
host2(config-router)#neighbor 10.2.25.7 remote-as 64720
```

From router Newport's perspective, router Salem is simply a member of AS 29:

```
host3(config)#router bgp 325
host3(config-router)#neighbor 10.2.25.6 remote-as 29
```

From router Mason's perspective, router Boston is simply a member of AS 29:

```
host4(config)#router bgp 413
```


host4(config-router)#**neighbor 10.3.3.2 remote-as 29**

bgp confederation identifier

- Use to establish a router as a member of the specified BGP confederation.
- To routers outside the confederation, the confederation appears as an autonomous system with an AS number the same as the confederation identifier.
- The new confederation identifier is used in open messages and in the AS path in update messages that are sent after you issue the command.

  To force sessions that are already up to use the new confederation identifier, you must use the **clear ip bgp** command to perform a hard clear.
- Use the **no** version to remove the sub-AS from the confederation.
- See *bgp confederation identifier*.

bgp confederation peers

- Use to enable EBGP sessions with routers in the peer sub-ASs; the EBGP sessions preserve local-pref, MED, and next-hop attributes.
- You can specify one or more individual sub-AS numbers, or you can issue the **filter-list** keyword and an AS-path access list (which is based on regular expressions) to specify a list of sub-AS numbers.
- If the remote AS of a peer appears in the specified list of sub-ASs or is identified by the filter list, then the peer is treated as being in the same confederation.
- This command takes effect immediately and bounces only those sessions whose peer type changed as a result of issuing the command.
- Use the **no** version to remove individually specified sub-ASs, all sub-ASs specified by the filter list, or all sub-ASs from the confederation.
- See *bgp confederation peers*.

ip bgp-confed-as-set new-format

- Use to specify that AS-confed-sets are displayed enclosed within square brackets rather than parentheses, and that the AS paths in the set are delimited by commas rather than spaces.
- Example

  host1(config)#**ip bgp-confed-as-set new-format**
- Use the **no** version to restore the default display within parentheses and with space-delimited ASs.
- See *ip bgp-confed-as-set new-format*.

Configuring Route Reflectors

Router reflection is an alternative to confederations as a strategy to reduce IBGP meshing. BGP specifies that a BGP speaker cannot advertise routes to an IBGP neighbor if the speaker learned the route from a different IBGP neighbor. A *route reflector* is a BGP speaker that advertises routes learned from each of its IBGP neighbors to its other IBGP neighbors; routes are reflected among IBGP routers that are not meshed. The route reflector's neighbors are called *route reflector clients*. The clients are neighbors only to the route reflector, not to each other. Each route reflector client depends on the route reflector to advertise its routes within the AS; each client also depends on the route reflector to pass routes to the client.

A route reflector and its clients are collectively referred to as a *cluster*. Clients peer only with a route reflector and do not peer outside their cluster. Route reflectors peer with clients and other route reflectors within the cluster; outside the cluster they peer with other reflectors and other routers that are neither clients nor reflectors. Route reflectors and nonclient routers must be fully meshed.

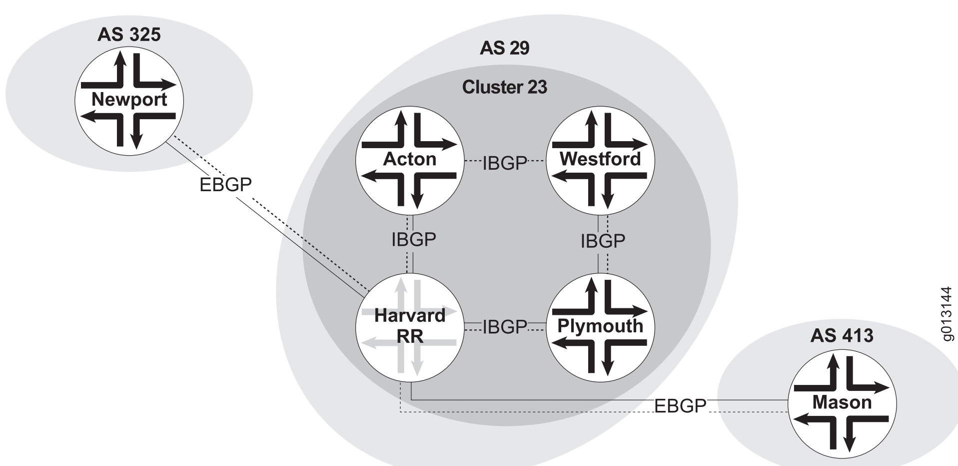
Clients and nonclients have no knowledge of route reflection; they operate as standard BGP peers and require no configuration. You simply configure the route reflectors.

Route reflectors advertise routes learned from:

- A nonclient peer only to clients
- A client peer to all nonclient peers and to all client peers except for the originator of the route
- An EBGP peer to all nonclient peers and all client peers

Figure 43 on page 147 illustrates a simple route reflection setup. Configured as a route reflector, Router Harvard reflects routes among its clients within Cluster 23: Routers Plymouth, Westford, and Acton. These route reflector clients see router Harvard and each other simply as IBGP neighbors. Router Newport in AS 325 and router Mason in AS 413 see router Harvard simply as an EBGP neighbor in AS 29.

Figure 43: Simple Route Reflection

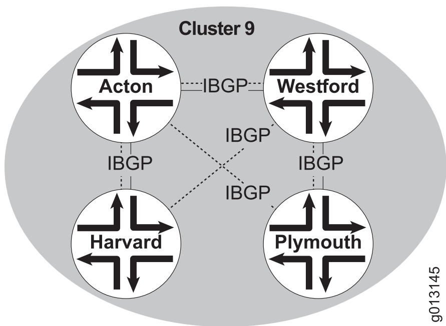


Route Reflection and Redundancy

Reliability and redundancy are important issues when using route reflection because the members of a cluster are not fully meshed. For example, if router Harvard in Figure 43 on page 147 goes down, all of its clients are isolated from networks outside the cluster. Having one or more redundant route reflectors in a cluster protects against such an occurrence.

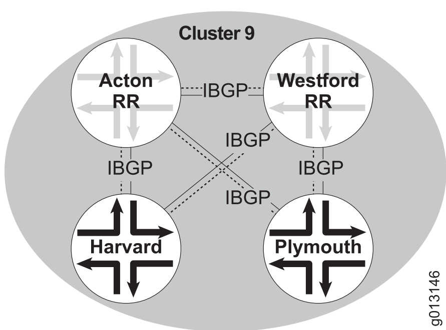
However, you cannot rely on logical redundancy alone. Consider the cluster shown in Figure 44 on page 148. The operator has attempted to provide redundancy in Cluster 9 by configuring two route reflectors, router Acton and router Westford. Unfortunately, router Harvard is physically isolated if its link to router Acton goes down, or if router Acton itself goes down. Similarly, router Plymouth is isolated if any problems develop with router Westford.

Figure 44: Route Reflection: Logical Redundancy



In Figure 45 on page 148, the operator has added physical redundancy to the cluster configuration. Now, loss of either one of the route reflectors does not isolate the reflector clients.

Figure 45: Route Reflection: Physical and Logical Redundancy



Route Reflection and Looping

BGP prevents looping *between* ASs by evaluating the AS-path attribute to determine a route's origin. Border routers reject routes they receive from external neighbors if the AS path indicates that the route originated within the border router's AS.

Route reflection creates the possibility of looping *within* an AS. Routes that originate within a cluster might be forwarded back to the cluster. Because this happens within a given AS, the AS-path attribute is of no use in detecting a loop.

Route reflectors add an *originator ID* to each route that identifies the originator of the route within the local AS by its router ID. If a router receives a route having the originator ID set to its own router ID, it rejects the route.

You can also use a *cluster list* to prevent looping. Each cluster has an identifying number, the cluster ID. For clusters with a single route reflector, the cluster ID is the router ID of the route reflector; otherwise you configure the cluster ID. The cluster list records the cluster ID of each cluster traversed by a route. When a route reflector passes a route from a client to a nonclient router outside the cluster, the reflector appends the cluster ID to the list. When a route reflector receives a route from a nonclient, it rejects the route if the list contains the local cluster ID.

What about routes that a client forwards out of the cluster? No cluster ID is needed, because clients can forward routes only to EBGP peers, that is, to peers outside the AS. Looping between ASs is prevented by the AS-path list.

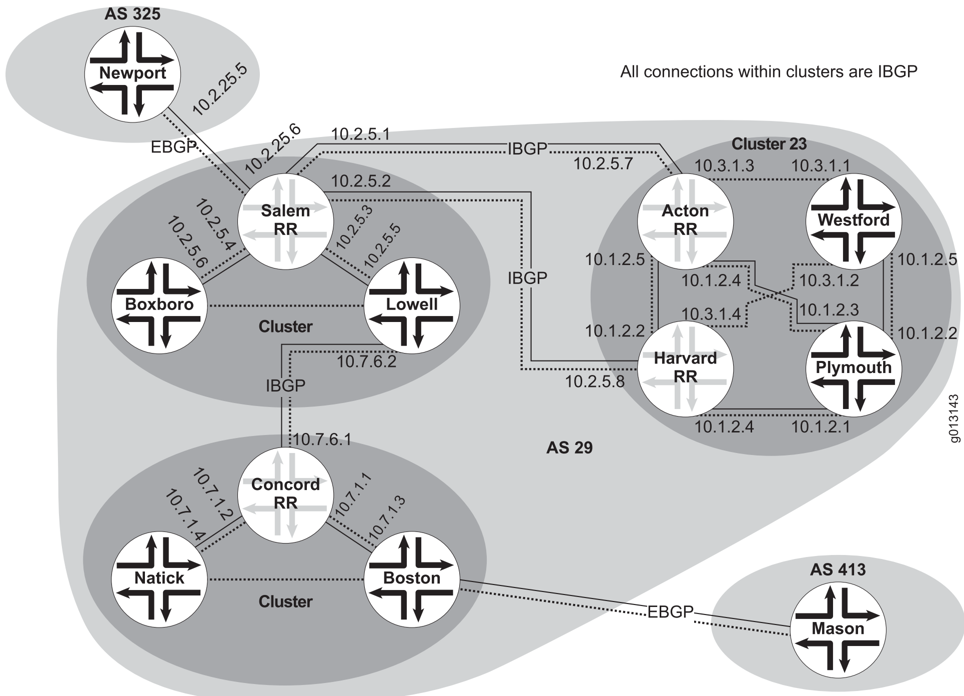
The following commands configure the route reflectors for the network topology shown in Figure 46 on page 150. You configure the other routers, whether nonclients or route reflector clients, as usual for IBGP and EBGP peers.

To configure router Salem as a route reflector:

```
host1(config)#router bgp 29
host1(config-router)#neighbor 10.2.5.5 remote-as 29
host1(config-router)#neighbor 10.2.5.5 route-reflector-client
host1(config-router)#neighbor 10.2.5.6 remote-as 29
host1(config-router)#neighbor 10.2.5.6 route-reflector-client
host1(config-router)#neighbor 10.2.5.7 remote-as 29
host1(config-router)#neighbor 10.2.5.8 remote-as 29
host1(config-router)#neighbor 10.2.25.5 remote-as 325
```

You do not configure a cluster ID, because router Salem is the only route reflector in this cluster.

Figure 46: BGP Route Reflection



To configure router Concord as a route reflector:

```
host2(config)#router bgp 29
host2(config-router)#neighbor 10.7.1.3 remote-as 29
host2(config-router)#neighbor 10.7.1.3 route-reflector-client
host2(config-router)#neighbor 10.7.1.4 remote-as 29
host2(config-router)#neighbor 10.7.1.4 route-reflector-client
host2(config-router)#neighbor 10.7.6.2 remote-as 29
```

You do not configure a cluster ID, because router Concord is the only route reflector in this cluster.

To configure router Acton as a route reflector:

```
host3(config)#router bgp 29
host3(config)#bgp cluster-id 23
host3(config-router)#neighbor 10.3.1.1 remote-as 29
host3(config-router)#neighbor 10.3.1.1 route-reflector-client
host3(config-router)#neighbor 10.1.2.3 remote-as 29
host3(config-router)#neighbor 10.1.2.3 route-reflector-client
host3(config-router)#neighbor 10.3.3.4 remote-as 29
host3(config-router)#neighbor 10.2.5.1 remote-as 29
```

You must configure a cluster ID, because router Acton and router Harvard are both route reflectors in this cluster.

To configure router Harvard as a route reflector:

```
host4(config)#router bgp 29
host4(config)#bgp cluster-id 23
```

```
host4(config-router)#neighbor 10.3.1.2 remote-as 29
host4(config-router)#neighbor 10.3.1.2 route-reflector-client
host4(config-router)#neighbor 10.1.2.1 remote-as 29
host4(config-router)#neighbor 10.1.2.1 route-reflector-client
host4(config-router)#neighbor 10.3.3.2 remote-as 29
host4(config-router)#neighbor 10.2.5.2 remote-as 29
```

You must configure a cluster ID, because router Harvard and router Acton are both route reflectors in this cluster.

bgp client-to-client reflection

- Use to reenable the reflector to reflect routes among all clients.
- Client-to-client reflection is enabled by default. If the route reflector's clients are fully meshed, you can disable reflection because it is not necessary.
- If client-to-client reflection is enabled (the default), clients of a route reflector cannot be members of a peer group.
- Example

  ```
  host1(config-router)#no bgp client-to-client reflection
  ```

- Changes apply automatically to any routes received after you issue the command. To advertise or withdraw routes that are already present in the BGP routing table, you must use the **clear ip bgp** command to issue a hard clear or an outbound soft clear.
- Use the **no** version to disable route reflection; use only if the route reflector's clients are fully meshed.
- See *bgp client-to-client reflection*.

bgp cluster-id

- Use to configure a cluster ID on the route reflectors if the BGP cluster has more than one route reflector. For clusters with a single reflector, the cluster ID is the reflector's router ID and does not have to be configured.
- You specify a cluster ID number or an IP address of a router acting as a route reflector.
- The new cluster ID is used in update messages sent after you issue the command. To force BGP to resend all routes with the new cluster ID, you must use the **clear ip bgp** command to perform a hard clear or a soft clear.
- Use the **no** version to cause BGP to use the router ID as the cluster ID.
- See *bgp cluster-id*.

neighbor route-reflector-client

- Use to configure the local router as the route reflector and the specified neighbor as one of its clients. The reflector and its clients constitute a cluster. BGP neighbors that are not specified as clients are nonclients.
- Route reflectors pass routes among the client routers.

- Route reflection eliminates the need for all IBPG peers to be fully meshed. The members of a cluster do not have to be fully meshed, but BGP speakers outside the cluster must be fully meshed.
- If client-to-client reflection is enabled (the default), clients of a route reflector cannot be members of a peer group.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command. You cannot override this inheritance for a peer group member.
- Changes apply automatically to any routes received after you issue the command. To advertise or withdraw routes that are already present in the BGP routing table, you must use the **clear ip bgp** command to issue a hard clear or an outbound soft clear.
- Use the **no** version to indicate that the neighbor is no longer a client. Use the **default** version to remove the explicit configuration from the peer or peer group and reestablish inheritance of the feature configuration.
- See *neighbor route-reflector-client*.

Configuring BGP Multicasting

The BGP multiprotocol extensions (MP-BGP) enable BGP to carry IP multicast routes used by the Protocol Independent Multicast (PIM) to build data distribution trees. (See *JunosE Multicast Routing Configuration Guide* for information about PIM.) You can configure a multicast routing topology different from your unicast topology to achieve greater control over network resources. This application of MP-BGP is often referred to as multicast BGP (MBGP).

The BGP multiprotocol extensions specify that BGP can exchange information within different types of *address families*:

- Unicast IPv4—If you do not explicitly specify the address family, the router is configured to exchange unicast IPv4 addresses by default.
- Multicast IPv4—If you specify the multicast IPv4 address family, you can use BGP to exchange routing information about how to reach a multicast source instead of a unicast destination. For information about BGP multicasting commands, see "Configuring BGP Routing" on page 3. For a general description of multicasting, see *JunosE Multicast Routing Configuration Guide*.
- VPN IPv4—If you specify the VPN-IPv4 (also known as VPNv4) address family, you can configure the router to provide IPv4 VPN services over an MPLS backbone. These VPNs are often referred to as BGP/MPLS VPNs.
- Unicast IPv6—If you specify the IPv6 unicast address family, you can configure the router to exchange unicast IPv6 routes. For a description of IPv6, see *JunosE IP, IPv6, and IGP Configuration Guide*.
- Multicast IPv6—If you specify the multicast IPv6 address family, you can use BGP to exchange routing information about how to reach an IPv6 multicast source instead of

an IPv6 unicast destination. For a general description of multicasting, see *JunosE Multicast Routing Configuration Guide*.

- VPN IPv6—If you specify the VPN-IPv6 address family, you can configure the router to provide IPv6 VPN services over an MPLS backbone. These VPNs are often referred to as BGP/MPLS VPNs.
- L2VPN—If you specify the L2VPN address family, you can configure the PE router for VPLS L2VPNs or VPWS L2VPNs to exchange layer 2 network layer reachability information (NLRI) for all VPLS or VPWS instances. Optionally, you can use the **signaling** keyword with the **address-family** command for the L2VPN address family to specify BGP signaling of L2VPN reachability information. Currently, you can omit the **signaling** keyword with no adverse effects. For a description of VPLS, see “Configuring VPLS” on page 613. For a description of VPWS, see “Configuring VPWS” on page 677.
- VPLS—If you specify the VPLS address family, you can configure the router to exchange layer 2 NLRI for a specified VPLS instance. For a description of VPLS, see “Configuring VPLS” on page 613.
- VPWS—If you specify the VPWS address family, you can configure the PE router to exchange layer 2 NLRI for a specified VPWS instance. For a description of VPWS, see “Configuring VPWS” on page 677.

As discussed in “Understanding BGP Command Scope” on page 18, BGP configuration commands fall into five categories. If you specify the multicast address family, from within the Address Family Configuration mode you can issue the commands listed in Table 7 on page 19 to configure parameters that affect the multicast address family globally. You can issue the commands listed in Table 9 on page 21 to configure a peer or peer group that you have activated in the multicast address family without affecting those configuration parameters for any other address family within which the peer or peer group is activated.

If you issue any of the commands listed in Table 8 on page 20 from within the default IPv4 unicast address family to configure a peer or peer group, you can apply those configuration values to the same entity in the multicast address family by activating the peer or peer group in the multicast address family.

Example To add a peer to the multicast routing table, first add the peer to the unicast routing table, and then copy it to the multicast routing table.

```
host1(config)#router bgp 22
host1(config-router)#neighbor 192.168.55.122 remote-as 33
host1(config-router)#address-family ipv4 multicast
host1(config-router-af)neighbor 192.168.55.122 activate
```

address-family

- Use to configure the router to exchange IPv4 or IPv6 addresses by creating the specified address family.
- IPv4 addresses can be exchanged in unicast, multicast, or VPN mode. IPv6 addresses can be exchanged in unicast mode.

- The default setting is to exchange IPv4 addresses in unicast mode from the default router.
- This command takes effect immediately.
- Examples

  host1:vr1(config-router)#**address-family ipv4 multicast**
  host1:vr1(config-router)#**address-family vpnv4**
  host1:vr1(config-router)#**address-family ipv4 unicast vrf vr2**

- Use the **no** version to disable the exchange of a type of prefix.
- See *address-family*.

exit-address-family

- Use to exit Address Family Configuration mode and access Router Configuration mode.
- Example

  host1:vr1(config-router-af)#**exit-address-family**

- There is no **no** version.
- See *exit-address-family*.

neighbor activate

- Use to specify a peer with which routes of the current address family are exchanged.
- A peer can be activated in more than one address family. By default, a peer is activated only for the IPv4 unicast address family.
- The peer must be created in unicast IPv4 or VPN IPv4 before you can activate it in another address family.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer.
- The address families that are actively exchanged over a BGP session are negotiated when the session is established.
- This command takes effect immediately. If dynamic capability negotiation was not negotiated with the peer, the session is automatically bounced so that the exchanged address families can be renegotiated in the open messages when the session comes back up.

  If dynamic capability negotiation was negotiated with the peer, BGP sends a capability message to the peer to advertise or withdraw the multiprotocol capability for the address family in which this command is issued. If a neighbor is activated, BGP also sends the full contents of the BGP routing table of the newly activated address family.

- Example

  host1:vr1(config-router-af)#**neighbor 192.168.1.158 activate**

- Use the **no** version to indicate that routes of the current address family must not be exchanged with the peer.
- See *neighbor activate*.

Monitoring BGP Multicast Services

To display values from the BGP multicast routing table, use the show BGP commands with the **ipv4 multicast** keyword. For more information about displaying BGP parameters, see "Monitoring BGP" on page 159.

Using BGP Routes for Other Protocols

You can use the **ip route-type** or **ipv6 route-type** command to specify whether BGP IPv4 or IPv6 unicast routes are available only for unicast routing protocols or for both unicast and multicast routing protocols to perform RPF checks. Routes available for unicast routing protocols appear in the unicast view of the routing table, whereas routes available for multicast routing protocols appear in the multicast view of the routing table.

Typically you use MP-BGP to learn the RPF routes for multicast protocols, especially if the topology for multicast networks differs from that for unicast networks. However, you might use this command if you do not want to run multicast MP-BGP, or if you are running BGP between CE routers in a given BGP/MPLS VPN (the current specification does not provide a way to transmit multicast MP-BGP routes across a BGP/MPLS VPN core).

ip route-type

ipv6 route-type

- Use to specify whether BGP routes are available for other unicast protocols or both unicast and multicast protocols.
- You cannot specify that BGP routes are available only for multicast protocols.
- Use the **show ip route** or **show ipv6 route** command to view the routes available for unicast protocols.
- Use the **show ip rpf-route** or **show ipv6 rpf-route** command to view the routes available for multicast protocols. It does not display routes available only to unicast protocols.
- By default, BGP IPv4 and IPv6 unicast routes are available only for other unicast routing protocols.
- Example 1

  host1(config)#**router bgp 100**
  host1(config-router)#**ipv6 route-type both**

- Example 2

  host1(config)#**router bgp 100**
  host1(config-router)#**address-family ipv4 unicast vrf v1**
  host1(config-router-af)#**ip route-type both**

- Use the **no** version to restore the default value, unicast.

- See *ip route-type*.
- See *ipv6 route-type*.

Configuring BGP/MPLS VPNs

The BGP multiprotocol extensions enable the exchange of BGP information within different types of address families. The VPN IPv4 address family enables you to configure the router to provide IPv4 VPN services over an MPLS backbone. These VPNs are often referred to as BGP/MPLS VPNs. For detailed information, see "Configuring BGP-MPLS Applications" on page 389.

Testing BGP Policies

You can analyze and check your BGP routing policies on your network before you implement the policies. Use the **test ip bgp neighbor** and **test bgp ipv6 neighbor** commands to test the outcome of a BGP policy. The commands output display the routes that are advertised or accepted if the specified policy is implemented.

BGP routes must be present in the forwarding table for this command to work properly. If you run the policy test on incoming routes, soft reconfiguration (configured with the **neighbor soft reconfiguration in** command) must be in effect.

NOTE: You can use the standard redirect operators to redirect the test output to network or local files. See *JunosE System Basics Configuration Guide*.

The output of these commands is always speculative. It does not reflect the current state of the router.

test bgp ipv6 neighbor

test ip bgp neighbor

- Use to test the effect of BGP policies on a router without implementing the policy.
- You can apply the test to routes advertised to peers or received from peers.
- You can test the following kinds of policies: distribute lists, filter lists, prefix lists, prefix trees, or route maps. If you do not specify a policy, then the test uses whatever policies are currently in effect on the router.

NOTE: If you test the current policies, the results might vary for routes learned before the current policies were activated if you did not clear the forwarding table when the policies changed.

- The following three items apply to the **test ip bgp neighbor** command only:
  - The *address-family identifier* for the route is the same as is used for identifying the neighbor.

- If you do not specify a route, the test is performed for all routes associated with the *address-family identifier*.
- Specifying only an address and mask without a route distinguisher causes all routes sharing the address and mask to be taken into account. Specifying only an address causes a best match to be performed for the route.

- If you completely specify a route with IP address, mask, and route distinguisher, the command displays detailed route information. Otherwise only summary information is shown. Use the **fields** option to select particular fields of interest.
- If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command unless it is overridden for a specific peer.
- You can set a weight value for inbound routes filtered with a filter list.
- Example

  host1#**test ip bgp neighbor 10.12.54.21 advertised-routes distribute-list boston5 fields all**
- There is no **no** version.
- See *test bgp ipv6 neighbor*.
- See *test ip bgp neighbor*.

CHAPTER 2

Monitoring BGP

This chapter describes the commands you can use to monitor and troubleshoot Border Gateway Protocol (BGP) on E Series routers.

NOTE: The E120 router and E320 router output for **monitor** and **show** commands is identical to output from other E Series routers, except that the E120 and E320 router output also includes information about the adapter identifier in the interface specifier (*slot/adapter/port*).

This chapter contains the following sections:

Setting a Baseline on All BGP Statistics

You can set a baseline for all BGP statistics.

The system implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.

To set a baseline for BGP statistics:

- Issue the **baseline ip bgp** command:

  host1#**baseline ip bgp**

Related Documentation

- *baseline ip bgp*

Enabling Display of BGP Logs

To display information about BGP logs for inbound or outbound events, or both.

- Issue the **debug ip bgp** command:

  host1#**debug ip bgp**

Related Documentation

- Disabling Display of BGP Logs on page 208
- *debug ip bgp*
- *undebug ip bgp*

Setting the Default Output Fields While Displaying Summarized Status of BGP Neighbors

Purpose Specify fields that are displayed by default by a subsequently issued **show ip bgp summary** command.

You can use the **intro** keyword to enable the display of introductory information about BGP attributes. The order in which you specify the fields has no effect on the order in which they are displayed.

Action To specify the default output fields:

```
host1:pe2(config-router)#default-fields peer remote-as state
messages-received messages-sent up-down-time
host1#show ip bgp summary
                                           Messages    Messages
Neighbor AS   State        Up/down time   Sent        Received
1.1.1.1  100  Established  00:07:55       94          92
```

Meaning Table 22 on page 161 lists the **show bgp summary** command output fields.

Table 22: show bgp summary Output Fields

Field Name	Field Description
Neighbor	BGP neighbors
AS	AS number of the peer
State	State of the connection
Up/down time	Time the connection has been up or down
Messages sent	Number of messages sent to peer
Messages received	Number of messages received from peer

Related Documentation

- Monitoring the Status of All BGP Neighbors on page 203
- *default-fields peer*

Setting the Default BGP Routing Table Output Fields

Purpose Specify fields that are displayed by default by any subsequently issued **show ip bgp** command that displays the BGP routing table.

You can use the **intro** keyword to enable the display of introductory information about BGP attributes. The order in which you specify the fields has no effect on the order in which they are displayed.

Action To specify the default output fields while displaying the BGP routes:

```
host1:pe2(config-router)#default-fields route intro next-hop med loc-pref
weight as-path
host1:pe2#show ip bgp vpnv4 all
Local BGP identifier 2.2.2.2, local AS 100
 6 routes (388 bytes)
 7 destinations (560 bytes) of which 0 have a route
 0 routes selected for route table installation
 6 path attribute entries (936 bytes)
 Local-RIB version 74. FIB version 74.

Prefix           Next-hop    MED  LocPrf  Weight  AS-path
99.99.99.11/32   1.1.1.1     1    100     0       65011
99.99.99.12/32   1.1.1.1     0    100     0       empty
99.99.99.13/32   1.1.1.1     2    100     0       empty
99.99.99.21/32   21.21.21.2  1    0               65021
99.99.99.22/32   22.22.22.2  0    32768           empty
99.99.99.23/32   23.23.23.2  2    32768           empty
```

Meaning Table 23 on page 162 lists the **show ip bgp** command output fields.

Table 23: show ip bgp Output Fields

Field Name	Field Description
Local BGP identifier	BGP router ID of the local router
local AS	Local autonomous system number
routes	Total number of routes stored in the BGP routing table and amount of memory consumed by routes. If several peers have advertised a route to the same prefix, all routes are included in this count.
destinations	Number of routes to unique prefixes stored in the BGP routing table and amount of memory consumed by routes. If several peers have advertised a route to the same prefix, only the best route is included in this count.
routes selected for route table installation	Number of routes in the BGP routing table that have been inserted into the IP routing table, plus prefixes for which there are currently no routes but which have had to be withdrawn from peers to which these prefixes may been previously advertised.
path attribute entries	Number of distinct path attributes stored in BGP's internal path attributes table. If BGP receives two routes for different prefixes but with identical path attributes, BGP will create only one entry in its internal path attribute table and share it between the two routes to conserve memory.
Local-RIB version	Number that is increased by one each time a route in that RIB is added, removed or modified.

Table 23: show ip bgp Output Fields *(continued)*

Field Name	Field Description
FIB version	Number that is increased by one each time BGP updates the routes in the IP routing table based on changes in the local RIB. The FIB version matches the local-RIB version when BGP has finished updating the routes in the IP route table. The FIB version is less than the local-RIB version when BGP is still in the process of updating the IP routing table.
Prefix	Prefix for the routing table entry
Peer	Peer from which route was learned
Next hop IP address	IP address of the next router that is used when a packet is forwarded to the destination network
MED	Multiexit discriminator for the route
LocPrf	Local preference for the route
Weight	Weight of the route
Origin	Origin of the route
AS path	AS path through which this route has been advertised
Communities	Community number associated with the route
Stale	Routes that have gone stale due to peer restart
unicast/multicast routes selected for route table installation	Number of unicast routes in the BGP routing table that have been inserted into the IP routing table that are also available for use in the multicast view of the IP routing table
unicast/multicast tunnel-usable routes selected for route table installation	Number of unicast and multicast routes in the BGP routing table that have been inserted into the IP routing table that are also available for use in the IP tunnel routing table
tunnel-only routes selected for tunnel-route table installation	Number of routes in the BGP routing table that have been inserted into the IP tunnel routing table
path attribute entries	Number of distinct path attributes stored in BGP's internal path attributes table. If BGP receives two routes for different prefixes but with identical path attributes, BGP will create only one entry in its internal path attribute table and share it between the two routes to conserve memory.
Statistics baseline set	Timestamp indicating when the statistics baseline was last set

Related Documentation

- Monitoring the BGP Routing Table on page 164
- *default-fields route*

Monitoring the AS-Path Access Lists for IP

Purpose Display information about AS-path access lists.

Action To display information about AS-path access lists:

```
host1#show ip as-path-access-list
AS Path Access List 1:
    permit  .*
AS Path Access List 2:
    deny  .*
AS Path Access List 3:
    permit  _109_
    deny  .*
AS Path Access List 4:
    permit  _109$
    deny  .*
AS Path Access List 10:
    deny  _109$
    permit  ^108_
    deny  .*
```

Meaning Table 24 on page 164 lists the **show ip as-path-access-list** command output fields.

Table 24: show ip as-path-access-list Output Fields

Field Name	Field Description
As Path Access List	Name of an AS-Path access list
permit, deny	Condition statement for routes matching the condition

Related Documentation

- *show ip as-path-access-list*

Monitoring the BGP Routing Table

Purpose Display the BGP IPv4 routing table or BGP IPv6 routing table.

The **show ip bgp** and **show bgp ipv6** commands display similar information.

Action To display information about routes in the IPv6 multicast address family:

```
host1#show bgp ipv6 multicast
Local BGP identifier 10.13.13.13, local AS 400
  4 routes (160 bytes)
  4 destinations (288 bytes) of which 4 have a route
  4 routes selected for route table installation
  3 path attribute entries (456 bytes)
  Local-RIB version 31. FIB version 31.
```

```
Status codes: > best, * invalid, s suppressed, d dampened, r rejected,
              a auto-summarized

  Prefix                 Peer            Next-hop
MED  LocPrf   Weight  Origin
  ::103.103.103.0/120     103.103.103.3   ::103.103.103.3       0
0       inc.
> 3ffe:0:0:1::/64        11.11.11.11     ::101.101.101.1       0    100
0       inc.
> 3ffe:0:0:3::/64        103.103.103.3   ::103.103.103.3       0
0       inc.
> 3ffe:0:1:1::/64        12.12.12.12     ::102.102.102.2       0    100
0       inc.
```

To display information about routes for the specified IPv4 prefix:

```
host1:pe1#show ip bgp 10.88.88.1
BGP route information for prefix 10.88.88.1/32
  Network route (best route
    Advertised to both internal and external peers
    Address Family Identifier (AFI) is ip-v4
    Subsequent Address Family Identifier (SAFI) is unicast
    Next hop IP address is 0.0.0.0 (metric 2)
    Multi-exit discriminator is 1
    Local preference is not present
    Weight is 32768
    Origin is IGP
    AS path is empty
    Extended communities empty
```

To display information about routes for the specified IPv6 prefix:

```
host1:pe1#show bgp ipv6 2001:0430::1/128

BGP route information for prefix 2001:1::1/128
  Received route learned from internal peer 2.2.2.2 (best route)
    Route placed in IP forwarding table
    Best to advertise to external peers
    Address Family Identifier (AFI) is ip-v6
    Subsequent Address Family Identifier (SAFI) is unicast
    MPLS in-label is none
    MPLS out-label is 17
    Next hop IP address is ::ffff:2.2.2.2 (metric 3)
    Multi-exit discriminator is 0
    Local preference is 100
    Weight is 0
    Origin is IGP
    AS path is 65021
```

To display information about next hop routers for VRF PE 11 in the IPv4 VPN address family:

```
host1:pe1#show ip bgp vpnv4 vrf pe11 next-hops
Indirect next-hop 11.11.11.2
    Resolution in IP route table of VR pe11
Reachable (metric 0)
IP indirect next-hop index 35
Direct next-hop ATM2/0.11 (11.11.11.2)
    Resolution in IP tunnel-route table of VR pe11
        Not reachable
      Reference count is 1
```

```
Indirect next-hop 2.2.2.2
    Resolution in IP route table of VR pe1
IP indirect next-hop index 123
Reachable (metric 100)
    Direct next-hop POS4/0 (10.10.10.1)
                POS4/1 (12.12.12.1)

Resolution in IP tunnel-route table of VR pe1
MPLS indirect next-hop index 578
Reachable (metric 100)
    Direct next-hop Push 23, POS4/0 (10.10.10.1)
                Push 43, POS4/1 (12.12.12.1)

Reference count is 1
```

To display information about routes in the route-target address family:

```
host1:pe1#show ip bgp route-target signaling
Local BGP identifier 13.13.13.13, local AS 100
  4 routes (240 bytes)
  3 destinations (228 bytes) of which 3 have a route
  3 routes selected for route tables installation
  0 unicast/multicast routes selected for route table installation
  0 unicast/multicast tunnel-usable routes selected for route table installation

  0 tunnel-only routes selected for tunnel-route table installation
  10 path attribute entries (1520 bytes)
  Local-RIB version 19. FIB version 19.
Status codes: > best, * invalid, s suppressed, d dampened, r rejected,
              a auto-summarized
  Prefix            Peer            Next-hop         MED LocPrf Weight Origin
> 0:0:0/0           12.12.12.12     12.12.12.12             100      0 IGP
> 100:100:1/96      11.11.11.11     11.11.11.11             100      0 IGP
  100:100:1/96      14.14.14.14     14.14.14.14             100      0 IGP
> 100:100:2/96      11.11.11.11     11.11.11.11             100      0 IGP
```

To display information about routes in the route-target address family corresponding to the specified RT-MEM-NLRI:

```
host1:pe1#show ip bgp route-target signaling 100:100:1/96
BGP route information for prefix 100:100:1/96
  Received route learned from internal peer 11.11.11.11 (best route)
    Route not placed in IP forwarding table
    Best to advertise to both internal and external peers
    Address Family Identifier (AFI) is ip-v4
    Subsequent Address Family Identifier (SAFI) is route-target-signaling
    Next hop IP address is 11.11.11.11 (metric 0)
    Multi-exit discriminator is not present
    Local preference is 100
    Weight is 0
    Origin is IGP
    AS path is empty
  Received route learned from internal peer 14.14.14.14
    Route not placed in IP forwarding table
    Do not advertise to any peers
    Address Family Identifier (AFI) is ip-v4
    Subsequent Address Family Identifier (SAFI) is route-target-signaling
    Next hop IP address is 14.14.14.14 (metric 0)
    Multi-exit discriminator is not present
    Local preference is 100
    Weight is 0
```



```
    Origin is IGP
    AS path is empty
```

To display information about network routes in the route-target address family:

```
host1:pe1#show ip bgp route-target signaling network
Prefix               Weight Route-map                              Backdoor
102:111:34/96                                                      No
1111111111:23:1/96                                                 No
```

To display information about network routes in the route-target address family corresponding to the specified RT-MEM-NLRI:

```
host1:pe1#show ip bgp route-target signaling network 102:111:34
Prefix               Weight Route-map                              Backdoor
102:111:34/96                                                      No
```

To display filtered information about all networks in the BGP routing table:

```
host1:pe1#show ip bgp fields peer next-hop next-hop-cost
Prefix               Peer             Next-hop       Next-hop-cost
11.11.11.11/32       3.3.3.3          3.3.3.3        Unreachable
11.11.11.11/32       4.4.4.4          4.4.4.4        Unreachable
22.22.22.22/32       3.3.3.3          3.3.3.3        Unreachable
22.22.22.22/32       4.4.4.4          4.4.4.4        Unreachable
33.33.33.33/32       3.3.3.3          3.3.3.3        Unreachable
44.44.44.44/32       4.4.4.4          4.4.4.4        Unreachable
55.55.55.55/32       0.0.0.0          0.0.0.0        0
66.66.66.66/32       6.6.6.6          6.6.6.6        Unreachable
77.77.77.77/32       57.57.57.7       57.57.57.7     1
88.88.88.88/32       57.57.57.7       57.57.57.7     1
```

To display filtered information about routes that are stale due to peer restart:

```
host1:pe1#show ip bgp fields best peer next-hop stale
Prefix               Stale Peer             Next-hop
10.22.22.1/32        stale 10.12.12.2       10.12.12.2
10.22.22.2/32        stale 10.12.12.2       10.12.12.2
10.22.22.3/32        stale 10.12.12.2       10.12.12.2
10.33.33.1/32              10.13.13.3       10.13.13.3
10.33.33.2/32              10.13.13.3       10.13.13.3
10.33.33.3/32              10.13.13.3       10.13.13.3
```

To display introductory information about BGP attributes:

```
host1:pe1#show ip bgp 0.0.0.0 /0 fields intro
Local BGP identifier 192.168.254.79, local AS 6730
  201058 routes (12063492 bytes)
  201540 destinations (15317040 bytes) of which 201058 have a route
  193909 routes selected for route tables installation
  0 unicast/multicast routes selected for route table installation
  0 unicast/multicast tunnel-usable routes selected for route table
installation
  0 tunnel-only routes selected for tunnel-route table installation
  35097 path attribute entries (5334744 bytes)
  Local-RIB version 20969483. FIB version 20969483.
  Statistics baseline set WED JUL 12 2006 10:31:53 METDST
  ...
```


To display all routes with a prefix that is equal to or more specific than the specified prefix:

```
host1#show ip bgp 12.2.0.0 255.255.0.0 longer-prefixes
Local router ID 192.168.1.153, local AS 100
  72074 paths, 72074 distinct prefixes (5189328 bytes used)
  72074 paths selected for route table installation
  21685 path attribute entries (2965327 bytes used)

  Prefix          Peer       Next-hop    MED CalPrf  Weight  Origin
> 12.2.6.0/24     10.5.0.48  10.5.0.48    100          100     IGP
> 12.2.7.0/24     10.5.0.48  10.5.0.48    100          100     IGP
> 12.2.76.0/24    10.5.0.48  10.5.0.48    100          100     IGP
> 12.2.88.0/22    10.5.0.48  10.5.0.48    100          100     IGP
> 12.2.97.0/24    10.5.0.48  10.5.0.48    100          100     IGP
> 12.2.99.0/24    10.5.0.48  10.5.0.48    100          100     IGP
> 12.2.109.0/24   10.5.0.48  10.5.0.48    100          100     IGP
> 12.2.169.0/24   10.5.0.48  10.5.0.48    100          100     IGP
```

Meaning Table 25 on page 168 lists the **show ip bgp** command output fields.

Table 25: show ip bgp Output Fields

Field Name	Field Description
Local BGP identifier	BGP router ID of the local router
local AS	Local autonomous system number
routes	Total number of routes stored in the BGP routing table and amount of memory consumed by routes. If several peers have advertised a route to the same prefix, all routes are included in this count.
destinations	Number of routes to unique prefixes stored in the BGP routing table and amount of memory consumed by routes. If several peers have advertised a route to the same prefix, only the best route is included in this count.
routes selected for route table installation	Number of routes in the BGP routing table that have been inserted into the IP routing table, plus prefixes for which there are currently no routes but which have had to be withdrawn from peers to which these prefixes may been previously advertised.
path attribute entries	Number of distinct path attributes stored in BGP's internal path attributes table. If BGP receives two routes for different prefixes but with identical path attributes, BGP will create only one entry in its internal path attribute table and share it between the two routes to conserve memory.
Local-RIB version	Number that is increased by one each time a route in that RIB is added, removed or modified.

Table 25: show ip bgp Output Fields *(continued)*

Field Name	Field Description
FIB version	Number that is increased by one each time BGP updates the routes in the IP routing table based on changes in the local RIB. The FIB version matches the local-RIB version when BGP has finished updating the routes in the IP route table. The FIB version is less than the local-RIB version when BGP is still in the process of updating the IP routing table.
Prefix	Prefix for the routing table entry
Peer	Peer from which route was learned
Next hop IP address	IP address of the next router that is used when a packet is forwarded to the destination network
MED	Multiexit discriminator for the route
LocPrf	Local preference for the route
Weight	Weight of the route
Origin	Origin of the route
AS path	AS path through which this route has been advertised
Communities	Community number associated with the route
Stale	Routes that have gone stale due to peer restart
unicast/multicast routes selected for route table installation	Number of unicast routes in the BGP routing table that have been inserted into the IP routing table that are also available for use in the multicast view of the IP routing table
unicast/multicast tunnel-usable routes selected for route table installation	Number of unicast and multicast routes in the BGP routing table that have been inserted into the IP routing table that are also available for use in the IP tunnel routing table
tunnel-only routes selected for tunnel-route table installation	Number of routes in the BGP routing table that have been inserted into the IP tunnel routing table
path attribute entries	Number of distinct path attributes stored in BGP's internal path attributes table. If BGP receives two routes for different prefixes but with identical path attributes, BGP will create only one entry in its internal path attribute table and share it between the two routes to conserve memory.
Statistics baseline set	Timestamp indicating when the statistics baseline was last set
Route-map	Indicates whether network-route filtering is enabled

Table 25: show ip bgp Output Fields *(continued)*

Field Name	Field Description
Backdoor	Indicates whether an IGP backdoor route is favored over an EBGP route
CalPrf	Calculated preference

Related Documentation

- *show bgp ipv6*
- *show ip bgp*

Monitoring Advertised BGP Routes

Purpose Display the routes in the specified neighbor's or peer group's Adj-RIBs-Out table for IPv4 or IPv6.

For peers, display the attributes associated with the route before the application of any outbound policy. For peer groups, display the attributes associated with the route after the application of any outbound policy; that is, the actual advertised attributes. Report whether the indirect next hop of a route is unreachable; if not, display the IGP cost to the indirect next hop.

The **show ip bgp advertised-routes** and **show bgp ipv6 advertised-routes** commands displays similar information.

NOTE: You must first enable storage of routes to the Adj-RIBs-Out tables with the **no rib-out disable** command or the **no neighbor rib-out disable** command. Otherwise, this command returns an error message.

Action To display routes in the specified neighbor's or peer group's Adj-RIBs-Out table for IPv4:

host1#**show ip bgp neighbors 5.72.116.1 advertised-routes**

```
Local BGP identifier 2.2.2.2, local AS 2222
  0 routes (0 bytes used), 0 distinct destinations (0 bytes used)
  0 routes selected for route table installation
  0 path attribute entries (0 bytes used)

Status codes: > best, * invalid, s suppressed, d dampened, r rejected

  Prefix              Peer            Next-hop     MED LocPrf Weight Origin
> 0.0.0.0/0          5.72.116.1      5.72.1.1              0       IGP
> 10.10.0.87/32      5.72.116.1      5.72.1.1              0       inc.
> 13.13.13.13/32     5.72.116.1      5.72.1.1              0       IGP
> 33.0.0.0/16        0.0.0.0         5.72.1.1        1      32768   inc.
> 33.0.0.0/24        0.0.0.0         5.72.1.1        1      32768   inc.
> 44.44.0.0/16       5.72.116.1      5.72.1.1              0       inc.
```

Meaning Table 26 on page 171 lists the **show ip bgp advertised-routes** command output fields.

Table 26: show ip bgp advertised-routes Output Fields

Field Name	Field Description
Local BGP identifier	BGP router ID of the local router
local AS	Local autonomous system number
routes	Total number of routes stored in the BGP routing table and amount of memory consumed by routes. If several peers have advertised a route to the same prefix, all routes are included in this count.
distinct destinations	Number of routes to unique prefixes stored in the BGP routing table. If several peers have advertised a route to the same prefix, only the best route is included in this count.
routes selected for route table installation	Number of routes in the BGP routing table that have been inserted into the IP routing table
path attribute entries	Number of distinct path attributes stored in BGP's internal path attributes table. If BGP receives two routes for different prefixes but with identical path attributes, BGP will create only one entry in its internal path attribute table and share it between the two routes to conserve memory.
Prefix	Prefix for the routing table entry
Peer	IP address of BGP peer
Next hop IP address	IP address of the next hop
MED	Multiexit discriminator for the route
LocPrf	Local preference for the route
Weight	Assigned path weight
Origin	Origin of the route

Related Documentation

- *show bgp ipv6 advertised-routes*
- *show ip bgp advertised-routes*

Monitoring BGP Aggregate Addresses

Purpose Display information about aggregate addresses.

The **show ip bgp aggregate-address** and **show bgp ipv6 aggregate-address** commands display similar information.

Action To display information about aggregate addresses:

```
host1# show bgp ipv6 aggregate-address
Prefix                AS set  Summ only  Attribute map   Advertise map     Suppress
 map
3ffe::/48             No      No         None            None              None
```

Meaning Table 27 on page 172 lists the **show bgp ipv6 aggregate-address** command output fields.

Table 27: show bgp ipv6 aggregate-address Output Fields

Field Name	Field Description
Prefix	Prefix of the aggregate address
AS set	ASs in the AS-set path
Summary only	Displays a summary of aggregate address information
Attribute map	Displays the attribute maps for aggregate addresses
Advertise map	Displays the advertise maps for aggregate addresses
Suppress map	Displays the suppressed maps for the aggregate addresses

Related Documentation

- *show bgp ipv6 aggregate-address*
- *show ip bgp aggregate-address*

Monitoring BGP Routes with Nonnatural Network Masks

Purpose Display information about routes that have nonnatural network masks. Report whether the indirect next hop of a route is unreachable; if not, display the IGP cost to the indirect next hop.

Action To display information about routes that have nonnatural network masks:

```
host1#show ip bgp cidr-only

Local BGP identifier 111.111.111.111, local AS 444
  0 routes (0 bytes used), 0 distinct destinations (0 bytes used)
  0 routes selected for route table installation
  0 path attribute entries (0 bytes used)

Status codes: > best, * invalid, s suppressed, d dampened, r rejected

  Prefix         Peer       Next-hop     MED LocPrf  Weight  Origin
  33.0.0.0/24   5.72.1.1   5.72.1.1     1           0       inc.
> 44.44.0.0/24  0.0.0.0    192.168.1.1  1           32768   inc.
```

Meaning Table 28 on page 173 lists the **show ip bgp cidr-only** command output fields.

Table 28: show ip bgp cidr-only Output Fields

Field Name	Field Description
Local BGP identifier	BGP router ID of the local router
local AS	Local autonomous system number
routes	Total number of routes stored in the BGP routing table and amount of memory consumed by routes. If several peers have advertised a route to the same prefix, all routes are included in this count.
distinct destinations	Number of routes to unique prefixes stored in the BGP routing table. If several peers have advertised a route to the same prefix, only the best route is included in this count.
routes selected for route table installation	Number of routes in the BGP routing table that have been inserted into the IP routing table
path attribute entries	Number of distinct path attributes stored in BGP's internal path attributes table. If BGP receives two routes for different prefixes but with identical path attributes, BGP will create only one entry in its internal path attribute table and share it between the two routes to conserve memory.
Prefix	Prefix for the routing table entry
Peer	IP address of BGP peer
Next hop IP address	IP address of the next hop
MED	Multiexit discriminator for the route
LocPrf	Local preference for the route
Weight	Assigned path weight
Origin	Origin of the route

Related Documentation

- *show ip bgp cidr-only*

Monitoring BGP Routes in a Community

Purpose Display all routes that are members of the specified BGP community. Report whether the indirect next hop of a route is unreachable; if not, display the IGP cost to the indirect next hop. Does not accept regular expressions.

The **show ip bgp community** and **show bgp ipv6 community** commands display similar information.

Action To display all routes that are members of the specified BGP community:

NOTE:

Specify the community number in AA:NN format:

- AA—Number that identifies the autonomous system
- NN—Number that identifies the community within the autonomous system

```
host1#show ip bgp community 999:999

Local router ID 192.168.1.153, local AS 100
  40845 paths, 40845 distinct prefixes (2940840 bytes used)
  40845 paths selected for route table installation
  13651 path attribute entries (1864908 bytes used)

  Prefix         Peer       Next-hop        MED CalPrf Weight Origin
> 24.0.0.0/12    10.5.0.48  10.5.0.48            100    100    IGP
> 24.4.252.0/22  10.5.0.48  10.5.0.48            100    100    IGP
> 24.6.0.0/23    10.5.0.48  10.5.0.48            100    100    IGP
> 24.6.11.0/24   10.5.0.48  10.5.0.48            100    100    IGP
```

Meaning Table 29 on page 174 lists the **show ip bgp community** command output fields.

Table 29: show ip bgp community Output Fields

Field Name	Field Description
Local router ID	BGP router ID of the local router
local AS	Local autonomous system number
routes	Total number of routes stored in the BGP routing table and amount of memory consumed by routes. If several peers have advertised a route to the same prefix, all routes are included in this count.
paths	Total number of routes stored in the BGP routing table. If several peers have advertised a route to the same prefix, all routes are included in this count.
distinct prefixes	Number of routes to unique prefixes stored in the BGP routing table. If several peers have advertised a route to the same prefix, only the best route is included in this count.
routes selected for route table installation	Number of routes in the BGP routing table that have been inserted into the IP routing table
path attribute entries	Number of distinct path attributes stored in BGP's internal path attributes table. If BGP receives two routes for different prefixes but with identical path attributes, BGP will create only one entry in its internal path attribute table and share it between the two routes to conserve memory.
Prefix	Prefix for the routing table entry

Table 29: show ip bgp community Output Fields *(continued)*

Field Name	Field Description
Peer	IP address of BGP peer
Next hop IP address	IP address of the next hop
MED	Multiexit discriminator for the route
CalPrf	Calculated preference for the route
Weight	Assigned path weight
Origin	Origin of the route

Related Documentation

- *show bgp ipv6 community*
- *show ip bgp community*

Monitoring BGP Community Routes in the Community List

Purpose

Display all routes that are members of communities on the specified BGP community list. Report whether the indirect next hop of a route is unreachable; if not, display the IGP cost to the indirect next hop. Accepts regular expressions.

The **show ip bgp community-list** and **show bgp ipv6 community-list** commands display similar information.

Action

To display all routes that are members of communities in the specified BGP community list:

host1#**show ip bgp community-list 1 fields peer communities**

```
Local router ID 192.168.1.153, local AS 100
  72077 paths, 72077 distinct prefixes (5189544 bytes used)
  72077 paths selected for route table installation
  21627 path attribute entries (2957324 bytes used)

Prefix              Peer            Communities
3.0.0.0/8           10.5.0.48       777:777 888:888
4.0.0.0/8           10.5.0.48       777:777 888:888
4.17.106.0/24       10.5.0.48       777:777 888:888
4.17.115.0/24       10.5.0.48       777:777 888:888
6.0.0.0/8           10.5.0.48       777:777 888:888
9.2.0.0/16          10.5.0.48       777:777 888:888
9.20.0.0/17         10.5.0.48       777:777 888:888
12.0.0.0/8          10.5.0.48       777:777 888:888
```

Meaning

Table 30 on page 176 lists the **show ip bgp community-list** command output fields.

Table 30: show ip bgp community-list Output Fields

Field Name	Field Description
Local router ID	BGP router ID of the local router
local AS	Local autonomous system number
routes	Total number of routes stored in the BGP routing table and amount of memory consumed by routes. If several peers have advertised a route to the same prefix, all routes are included in this count.
paths	Total number of routes stored in the BGP routing table. If several peers have advertised a route to the same prefix, all routes are included in this count.
distinct prefixes	Number of routes to unique prefixes stored in the BGP routing table. If several peers have advertised a route to the same prefix, only the best route is included in this count.
paths selected for route table installation	Number of routes in the BGP routing table that have been inserted into the IP routing table
path attribute entries	Number of distinct path attributes stored in BGP's internal path attributes table. If BGP receives two routes for different prefixes but with identical path attributes, BGP will create only one entry in its internal path attribute table and share it between the two routes to conserve memory.
Prefix	Prefix for the routing table entry
Peer	IP address of BGP peer
Communities	Community number in AA:NN format: • AA—Number that identifies the autonomous system • NN—Number that identifies the community within the autonomous system

Related Documentation

- *show bgp ipv6 community-list*
- *show ip bgp community-list*

Monitoring Dampened BGP Routes

Purpose

Display information about dampened routes. Report whether the indirect next hop of a route is unreachable; if not, display the IGP cost to the indirect next hop.

The **show ip bgp dampened-paths** and **show bgp ipv6 dampened-paths** commands display similar information.

Action To display information about dampened routes:

```
host1#show ip bgp dampened-paths
Local router ID 192.168.1.218, local AS 100
  Route flap dampening is enabled
  Decay half-life is 10 minutes while reachable, 20 minutes while unreachable
  Cuttoff threshold is 2000, reuse threshold is 750
  Maximum hold-down time is 20 minutes
  60 paths have active route flap histories (4560 bytes used)
  11 paths are suppressed
                                               Figure   Time until
Prefix           Peer      Status              of Merit Reuse/Remove
24.31.128.0/19    10.2.1.48 Suppressed/Reachable   2681    00:17:00
24.93.128.0/19    10.2.1.48 Suppressed/Reachable   2681    00:17:00
24.95.0.0/19      10.2.1.48 Suppressed/Reachable   2681    00:17:00
128.192.0.0/16    10.2.1.48 Available              1997    00:15:08
148.161.0.0/16    10.2.1.48 Available              1997    00:15:10
164.81.0.0/16     10.2.1.48 Available              1997    00:15:11
192.29.60.0/24    10.2.1.48 Available              1997    00:15:12
192.58.228.0/24   10.2.1.48 Available              1997    00:15:15
192.88.8.0/24     10.2.1.48 Available              1997    00:15:17
192.107.253.0/24 10.2.1.48 Suppressed/Unreachable 4331    00:19:42
192.195.44.0/24   10.2.1.48 Suppressed/Reachable   2923    00:19:15
192.195.49.0/24   10.2.1.48 Suppressed/Reachable   2923    00:19:15
192.195.50.0/24   10.2.1.48 Suppressed/Reachable   2923    00:19:15
192.197.150.0/24 10.2.1.48 Available              1997    00:15:25
192.222.89.0/24   10.2.1.48 Suppressed/Unreachable 2788    00:19:42
204.17.195.0/24   10.2.1.48 Suppressed/Reachable   2923    00:17:20
204.52.186.0/24   10.2.1.48 Available              1997    00:15:26
204.68.178.0/24   10.2.1.48 Available              1000    00:19:38
204.101.0.0/16    10.2.1.48 Available              1997    00:15:29
204.128.227.0/24 10.2.1.48 Suppressed/Reachable   2923    00:17:16
204.146.24.0/22   10.2.1.48 Available              1997    00:15:30
204.146.24.0/24   10.2.1.48 Available              1997    00:15:30
```

Meaning Table 31 on page 177 lists the **show ip bgp dampened-paths** command output fields.

Table 31: show ip bgp dampened-paths Output Fields

Field Name	Field Description
Local router ID	BGP router ID of the local router
local AS	Number of the local AS
Route flap dampening	Status of route flap dampening (enabled or disabled)
Decay half-life	Time (in minutes) after which a penalty is decreased. After the route has been assigned a penalty, the penalty is decreased by half after the half-life period (which is 15 minutes by default).
Cutoff threshold	Value of the penalty for a flapping route below which the route is unsuppressed
Reuse threshold	Time (in hours:minutes:seconds) after which the path will be made available

Table 31: show ip bgp dampened-paths Output Fields *(continued)*

Field Name	Field Description
Maximum hold-down time	Interval, in seconds, after not receiving a keepalive message that the software declares a peer dead
route flap history	Status of route flap history for route paths
Prefix	Prefix for the IP address
Peer	IP address of BGP peer
Status	Status of route dampening of the route path
Figure of Merit	A measure of the route's stability. Higher values indicate more recent route flap activity or less stability.
Time until Reuse/Remove	Time until the route is either reused (if currently suppressed) or its history entry is removed (if currently available)

Related Documentation

- *show bgp ipv6 dampened-paths*
- *show ip bgp dampened-paths*

Monitoring BGP Routes with Matching AS Paths and AS-Path Access Lists

Purpose Display all routes whose AS path matches the specified AS-path access list. Report whether the indirect next hop of a route is unreachable; if not, display the IGP cost to the indirect next hop.

The **show ip bgp filter-list** and **show bgp ipv6 filter-list** commands display similar information.

Action To display all routes whose AS path matches the specified AS-path access list:

```
host1#show ip bgp filter-list 1
Local router ID 192.168.1.153, local AS 100
  72080 paths, 72080 distinct prefixes (5189760 bytes used)
  72080 paths selected for route table installation
  21667 path attribute entries (2962828 bytes used)

  Prefix        Next-hop  MED CalPrf Weight AS-path
> 6.0.0.0/8     10.5.0.48   100        100  11488 701 7018 7170
> 12.0.0.0/8    10.5.0.48   100        100  11488 701 1740 7018
> 12.1.248.0/24 10.5.0.48   100        100  11488 701 7018 13391
> 12.2.6.0/24   10.5.0.48   100        100  11488 701 7018 11101
> 12.2.7.0/24   10.5.0.48   100        100  11488 701 7018 11101
> 12.2.76.0/24  10.5.0.48   100        100  11488 701 7018 11812
> 12.2.99.0/24  10.5.0.48   100        100  11488 701 7018 10656
> 12.2.109.0/24 10.5.0.48   100        100  11488 701 7018 10656
> 12.2.169.0/24 10.5.0.48   100        100  11488 701 7018 11806
> 12.4.114.0/24 10.5.0.48   100        100  11488 701 7018 14065
```



```
> 12.4.119.0/24 10.5.0.48   100        100  11488 701 7018 14065
> 12.4.175.0/24 10.5.0.48   100        100  11488 701 7018 11895
> 12.4.196.0/22 10.5.0.48   100        100  11488 701 7018 12163
> 12.5.48.0/21  10.5.0.48   100        100  11488 701 7018 12163
> 12.5.164.0/24 10.5.0.48   100        100  11488 701 7018 11134
> 12.6.42.0/23  10.5.0.48   100        100  11488 701 7018 11090
```

Meaning Table 32 on page 179 lists the **show ip bgp filter-list** command output fields.

Table 32: show ip bgp filter-list Output Fields

Field Name	Field Description
Local router ID	BGP router ID of the local router
local AS	Local autonomous system number
paths	Total number of routes stored in the BGP routing table. If several peers have advertised a route to the same prefix, all routes are included in this count.
distinct prefixes	Number of routes to unique prefixes stored in the BGP routing table. If several peers have advertised a route to the same prefix, only the best route is included in this count.
paths selected for route table installation	Number of routes in the BGP routing table that have been inserted into the IP routing table
path attribute entries	Number of distinct path attributes stored in BGP's internal path attributes table. If BGP receives two routes for different prefixes but with identical path attributes, BGP will create only one entry in its internal path attribute table and share it between the two routes to conserve memory.
Prefix	Prefix for the routing table entry
Next hop IP address	IP address of the next router that is used when a packet is forwarded to the destination network
MED	Multiexit discriminator for the route
CalPrf	Calculated preference for the route
Weight	Weight of the route
AS path	Autonomous system path

Related Documentation

- *show bgp ipv6 filter-list*
- *show ip bgp filter-list*

Monitoring BGP Flap Statistics

Purpose Display information about BGP flap statistics

The **show ip bgp flap-statistics** and **show bgp ipv6 flap-statistics** commands display similar information.

Action To display information about flap statistics:

```
host1#show ip bgp flap-statistics
Local BGP identifier 192.168.1.232, local AS 100
  Route flap dampening is enabled
  Default decay half-life is 15 minutes
  Default cutoff threshold is 2000, default reuse threshold is 750
  Default maximum hold-down time is 60 minutes
  307 paths have active route flap histories (27016 bytes used)
  5 paths are suppressed
                                                  Figure      Time until
Prefix              Peer             Status       of Merit     Reuse/Remove
24.201.0.0/18       192.168.1.158    Available     925         00:58:23
24.201.64.0/18      192.168.1.158    Available     925         00:58:23
52.128.224.0/19     192.168.1.158    Available     750         00:54:12
61.8.0.0/19         192.168.1.158    Available     993         00:59:53
61.8.30.0/24        192.168.1.158    Available     993         00:59:53
62.229.73.0/24      192.168.1.158    Unreachable   925         00:58:23
63.69.150.0/24      192.168.1.158    Available     750         00:54:12
```

Meaning Table 33 on page 180 lists the **show ip bgp flap-statistics** command output fields.

Table 33: show ip bgp flap-statistics Output Fields

Field Name	Field Description
Local BGP identifier	BGP router ID of the local router
local AS	Local autonomous system number
Route flap dampening	Status of route flap dampening (enabled or disabled)
Default decay half-life	Time (in minutes) after which a penalty is decreased. After the route has been assigned a penalty, the penalty is decreased by half after the half-life period (which is 15 minutes by default).
Default cutoff threshold	Value of the penalty for a flapping route below which the route is unsuppressed
Default reuse threshold	Time (in hours:minutes:seconds) after which the path will be made available
Default maximum hold-down time	Interval, in seconds, after not receiving a keepalive message that the software declares a peer dead
route flap history	Status of route flap history for route paths

Table 33: show ip bgp flap-statistics Output Fields *(continued)*

Field Name	Field Description
Prefix	Prefix for the routing table entry
Peer	IP address of BGP peer
Status	Status of route dampening of the route path
Figure of Merit	A measure of the route's stability. Higher values indicate more recent route flap activity or less stability.
Time until Reuse/Remove	Time until the route is either reused (if currently suppressed) or its history entry is removed (if currently available)

Related Documentation

- *show bgp ipv6 flap-statistics*
- *show ip bgp flap-statistics*

Monitoring BGP Routes with Inconsistent AS Paths

Purpose Display information about routes that have inconsistent AS paths. Report whether the indirect next hop of a route is unreachable; if not, display the IGP cost to the indirect next hop.

The **show ip bgp inconsistent-as** and **show bgp ipv6 inconsistent-as** commands display similar information.

Action To display information about routes that have inconsistent AS paths:

```
host1#show ip bgp inconsistent-as
Local BGP identifier 192.168.1.10, local AS 123
  0 routes (0 bytes used), 0 distinct destinations (0 bytes used)
  0 routes selected for route table installation
  0 path attribute entries (0 bytes used)

Status codes: > best, * invalid, s suppressed, d dampened, r rejected

  Prefix             Next-hop         MED  LocPrf Weight AS-path
> 4.0.0.0/8          0.0.0.0            1          32768 empty
  4.0.0.0/8          192.168.1.1        0 11488     701 1
```

Meaning Table 34 on page 181 lists the **show ip bgp inconsistent-as** command output fields.

Table 34: show ip bgp inconsistent-as Output Fields

Field Name	Field Description
Local BGP identifier	BGP router ID of the local router
local AS	Local autonomous system number

Table 34: show ip bgp inconsistent-as Output Fields *(continued)*

Field Name	Field Description
routes	Total number of routes stored in the BGP routing table and amount of memory consumed by routes. If several peers have advertised a route to the same prefix, all routes are included in this count.
distinct destinations	Number of routes to unique prefixes stored in the BGP routing table. If several peers have advertised a route to the same prefix, only the best route is included in this count.
routes selected for route table installation	Number of routes in the BGP routing table that have been inserted into the IP routing table
path attribute entries	Number of distinct path attributes stored in BGP's internal path attributes table. If BGP receives two routes for different prefixes but with identical path attributes, BGP will create only one entry in its internal path attribute table and share it between the two routes to conserve memory.
Prefix	Prefix for the routing table entry
Next hop	IP address of the next hop
MED	Multiexit discriminator for the route
LocPrf	Local preference for the route
Weight	Assigned path weight
Origin	Origin of the route
AS-path	AS path through which this route bas been advertised

Related Documentation

- *show bgp ipv6 inconsistent-as*
- *show ip bgp inconsistent-as*

Monitoring BGP Neighbors

Purpose Display information about BGP neighbors.

The **show ip bgp neighbors** and **show bgp ipv6 neighbors** commands display similar information.

Action To display information about BGP neighbors:

- host1#**show ip bgp neighbors**

```
BGP neighbor ID 10.2.1.48, remote AS 11488 (external peer)
  Remote router ID is 172.31.1.48, negotiated BGP version is 4
  Administrative status is Start, connection state is Established
```

```
  Reason for last reset was tcp connection error
  TCP error code 60 (Connection timed out)
  Connection has been established 1 time, up for 0 17:42:31
  Options:
    Default originate is disabled
    EBGP multi-hop is enabled
    IBGP single-hop is disabled
    Next hop self is disabled
    seconds
Policy:
    Neighbor weight is 100
 Timers:
    Connect retry interval is 120 seconds
    Minimum route advertisement interval is 30 seconds
 Minimum AS origination interval is 10 seconds
    Configured keep-alive interval is 30 seconds, negotiated 30
       seconds
    Configured hold time is 90 seconds, negotiated 90
 TCP connection:
    Local IP address is 192.168.1.218, local port is 1024
    Remote IP address is 10.2.1.48, remote port is 179
 Statistics:
    Total of 4100 messages sent, 44913 messages received
    2053 update messages sent, 42785 update messages received
    0 00:00:17 since last update message was received
```

- Fields relevant to multiprotocol extensions:

```
Multi-protocol extensions negotiation:
    ip-v4 unicast: sent, received, used
    ip-v6 unicast-labeled: sent, received, used
```

- For the graceful restart capability, additional information is presented.
  - Fields concerning graceful restart attributes that apply to peers as a whole (for all address families):

```
Graceful restart negotiation:
    Sent restart time is 120 seconds
    Sent restart state bit is zero (we are not restarting)
    Received restart time is 120 seconds
    Received restart state bit is zero (peer is not restarting)
    Maximum time for keeping stale paths is 360 seconds
```

  - Fields concerning attributes that apply to peers a particular address family:

```
Peer is capable of preserving forwarding stat(3)
Peer preserved forwarding state during last restart
We have received an end-of-rib marker from the peer
We have sent an end-of-rib marker to the peer
```

  - Fields relevant if the peer is currently restarting:

```
  Graceful restart waiting for the session to come back up
    Restart-time advertised by the peer is 120 seconds
    Remaining time for the peer to come back up is 117 seconds
    Remaining time for keeping stale routes from the peer is 357 seconds
```

  - Fields relevant during reconvergence after the peer has restarted:

```
  Graceful restart negotiation:
    Sent restart time is 120 seconds
    Sent restart state bit is zero (we are not restarting)
    Received restart time is 120 seconds
```

```
Received restart state bit is zero (peer is not restarting)
Maximum time for keeping stale paths is 300 seconds
Remaining time for keeping stale routes from the peer is 297 seconds
```

- For BFD, additional information is presented.
  - Fields relevant to BFD when BFD is not configured:

    ```
    BFD is disabled
    ```

  - Fields relevant to BFD when BFD is configured for an IBGP peer:

    ```
    BFD is enabled but not supported (IBGP neighbor)
    ```

  - Fields relevant to BFD when BFD is configured for a multihop EBGP peer:

    ```
    BFD is enabled but not supported (multi-hop EBGP neighbor)
    ```

  - Fields relevant to BFD when BFD is configured but the BGP session is not established:

    ```
    BFD is enabled:
         Single-hop IPv4 BFD session to 1.2.3.4
         Minimum transmit interval is 300 ms
         Minimum receive interval is 300 ms
         Multiplier is 3
         Waiting for BGP to become established before initiating BFD session
    ```

  - Fields relevant to BFD when BFD is configured, the BGP session is established, but the BFD protocol session is not up:

    ```
    BFD is enabled:
         Single-hop IPv4 BFD session to 1.2.3.4
         Minimum transmit interval is 300 ms
         Minimum receive interval is 300 ms
         Multiplier is 3
         BFD session is down
    ```

  - Fields relevant to BFD when BFD is configured, the BGP session is established, and the BFD protocol session is up:

    ```
    BFD is enabled:
         Single-hop IPv4 BFD session to 1.2.3.4
         Minimum transmit interval is 300 ms
         Minimum receive interval is 300 ms
         Multiplier is 3
         BFD session is up for 00:00:50
         Negotiated detection time is 900 ms
    ```

- Fields relevant to conditional advertisement:

  ```
  Advertise-map is advertisetoR1
      Condition-map: trigger1
      Sequence: 5
      Status: Withdraw
  Advertise-map is alternatetoR1
      Condition-map: trigger2
      Sequence: 10
      Status: Advertise
  ```

Meaning Table 35 on page 185 lists the **show ip bgp neighbors** command output fields.

Table 35: show ip bgp neighbors Output Fields

Field Name	Field Description
BGP neighbor ID	BGP identifier of the BGP neighbor
remote AS	Remote AS of the BGP neighbor
Description	Textual description of the BGP neighbor
Member of peer group	Name of the peer group of which this BGP neighbor is a member
Remote router ID	Router ID of the remote router
negotiated BGP version	BGP version being used to communicate with the neighbor
Administrative status	Desired state of the peer connection
Connection state	Current state of the BGP connection
Connection has been established	Time that TCP connection was established
Reason for last reset	Reason for last reset of the BGP session
TCP error code	TCP connection error type
Default originate	Status of default originate (enabled or disabled)
EBGP multi-hop	Status of EBGP multihop (enabled or disabled)
IBGP single-hop	Status of IBGP single hop (enabled or disabled)
Next hop self	Status of next-hop self (enabled or disabled)
Route reflector status	Identifies the neighbor as a route-reflector client
Neighbor weight	Weight of routes from the BGP neighbor
Incoming update distribute list	Distribute list for incoming routes, if configured
Outgoing update distribute list	Distribute list for outgoing routes, if configured
Incoming update filter list	Update filter list for incoming routes, if configured
Outgoing update filter list	Update filter list for outgoing route, if configured
Weight filter list	Weight filter list for routes, if configured
Incoming route map	Incoming route map, if configured

Table 35: show ip bgp neighbors Output Fields *(continued)*

Field Name	Field Description
Outgoing route map	Outgoing route map, if configured
Connect retry interval	Time between a BGP peer’s attempts to reestablish a connection to the neighbor
Minimum route advertisement interval	Minimum time between route advertisements
Minimum AS origination interval	Minimum time between advertisement of changes within the speaker’s AS
Configured keep-alive interval	Frequency of keep-alive messages generated
Negotiated keepalive interval	Negotiated frequency of keep-alive messages generated
Configured hold time	Configured maximum time allowed between received messages
Negotiated hold time	Negotiated maximum time allowed between received messages
Configured update source IP address	IP address used when sending update messages
Local IP address	Local IP address used for TCP communication to this peer
Local port	Local TCP port number used for TCP communication to this peer
Remote IP address	Remote IP address used for TCP communication to this peer
Remote port	Remote IP address used for TCP communication to this peer
Total messages sent	Total BGP messages sent to this neighbor
Total messages received	Total BGP messages received from this neighbor
Total update messages sent	Total BGP update messages sent to this neighbor
Total update messages received	Total BGP update messages received from this neighbor
Time since last update message was received	Time since last BGP update message was received from this neighbor
Address Family dependent capabilities	Lists type of ORF send and receive capability per address family and whether it is advertised (configured) or received

Table 35: show ip bgp neighbors Output Fields *(continued)*

Field Name	Field Description
Maximum number of ORF entries	Limit of ORF entries that will be accepted from the neighbor
Capability advertisement	Lists whether the specific capability (capabilities option, deprecated dynamic capability negotiation, dynamic capability negotiation, multiprotocol extensions, route refresh, route refresh (Cisco proprietary), four octet AS numbers, and graceful restart) has been sent, received, or both
Multi-protocol extensions negotiation	Lists the relevant address family and whether it has been sent, received, or used
BFD	Status of BFD configuration, enabled, enabled but not supported because the peer is an IBGP neighbor a multihop EBGP neighbor, or disabled
BFD session	Type and address of peer to which BFD session is established
Minimum transmit interval	Desired interval between BFD packets transmitted to members of peer group
Minimum receive interval	Desired interval between BFD packets received from members of peer group
Multiplier	Number of BFD packets that can be missed before declaring BFD session down
Negotiated detection time	Interval between BFD packets negotiated by peers
Advertise-map	Name of route map that specifies routes to be advertised when routes in conditional route maps are matched
Condition-map	Name of route map that specifies routes to be matched by routes in the BGP routing table
Sequence	Position of the specified advertise route map in a list of advertise route maps configured for a particular peer within the same address-family. A lower sequence number has a higher priority; that route map is processed before one with a higher sequence number.
Status	Status of the routes specified by the route map, advertise (route map condition has been met) or withdraw (route map condition has not been met; regardless of this status, the specified routes might be governed by another route map with a lower sequence number and actually advertised or not according to that map

Related Documentation

- *show bgp ipv6 neighbors*
- *show ip bgp neighbors*

Monitoring Dampened BGP Routes of Specified Neighbors

Purpose

Display information about routes with a dampening history for the specified BGP neighbor. Report whether the indirect next hop of a route is unreachable; if not, display the IGP cost to the indirect next hop.

The **show ip bgp neighbors dampened-routes** and **show bgp ipv6 neighbors dampened-routes** commands display similar information.

Action

To display information about routes with a dampening history for the specified BGP neighbor:

```
host1#show ip bgp neighbors 192.168.1.158 dampened-routes
Local BGP identifier 192.168.1.232, local AS 100
  120 routes (5760 bytes used), 94 distinct destinations (9024 bytes used)
  67 routes selected for route table installation
  23 path attribute entries (3450 bytes used)

Status codes: > best, * invalid, s suppressed, d dampened, r rejected

  Prefix             Peer            Next-hop MED LocPrf Weight Origin
 d12.8.12.0/24       192.168.1.158   192.168.1.1             0 IGP
 d24.48.12.0/24      192.168.1.158   192.168.1.1             0 IGP
 d24.72.12.0/24      192.168.1.158   192.168.1.1             0 inc.
 d24.116.12.0/23     192.168.1.158   192.168.1.1             0 IGP
 d24.143.12.0/24     192.168.1.158   192.168.1.1             0 IGP
 d24.154.12.0/24     192.168.1.158   192.168.1.1             0 inc.
 d24.216.12.0/24     192.168.1.158   192.168.1.1             0 IGP
 d24.240.12.0/24     192.168.1.158   192.168.1.1             0 IGP
 d24.244.12.0/22     192.168.1.158   192.168.1.1             0 IGP
 d24.246.12.0/22     192.168.1.158   192.168.1.1             0 IGP
 d61.0.12.0/24       192.168.1.158   192.168.1.1             0 IGP
 d61.11.12.0/24      192.168.1.158   192.168.1.1             0 IGP
 d62.74.12.0/22      192.168.1.158   192.168.1.1             0 IGP
 d62.76.12.0/22      192.168.1.158   192.168.1.1             0 IGP
 d63.65.12.0/24      192.168.1.158   192.168.1.1             0 inc.
 d63.73.12.0/24      192.168.1.158   192.168.1.1             0 IGP
```

Meaning

Table 36 on page 188 lists the **show ip bgp neighbors dampened-routes** command output fields.

Table 36: show ip bgp neighbors dampened-routes Output Fields

Field Name	Field Description
Local BGP identifier	BGP router ID of the local router
local AS	Local autonomous system number
routes	Total number of routes stored in the BGP routing table and amount of memory consumed by routes. If several peers have advertised a route to the same prefix, all routes are included in this count.

Table 36: show ip bgp neighbors dampened-routes Output Fields *(continued)*

Field Name	Field Description
distinct destinations	Number of routes to unique prefixes stored in the BGP routing table. If several peers have advertised a route to the same prefix, only the best route is included in this count.
routes selected for route table installation	Number of routes in the BGP routing table that have been inserted into the IP routing table
path attribute entries	Number of distinct path attributes stored in BGP's internal path attributes table. If BGP receives two routes for different prefixes but with identical path attributes, BGP will create only one entry in its internal path attribute table and share it between the two routes to conserve memory.
Prefix	Prefix for the routing table entry
Peer	IP address of BGP peer
Next hop IP address	IP address of the next hop
MED	Multiexit discriminator for the route
LocPrf	Local preference for the route
Weight	Assigned path weight
Origin	Origin of the route

Related Documentation

- *show bgp ipv6 neighbors dampened-routes*
- *show ip bgp neighbors dampened-routes*

Monitoring BGP Paths of Neighbors

Purpose Display path information for the specified BGP neighbor.

The **show ip bgp neighbors paths** and **show bgp ipv6 neighbors paths** commands display similar information.

Action To display path information for the specified BGP neighbor.

```
host1#show ip bgp neighbors 1.02.3.4 paths
Address      Refcount  Origin  Next-hop          AS-path
0xC384BD0    1          IGP     192.168.1.1       11488 701 2853 5515 764
0xC384C40    1          IGP     192.168.1.1       11488 701 4183
0xC384CB0    1         IGP     192.168.1.1      11488 701 1239 1833 1833 1833 1299
 8308
0xC384D20    1          IGP     192.168.1.1       11488 701 6453 786
0xC384D90    1          IGP     192.168.1.1       11488 701 6453 1103 1103
```

```
0xC384E00   1        IGP     192.168.1.1      11488 701 6762 9116 9116 9116 6888
 6888
0xC384E70   1        IGP     192.168.1.1      11488 701 6453 8297 6758
0xC384EE0   1        IGP     192.168.1.1      11488 701 5511 3215
0xC384F50   1        IGP     192.168.1.1      11488 701 3561 5683 5551
0xC384FC0   1        IGP     192.168.1.1      11488 701 1239 1755 1273 8793 8793
 8793
0xC385030   1        IGP     192.168.1.1      11488 701 5705 5693
```

Meaning Table 37 on page 190 lists the **show ip bgp neighbors paths** command output fields.

Table 37: show ip bgp neighbors paths Output Fields

Field Name	Field Description
Address	Hexadecimal number that uniquely identifies the path attributes
Refcount	Number of routes that share the path attributes
Origin	Value of the origin path attribute
Next-hop	Value of the next-hop path attribute
AS-path	Value of the AS-path attribute

Related Documentation

- *show bgp ipv6 neighbors paths*
- *show ip bgp neighbors paths*

Monitoring Prefix List Outbound Route Filters Received from the BGP Neighbor

Purpose Display prefix-list outbound route filters received from the BGP neighbor.

Action To display prefix-list outbound route filters received from the specified neighbor:

```
host1#show ip bgp neighbors 192.168.1.158 received prefix-filter
ip prefix-list filter 192.168.1.158 for address family ipv4:unicast
   seq 5 permit 10.1.1.1/32
   seq 10 permit 10.1.1.2/32
   seq 15 permit 10.1.1.3/32
```

Meaning Table 38 on page 190 lists the **show ip bgp neighbors received prefix-filter** command output fields.

Table 38: show ip bgp neighbors received prefix-filter

Field Name	Field Description
seq	Sequence number of the entry in the prefix list
permit, deny	Condition statement for addresses matching the listed address

Related Documentation

- *show ip bgp neighbors*

Monitoring Routes Originating from a BGP Neighbor Before Application of Inbound Policy

Purpose

Display routes originating from the specified BGP neighbor before inbound policy is applied. Report whether the indirect next hop of a route is unreachable; if not, display the IGP cost to the indirect next hop.

The **show ip bgp neighbors received-routes** and **show bgp ipv6 neighbors received-routes** commands display similar information.

Action

To display routes originating from the specified BGP neighbor before inbound policy is applied:

```
host1#show ip bgp neighbors 192.168.1.158 received-routes
Local BGP identifier 111.111.111.111, local AS 444
  0 routes (0 bytes used), 0 distinct destinations (0 bytes used)
  0 routes selected for route table installation
  0 path attribute entries (0 bytes used)

Status codes: > best, * invalid, s suppressed, d dampened, r rejected

Prefix          Peer          Next-hop      MED LocPrf Weight Origin
>0.0.0.0/0      192.168.1.158  192.168.1.158              0       IGP
>13.13.13.13/32 192.168.1.158  192.168.1.158       0      0       IGP
```

Meaning

Table 39 on page 191 lists the **show ip bgp neighbors received-routes** command output fields.

Table 39: show ip bgp neighbors received-routes Output Fields

Field Name	Field Description
Local BGP identifier	BGP router ID of the local router
local AS	Local autonomous system number
routes	Total number of routes stored in the BGP routing table and amount of memory consumed by routes. If several peers have advertised a route to the same prefix, all routes are included in this count.
distinct destinations	Number of routes to unique prefixes stored in the BGP routing table. If several peers have advertised a route to the same prefix, only the best route is included in this count.
routes selected for route table installation	Number of routes in the BGP routing table that have been inserted into the IP routing table
path attribute entries	Number of distinct path attributes stored in BGP's internal path attributes table. If BGP receives two routes for different prefixes but with identical path attributes, BGP will create only one entry in its internal path attribute table and share it between the two routes to conserve memory.

Table 39: show ip bgp neighbors received-routes Output Fields *(continued)*

Field Name	Field Description
Prefix	Prefix for the routing table entry
Peer	IP address of BGP peer
Next hop	IP address of the next hop
MED	Multiexit discriminator for the route
LocPrf	Local preference for the route
Weight	Assigned path weight
Origin	Origin of the route

Related Documentation

- *show bgp ipv6 neighbors received-routes*
- *show ip bgp neighbors received-routes*

Monitoring Routes Originating from a BGP Neighbor After Application of Inbound Policy

Purpose

Display all routes that originate from the specified BGP neighbor after inbound policy is applied. Report whether the indirect next hop of a route is unreachable; if not, display the IGP cost to the indirect next hop.

The **show ip bgp neighbors routes** and **show bgp ipv6 neighbors routes** commands display similar information.

Action

To all routes that originate from the specified BGP neighbor after inbound policy is applied:

```
host1#show bgp ipv6 neighbors 12.12.12.12 routes
Local BGP identifier 11.11.11.11, local AS 400
  5 routes (200 bytes)
  5 destinations (360 bytes) of which 5 have a route
  5 routes selected for route table installation
  4 path attribute entries (608 bytes)
  Local-RIB version 33. FIB version 33.

Status codes: > best, * invalid, s suppressed, d dampened, r rejected,
              a auto-summarized

  Prefix              Peer          Next-hop           MED LocPrf Weight Origin
> 3ffe:0:1:1::/64      12.12.12.12    ::102.102.102.2      0     100      0 inc.
```

Meaning

Table 40 on page 193 lists the **show bgp ipv6 neighbors routes** command output fields.

Table 40: show bgp ipv6 neighbors routes Output Fields

Field Name	Field Description
Local BGP identifier	BGP router ID of the local router
local AS	Local autonomous system number
routes	Total number of routes stored in the BGP routing table and amount of memory consumed by routes. If several peers have advertised a route to the same prefix, all routes are included in this count.
destinations	Number of routes to unique prefixes stored in the BGP routing table and amount of memory consumed by routes. If several peers have advertised a route to the same prefix, only the best route is included in this count.
routes selected for route table installation	Number of routes in the BGP routing table that have been inserted into the IP routing table, plus prefixes for which there are currently no routes but which have had to be withdrawn from peers to which these prefixes may been previously advertised.
path attribute entries	Number of distinct path attributes stored in BGP's internal path attributes table. If BGP receives two routes for different prefixes but with identical path attributes, BGP will create only one entry in its internal path attribute table and share it between the two routes to conserve memory.
Local-RIB version	Number that is increased by one each time a route in that RIB is added, removed or modified.
FIB version	Number that is increased by one each time BGP updates the routes in the IP routing table based on changes in the local RIB. The FIB version matches the local-RIB version when BGP has finished updating the routes in the IP route table. The FIB version is less than the local-RIB version when BGP is still in the process of updating the IP routing table.
Prefix	Prefix for the routing table entry
Peer	IP address of BGP peer
Next hop	IP address of the next hop
MED	Multiexit discriminator for the route
LocPrf	Local preference for the route
Weight	Assigned path weight
Origin	Origin of the route

Related Documentation

- *show bgp ipv6 neighbors routes*
- *show ip bgp neighbors routes*

Monitoring Networks in an Autonomous System

Purpose

Display information about networks in an AS.

The **show ip bgp network** and **show bgp ipv6 network** commands display similar information.

Action

To display information about networks in an AS:

```
host1#show bgp ipv6 network
Prefix                        Weight   Route-map     Backdoor
3ffe:0:0:2::/64                                      No
```

Meaning

Table 41 on page 194 lists the **show bgp ipv6 network** command output fields.

Table 41: show bgp ipv6 network Output Fields

Field Name	Field Description
Prefix	Prefix for the routing table entry
Weight	Assigned path weight
Route-map	Indicates whether network-route filtering is enabled
Backdoor	Indicates whether an IGP backdoor route is favored over an EBGP route

Related Documentation

- *show bgp ipv6 network*
- *show ip bgp network*

Monitoring BGP Next Hops

Purpose

Display information about all indirect next hops or a particular next hop.

The **show ip bgp next-hops** and **show bgp ipv6 next-hops** commands display similar information.

Action

To display information about all indirect next hops or a particular indirect next hop:

```
host1#show ip bgp next-hops
Indirect next-hop 4.4.4.4
  Reachable (metric 2)
  Direct next-hop atm2/0.34 (34.34.34.4)
  Reference count is 3

Indirect next-hop ::ffff:2.2.2.2
  MPLS stacked label 17
```

```
  Reachable (metric 3)
  Direct next-hop  tun mpls:vpnInL17-23
  Reference count is 1

Indirect next-hop 5.5.5.5
  Reachable (metric 2)
  Direct next-hop atm2/0.35 (35.35.35.5)
  Reference count is 3

Indirect next-hop 6.6.6.6
  Reachable (metric 3)
  Direct next-hop atm2/0.34 (34.34.34.4)
                  atm2/0.35 (35.35.35.5)
  Reference count is 3

Indirect next-hop 13.13.13.1
  Not reachable
  Reference count is 2
```

Meaning Table 42 on page 195 lists the **show ip bgp next-hops** command output fields.

Table 42: show ip bgp next-hops Output Fields

Field Name	Field Description
Indirect next-hop	BGP next-hop attribute as received in the BGP update message
Reachable	Indicates whether or not the indirect next hop is reachable.
Metric	Metric of the BGP indirect next hop
Direct next-hop	IP interface and next-hop IP address that resolve the BGP indirect next hop; the direct next hop can also be an IP indirect next hop or an MPLS indirect next hop when chains of next hops are in use
Reference count	Number of label mappings of BGP routes that use this next hop

Related Documentation

- *show bgp ipv6 next-hops*
- *show ip bgp next-hops*

Monitoring BGP Paths

Purpose Display information about the most common BGP path attributes.

The **show ip bgp paths** and **show bgp ipv6 paths** commands display similar information.

Action To display information about BGP paths:

```
host1#show bgp ipv6 paths
Address     Refcount  Metric  AS-path
0x4B311118  1                 100
0x4C548224  1         0       100
0x4C548530  1         0       200
0x4C548704  2         0       300
```

BGP internally maintains additional attributes that are not displayed—for example, the MED, local preference, and communities attributes.

Meaning Table 43 on page 196 lists the **show bgp ipv6 paths** command output fields.

Table 43: show bgp ipv6 paths Output Fields

Field Name	Field Description
Address	Hexadecimal number that uniquely identifies the path attributes
Refcount	Number of routes that share the path attributes
Origin	Value of the origin path attribute
Next hop	Value of the next-hop path attribute
AS-path	Value of the AS-path attribute

Related Documentation

- *show bgp ipv6 paths*
- *show ip bgp paths*

Monitoring BGP Peer Groups

Purpose Display information about BGP peer groups.

The **show ip bgp peer-group** and **show bgp ipv6 peer-group** commands display similar information.

Action To display information about BGP peer groups:

```
host1#show ip bgp peer-group
BGP peer-group leftcoast, remote AS 200
  Peer-group members are external peers
  Local AS 100
  Administrative status is Start
  EBGP multi-hop is disabled
  IBGP single-hop is disabled
  BFD is enabled:
     Single-hop IPv4 BFD session
     Minimum transmit interval is 300 ms
     Minimum receive interval is 300 ms
     Multiplier is 3
  Maximum update message size is 1024 octets
  Neighbor weight is 0
  Connect retry interval is 10 seconds initially
  Configured keep-alive interval is 30 seconds
  Configured hold time is 90 seconds
  Minimum route advertisement interval is 30 seconds
  Minimum AS origination interval is 10 seconds
  Graceful restart negotiation:
    Restart time is 120 seconds
    Stale paths time is 360 seconds
```



```
Configuration for address family ipv4:unicast
  RIB-out is disabled
  Default originate is disabled
  Next hop self is disabled
  Next hop unchanged is disabled
  Don't send communities
  Inbound soft reconfiguration is disabled
  Private AS number stripping is disabled
  Override site AS with provider AS is disabled
  No loops in the received AS-path are allowed
  Members: 10.2.2.2 10.3.3.3
```

- Fields relevant to conditional advertisement:

```
Advertise-map is advertisetoR1
    Condition-map: trigger1
    Sequence: 5
    Status: Withdraw
Advertise-map is alternatetoR1
    Condition-map: trigger2
    Sequence: 10
    Status: Advertise
```

Meaning Table 44 on page 197 lists the **show ip bgp peer-group** command output fields.

Table 44: show ip bgp peer-group Output Fields

Field Name	Field Description
BGP peer group	Name of a BGP peer group
remote AS	Remote AS of the BGP neighbor
local AS	Local autonomous system number
Administrative status	Desired state of the peer connection
EBGP multi-hop	Status of EBGP multihop for the peer group (enabled or disabled)
IBGP single-hop	Status of IBGP single hop for the peer group (enabled or disabled)
BFD	Status of BFD configuration for the peer group (enabled or disabled)
BFD session	Type and address of peer to which BFD session is established
Description	Textual description of the BGP neighbor
Members	Name of the peer group of which this BGP neighbor is a member
Default originate	IP addresses of the members of the BGP peer group
Minimum transmit interval	Desired time interval between BFD packets transmitted to members of peer group

Table 44: show ip bgp peer-group Output Fields *(continued)*

Field Name	Field Description
Minimum receive interval	Desired time interval between BFD packets received from members of peer group
Multiplier	Number of BFD packets that can be missed before declaring BFD session down
Next hop self	Status of next-hop self information for the peer group (enabled or disabled)
Peers are route reflector clients	BGP peer group is configured as a route reflector. This field does not appear when route reflectors are not configured.
weight	Neighbor weights assigned to BGP peer groups
Incoming update distribute list	Distribute list for incoming routes, if configured
Outgoing update distribute list	Distribute list for outgoing routes, if configured
Incoming update filter list	Update filter list for incoming routes, if configured
Outgoing update filter list	Update filter list for outgoing route, if configured
Weight filter list	Weight filter list for routes, if configured
Incoming route map	Incoming route map, if configured
Outgoing route map	Outgoing route map, if configured
Minimum route advertisement interval	Minimum time between route advertisements
Configured update source IP address	IP address used when sending update messages
Advertise-map	Name of route map that specifies routes to be advertised when routes in conditional route maps are matched
Condition-map	Name of route map that specifies routes to be matched by routes in the BGP routing table
Sequence	Position of the specified advertise route map in a list of advertise route maps configured for a particular peer within the same address-family. A lower sequence number has a higher priority; that route map is processed before one with a higher sequence number.

Table 44: show ip bgp peer-group Output Fields *(continued)*

Field Name	Field Description
Status	Status of the routes specified by the route map, advertise (route map condition has been met) or withdraw (route map condition has not been met; regardless of this status, the specified routes might be governed by another route map with a lower sequence number and actually advertised or not according to that map

Related Documentation

- *show bgp ipv6 peer-group*
- *show ip bgp peer-group*

Monitoring BGP Routes with Matching AS Paths and Regular Expressions for Single Regular Expressions

Purpose Display information about BGP routes whose AS path matches the specified regular expression. Accepts a single regular expression element. Report whether the indirect next hop of a route is unreachable; if not, display the IGP cost to the indirect next hop.

Regular expressions match numbers for which the specified path is a substring—for example, if you specify 20, 200 matches because 20 is a substring of 200. You can disallow substring matching by using the underscore (_) metacharacter to constrain matching to the specified pattern, for example, _20_.

The **show ip bgp quote-regexp** and **show bgp ipv6 quote-regexp** commands display similar information.

Action To display information about routes whose AS path matches the specified regular expression:

```
host1#show ip bgp quote-regexp ^200
Local router ID 192.168.1.232, local AS 100
  6 paths, 3 distinct prefixes (324 bytes used)
  3 paths selected for route table installation
  7 path attribute entries (872 bytes used)

Prefix        Next-hop  MED  CalPrf  Weight  AS-path
10.99.1.2/32  10.1.1.2       100     100     200
10.99.1.3/32  10.1.1.2       100     100     200 10
10.99.1.4/32  10.1.1.2       100     100     200 10 20
```

NOTE: For single regular expressions without any spaces in them, you can use either **show ip bgp regexp** or **show ip bgp quote-regexp** with the same results.

You must enclose all regular expression elements within quotation marks (“ element”) when the regular expressions contain one or more spaces. To display information about routes whose AS path matches the specified regular expression and also has spaces within the regular expression element:

```
host1#show ip bgp quote-regexp " 10 20"
Local router ID 192.168.1.232, local AS 100
  6 paths, 3 distinct prefixes (324 bytes used)
  3 paths selected for route table installation
  7 path attribute entries (872 bytes used)

Prefix         Next-hop   MED  CalPrf  Weight  AS-path
10.99.1.4/32   10.1.1.2        100     100     200 10 20
```

Because the show **ip bgp quote-regexp** command accepts only one string as an argument to the regular expression, output filtering is possible. To display information about routes whose AS path matches the specified regular expression with output filtering:

```
host1#show ip bgp quote-regexp ^200 | begin Prefix

Prefix         Next-hop   MED  CalPrf  Weight  AS-path
10.99.1.2/32   10.1.1.2        100     100     200
10.99.1.3/32   10.1.1.2        100     100     200 10
10.99.1.4/32   10.1.1.2        100     100     200 10 20
```

Meaning Table 45 on page 200 lists the **show ip bgp quote-regexp** command output fields.

Table 45: show ip bgp quote-regexp Output Fields

Field Name	Field Description
Local router ID	BGP router ID of the local router
local AS	Local autonomous system number
paths	Total number of routes stored in the BGP routing table. If several peers have advertised a route to the same prefix, all routes are included in this count.
distinct prefixes	Number of routes to unique prefixes stored in the BGP routing table. If several peers have advertised a route to the same prefix, only the best route is included in this count.
paths selected for route table installation	Number of routes in the BGP routing table that have been inserted into the IP routing table
path attribute entries	Number of distinct path attributes stored in BGP's internal path attributes table. If BGP receives two routes for different prefixes but with identical path attributes, BGP will create only one entry in its internal path attribute table and share it between the two routes to conserve memory.
Prefix	Prefix for the routing table entry
Next hop IP address	IP address of the next router that is used when a packet is forwarded to the destination network
MED	Multiexit discriminator for the route

Table 45: show ip bgp quote-regexp Output Fields *(continued)*

Field Name	Field Description
CalPrf	Calculated preference for the route
Weight	Weight of the route
AS path	Autonomous system path

Related Documentation

- Monitoring BGP Routes with Matching AS Paths and Regular Expressions for Multiple Regular Expressions on page 201
- *show bgp ipv6 quote-regexp*
- *show ip bgp quote-regexp*

Monitoring BGP Routes with Matching AS Paths and Regular Expressions for Multiple Regular Expressions

Purpose Display information about BGP routes whose AS path matches the specified regular expression elements. Accepts one or more regular expression elements. Report whether the indirect next hop of a route is unreachable; if not, display the IGP cost to the indirect next hop.

Regular expressions match numbers for which the specified path is a substring—for example, if you specify 20, 200 matches because 20 is a substring of 200. You can disallow substring matching by using the underscore (_) metacharacter to constrain matching to the specified pattern, for example, _20_.

The **show ip bgp regexp** and **show bgp ipv6 regexp** commands display similar information.

Action To display information about routes whose AS path matches the specified regular expression element:

```
host1#show ip bgp regexp ^200
Local router ID 192.168.1.232, local AS 100
  6 paths, 3 distinct prefixes (324 bytes used)
  3 paths selected for route table installation
  7 path attribute entries (872 bytes used)

Prefix        Next-hop  MED  CalPrf  Weight  AS-path
10.99.1.2/32  10.1.1.2       100     100     200
10.99.1.3/32  10.1.1.2       100     100     200 10
10.99.1.4/32  10.1.1.2       100     100     200 10 20
```

NOTE: For single regular expressions without any spaces in them, you can use either **show ip bgp regexp** or **show ip bgp quote-regexp** with the same results.

To display information about routes whose AS path matches the specified regular expression and also has spaces within the regular expression element:

```
host1#show ip bgp regexp 10 20
Local router ID 192.168.1.232, local AS 100
  6 paths, 3 distinct prefixes (324 bytes used)
  3 paths selected for route table installation
  7 path attribute entries (872 bytes used)

Prefix          Next-hop    MED   CalPrf  Weight  AS-path
10.99.1.4/32    10.1.1.2          100     100     200 10 20
```

The **show ip bgp regexp** command accepts multiple strings as arguments. If you try to apply output filtering, the command interprets the filter information as a regular expression and fails. To display information about routes whose AS path matches the specified regular expression with output filtering:

```
host1#show ip bgp regexp ^200 | begin Prefix
% invalid regular expression
```

Meaning Table 46 on page 202 lists the **show ip bgp regexp** command output fields.

Table 46: show ip bgp regexp Output Fields

Field Name	Field Description
Local router ID	BGP router ID of the local router
local AS	Local autonomous system number
paths	Total number of routes stored in the BGP routing table. If several peers have advertised a route to the same prefix, all routes are included in this count.
distinct prefixes	Number of routes to unique prefixes stored in the BGP routing table. If several peers have advertised a route to the same prefix, only the best route is included in this count.
paths selected for route table installation	Number of routes in the BGP routing table that have been inserted into the IP routing table
path attribute entries	Number of distinct path attributes stored in BGP's internal path attributes table. If BGP receives two routes for different prefixes but with identical path attributes, BGP will create only one entry in its internal path attribute table and share it between the two routes to conserve memory.
Prefix	Prefix for the routing table entry
Next hop IP address	IP address of the next router that is used when a packet is forwarded to the destination network
MED	Multiexit discriminator for the route
CalPrf	Calculated preference for the route

Table 46: show ip bgp regexp Output Fields *(continued)*

Field Name	Field Description
Weight	Weight of the route
AS path	Autonomous system path

Related Documentation

- Monitoring BGP Routes with Matching AS Paths and Regular Expressions for Single Regular Expressions on page 199
- *show bgp ipv6 regexp*
- *show ip bgp regexp*

Monitoring the Status of All BGP Neighbors

Purpose Display summarized status of all BGP neighbors.

The **show ip bgp summary** and **show bgp ipv6 summary** commands display similar information.

Action To display summarized status of all BGP neighbors:

```
host1#show bgp ipv6 summary
Local router ID 10.13.13.13, local AS 400
  Administrative state is Start
  BGP Operational state is Up
  Shutdown in overload state is disabled
  Default local preference is 100
  IGP synchronization is disabled
  Default originate is disabled
  Always compare MED is disabled
  Compare MED within confederation is disabled
  Advertise inactive routes is disabled
  Advertise best external route to internal peers is disabled
  Enforce first AS is disabled
  Missing MED as worst is disabled
  Route flap dampening is disabled
  Maximum number of equal-cost EBGP paths is 2
  Maximum number of equal-cost IBGP paths is 2
  Log neighbor changes is disabled
  Fast External Fallover is disabled
  No maximum received AS-path length
  BGP administrative distances are 20 (ext), 200 (int), and 200 (local)
  Client-to-client reflection is enabled
  Cluster ID is 10.13.13.13
  Route-target filter is enabled
  Default IPv4-unicast is enabled
  Redistribution of iBGP routes is disabled
  Graceful restart is globally disabled
  Global graceful-restart restart time is 120 seconds
  Global graceful-restart stale paths time is 360 seconds
  Graceful-restart path selection defer time is 360 seconds
  This platform supports only the receiver role of graceful restart
  Route Distinguisher: 100:11
```



```
  Import route map: test2-import-map
  Export route map: test1-export-map (cannot filter routes)
  Global import route map: test3-global-import-map
  103 routes imported from global table (max 5000 routes allowed)
  Global export route map: test4-global-export-map
  Local-RIB version 7. FIB version 7.
                                                Messages  Messages  Prefixes
Neighbor           AS State       Up/down time      Sent  Received  Received
11.11.11.11       400 Established     00:36:19        78        81         2
12.12.12.12       400 Established     00:36:21        78        78         1
103.103.103.3     300 Established     00:36:34        85        80         2
```

To display the status of next hop reachability checking by specifying vpnv4:

```
host1#show ip bgp vpnv4 all summary
Local router ID 10.13.5.19, local AS 100
  Administrative state is Start
  BGP Operational state is Up
 ...
  Default IPv4-unicast is enabled
  Redistribution of iBGP routes is disabled
  Check reachability of next-hops for VPN routes is enabled
 ...
```

To display the status of fields related to enabling local AS numbers to be received in routes:

```
host1#show ip bgp summary fields remote-as state rib-version
send-queue-length more-in-queue
                                            Send More
Neighbor           AS State         RIB Ver    Q InQ
2.2.2.2           100 Established         2    0 no
```

Meaning Table 47 on page 204 lists the **show bgp ipv6 summary** command output fields.

Table 47: show bgp ipv6 summary Output Fields

Field Name	Field Description
Local router ID	Router ID of the local router
local AS	Local autonomous system number
Administrative state	BGP administrative state, start or stop
BGP Operational state	Operational state, up, down, or overload
Shutdown in overload state	Status, enabled or disabled
Default local preference	Default value for local preference
IGP synchronization	Synchronization status, enabled or disabled
Default originate	Whether network 0.0.0.0 is redistributed into BGP (enabled) or not (disabled)
Auto-summary	Status of auto summarization of routes redistributed into BGP

Table 47: show bgp ipv6 summary Output Fields *(continued)*

Field Name	Field Description
Always compare MED	Status, enabled or disabled
Compare MED within confederation	Status, enabled or disabled
Advertise inactive routes	Status, enabled or disabled
Advertise best external route to internal peer	Status, enabled or disabled
Enforce first AS	Status, enabled or disabled
Missing MED as worst	Status, enabled or disabled
Route flap dampening	Status, enabled or disabled
Maximum number of equal-cost EBGP paths	Number of paths
Maximum number of equal-cost IBGP paths	Number of paths
Log neighbor changes	Status, enabled or disabled
Fast External Fallover	Status, enabled or disabled
No maximum received AS-path length	Indicates whether limit is set for AS-path length and, if set, the limit
BGP administrative distances	Distances for external, internal, and local BGP routes
Router is a route reflector	Indicates whether the router has been configured as a route reflector
Client-to-client reflection	Whether client-to-client reflection is configured (enabled) or not (disabled)
Cluster ID	Identifying number for cluster ID
Route-target filter	Status, enabled or disabled
Default IPv4-unicast	Status, enabled or disabled
Redistribution of iBGP routes	Status, enabled or disabled
Check reachability of next-hops for VPN routes	Status, enabled or disabled

Table 47: show bgp ipv6 summary Output Fields *(continued)*

Field Name	Field Description
Graceful restart	Status, enabled or disabled
Global graceful-restart restart time	Time in seconds
Global graceful-restart stale paths time	Time in seconds
Graceful-restart path selection defer time	Time in seconds
Route Distinguisher	RD assigned to the VRF
Confederation ID	Confederation ID
Confederation peers	Confederation peers
Import route map	Route map associated with the VRF that filters and modifies routes imported to the VRF from the global BGP VPN RIB. The map applies to both IPv4 and IPv6 routes, unless the field name is preceded by IPv4 (applies the map to only IPv4 routes) or IPv6 (applies the map to only IPv6 routes).
Export route map	Route map associated with the VRF that modifies and filters routes exported by the VRF to the global BGP VPN RIB. The map applies to both IPv4 and IPv6 routes, unless the field name is preceded by IPv4 (applies the map to only IPv4 routes) or IPv6 (applies the map to only IPv6 routes). The can filter routes text appears only if the filter keyword was issued for export map.
Global import route map	Route map associated with the VRF that modifies routes imported to the VRF from the global BGP non-VPN RIB. The map applies to both IPv4 and IPv6 routes, unless the field name is preceded by IPv4 (applies the map to only IPv4 routes) or IPv6 (applies the map to only IPv6 routes).
routes imported from global table	Number of routes imported from the global BGP non-VPN RIB; also lists the maximum number of routes that can be imported
Global export route map	Route map associated with the VRF that modifies routes exported by the VRF to the global BGP non-VPN RIB. The map applies to both IPv4 and IPv6 routes, unless the field name is preceded by IPv4 (applies the map to only IPv4 routes) or IPv6 (applies the map to only IPv6 routes).
Local-RIB version	Number that is increased by one each time a route in that RIB is added, removed or modified.

Table 47: show bgp ipv6 summary Output Fields *(continued)*

Field Name	Field Description
FIB version	Number that is increased by one each time BGP updates the routes in the IP routing table based on changes in the local RIB. The FIB version matches the local-RIB version when BGP has finished updating the routes in the IP route table. The FIB version is less than the local-RIB version when BGP is still in the process of updating the IP routing table.
Neighbor	BGP neighbors
AS	AS number of the peer
Ver	Negotiated BGP version number
State	State of the connection
Up/down time	Time the connection has been up or down
Messages sent	Number of messages sent to peer
Messages received	Number of messages received from peer
Prefixes received	Number of prefixes received from peer
Rib Ver	Last RIB version queued to be sent to peer
Send Q	Number of messages queued to be sent to peer
More InQ	Status indicating whether any messages are waiting to be sent to peer

Related Documentation

- *show bgp ipv6 summary*
- *show ip bgp summary*

Monitoring the Routes Permitted by IP Community Lists

Purpose

Display community list information.

The display varies if you issue the **ip bgp community new-format** command.

Action

To display community list information without issuing the **ip bgp community new-format** command:

```
host1#show ip community-list
Community List 1:
    permit  81200109
    permit  81200110
    permit  81200108
```

```
Community List 2:
    deny  81200109
    permit  81200110
    permit  81200108
Community List 4:
    permit local-as
Community List 5:
    permit no-advertise
Community List 6:
    permit no-export
Community List 7:
    permit internet
```

To display the community list information by issuing the **ip bgp community new-format** command:

```
host1#show ip community-list
Community List 1:
    permit 1239:1005
    permit 1239:1006
    permit 1239:1004
Community List 2:
    deny 1239:1005
    permit 1239:1006
    permit 1239:1004
Community List 4:
    permit local-as
Community List 5:
    permit no-advertise
Community List 6:
    permit no-export
Community List 7:
    permit internet
```

Meaning Table 48 on page 208 lists the **show ip community-list** command output fields.

Table 48: show ip community-list Output Fields

Field Name	Field Description
Community List 1	Name of the community list
permit, deny	Condition statement for routes matching the condition

Related Documentation

- *show ip community-list*

Disabling Display of BGP Logs

To disable the display of information about BGP logs that was previously enabled with the **debug ip bgp** command.

- Issue the **undebug ip bgp** command:

  host1#**undebug ip bgp**

Related Documentation

- Enabling Display of BGP Logs on page 160
- *debug ip bgp*
- *undebug ip bgp*

PART 2

Multiprotocol Layer Switching

CHAPTER 3

MPLS Overview

This chapter describes Multiprotocol Label Switching (MPLS) and contains the following sections:

MPLS Overview

In conventional IP routing, as a packet traverses from one router to the next through a network, each router analyzes the packet's header and performs a network layer routing table lookup to choose the next hop for the packet. In conventional IP forwarding, the router looks for the address in its forwarding table with the longest match (best match) for the packet's destination address. All packets forwarded to this longest match are considered to be in the same forwarding equivalence class (FEC).

MPLS is a hybrid protocol that integrates network layer routing with label switching to provide a layer 3 network with traffic management capability. MPLS provides traffic-engineering capabilities that make effective use of network resources while maintaining high bandwidth and stability. MPLS enables service providers to provide their customers with the best service available given the provider's resources, with or without traffic engineering. MPLS is the foundation for layer 3 and layer 2 VPNs.

The two basic components of MPLS are label distribution and data mapping.

- Label distribution is the set of actions MPLS performs to establish and maintain a label-switched path (LSP), also known as an *MPLS tunnel*.
- Data mapping is the process of getting data packets onto an established LSP.

Related Documentation

- MPLS Terms and Acronyms on page 216
- Terminology for MPLS Topics on page 214

Terminology for MPLS Topics

Certain terms used with MPLS, such as the names of messages, are often expressed in the RFCs and other sources either with initial uppercase letters or all uppercase letters. For improved readability, those terms are represented in lowercase in this chapter. Table 49 on page 215 lists the terms and some of their variant spellings.

Table 49: Conventions for MPLS Terms

In This Chapter	In RFCs and Other Sources	
ack	Ack	ACK
bundle	Bundle	–
hello	Hello	HELLO
initialization	Initialization	INITIALIZATION
keepalive	Keepalive	KEEPALIVE
label mapping	Label Mapping	LABEL_MAPPING
label release	Label Release	LABEL_RELEASE
label request	Label Request	LABEL_REQUEST
label request abort	Label Request Abort	LABEL_REQUEST_ABORT
label withdrawal	Label Withdrawal	LABEL_WITHDRAWAL
message ack	message_Ack	MESSAGE_ACK
message ID	message_ID	MESSAGE_ID
srfresh	Srefresh	–
path	Path	PATH
patherr	PathErr	PATHERR
pathtear	PathTear	PATHTEAR
resv	Resv	RESV
resvconf	ResvConf	RESVCONF
resverr	ResvErr	RESVERR
resvtear	ResvTear	RESVTEAR
targeted hello	Targeted Hello	TARGETED_HELLO

Related Documentation

- MPLS Terms and Acronyms on page 216
- MPLS Overview on page 214

MPLS Terms and Acronyms

Table 50 on page 216 defines terms and acronyms that are used in this discussion of MPLS.

Table 50: MPLS Terms and Acronyms

Term	Definition
Admission control	Accounting mechanism that tracks resource information. Prevents requests from being accepted if sufficient resources are not available.
BGP	Border Gateway Protocol, which provides loop-free interdomain routing between autonomous systems (ASs) and can act as a label distribution protocol for MPLS.
Branch node	An LSR in a point-to-multipoint LSP that is not an ingress node or an egress node. A branch node can be connected to other branch nodes, an ingress node, or an egress node.
Constraint-based routing	A mechanism to establish paths based on certain criteria (explicit route, QoS parameters). The standard routing protocols can be enhanced to carry additional information to be used when running the route calculation.
E-LSP	EXP-inferred-PSC LSP. The EXP field of the MPLS Shim Header is used to determine the per-hop behavior applied to the packet.
Explicit routing	A subset of constraint-based routing where the constraint is an explicit route
FEC	Forwarding equivalence class—Group of IP packets forwarded over the same path with the same path attributes applied
Label Distribution Protocol	• A particular label distribution protocol used for label distribution among the routers in an MPLS domain; represented by the acronym LDP • In lowercase—label distribution protocol—a generic term for any of several protocols that distribute labels among the routers in an MPLS domain, including BGP, LDP, and RSVP-TE. This usage is **not** represented in this text by the acronym, LDP.
Leaf	Egress LSRs in a point-to-multipoint LSP. It is also referred as a leaf node.
LDP	Label Distribution Protocol—A particular protocol used for label distribution among the routers in an MPLS domain This text does **not** use LDP to refer to the generic class of label distribution protocols.
LER	Label edge router—A label-switching router serving as an ingress or egress nodes

Table 50: MPLS Terms and Acronyms *(continued)*

Term	Definition
LSP	Label-switched path—The path traversed by a packet that is routed by MPLS. Some LSPs act as tunnels.
LSP priority level	A priority that indicates the importance of one LSP relative to another LSP. LSPs having higher priorities can preempt LSPs having lower priorities. Priorities range from 0 through 7 in order of *decreasing* priority.
L-LSP	Label-only-inferred-PSC LSP. The label value, and possibly the EXP-bits, are used to determine the per-hop behavior applied to the packet.
LSR	Label-switching router—An MPLS node that can forward layer 3 packets based on their labels
MPLS	Multiprotocol Label Switching—Set of techniques enabling forwarding of traffic using layer 2 and layer 3 information
MPLS edge node	MPLS node that connects an MPLS domain with a node outside the domain that either does not run MPLS or is in a different domain
MPLS egress node	MPLS edge node in the role of handling traffic as it leaves an MPLS domain
MPLS ingress node	MPLS edge node in the role of handling traffic as it enters an MPLS domain
MPLS label	Label carried in a packet header that represents a packet's forwarding equivalence class
MPLS node	A router running MPLS. An MPLS node is aware of MPLS control protocols, operates one or more L3 routing protocols, and is capable of forwarding packets based on labels. Optionally, an MPLS node can be capable of forwarding native L3 packets.
Point-to-multipoint tunnel	The series of LSRs and links that form the path from an ingress LSR to all of its egress LSRs. Each tunnel is uniquely identified by a session object.
Point-to-multipoint LSP	An RSVP-TE LSP with a single ingress LSR and one or more egress LSRs. Incoming data is replicated at the branch nodes.
Provider edge router	PE—An LER at the edge of a service provider core that provides ingress to or egress from a VPN
Provider core router	P—An LSR within a service provider core that carries traffic for a VPN
RSVP	Resource Reservation Protocol; E Series routers do not support RSVP

Table 50: MPLS Terms and Acronyms *(continued)*

Term	Definition
RSVP-TE	Resource Reservation Protocol enhanced to support MPLS traffic engineering; E Series routers support RSVP-TE
Sub-LSP	The portion of the LSP from one LSR to another LSR in a point-to-multipoint tunnel.
Traffic engineering	The ability to control the path taken through a network or portion of a network based on a set of traffic parameters (bandwidth, QoS parameters, and so on). Traffic engineering (TE) enables performance optimization of operational networks and their resources.
Tunnel	LSP that is used by an IGP to reach a destination, or an LSP that uses traffic engineering

Related Documentation

- MPLS Overview on page 214
- Terminology for MPLS Topics on page 214

MPLS Features

The following major features are currently supported by MPLS:

- BFD fast failure detection for RSVP-TE adjacencies
- Differentiated services
- Interface support
  - ATM AAL5 (RSVP-TE only)
  - ATM1483 (point-to-point AAL5SNAP only)
  - Ethernet/VLAN
  - GRE
  - Multilink PPP
  - POS (PPP over HDLC)
  - PPP
  - SLEP (Cisco HDLC)
- Label stacking
  - Virtual Private Networks (VR-based and BGP-based)
  - Layer 2 Services over MPLS
- LER functionality
- LSR functionality

- Spoof checking
- LDP graceful restart
- ECMP
- Topology-driven LSPs (LDP) including support of LDP over RSVP tunnels
- Traffic engineering (RSVP-TE)
  - Constraint-based explicit routing
  - Statically configured explicit routing
  - Hop-by-hop routing
  - Admission control and bandwidth enforcement
  - Tunnels used by IGP as next hops in SPF calculation
  - Tunnel ingress controlled failure recovery
  - Facility back-up-style fast-route
- Traffic support
  - Layer 2 frames: ATM, Ethernet, Frame Relay, HDLC, PPP, VLAN
  - Layer 3 datagrams: IPv4, IPv6
- Point-to-multipoint LSP support
  - Data replication at branch nodes
  - E Series routers as egress LSRs

Related Documentation

- MPLS Overview on page 214
- Terminology for MPLS Topics on page 214
- MPLS Terms and Acronyms on page 216

MPLS Platform Considerations

For information about modules that support MPLS on the ERX7xx models, ERX14xx models, and the ERX310 Broadband Services Router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support BGP.

For information about modules that support MPLS on E120 and E320 Broadband Services Routers:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.

- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support MPLS.

Related Documentation

- *Interface Types and Specifiers*

MPLS References

For more information about the MPLS protocol, consult the following resources:

- *JunosE Release Notes*, *Appendix A*, *System Maximums*—Refer to the Release Notes corresponding to your software release for information about maximum values.
- Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)—draft-ietf-mpls-in-ip-or-gre-03.txt (September 2003 expiration)
- LDP IGP Synchronization—draft-jork-ldp-igp-sync-01.txt (August 2005 expiration)
- Detecting Data Plane Failures in Point-to-Multipoint Multiprotocol Label Switching (MPLS) - Extensions to LSP Ping—draft-ietf-mpls-p2mp-lsp-ping-08.txt (February 2010 expiration)
- RFC 2104—HMAC: Keyed-Hashing for Message Authentication (February 1997)5.2
- RFC 2205—Resource ReSerVation Protocol (RSVP) -- Version 1, Functional Specification (September 1997)
- RFC 2209—Resource ReSerVation Protocol (RSVP) -- Version 1, Message Processing Rules (September 1997)
- RFC 2210—The Use of RSVP with IETF Integrated Services (September 1997)
- RFC 2211—Specification of the Controlled-Load Network Element Service (September 1997)
- RFC 2474—Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers (December 1998)
- RFC 2475—An Architecture for Differentiated Services (December 1998)
- RFC 2597—Assured Forwarding PHB Group (June 1999)
- RFC 2685—Virtual Private Networks Identifier (September 1999)
- RFC 2702—Requirements for Traffic Engineering over MPLS (September 1999)
- RFC 2747—RSVP Cryptographic Authentication (January 2000)
- RFC 2836—Per Hop Behavior Identification Codes (May 2000)
- RFC 4760—Multiprotocol Extensions for BGP-4 (January 2007)
- RFC 2961—RSVP Refresh Overhead Reduction Extensions (April 2001)
- RFC 3031—Multiprotocol Label Switching Architecture (January 2001)
- RFC 3032—MPLS Label Stack Encoding (January 2001)
- RFC 3035—MPLS using LDP and ATM VC Switching (January 2001)

- RFC 3036—LDP Specification (January 2001)
- RFC 3037—LDP Applicability (January 2001)
- RFC 3097—RSVP Cryptographic Authentication -- Updated Message Type Value (April 2001)
- RFC 3107—Carrying Label Information in BGP-4 (May 2001)
- RFC 3140—Per Hop Behavior Identification Codes (June 2001)
- RFC 3209—RSVP-TE: Extensions to RSVP for LSP Tunnels (December 2001)
- RFC 3210—Applicability Statement for Extensions to RSVP for LSP-Tunnels (December 2001)
- RFC 3246—An Expedited Forwarding PHB (Per-Hop Behavior) (March 2002)
- RFC 3270—Multi-Protocol Label Switching (MPLS) Support of Differentiated Services (May 2002)
- RFC 3443—Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks (January 2003)
- RFC 3471—Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description (January 2003)
- RFC 3473—Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions (January 2003)
- RFC 3478—Graceful Restart Mechanism for Label Distribution Protocol (February 2003)
- RFC 3479—Fault Tolerance for the Label Distribution Protocol (LDP) (February 2003)
- RFC 3564—Requirements for support of Differentiated Services-aware MPLS Traffic Engineering (July 2003)
- RFC 4090—Fast Reroute Extensions to RSVP-TE for LSP Tunnels (May 2005)
- RFC 4364—BGP/MPLS IP Virtual Private Networks (VPNs) (February 2006)
- RFC 4379—Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures (February 2006)
- RFC 4661—Signaling Requirements for Point-to-Multipoint Traffic-Engineered MPLS Label Switched Paths (LSPs) (April 2006)
- RFC 4875—Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs) (May 2007)

NOTE: **IETF drafts are valid for only 6 months from the date of issuance. They must be considered as works in progress. Please refer to the IETF Website at http://www.ietf.org for the latest drafts.**

Related Documentation

- MPLS Overview on page 214

MPLS Label Switching and Packet Forwarding Overview

MPLS is not a routing protocol; it works with layer 3 routing protocols (BGP, IS-IS, OSPF) to integrate network layer routing with label switching. An MPLS FEC consists of a set of packets that are all forwarded in the same manner by a given label-switching router (LSR). For example, all packets received on a particular interface might be assigned to a FEC. MPLS assigns each packet to a FEC only at the LSR that serves as the ingress node to the MPLS domain. A label distribution protocol binds a label to the FEC. Each LSR uses the label distribution protocol to signal its forwarding peers and distribute its labels to establish an LSP. The label distribution protocol enables negotiation with the downstream LSRs to determine what labels are used on the LSP and how they are employed.

Labels represent the FEC along the LSP from the ingress node to the egress node. The label is prepended to the packet when the packet is forwarded to the next hop. Each label is valid only between a pair of LSRs. A downstream LSR reached by a packet uses the label as an index into a table that contains both the next hop and a different label to prepend to the packet before forwarding. This table is usually referred to as a label information base (LIB).

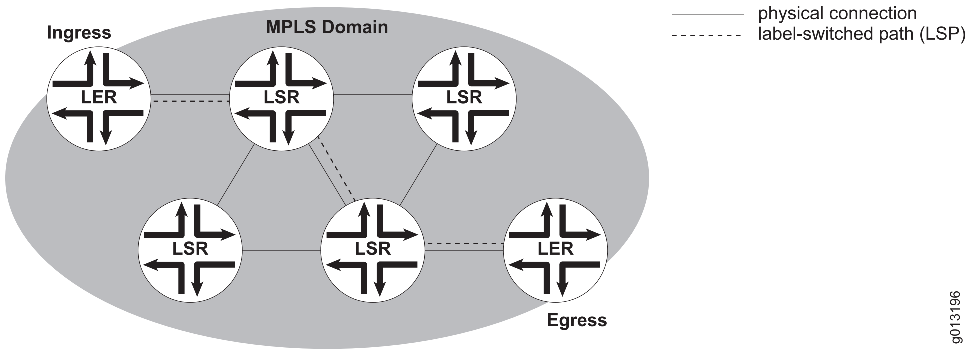
The LSR that serves as the egress MPLS node uses the label as an index into a table that has the information necessary to forward the packet from the MPLS domain. The forwarding actions at the egress LSR can be any of the following:

- Forward the packet based on the inner header exposed after popping the label. This can be accomplished either by doing a routing table lookup or forwarding based on the exposed inner MPLS label.
- Forward the packet to a particular neighbor as directed by the table entry, for example in a Martini layer 2 transport case.

NOTE: **Forwarding of traffic for labeled IPv6 unicast routes with native IPv6 next-hop addresses does not work.**

Figure 47 on page 223 shows a simple MPLS domain, consisting of multiple LSRs. The LSRs serving as ingress and egress nodes are also referred to as *label edge routers* (LERs). The ingress router is sometimes referred to as the *tunnel head end*, or the *head-end* router. The egress router is sometimes referred to as the *tunnel tail end*, or the *tail-end* router. LSPs are *unidirectional*, carrying traffic only in the downstream direction from the ingress node to the egress node.

Figure 47: Simple MPLS Domain



MPLS LSRs

Each LSR, also known as an MPLS node, must have the following:

- At least one layer 3 routing protocol
- A label distribution protocol
- The ability to forward packets based on their labels

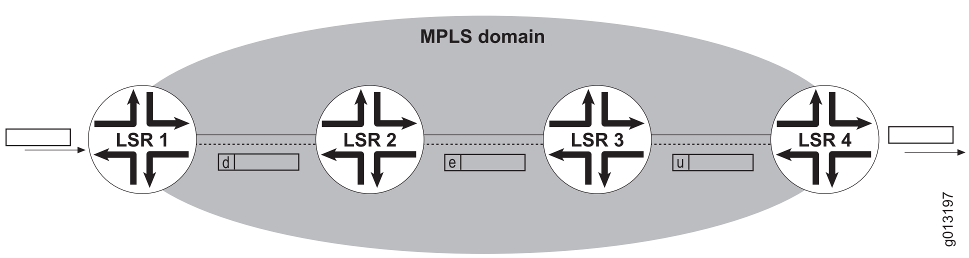
The router can use BGP, IS-IS, or OSPF as its layer 3 routing protocol, and BGP, LDP, or RSVP-TE as its label distribution protocol.

MPLS Label Switching: Push, Look Up, and Pop

MPLS can label packets by using the existing layer 2 header or an encapsulation header that carries the MPLS label. During LSP negotiation, the LSRs in an MPLS domain agree on a labeling method. Labels have only local meaning; that is, meaning for two LSR peers. Each pair of LSRs—consisting of a label originator and a label acceptor—must use a label distribution protocol to agree on the label-to-FEC binding.

Because of the local label assignment, packet labels typically change at each segment in the LSP path, as shown in Figure 48 on page 223. The ingress node, LSR 1, receives an unlabeled data packet and prepends label *d* to the packet. LSR 2 receives the packet, removes label *d* and uses it as an index in its forwarding table to find the next label. LSR 2 prepends label *e* to the packet. LSR 3 does the same thing, removing label *e* and prepending label *u*. Finally, the egress node, LSR 4, removes label *u* and determines where to forward the packet outside the MPLS domain.

Figure 48: Label Switching



Any packet can carry multiple labels. The labels are stacked in a last-in-first-out order. Each LSR forwards packets based on the outermost (top) label in the stack. An LSR *pushes* a label onto the stack when it prepends the label to a packet header. It *pops* the label when it pulls the label off the stack and compares it with the forwarding table. On determining the label for the next segment of the LSP, the LSR pushes the new label on the stack. A label swap consists of a pop, lookup, and push.

When the egress router, such as LSR 4 in Figure 48 on page 223, receives a packet, it may perform two lookups: it looks up the label and determines that the label must be popped, then it does another lookup based on the exposed header to determine where to forward the packet. This behavior is known as *ultimate hop popping*, and was the only possible action for the JunosE implementation before Release 7.3.0.

Beginning with JunosE Release 7.3.0, an alternative behavior, known as *penultimate hop popping* (PHP), is the default when RSVP-TE is the signaling protocol. Beginning with JunosE Release 8.1.0, PHP is also the default when LDP is the signaling protocol. PHP reduces the number of lookups performed by the LER. In PHP, the LER requests its upstream neighbor (the penultimate hop) to pop the outermost label and send just the packet to the LER. The LER then performs only the lookup for the packet. The request to perform PHP is signaled by the LER when it includes an implicit null label in the label mapping message that it sends to its upstream neighbor. The implicit null label never appears in the encapsulation.

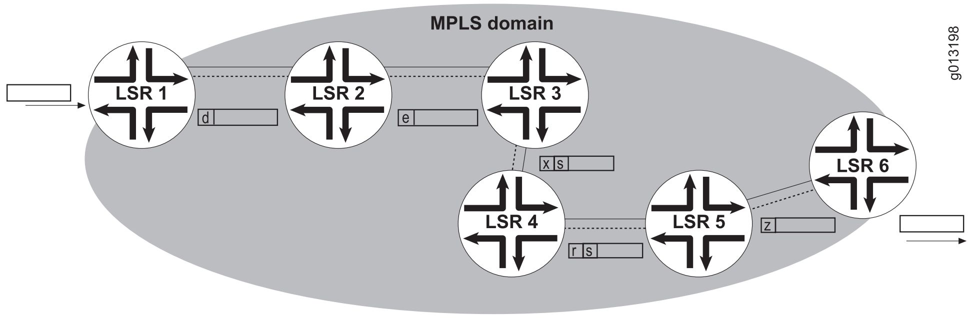
You can still achieve ultimate hop popping by configuring the egress router to advertise an explicit null label to its upstream neighbor. This advertisement, performed by LDP or RSVP-TE, ensures that all MPLS packets traversing the LSP to the egress router include a label. Alternatively, you can configure the egress router to advertise real (non-null) labels, and achieve the same result.

Regardless of whether the LSR advertises the implicit null label to achieve PHP on an upstream neighbor, if the LSR receives a PHP request from a downstream neighbor, then the LSR does perform the PHP for its neighbor.

MPLS Label Stacking

Figure 49 on page 225 shows an LSP that uses label stacking. The ingress node, LSR 1, receives an unlabeled data packet and prepends label *d* to the packet. LSR 2 receives the packet, removes label *d* and uses it as an index in its forwarding table to find the next label, prepending label *e* to the packet. LSR 3 removes label *e* and prepends label *s* (negotiated with LSR 5) to the packet. LSR 3 pushes label *x* on top of label *s*. LSR 4 pops the top (outermost) label, *x*, and pushes label *r* on top of label *s*. LSR 5 pops label *r*, determines that it must pop label *s*, and pushes label *z* on the empty stack. Finally, the egress node, LSR 6, removes label *z* and determines where to forward the packet outside the MPLS domain.

Figure 49: Label Stacking



The configuration shown in Figure 49 on page 225 is an example of an LSP within an LSP (a tunnel within a tunnel). The first LSP consists of LSR 1, LSR 2, LSR 3, LSR 5, and LSR 6. The second LSP consists of LSR 3, LSR 4, and LSR 5. The two LSPs have different ingress and egress points. LSR 1 and LSR 6 are LERs. Less obviously, LSR 3 and LSR 5 are also LERs, but for the internal LSP.

NOTE: Label stacking is typically employed for LSR peers that are not directly connected. Figure 49 on page 225 is a simplified example to illustrate the concept of label stacking.

MPLS Labels and Label Spaces

MPLS uses labels from either the *platform label space* or the *interface label space*. ATM AAL5 interfaces always use labels from only the interface label space. For every interface using the interface label space, you must define the range available to the router for labels in the interface label space. All other interface types always use labels from only the platform label space. You cannot configure the range for the platform label space.

The platform label space is a large, single, unconfigurable pool of labels that can be shared by the platform—all MPLS interfaces on a given virtual router. By contrast, interface labels enable you to effectively create multiple smaller pools of labels, each used only by a particular interface. When you configure interface labels, you restrict only a given interface to a particular range of labels. Other interfaces in that VR can still use labels from that space unless you restrict them in turn to a different range of interface labels.

In the interface label space, MPLS selects labels from interface resources, a VPI/VCI combination. You configure a VPI range and a VCI range available to the labels. When an upstream LSR requests a label, the downstream LSR allocates a VPI/VCI combination for use as a label between these two peers. Allocating labels on a per interface basis is necessary because the VPI/VCI ranges are limited. This enables you to use the same label on different interfaces without conflict.

When you use the platform label space, the MPLS ingress node places labels in *shim headers* between the link-layer header and the payload. The shim header includes the following bits (Figure 50 on page 226):

- Label bits—Twenty bits
- EXP bits—Three bits for class of service information; these bits are variously called the experimental bits, class of service (CoS) bits, or type of service (ToS) bits. The EXP bits are mapped from the IP packet at the ingress node and are mapped back into the IP packet at the egress node.
- S bit—One bit to indicate whether the label is on the bottom of the label stack.
- TTL bits—Eight bits for a time-to-live indicator. The TTL bits are mapped from the IP packet at the ingress node. The TTL bits in the shim header are decremented at each hop. The bits are mapped back into the IP packet at the egress node. See "TTL Processing in the Platform Label Space Overview" on page 226 for more information.

Figure 50: Shim Header

If you configure an MPLS interface to use the interface label space, the VPI/VCI combinations are used as labels, so there is no need to place them within a shim header. As the data travels along the LSP, the LSRs examine only the VPI/VCI combination. The shim header is used only to carry the TTL bits to the egress, and is not visible to intermediate LSRs. The ingress node learns the total hop count from signaling and then uses that count to decrement the TTL to the correct final value. The TTL is then carried in the shim header to the egress node without modification, arriving with the correct count.

Related Documentation

- MPLS Label Distribution Methodology on page 231
- IP Data Packet Mapping onto MPLS LSPs Overview on page 233
- MPLS Forwarding and Next-Hop Tables Overview on page 237
- MPLS Interfaces and Interface Stacking Overview on page 240
- TTL Processing in the Platform Label Space Overview on page 226
- Topology-Driven LSPs Overview on page 259

TTL Processing in the Platform Label Space Overview

JunosE MPLS TTL processing is compliant with RFC 3443. The details of TTL processing vary with the tunnel model that is configured for TTL processing, pipe or uniform.

To keep backward compatibility with earlier JunosE releases, you do not use the **mpls tunnel-model** command to configure the tunnel model for TTL processing, That command is used instead to configure the tunnel model for EXP bits processing. The default tunnel model varies between TTL and EXP processing; for EXP processing, the default tunnel model is pipe, while for TTL processing the default tunnel model is uniform.

You can issue the **no mpls ip propagate-ttl** command to change the TTL processing tunnel model from the default uniform model to the pipe model. Issue the **no mpls ip**

propagate-ttl local command to set the tunnel model to pipe for locally originated packets. Issue the **no mpls ip propagate-ttl forwarded** command to set the tunnel model to pipe for forwarded packets.

TTL Processing on Incoming MPLS Packets

The flow chart on Figure 51 on page 228 illustrates TTL processing on incoming MPLS packets. On a transit LSR or an egress LER, MPLS pops one or more labels and can push one or more labels. The incoming TTL of the packet is determined by the configured TTL processing tunnel model.

When all of the following conditions are met, the incoming TTL is set to the TTL value found in the immediate inner header:

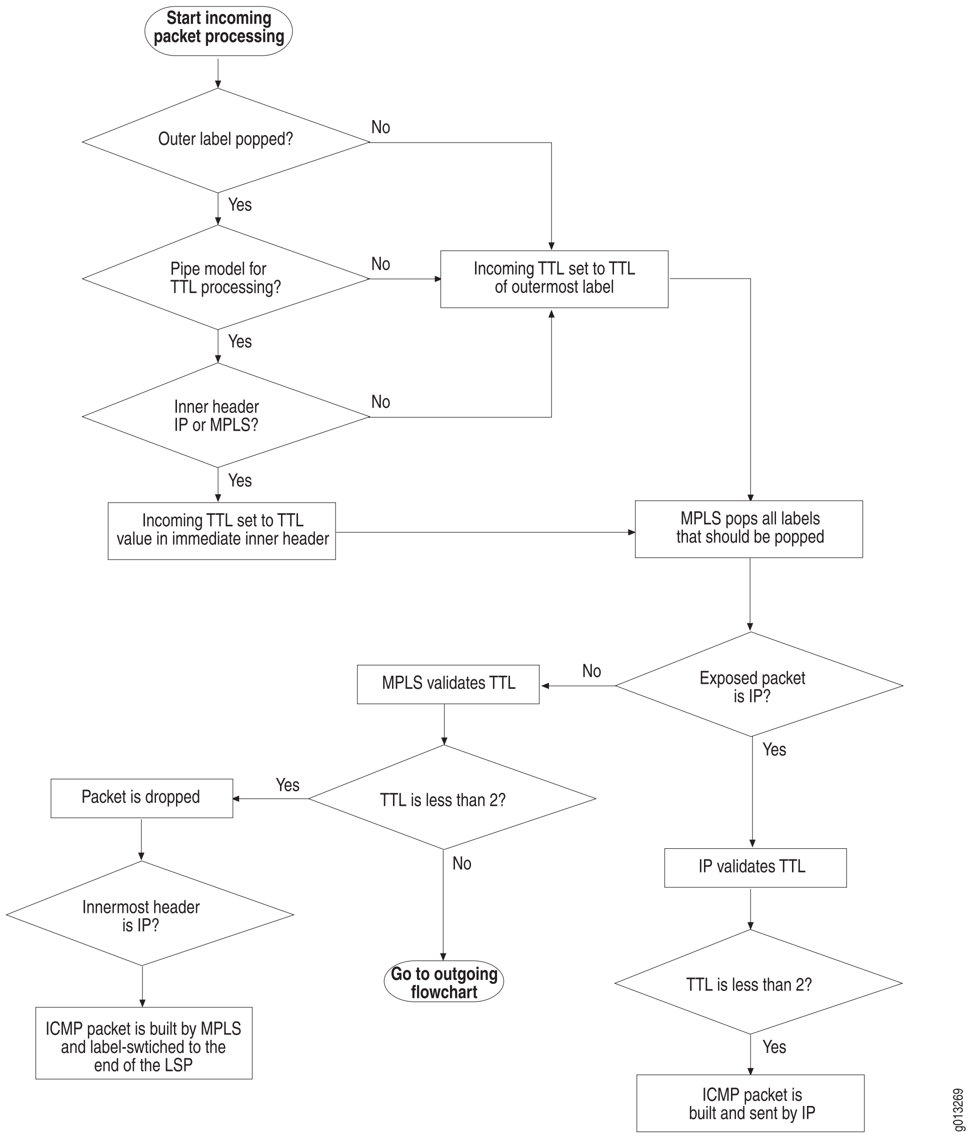
- The outer label is popped as opposed to being swapped
- The TTL processing model is configured to pipe
- The inner header is MPLS or IP

If any of those conditions is not met, then the incoming TTL is set to the TTL value found in the outermost label. In all cases, the TTL values of any further inner labels are ignored.

When an IP packet is exposed after MPLS pops all the labels that should be popped, MPLS passes the packet to IP for further processing, including TTL checking. When the uniform tunnel model for TTL processing is in effect, MPLS sets the TTL value of the IP packet to the incoming TTL value that was just set. In other words, the TTL value is copied from the outermost label to the IP packet. When the pipe model for TTL processing is in effect, the TTL value in the IP header is left unchanged.

If an IP packet is not exposed by the label popping, then MPLS performs the TTL validation. If the incoming TTL is less than 2, the packet is dropped. If innermost packet is IP, an ICMP packet is built and sent. If the TTL does not expire and the packet needs to be sent out, the outgoing TTL is determined by the rules for outgoing MPLS packets.

Figure 51: TTL Processing on Incoming MPLS Packets



TTL Processing on Outgoing MPLS Packets

The flow chart on Figure 52 on page 230 illustrates TTL processing on outgoing MPLS packets.

Rules for Processing on an LSR

On an LSR—where an MPLS packet is label-switched after processing on the line module—the TTL value in the swapped-to label is decremented by 1 from the incoming TTL value when the swapped-to label is not implicit-null. When the swapped-to label is implicit-null (for example, in a PHP configuration), the inner or exposed header's TTL is either left unchanged (when the **forwarded** option for the **mpls ip propagate-ttl** command has been configured) or is decremented by 1 from the incoming TTL value. If

MPLS needs to push more labels, it sets the TTL for each label according to the following LER rules, because for those labels the router effectively is an ingress LER.

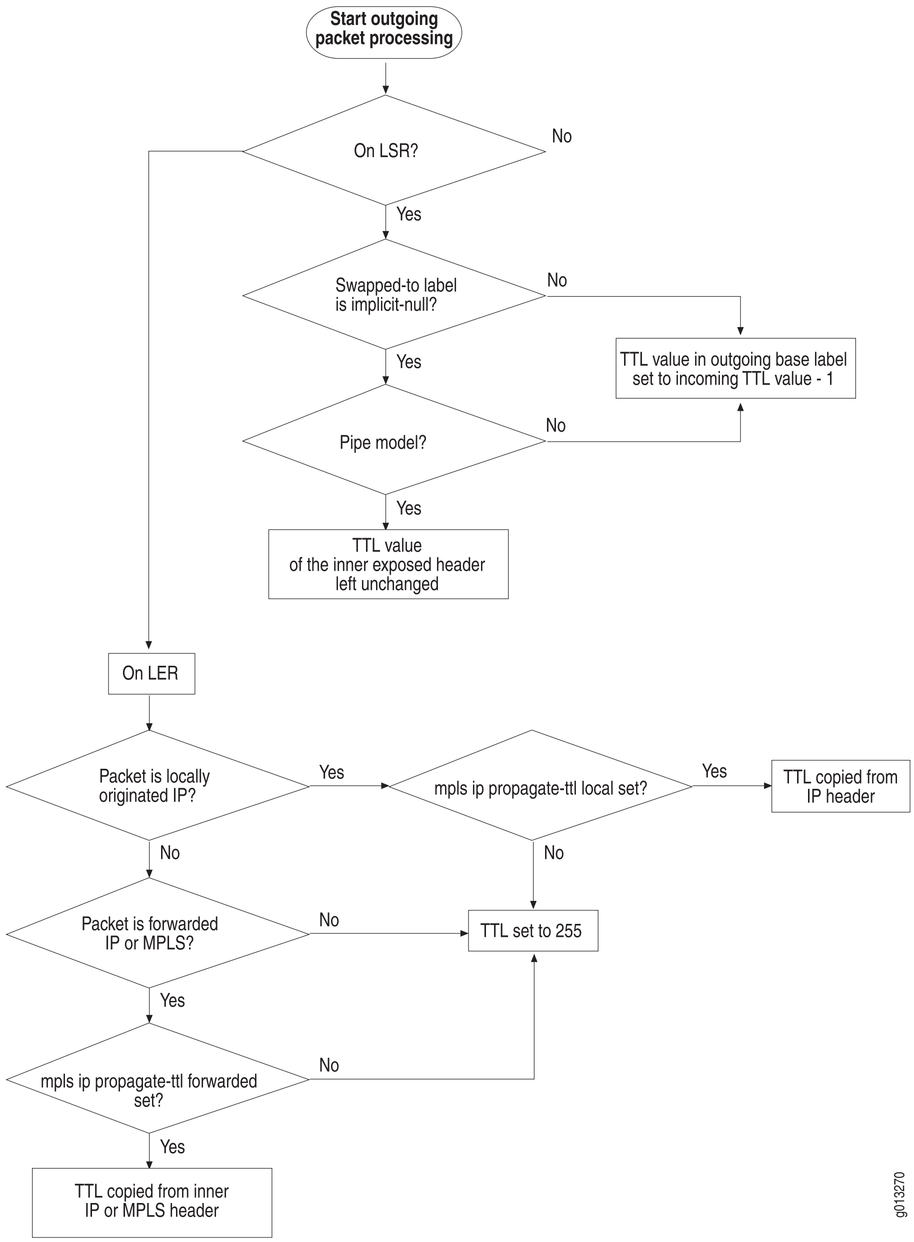
Rules for Processing on an LER

On an LER, when the packet is a locally originated IP packet, MPLS copies the TTL of all pushed MPLS labels from the IP header when the **local** option for the **mpls ip propagate-ttl** command has been configured. When the **no mpls ip propagate-ttl local** command has been configured, MPLS sets the TTL to 255.

When the packet is a forwarded IP or MPLS packet, MPLS copies the TTL of all pushed labels from the inner IP or MPLS header when the **forwarded** option for the **mpls ip propagate-ttl** command has been configured. When the **no mpls ip propagate-ttl forwarded** command has been configured, MPLS sets the TTL for these pushed labels to 255.

When the packet is neither IP nor MPLS, such as a Martini packet, MPLS sets the TTL of all pushed labels to 255.

Figure 52: TTL Processing on Outgoing MPLS Packets



MPLS Rules for TTL Expiration

MPLS takes the following actions when the TTL in a MPLS label of a received MPLS packet expires:

1. A TTL-expired ICMP packet is constructed.
2. The destination address of ICMP packet is set to the source address of the IP packet that was encapsulated in the MPLS packet.

3. The source address of ICMP packet is set to the router ID of the router on which the TTL expired.
4. The first 128 bytes of the MPLS packet including the IP payload encapsulated in the MPLS packet are copied into the payload of the ICMP packet, followed by the entire label stack of the original packet.

The ICMP packet is label-switched to the end of the LSP. From that location, the packet is forwarded back to the source of the IP packet. This behavior enables IP trace-route to work even when the LSR in the middle of the LSP does not have an IP route to the source address of the IP packet.

Related Documentation

- MPLS Label Distribution Methodology on page 231
- IP Data Packet Mapping onto MPLS LSPs Overview on page 233
- MPLS Label Switching and Packet Forwarding Overview on page 222
- MPLS Forwarding and Next-Hop Tables Overview on page 237
- MPLS Interfaces and Interface Stacking Overview on page 240
- Topology-Driven LSPs Overview on page 259

MPLS Label Distribution Methodology

The JunosE implementation of MPLS supports the following methods of label distribution:

- Downstream-on-demand, ordered control with RSVP-TE
- Downstream-unsolicited, independent control or ordered control with LDP; ordered control is the default. BGP accepts only downstream-unsolicited, ordered control

Downstream-on-demand means that MPLS devices do not signal a FEC-to-label binding until requested to do so by an upstream device. Upstream is the direction toward a packet's source; the ingress node in an MPLS domain is the farthest possible upstream node. Downstream is the direction toward a packet's destination; the egress node in an MPLS domain is the farthest possible downstream node. The egress node is sometime referred to as the *tunnel endpoint*.

Downstream-on-demand conserves labels in that they are not bound until they are needed and the LSR receives label mappings (also known as label bindings) from a neighbor that is the next hop to a destination; it is used when RSVP is the signaling protocol.

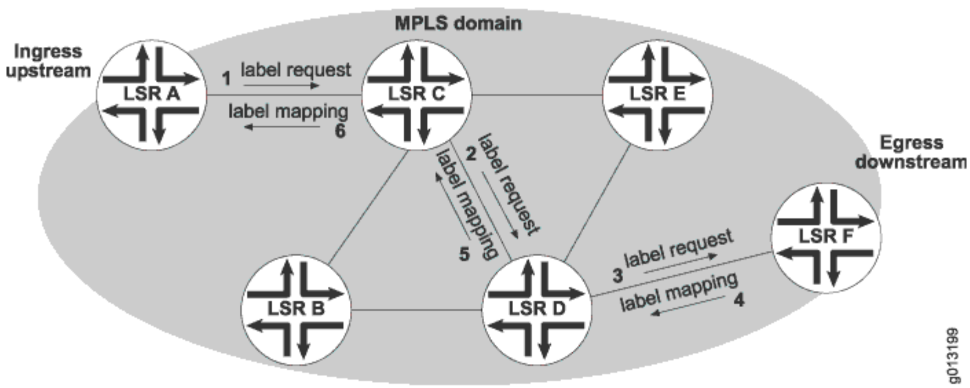
Ordered control means that an LSR does not advertise a label for a FEC unless it is the egress LSR for the FEC or until it has received a label for the FEC from its downstream peer. In this manner the entire LSP is established before MPLS begins to map data onto the LSP, preventing inappropriate (early) data mapping from occurring on the first LSR in the path.

An LSR is an egress LSR for a FEC when the FEC is its directly attached interface or when MPLS is not configured on the next-hop interface.

In Figure 53 on page 232, LSR A sends a label request to LSR C. Before LSR C responds, it sends its own request to LSR D. LSR D in turn makes a request for a label to LSR F. When LSR F returns an acceptable label to LSR D, that label is for use only between LSRs D and F. LSR D sends a label back to LSR C that this pair of LSRs will use. Finally, LSR C sends back to LSR A the label that they will use. This completes the establishment of the LSP.

Downstream-unsolicited means that MPLS devices do not wait for a request from an upstream device before signaling FEC-to-label bindings. As soon as the LSR learns a route, it sends a binding for that route to all peer LSRs, both upstream and downstream. Downstream-unsolicited does not conserve labels, because an LSR receives label mappings from neighbors that might not be the next hop for the destination; it is used by BGP or LDP when adjacent peers are configured to use the platform label space.

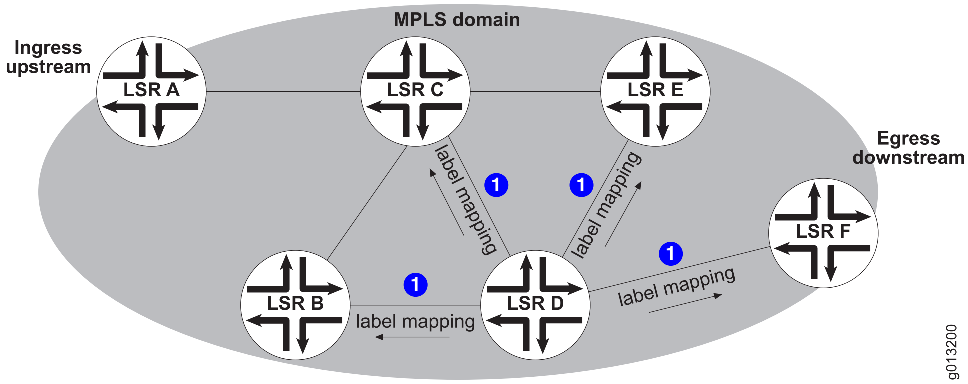
Figure 53: LSP Creation, Downstream-on-Demand, Ordered Control



Independent control means that the LSR sending the label acts independently of its downstream peer. It does not wait for a label from the downstream LSR before it sends a label to its peers. When an LSR advertises a label to an upstream neighbor before it has received a label for the FEC from the next-hop neighbor, the LSP is terminated at the LSR. Traffic for the destination cannot be label-switched all the way to the egress LSR. If no inner label is present, then the traffic is routed instead of switched.

In Figure 54 on page 233, LSR D learns a route to some prefix. LSR D immediately maps a label for this destination and sends the label to its peers, LSR B, LSR C, LSR E, and LSR F. In the topology-driven network, the LSPs are created automatically with each peer LSR.

Figure 54: LSP Creation, Downstream-Unsolicited, Independent Control



Related Documentation

- TTL Processing in the Platform Label Space Overview on page 226
- IP Data Packet Mapping onto MPLS LSPs Overview on page 233
- MPLS Label Switching and Packet Forwarding Overview on page 222
- MPLS Forwarding and Next-Hop Tables Overview on page 237
- MPLS Interfaces and Interface Stacking Overview on page 240
- Topology-Driven LSPs Overview on page 259

IP Data Packet Mapping onto MPLS LSPs Overview

IP packets are mapped onto LSPs by one of the following methods:

- RSVP-TE tunnels can be referenced directly by static routes that you configure. You can determine which routes (routes destined for which subnets) to direct through the LSP and issue the appropriate **ip route** commands, as shown in the following example:

  host1(config-if)#**ip route 10.15.21.16 tunnel mpls:1**

  You cannot create any static routes until the tunnel interface has been created. However, the tunnel does not have to be active before you create the static routes.

- RSVP-TE tunnels are announced to IS-IS and OSPF; the IGP then uses the tunnels as next hop interfaces for its SPF calculations. For this method, you must issue the **tunnel mpls autoroute announce** command. When the LSP is established, the ingress LSR announces the LSP endpoint to the IGP. This is also referred to as *registering* the LSP. The IGP then recalculates the shortest path for all routes destined for or beyond that endpoint. You can choose to register endpoints with both IS-IS and OSPF. The following is an example registration command:

  host1(config-if)#**tunnel mpls autoroute announce isis**

- For topology-driven LSPs, LDP can modify the IP routing table to use MPLS next hops in the routing table, replacing the regular IP next hops for the corresponding routes.
- For labeled BGP routes, BGP adds routes with MPLS next hops to the appropriate VR or VRF routing table.

When IP packets arrive at the ingress LER, they are looked up in the relevant IP forwarding table and then are forwarded into an LSP. Every IP route eventually points to an IP interface. The IP interface contains IP attributes that affect how the IP packet is forwarded. IPv4 routes point only to IPv4 interfaces and IPv6 routes point only to IPv6 interfaces.

Because IP routes cannot point directly to MPLS major interfaces, MPLS automatically creates up to four dynamic IP shared interfaces that are stacked on each MPLS major interface. When you issue the **mpls** command in Interface Configuration mode, the interfaces are created dynamically and provide the interfaces an IP route needs to point to. You can specify a profile (created with the **profile** command) to configure attributes for these interfaces with the **mpls create-dynamic-interfaces** command. You can use the same command to enable or disable the creation of specific interface types or all types.

Each dynamic interface is one of the following types:

- By default, MPLS creates one dynamic IPv4 interface per MPLS major interface for non-VPN traffic. This interface is used by default for VPN traffic as well.
- By default, but only if IPv6 is enabled in the virtual router, MPLS creates one dynamic IPv6 interface per MPLS major interface for non-VPN traffic. This interface is used by default for VPN traffic as well.
- If you configure it to do so, MPLS creates one dynamic IPv4 interface per MPLS major interface for VPN traffic. If this interface is not created, then the VPN traffic uses the default IPv4 interface for non-VPN traffic.

  Typically, you request the creation of separate IPv4 interfaces for VPN traffic only when you want the IPv4 interface for VPN traffic to have different attributes, such as a different IP policy, from the IPv4 interface for non-VPN traffic. When it is acceptable for the VPN traffic and the non-VPN traffic to receive the same IP treatment, then you do not need to create separate IPv4 interfaces for the VPN traffic.
- If you configure it to do so, but only if IPv6 is enabled in the virtual router, MPLS creates one dynamic IPv6 interface per MPLS major interface for VPN traffic. If this interface is not created, then the VPN traffic uses the default IPv6 interface for non-VPN traffic.

  Typically, you request the creation of separate IPv6 interfaces for VPN traffic only when you want the IPv6 interface for VPN traffic to have different attributes, such as a different IP policy, from the IPv6 interface for non-VPN traffic. When it is acceptable for the VPN traffic and the non-VPN traffic to receive the same IP treatment, then you do not need to create separate IPv6 interfaces for the VPN traffic.

IPv6 must be enabled in the parent virtual router so that IPv6 dynamic interfaces can be created over MPLS interfaces. Otherwise, IPv6 VPNs do not work correctly,

All VPN traffic sent onto or received from the same layer 2 interface uses the same IPv4 VPN or IPv6 VPN interface. Consequently, any policy attached to the interface applies to all that VPN traffic.

Related Documentation

- TTL Processing in the Platform Label Space Overview on page 226
- MPLS Label Switching and Packet Forwarding Overview on page 222

Statistics for IP Packets Moving On or Off MPLS LSPs

In the earlier architecture, the statistics for IP packets moving onto or off an LSP applied to the IP interface that was stacked on top of the LSP. Because IP interfaces are no longer stacked on LSPs in the current architecture, these statistics apply to one of the dynamic IP interfaces that is established on the same major interface the LSP is stacked on.

However, even though the IP traffic leaving the LSP as MPLS packets is associated with a dynamic IP interface, it does not use the queue associated with that dynamic IP interface. Instead, the IP traffic uses the queues at the layer 2 interface; the MPLS major interface on which the dynamic IP interface is created does not have its own queue. As a result, queue-related statistics do not increase with the IP traffic flow on the dynamic IP interfaces. This behavior can create some confusion when you examine the output from commands such as **show egress-queue rate interface ip**.

In the following sample output, the statistics of interest are those for the layer 2 interface, atm-vc ATM9/0.10. Traffic is present as indicated by the forwarded rate value for the layer 2 interface. If no IP traffic is present, the forwarded rate for the layer 2 interface has a value of 0.

```
host1:pe1# show egress-queue rates interface atm9/0.10
                              traffic   forwarded aggregate minimum  maximum
         interface             class      rate    drop rate  rate     rate
--------------------------- ----------- --------- --------- ------- ---------
atm-vc ATM9/0.10            best-effort    116032         0 4680000 149760000
ip ATM9/0.10                best-effort         0         0 4680000 149760000
ip ip19000001.mpls.ip       best-effort         0         0 4680000 149760000
ipv6 ipv619000001.mpls.ipv6 best-effort         0         0 4680000 149760000
  Queues reported:                      4
  Queues filtered (under threshold):   0
* Queues disabled (no rate period):    0
**Queues disabled (no resources):      0
  Total queues:                         4
```

You can use the **show mpls interface brief** command to display the MPLS major interfaces. You can then view the statistics for the major interface displayed by the **show ip interface** command, as show in the following display excerpt:

```
host1:pe1# show ip interface
null0 line protocol is up, ip is up
  ...
loopback0 line protocol is up, ip is up
  ...
ATM9/0.10 line protocol Atm1483 is up, ip is up
  ...
  In Received Packets 78, Bytes 5991
```



```
    Unicast Packets 29, Bytes 2469
    Multicast Packets 49, Bytes 3522
  In Policed Packets 0, Bytes 0
  In Error Packets 0
  In Invalid Source Address Packets 0
  In Discarded Packets 0
  Out Forwarded Packets 78, Bytes 5786
    Unicast Packets 78, Bytes 5786
    Multicast Routed Packets 0, Bytes 0
  Out Scheduler Dropped Packets 0, Bytes 0
  Out Policed Packets 0, Bytes 0
  Out Discarded Packets 0
  queue 0: traffic class best-effort, bound to ip ATM9/0.10
    Queue length 0 bytes
    Forwarded packets 1, bytes 52
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0
ip19000001.mpls.ip line protocol MplsMajor is up, ip is up
...
```

The **show mpls interface** command displays the queue associated with the MPLS major interface, but indicates it is bound to the layer 2 interface.

```
host1:pe1# show mpls interface
MPLS major interface ATM9/0.10
  ATM circuit type is 1483 LLC encapsulation
  Administrative state is enabled
  Operational state is up
  Operational MTU is 9180
  Received:
    1 packet
    136 bytes
    0 errors
    0 discards
    0 failed label lookups
  Sent:
    1 packet
    136 bytes
    0 errors
    0 discards
LDP information:
    10.10.10.1/24
    enabled with profile 'default'
     133 hello recv, 136 hello sent, 0 hello rej
     2 adj setup, 1 adj deleted,
        Session to 10.10.10.2 is operational (passive)
        Session statistics:
          4 label alloc, 4 label learned,
          4 accum label alloc, 4 accum label learned,
          last restart time = 00:01:49
          Rcvd: 0 notf, 8 msg, 4 mapping, 0 request
                0 abort, 0 release, 0 withdraw, 1 addr
                0 addr withdraw, 8 msgId
                0 bad mapping, 0 bad request, 0 bad abort, 0 bad release
                0 bad withdraw, 0 bad addr, 0 bad addr withdraw
                0 unknown msg type err
                last info err code = 0x00000000, 0 loop detected
          Sent: 0 notf, 8 msg, 4 mapping, 0 request
                0 abort, 0 release, 0 withdraw, 1 addr
                0 addr withdraw, 8 msgId
```



```
      Adjacency statistics:
        30 hello recv, 29 hello sent, 0 bad hello recv
        adj setup time = 00:02:19
        last hello recv time = 00:00:00, last hello sent time = 00:00:00
queue 0: traffic class best-effort, bound to atm-vc ATM9/0.10
  Queue length 0 bytes
  Forwarded packets 1, bytes 148
  Dropped committed packets 0, bytes 0
  Dropped conformed packets 0, bytes 0
  Dropped exceeded packets 0, bytes 0
```

Related Documentation

- MPLS Label Switching and Packet Forwarding Overview on page 222
- MPLS Forwarding and Next-Hop Tables Overview on page 237
- MPLS Interfaces and Interface Stacking Overview on page 240
- Topology-Driven LSPs Overview on page 259
- MPLS Packet Spoof Checking Overview on page 238

MPLS Forwarding and Next-Hop Tables Overview

An MPLS forwarding table determines how MPLS handles received MPLS packets. When an MPLS packet arrives on an MPLS major interface, MPLS looks up the outermost MPLS label of the received packet in the relevant MPLS forwarding table.

The entries in the MPLS forwarding table map labels to next hops. Each entry in the MPLS forwarding table points to an entry in the MPLS next-hop table. Each MPLS next hop points to either an interface or another MPLS next hop. The chain of MPLS next hops, which ends at an interface, informs MPLS which labels to push and where to send the MPLS packet.

For RSVP-TE tunnels, minor interfaces are created in addition to the forwarding table and next-hop table entries. One minor interface is created of each in/out segment of a tunnel. the purpose of these minor interfaces is to attach QoS and policy to an LSP.

MPLS forwarding tables consist of the following:

- One forwarding table for each MPLS virtual router. This table contains labels from the platform label space. When an MPLS packet arrives on an MPLS major interface that uses the platform label space, MPLS looks up the label in the MPLS forwarding table of the virtual router in which the major interface exists.
- One forwarding table for each MPLS major interface that uses the interface label space. This table contains labels from the interface label space of that major interface. When an MPLS packet arrives on an MPLS major interface that uses the interface label space, MPLS looks up the label in the MPLS forwarding table for that particular major interface.

The signaling protocols add entries to the MPLS forwarding tables. You cannot manually create an MPLS forwarding entry. The signaling protocols set the following attributes for each entry placed in the forwarding table:

- An MPLS in label that is matched against the outermost label of the received MPLS packet.
- The MPLS next hop, which specifies the actions to be performed on the MPLS packet. MPLS next hops can be chained together to create complex actions.
- A spoof check field that specifies the type of spoof checking is performed to determine whether the MPLS packet arrived from a legitimate source. See "MPLS Packet Spoof Checking Overview" on page 238 for more information.

See "Monitoring MPLS" on page 325, for information about enabling statistics collection for MPLS forwarding table entries.

Related Documentation

- MPLS Label Distribution Methodology on page 231
- MPLS Label Switching and Packet Forwarding Overview on page 222
- MPLS Packet Spoof Checking Overview on page 238
- IP and IPv6 Tunnel Routing Tables and MPLS Tunnels Overview on page 238

MPLS Packet Spoof Checking Overview

The MPLS forwarding table enables MPLS to determine whether an MPLS packet received from an upstream neighbor contains an MPLS label that was advertised to that neighbor. If not, the packet is dropped. Each entry in the forwarding table has a spoof check field that specifies the type of checking that must be performed for the associated in label. The signaling protocol (BGP, LDP, or RSVP) that populates the entry in the MPLS forwarding table sets the spoof check field.

MPLS supports the following types of spoof checking:

- Router spoof checking—MPLS packets are accepted only if they arrive on an MPLS major interface that is in the same virtual router as the MPLS forwarding table.
- Interface spoof checking—MPLS packets are accepted only if they arrive on the particular MPLS major interface identified in the spoof check field.

You can use the **show mpls forwarding** command to view the spoof check field for an MPLS forwarding table entry.

Related Documentation

- MPLS Forwarding and Next-Hop Tables Overview on page 237

IP and IPv6 Tunnel Routing Tables and MPLS Tunnels Overview

The IP and IPv6 tunnel routing tables contain routes that point only to tunnels, such as MPLS tunnels. The tunnel routing table is not used for forwarding. Instead, protocols resolve MPLS next hops by looking up the routes in the table. For example, BGP uses the table to resolve indirect next hops for labeled routes.

BGP, LDP, and RSVP-TE can contribute routes to the table. LDP adds all destinations that can be reached by means of labels learned from downstream LDP neighbors.

RSVP-TE adds only MPLS tunnel endpoints. BGP also adds routes to the tunnel table in certain cases. Routes added by any of these protocols include the effective metric.

For example, in a BGP/MPLS VPN topology, LDP or RSVP-TE adds routes to the tunnel routing table for all available tunnels. BGP performs a lookup in the tunnel routing table so that it can resolve indirect next hops.

You can clear the routes from the tunnel routing table. You might do this, for example, to reapply routing policies when the policies are changed.

Related Documentation

- MPLS Forwarding and Next-Hop Tables Overview on page 237
- Clearing and Refreshing IPv4 Dynamic Routes in the Tunnel Routing Table on page 329
- Clearing and Refreshing IPv6 Dynamic Routes in the Tunnel Routing Table on page 329
- Explicit Routing for MPLS Overview on page 239

Explicit Routing for MPLS Overview

MPLS offers two options for selecting routing paths:

- Hop-by-hop routing
- Explicit routing

In explicit routing, the route the LSP takes is defined by the ingress node. The path consists of a series of hops defined by the ingress LSR. Each hop can be a traditional interface, an autonomous system, or an LSP. A hop can be *strict* or *loose*.

A *strict* hop must be directly connected (that is, adjacent) to the previous node in the path. A *loose* hop is not necessarily directly connected to the previous node; whether it is directly connected is unknown.

The sequence of hops comprising an explicit routing LSP may be chosen in either of the following ways:

- Through a user-defined configuration, resulting in *configured* explicit paths. When you create the explicit route, you must manually configure each hop in the path.
- Through a routing protocol–defined configuration, resulting in *dynamic* explicit paths. When the routing protocol (IS-IS or OSPF) creates the explicit path, it makes use of the topological information learned from a link-state database in order to compute the entire path, beginning at the ingress node and ending at the egress node.

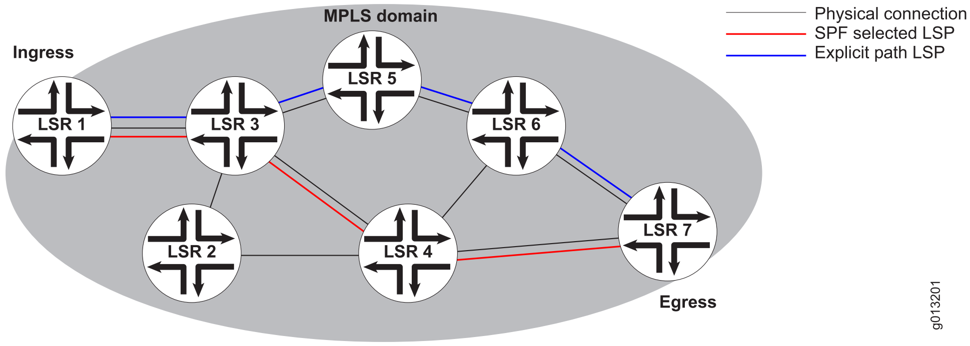
Consider the MPLS domain shown in Figure 55 on page 240. Without explicit path routing, the tunnel is created hop by hop along the following path:

LSR 1 –> LSR 3 –> LSR 4 –> LSR 7

Suppose LSR 5 and LSR 6 are underused and LSR 4 is overused. In this case you might choose to configure the following explicit path because it forwards the data better than the hop-by-hop path:

LSR 1 –> LSR 3 –> LSR 5 –> LSR 6 –> LSR 7

Figure 55: Explicit Routing in an MPLS Domain



Related Documentation

- Configuring Explicit Routing for MPLS on page 290

MPLS Interfaces and Interface Stacking Overview

The JunosE implementation of MPLS employs MPLS major, minor, and shim interfaces.

MPLS Major Interfaces

An MPLS major interface must be stacked on a layer 2 interface to send or receive MPLS packets on that interface. Each MPLS major interface exists in a particular virtual router.

MPLS major interfaces can use the platform label space or the interface label space. Which type of label space is used by the major interface is determined by the layer 2 interface on which the major interface is stacked. If the layer 2 interface is an ATM AAL5 interface, the major interface uses the interface label space. For all other layer 2 interface types, the major interface uses the platform label space.

When an MPLS packet arrives on the MPLS major interface, MPLS looks up the label of the received MPLS packet, the in label, in the MPLS forwarding table that is associated with the major interface. For major interfaces using the platform label space, the lookup is in the MPLS forwarding table of the VR. For major interfaces using the interface label space, the lookup is in the MPLS forwarding table of the major interface.

You use the **mpls** command in Interface Configuration mode to create or remove MPLS major interfaces. Some other commands create an MPLS major interface if it does not already exist.

You can configure the following attributes for each MPLS major interface:

- The administrative state, enabled or disabled, configured with the **mpls disable** command.
- The range of ATM VPIs used as interface labels for major interfaces stacked on ATM AAL5 interfaces, configured with the **mpls atm vpi range** command.
- The range of ATM VCIs used as interface labels for major interfaces stacked on ATM AAL5 interfaces, configured with the **mpls atm vci range** command.

MPLS Minor Interfaces

When you configure an LSP with the **interface tunnel mpls** command, RSVP-TE creates an MPLS minor interface to represent the head of the LSP. MPLS minor interfaces are also created by RSVP-TE on the transit and tail LSRs when the LSP is signaled. Only RSVP-TE creates MPLS minor interfaces. Neither BGP nor LDP create them.

These minor interfaces are used to associate policy or a QoS profile with an LSP (either on an LSR or an LER). This minor interface is created automatically by the signaling protocol. Minor interfaces are not saved in NVS. Use the **show mpls interface minor** command to view the minor interfaces.

The following attributes of the minor interface are set by RSVP-TE:

- The UID of the minor interface, assigned automatically when the interface is created.
- The operational state of the interface, up or down.
- Whether the interface is an ingress MPLS minor interface used to receive traffic or an egress MPLS minor interface used to transmit traffic.

MPLS Shim Interfaces

MPLS shim interfaces are stacked on layer 2 interfaces to provide layer 2 services over MPLS or to create local cross-connects by cross-connecting the layer 2 interface to another layer 2 interface. For more information about MPLS shim interfaces, see *Configuring Layer 2 Services over MPLS*.

Interface Stacking

MPLS interface stacking differs depending on whether the platform label space (Figure 56 on page 241) or the interface label space (Figure 57 on page 242) is used.

Figure 56: MPLS Interface Stacking for the Platform Label Space

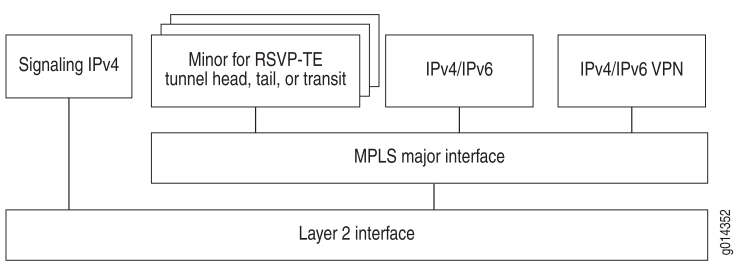
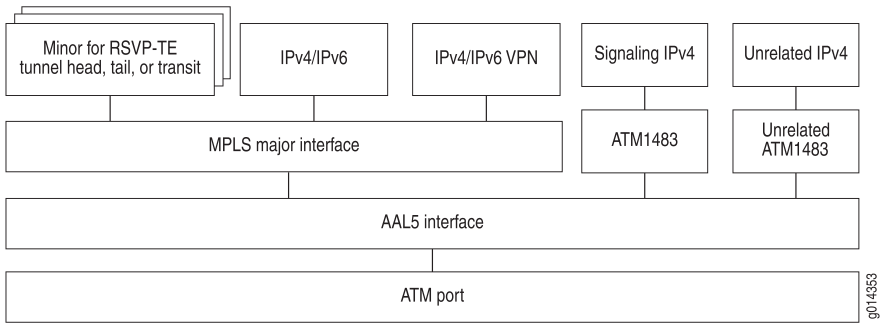


Figure 57: MPLS Interface Stacking for the Interface Label Space



Related Documentation

- MPLS Label Distribution Methodology on page 231
- MPLS Label Switching and Packet Forwarding Overview on page 222
- MPLS Label Distribution Protocols Overview on page 242

MPLS Label Distribution Protocols Overview

Label distribution protocols create and maintain the label-to-FEC bindings along an LSP from MPLS domain ingress to MPLS domain egress. A label distribution protocol is a set of procedures by which one LSR informs a peer LSR of the meaning of the labels used to forward traffic between them. It enables each peer to learn about the other peer's label mappings. The label distribution protocol provides the information MPLS uses to create the forwarding tables in each LSR in the MPLS domain.

NOTE: Label distribution protocols are sometimes referred to as signaling protocols. However, label distribution is a more accurate description of their function and is preferred in this text.

The following protocols are currently used for label distribution:

- BGP—Border Gateway Protocol
- LDP—Label Distribution Protocol
- RSVP-TE—Resource Reservation Protocol with traffic-engineering extensions that enable label binding and explicit route capability

  NOTE: To reduce confusion, this text uses the lowercase term, label distribution protocol, to refer to the generic class of protocols. The acronym, LDP, refers only to the particular protocol named Label Distribution Protocol.

BGP and LDP have no traffic-engineering capability and support only best-effort LSPs. LDP supports topology-driven MPLS networks in best-effort, hop-by-hop implementations. RSVP-TE is used primarily for MPLS applications that require traffic engineering (TE) or quality of service (QoS) capabilities, but they also support best-effort LSPs.

LDP Messages and Sessions

LDP creates reliable sessions by running over TCP. You do not have to explicitly configure LDP peers, because each LSR actively discovers all other LSRs to which it is directly connected. LDP is a *hard-state* protocol, meaning that after the LSP is established, it is assumed to remain in place until it has been explicitly torn down. This is in contrast to RSVP-TE, which is a *soft-state* protocol. See "RSVP-TE Messages and Sessions" on page 244.

LDP uses many messages to create LSPs, classified in the following four types:

- Discovery—To identify other LSRs
- Adjacency—To create, maintain, and end sessions between LSRs
- Label advertisement—To request, map, withdraw, and release labels
- Notification—To provide advisory and error information

Unlike the other LDP messages, the discovery process runs over UDP. Each LSR periodically broadcasts a link hello message to the well-known UDP port, 646. Each LSR listens on this port for link hello messages from other LSRs. In this manner, each LSR learns about all other LSRs to which it is directly connected, creating link hello adjacencies. When an LSR learns about another LSR, it establishes a TCP connection to the peer on well-known TCP port 646 and creates an LDP session on top of the TCP connection.

A transport address for the local peer is advertised in LDP discovery hello messages. Interfaces that use the platform label space default to the LSR router ID for the transport address. You can use the **mpls ldp discovery transport-address** command to specify an arbitrary IP address as the transport address.

LDP can also discover peers that are not directly connected if you provide the LSR with the IP address of one or more peers by means of an access list. The LSR sends targeted hello messages to UDP port 646 on each remote peer. If the targeted peer responds with a targeted hello message to the initiator, a targeted hello adjacency is created and session establishment can proceed.

In certain cases, a targeted hello adjacency to directly connected peers might be useful. If an LSR receives both a link hello message and a targeted hello message from the same initiator, only a single LDP session is established between the LSRs.

By default, because all LSRs listen on the well-known port, they all attempt to create a session with the originator. You can use the **mpls ldp link-hello disable** command to suppress the transmission of link hello messages. Thereafter, sessions are formed only with peers contacted with targeted hello messages.

The LDP peers exchange session initialization messages that include timer values and graceful-restart parameters. An LSR responds with a keepalive message if the values in the initialization message are acceptable. If any value is not acceptable, the LSR responds instead with an error notification message, terminating the session. After a session is established, LDP peers exchange keepalive messages that verify continued functioning of the LSR. Failure to receive an expected keepalive message causes an LSR to terminate the LDP session.

Label mapping and distribution use downstream-unsolicited, independent control.

With downstream-unsolicited, independent control, an LSR creates a label binding whenever it learns a new IGP route; the LSR sends a label mapping message immediately to all of its peer LSRs—upstream and downstream—without having received a label request message from any peer. The LSR sends the label mapping message regardless of whether it has received a label mapping message from a downstream LSR. This is the label distribution method employed in a topology-driven MPLS network.

A downstream LSR can send a label withdrawal message to recall a label that it previously mapped. If an LSR that has received a label mapping subsequently determines that it no longer needs that label, it can send a label release message that frees the label for use.

NOTE: **The LDP database can maintain up to 250 neighbors if you configure the hello and dead intervals (or hold timers) for IGP, such as OSPF or IS-IS, to be higher than their default values. If you set the hello and dead intervals (or hold timers) at their default values, the LDP neighbors start flapping (constantly go up and down) when more than 200 LDP neighbors are present.**

RSVP-TE Messages and Sessions

RSVP is described in RFC 2205. Multiple RFCs enable extensions to RSVP for traffic engineering. The router supports the extended version of RSVP, referred to as RSVP-TE.

RSVP-TE is “unreliable” because it does not use TCP to exchange messages. In contrast to LDP—a hard-state protocol—RSVP-TE is a *soft-state* protocol, meaning that much of the session information is embedded in a state machine on each LSR. The state machine must be refreshed periodically to avoid session termination. LSRs send path messages to downstream peers to create and refresh local path states. LSRs send resv messages to upstream peers in response to path messages to create and refresh local resv states. A session is ended if the state machine is not refreshed within the RSVP tunnel timeout period, which is determined as follows:

$$\text{RSVP tunnel timeout period in seconds} = \left\{[(\text{cleanup timeout factor} + 0.5) \times 1.5] \times \left(\frac{\text{refresh period}}{1000}\right)\right\}$$

For example, for the default values,

$$\text{RSVP tunnel timeout period in seconds} = \left\{[(3 + 0.5) \times 1.5] \times \left(\frac{30{,}000}{1000}\right)\right\} = 157.5$$

RSVP-TE messages carry *objects* consisting of type-length-values (TLVs). The *label request object* instructs the endpoint LSR to return an resv message to establish the LSP. The resv message contains the *label object*, the label used for the FEC. Both the path and resv messages carry the *record route object*, which records the route traversed by the message.

An upstream LSR sends a pathtear message when its path state times out as a result of not being refreshed. The pathtear message removes the path and resv states in each LSR as it proceeds downstream. Downstream LSRs similarly send the resvtear message when their resv state times out to remove the resv states in upstream LSRs.

If a downstream LSR determines that it received an erroneous path message, it sends a patherr message to the sender. If a reservation (label) request fails, the request initiator sends a resverr message to the downstream LSRs. Both of these messages are advisory and do not alter path or resv state.

RSVP-TE State Refresh and Reliability

RSVP-TE employs refresh messages to synchronize state between neighbors and to recover from lost RSVP-TE messages of other types. RSVP-TE by default does not provide for reliable transmission. Each node is responsible for the transmission of RSVP-TE messages to its neighbors and relies on periodic state refreshes to recover from lost messages.

RSVP-TE refresh reduction provides a means to increase reliability while reducing message handling and synchronization overhead. Issuing the **mpls rsvp refresh-reduction** command enables the following features:

- The message ID object is included in RSVP-TE messages to provide a unique message ID on a per-hop basis. In refresh messages, it indicates to the receiving node that the message is a refresh message, eliminating the need for the node to completely analyze the message and thus reducing the processing overhead at the receiver.
- RSVP-TE uses a message acknowledgment mechanism to support reliable message delivery on a per-hop basis. Messages that include the message ID object also include a request for acknowledgment. Upon receipt, the receiving node returns a message ack object, enabling the sending node to determine whether a message was lost and triggering a retransmission as necessary.
- Summary refresh (srefresh) messages refresh the state previously advertised in path or resv messages that included message ID objects. The srefresh message carries the unique message ID as state identifier, eliminating the need to send whole refresh messages and reducing the overhead needed to maintain RSVP-TE state synchronization. This method maintains RSVP-TE's ability to indicate when the state is lost and to adjust to changes in routing. The srefresh message can carry message IDs for multiple RSVP-TE sessions.

Issuing the **mpls rsvp message-bundling** command enables RSVP-TE to use bundle messages, each of which includes multiple standard RSVP-TE messages, to reduce the overall message processing overhead.

BGP Signaling

You can use BGP as a label distribution protocol both for IP routes and for VPN routes.

When BGP distributes a particular IP route, it can also distribute an MPLS label that has been mapped to that route, as described in RFC 3107. The MP-BGP extensions (RFC 4760) enable the BGP update message to carry both the route and the label mapping information for the route. The label is encoded into the NLRI field of the attribute, and the SAFI field is set to 4 to indicate that the NLRI contains a label. A BGP speaker can use BGP to send labels to a particular BGP peer only if that peer advertised the capability to process update messages with SAFI 4. BGP speakers advertise this capability only to peers for which the **neighbor send-label** command has been configured.

When BGP advertises labeled routes, it adds a label-to-next-hop mapping (cross-connect) to the MPLS forwarding table. This mapping consists of the in label that BGP allocates from the platform label space plus the MPLS next hop information related to the labeled route's next hop.

BGP can also distribute labels for VPN routes in BGP/MPLS VPNs. In a BGP/MPLS VPN network, BGP is used to exchange the routes of a particular VPN among the provider edge routers attached to the VPN. To ensure that routes from different VPNS remain distinct even if the VPNs use overlapping address spaces, an MPLS label is assigned to each route within the VPN. BGP distributes both the VPN routes and the associated MPLS label for each route.

The label mapping information for a particular VPN route is included in the same BGP update message that distributes the route. The label is encoded into the NLRI field of the attribute, and the SAFI field has a value of 128 to indicate that the NLRI contains both an RD (route distinguisher) and a label.

For more information on BGP capabilities, see *Configuring BGP Routing*. For more information on MP-BGP extensions, NLRIs, and BGP/MPLS VPNs, see *Configuring BGP-MPLS Applications*.

Related Documentation

- MPLS Label Distribution Methodology on page 231
- MPLS Label Switching and Packet Forwarding Overview on page 222
- MPLS Interfaces and Interface Stacking Overview on page 240
- Topology-Driven LSPs Overview on page 259

ECMP Labels for MPLS Overview

MPLS supports equal-cost multipath (ECMP) labels. A maximum of 16 MPLS paths is supported; 4 paths are available by default. On LERs, MPLS ECMP next hops can be used in the IP routing table for non-VPN and VPN routes. On LSRs, an incoming label can point to either an MPLS ECMP next hop or an IP ECMP.

The signaling protocol determines whether ECMP next hops are used. For example, LDP can learn multiple labels for a route from different downstream peers (or one label from a downstream peer that has parallel connections to the router). LDP then creates an MPLS ECMP next hop that can be used in the IP routing table. If LDP also advertises a label, then a forwarding entry is added to the MPLS forwarding table with the ECMP next hop.

MPLS Connectivity and ECMP

When an MPLS ECMP is part of the tunnel being explored by an MPLS echo request, the request packet takes one of the available ECMP paths. Probing FECs with different label stacks can yield different ECMP paths. However, you cannot guarantee complete coverage of all the ECMP paths.

You can use MPLS trace to determine which paths are present on an MPLS LSR. When the TTL expires on an MPLS LSR, the echo reply that is returned includes a downstream mapping TLV. This TLV contains all the downstream mappings of the LSR on which the TTL expired, if that feature is supported by the LSR. You can use the **detail** version of the **trace mpls** commands to display these downstream mappings.

Supported TLVs

Table 51 on page 247 lists the TLVs supported by the MPLS LSP ping feature. Table 52 on page 248 lists the sub-TLVs supported for the Target FEC Stack TLV.

Table 51: TLVs Supported by MPLS LSP ping

Type Number	Value	Comments
1	Target FEC Stack	Multiple FEC stack sub-TLVs are not supported. A single LSP ping message cannot have more than one target FEC stack TLV.
2	Downstream Mapping	Only the IPv4 (numbered or unnumbered) downstream address type is supported. Flag I for the Interface and Label Stack object is supported. Flag N, to treat the packet as a non-IP packet, is not supported. An MPLS LSP trace echo request includes this TLV. This TLV contains the downstream address all-routers-multicast; that is the well-known IP address 224.0.0.2. Validation of the downstream address is not performed. Verification of the downstream address is not performed on receipt of an MPLS echo request that contains this TLV. In an MPLS echo reply, multipath information is not supported in this TLV; the multipath type is always set to 0 in the reply. However, the reply includes one downstream mapping TLV for each downstream path.
3	Pad	This TLV is included in the MPLS echo request packet. The TLV can specify either "Do not reply" or "Reply via an IPv4/IPv6 UDP packet."
7	Interface and Label Stack	This TLV is generated if requested by the received downstream mapping TLV.

Table 51: TLVs Supported by MPLS LSP ping *(continued)*

Type Number	Value	Comments
9	Errored TLVs	This TLV is generated if an error is encountered while parsing one of the received TLVs.
10	Reply TOS Byte	–
11	P2MP Responder Identifier	This TLV is included in the MPLS echo request packet to validate whether the IP address specified in the TLV is an IP address of one of the interfaces in the router. Four sub-TLVs are defined for inclusion in the P2MP Responder Identifier TLV in the echo request message.
12	Echo Jitter	This TLV is included in the LSP ping message (echo request) to enable the egress node of a point-to-multipoint LSP to delay the transmission of the response by a time interval that is limited by the value specified in this TLV. In JunosE Software, the delay is set to a maximum of 30 seconds.

Table 52: Sub-TLVs Supported for the Target FEC Stack TLV

Subtype Number	Value	Comments
1	LDP IPv4 prefix	–
2	LDP IPv6 prefix	–
3	RSVP IPv4 LSP	–
6	VPN IPv4 prefix	–
7	VPN IPv6 prefix	–
8	L2 VPN endpoint	For VPLS and VPWS
10	FEC 128 pseudowire	For Martini encapsulation
17	RSVP P2MP IPv4 Session	For identification of the point-to-multipoint LSP for which you want to verify the data plane

Related Documentation

- MPLS Label Switching and Packet Forwarding Overview on page 222

MPLS Connectivity Verification and Troubleshooting Methods

In IP networks, the **ping** and **traceroute** commands enable you to verify network connectivity and find broken links or loops. In MPLS-enabled networks, you can use the **ping** command to determine whether IP connectivity exists to a destination even when the ping packets must traverse multiple LSPs. You can use the **traceroute** command to determine the labels that data packets use when traversing LSPs to the destination.

In an MPLS-enabled network, however, you cannot use these IP commands to determine MPLS connectivity to a destination.

You can use the MPLS ping and trace features to detect data plane failures in LSPs. Specific **mpls ping** and **trace mpls** commands enable you to target different types of MPLS applications and network topologies. The various **ping mpls** and **trace mpls** commands send UDP packets, known as MPLS echo requests, to the egress LSR of MPLS packets in a given FEC. Each echo request is forwarded along the same data path as the MPLS packets in that FEC.

The echo request packets use a destination address in the 127.0.0.0/8 range and port 3503. The default address is 127.0.0.1. This address range prevents IP from forwarding the packet, so that the echo request must follow the MPLS data path. This behavior is different from that of the IP **ping** and **traceroute** commands, which send ICMP packets to the actual destination.

Each MPLS echo request packet contains information about the FEC stack that is being validated. LSRs that receive an MPLS echo request respond with MPLS echo reply packets. (Even when MPLS is not enabled on that router, echo reply packets are sent by E Series routers that receive an echo request packet. This situation is a transient condition when the router is receiving labeled packets. A return code in the echo replies indicates to the sending router that no label mapping exists on the receiving router.)

The **ping mpls** commands perform a basic connectivity check. When the echo request exits the tunnel at the egress LSR, the LSR sends the packet to the control plane. The egress router validates the FEC stack to determine whether that LSR is the actual egress for the FEC. The egress router sends an echo reply packet back to the source address of the echo request packet. The egress router can send the packet back by means of either the IP path or the MPLS path.

The **trace mpls** commands isolate faults in the LSP. For these commands, successive echo request packets are sent along the path. The first packet has a TTL of one; the TTL value is incremented by one for each successive packet. The first packet therefore reaches only the next hop on the path; the second packet reaches the next router after that. Echo request packets are sent until either an echo reply is received from the egress router for the FEC or a TTL of 32 is reached.

When a TTL expires on an LSR, that LSR sends an echo reply packet back to the source. For transit routers, the echo reply indicates that downstream mapping exists for the FEC, meaning that the packet would have been forwarded if the TTL had not expired. The egress router sends an echo reply packet verifying that it is the egress.

Although you cannot send IPv6 UDP packets for MPLS ping, you can use the **ping mpls l3vpn** command with an IPv6 prefix to investigate IPv6 VPNs.

Related Documentation

- ECMP Labels for MPLS Overview on page 246
- Verifying and Troubleshooting MPLS Connectivity on page 377
- Packet Flow Examples for Verifying MPLS Connectivity on page 379

- *ping mpls ip*
- *ping mpls l2transport*
- *ping mpls l3vpn*
- *ping mpls rsvp tunnel*
- *ping mpls vpls*
- *trace mpls ip*
- *trace mpls l2transport*
- *trace mpls l3vpn*
- *trace mpls rsvp tunnel*
- *trace mpls vpls*

Point-to-Multipoint LSPs Connectivity Verification at Egress Nodes Overview

JunosE Software enables you to use the MPLS ping feature to detect data plane failures in point-to-multipoint MPLS LSPs. The implementation of the RSVP-TE protocol in JunosE Software was previously enhanced to enable you to configure E Series routers as tunnel-tail or egress nodes for point-to-multipoint MPLS LSPs. You can use the point-to-multipoint LSP ping functionality, which is available in routers other than E Series routers, to send echo request packets and receive echo reply packets to test network connectivity of point-to-multipoint RSVP-TE MPLS LSPs that contain E Series routers as egress nodes. You cannot use the **ping** commands for other point-to-multipoint MPLS applications, such as LDP LSPs because E Series routers do not support such functionality. In addition, because E Series routers do not support ingress, transit, or branch label-switched routers (LSR) roles for point-to-multipoint LSPs, they do not support the point-to-multipoint MPLS ping feature for ingress, transit, or branch nodes.

Related Documentation

- TLVs and Sub-TLVs Supported for Point-to-Multipoint LSPs Connectivity Verification at Egress Nodes on page 252
- Ping Extensions for Point-to-Multipoint LSPs Connectivity Verification at Egress Nodes on page 251
- Verifying and Troubleshooting MPLS Connectivity on page 377

Ping Extensions for Point-to-Multipoint LSPs Connectivity Verification at Egress Nodes

To enable detection of data plane failures at egress nodes in point-to-multipoint LSPs, the MPLS ping extensions in point-to-multipoint LSPs define the following new sub-type-length-values (TLVs) for the Target FEC Stack TLV and new TLVs:

- RSVP P2MP IPv4 Session sub-TLV
- P2MP Responder Identifier TLV
- Echo Jitter TLV

RSVP P2MP IPv4 Session Sub-TLV Overview

The RSVP P2MP IPv4 Session TLV identifies the point-to-multipoint LSP for which you are verifying the data plane. This TLV has a type number of 17. To identify the point-to-multipoint LSP for which you are running diagnostic connectivity checks, the echo request message must carry a Target FEC Stack TLV that contains an RSVP P2MP IPv4 Session sub-TLV. The point-to-multipoint LSP ping functionality performs necessary validation with RSVP-TE before sending the response to the source. For other sub-TLVs defined for identifying point-to-multipoint LDP MPLS LSP, the ping feature sends an error response.

P2MP Responder Identifier TLV Overview

The point-to-multipoint MPLS LSP ping extensions enable a specific egress node of the point-to-multipoint MPLS LSP to be selected to verify that the data plane of the path to the particular egress node does not possess any failures. Use the new P2MP Responder Identifier TLV and associated rules for processing the LSP ping message (echo request) that contain this new TLV to validate whether the IP address specified in the TLV is an IP address of one of the interfaces in the router.

- If the IP address in the TLV matches the IP address assigned to one of the interfaces, the point-to-multipoint MPLS LSP ping feature sends the success response to the originator.
- If the IP address specified in the TLV does not match any of the IP addresses assigned to the interfaces, no response is sent to the originator.

If errors exist in the syntax of TLVs in the message received or if the router to which echo request packets are sent is not an egress node for the point-to-multipoint MPLS LSP, the echo response is sent to the originator, regardless of the presence of the P2MP Responder Identifier TLV in the request packet.

IETF draft, Detecting Data Plane Failures in Point-to-Multipoint Multiprotocol Label Switching (MPLS) - Extensions to LSP Ping (draft-ietf-mpls-p2mp-lsp-ping-08.txt) (February 2010 expiration), recommends a particular type value for the P2MP Responder Identifier TLV.

The type value used by point-to-multipoint LSPs in Junos OS differs from the type value specified in the IETF draft. To enable interoperability with routers running Junos OS (which are often employed as the ingress, transit, or branch nodes in point-to-multipoint LSPs), the point-to-multipoint MPLS LSP ping feature in JunosE Software interprets both type values to identify the TLV (P2MP Responder Identifier).

Echo Jitter TLV Overview

The point-to-multipoint MPLS LSP ping extensions enable the initiator or ingress of the ping operation to request the egress nodes not to send responses immediately after the echo request message is received, but delay the response by a certain period of time. Use the new Echo Jitter TLV and associated rules for processing the LSP ping message (echo request) that contains this the Echo Jitter TLV to delay the transmission of the response by a time interval that is limited by the value specified in the Echo Jitter TLV.

The point-to-multipoint MPLS LSP ping functionality in JunosE Software supports a delay of up to 30 seconds. If the value specified in the Echo Jitter TLV is not within the supported range, the egress node uses a value within the range as the time to wait to send the echo response.

IETF draft, Detecting Data Plane Failures in Point-to-Multipoint Multiprotocol Label Switching (MPLS) - Extensions to LSP Ping, uses a particular type value for the Echo Jitter TLV. The point-to-multipoint MPLS LSP ping functionality in JunosE Software interprets the suggested type value in the draft to denote the new TLV (Echo Jitter).

Traceroute Overview

The point-to-multipoint LSP ping feature enables the egress node to respond to the traceroute requests that the ingress node initiates in the same manner as the egress node responds to a ping request. No additional or separate configuration is needed to enable the path to the egress nodes to be traced by ingress nodes in point-to-multipoint LSPs.

Related Documentation

- TLVs and Sub-TLVs Supported for Point-to-Multipoint LSPs Connectivity Verification at Egress Nodes on page 252
- Point-to-Multipoint LSPs Connectivity Verification at Egress Nodes Overview on page 250
- Verifying and Troubleshooting MPLS Connectivity on page 377

TLVs and Sub-TLVs Supported for Point-to-Multipoint LSPs Connectivity Verification at Egress Nodes

To enable detection of data plane failures using the **ping mpls** and **trace mpls** commands at egress nodes of point-to-multipoint LSPs, JunosE Software supports two new TLVs, Echo Jitter and P2MP Responder Identifier. Also, a sub-TLV, RSVP P2MP IPv4 Session, is supported in the Target FEC Stack TLV to verify MPLS connectivity to egress nodes of point-to-multipoint LSPs.

Echo Jitter TLV Operations

The initiator (ingress) of a ping request might require the responding egress to introduce a random delay (or jitter) before forwarding the response. The delay period enables the responses from multiple egresses to be spread over a time period. This mechanism is very useful in situations when the entire LSP tree is being pinged because it helps the ingress (and nearby routers) node from being flooded with a number of responses, or from discarding responses if any rate limits are applied on the incoming traffic.

In JunosE Software, the delay is set to a maximum of 30 seconds. The ingress node informs the egresses of this time interval limitation by supplying a value in the Echo Jitter TLV in the echo request message. If this TLV is present in the echo request packet, the responding egress node delays sending a response for a random amount of time between zero milliseconds and 30 seconds that is predefined for this TLV. If the TLV is not contained in the echo request packet, the responding egress node does not create any additional delay in responding to the echo request. The Echo Jitter TLV is valid only in an echo request message. If this TLV is included in an echo response message, it is ignored. The Echo Jitter TLV is assigned the TLV type value of 12.

P2MP Responder Identifier TLV Operations

You can select the egress node in a point-to-multipoint LSP for which you want to trace the path from an ingress node and detect network failures by including a P2MP Responder Identifier TLV in the echo request packet sent to the egress node. The initiator can determine whether only the node identified in the TLV must respond or any node on the path to the selected egress node in the TLV needs to respond. If you do not include the P2MP Responder Identifier TLV in an echo request packet, all egress nodes in the path to the ingress node respond to echo request packets.

If the node is an egress of the P2MP LSP, the node checks whether it has received a P2MP Responder Identifier TLV in an echo request. If a P2MP Responder Identifier TLV is contained in the received echo request, the node uses the sub-TLVs contained in this TLV to determine whether it must respond to the request. If the P2MP Responder Identifier TLV is not present (or does not contain any sub-TLVs), the egress node responds to the echo request depending on the setting of the Response Type field in the echo message.

The P2MP Responder Identifier TLV is assigned a type number of 11. The P2MP Responder Identifier TLV is valid only in an echo request message.

- If this TLV is included in an echo response message, it is ignored.
- If no sub-TLVs are present, this TLV is processed as though it were not included.
- If more than one sub-TLV is present in this TLV, they are processed based on their subtype numbers and subsequent sub-TLVs are ignored.

Four sub-TLVs are defined for inclusion in the P2MP Responder Identifier TLV in the echo request message. Table 53 on page 253 lists the sub-TLVs supported for the P2MP Responder Identifier TLV.

Table 53: Sub-TLVs Supported for the P2MP Responder Identifier TLV

Subtype Number	Value	Comments
1	IPv4 Egress Address P2MP Responder Identifier	The IPv4 address in this sub-TLV is the IPv4 address of the egress node and does not specify the IPv4 address of a branch or intermediate node.

Table 53: Sub-TLVs Supported for the P2MP Responder Identifier TLV *(continued)*

Subtype Number	Value	Comments
2	IPv4 Node Address P2MP Responder Identifier	The IPv6 address in this sub-TLV is the IPv6 address of the egress node and does not specify the IPv6 address of a branch or intermediate node.
3	IPv6 Egress Address P2MP Responder Identifier	The IPv4 address in the sub-TLV might be of any physical interface or the router ID of the node itself.
4	IPv6 Node Address P2MP Responder Identifier	The IPv6 address in the sub-TLV might be of any physical interface or the router ID of the node itself.

The echo response is always controlled by the Response Type field in the echo message and also depends on whether the responding node is part of the point-to-multipoint LSP that is denoted in the Target FEC Stack TLV. The following sections describe the sub-TLVs of the P2MP Responder Identifier TLV, which are additional influencing factors to those requirements and are not a replacement for those requirements:

- Egress Address P2MP Responder Identifier Sub-TLVs on page 254
- Node Address P2MP Responder Identifier Sub-TLVs on page 254

Egress Address P2MP Responder Identifier Sub-TLVs

You can use the IPv4 or IPv6 Egress Address P2MP Responder Identifier sub-TLVs in an echo request that contains the RSVP P2MP Session or Multicast LDP FEC Stack sub-TLV. An egress node that receives an echo request with this sub-TLV present responds only if the node lies on the path to the address in the sub-TLV. The address in this sub-TLV is the address of the egress node and does not specify the address of a branch or intermediate node. This address is made available to the nodes upstream of the target node, using signaling protocols, such as RSVP. This sub-TLV may be used to trace a specific egress node in a point-to-multipoint LSP.

Node Address P2MP Responder Identifier Sub-TLVs

You can use the IPv4 or IPv6 Node Address P2MP Responder Identifier sub-TLVs in an echo request that contains the RSVP P2MP Session or Multicast LDP FEC Stack sub-TLV. A node that receives an echo request with this sub-TLV present responds only if the address in the sub-TLV corresponds to any address that is local to the node. This address in the sub-TLV might be of any physical interface or the router ID of the node itself. The address in this sub-TLV can be the address of any transit, branch, or egress node for that point-to-multipoint LSP.

Related Documentation

- Troubleshooting MTU Problems in Point-to-Point LSPs on page 386

- Ping Extensions for Point-to-Multipoint LSPs Connectivity Verification at Egress Nodes on page 251
- Point-to-Multipoint LSPs Connectivity Verification at Egress Nodes Overview on page 250
- Verifying and Troubleshooting MPLS Connectivity on page 377

LDP Discovery Mechanisms

LDP uses two different mechanisms for peer discovery. Peer discovery removes the need to explicitly configure the label-switching peers for an LSR.

- LDP uses the basic discovery mechanism to discover directly connected LDP peers.
- LDP uses the extended discovery mechanism to discover peers that are not directly connected.

LDP Basic Discovery Mechanism

To discover directly connected peers, LSRs periodically send out LDP link hellos on the interface. The link hellos are contained in UDP packets that are addressed to the well-known LDP discovery port, 646. The destination address for the ports is 224.0.0.2. Using this port and address ensures that the hellos are sent to all routers on the interface's subnet.

The link hello includes the LDP identifier for the label space that the LSR intends to use for the interface. In the JunosE implementation, this is always the platform label space, so the LDP identifier specifies the LSR ID and a value of 0 for the label space. The link hello also includes other information, such as the hello hold time configured on the interface. The hello hold time specifies how long an LSR maintains a record of hellos received from potential peers.

When an LSR receives a link hello, it identifies the sending LSR as a potential LDP peer on that interface. The LSRs form a hello adjacency to keep track of each other.

The basic discovery mechanism is enabled by default when you enable LDP on an interface. You can configure the link hellos in the LDP profile with the **hello hold-time** and **hello interval** commands. You can configure a transport IP address to be globally included in link hellos with the **mpls ldp discovery transport-address** command.

LDP Extended Discovery Mechanism

To discover LDP peers that are not directly connected, LSRs periodically send out LDP targeted hellos to potential peers. The targeted hellos are contained in UDP packets that are addressed to the well-known LDP discovery port, 646. The destination address for the ports is a specific targeted address. LDP sends targeted hellos when you configure one or more IP addresses in a targeted-hello send list. In a layer 2 Martini circuit, targeted hellos are automatically sent to the remote PE neighbor (the base tunnel endpoint). See *Configuring Layer 2 Services over MPLS* for information about layer 2 circuits.

The targeted hello includes the LDP identifier for the label space that the LSR intends to use. In the JunosE implementation, this is always the platform label space, so the LDP identifier specifies the LSR ID and a value of 0 for the label space. The targeted hello also includes other information, such as the targeted-hello hold time, which is configured globally. The targeted-hello hold time configures how long an LSR waits for another targeted hello from its peer before declaring the adjacency to be down.

Unlike basic discovery, where hellos are sent by all LSRs, extended discovery is initiated by one LSR that targets a specific LSR. The initiating LSR periodically sends targeted hellos to the targeted LSR. The targeted LSR then determines whether to respond to the targeted hello or to ignore it. If the targeted LSR responds to the sender, it does so by periodically sending targeted hellos to the initiating LSR. The exchange of targeted hellos constitutes a hello adjacency for the two LSRs.

Targeted hello values are configured globally with the **mpls ldp targeted-hello holdtime**, **mpls ldp targeted-hello interval**, **mpls ldp targeted-hello receive list**, and **mpls ldp targeted hello send list** commands.

Related Documentation

- MPLS Label Switching and Packet Forwarding Overview on page 222
- MPLS Label Distribution Protocols Overview on page 242

MPLS Traffic Engineering Overview

MPLS traffic engineering (TE) is the ability to establish LSPs according to particular criteria (*constraints*) in order to meet specific traffic requirements rather than relying on the path chosen by the conventional IGP. The constraint-based IGP examines the available network resources and calculates the shortest path for a particular tunnel that has the resources required by that tunnel. Traffic engineering enables you to make the best use of your network resources by reducing overuse and underuse of certain links.

Constraint-based routing (CR) makes traffic engineering possible by considering resource requirements and resource availability rather than merely the shortest path calculations. Constraints are determined at the edge of the network and include criteria such as required values for bandwidth or required explicit paths. You can use RSVP-TE as the label distribution protocol for traffic engineering. The IGP propagates resource information throughout its network. RSVP-TE employs downstream-on-demand, ordered control for label mapping and distribution.

Explicit routing specifies a list or group of nodes (hops) that must be used in setting up the tunnels. CR explicit paths can be *strict* or *loose*. Strict paths specify an exact physical path, including every physical node. Loose paths include hops that have local flexibility; the hop can be a traditional interface, an autonomous system, or an LSP.

LSP Backup

You can configure multiple LSPs to the same destination. By configuring different tunnel metrics for these LSPs, you can force a ranking or priority of use for the LSPs. In this scenario, all the configured LSPs are up and active. If the LSP in use develops problems and goes down, traffic is diverted to the LSP having the next best metric.

Path Option

You can configure multiple paths for an LSP with the **tunnel mpls path-option** command. Each path option has an identifying number; the lower the number the higher the preference for that path option. In this scenario, only a single LSP is up and active at a time. If the path option currently in use by an LSP goes down, MPLS tries to reroute the tunnel using the path option with the next highest preference. In certain circumstances—for example, when a tunnel is preempted by another—MPLS first attempts to reroute the tunnel with the current path option.

Reoptimization

You can use the traffic-engineering reoptimization capability to ensure that the best path is being used. Suppose the current path goes down and MPLS switches to an alternate path that is not as good as the failed path. You can have MPLS periodically search—according to a specified schedule—for a path better than the alternate by configuring the reoptimization timer. For example, you might configure MPLS to search for a better path every 10 minutes; if it finds a better path, it switches.

On the other hand, you might be concerned about route flapping. If a path goes down and then comes back up, perhaps it will continue to do so. In this case, you might not ever want to go back to a path that goes down. To accomplish this, you can configure reoptimization to never occur.

When you do not want the initial path to change—that is, when you want to pin the route—you can disable reoptimization globally by setting the timer to 0. Alternatively, you can disable reoptimization on a per-tunnel basis by using the **lockdown** option with the **tunnel mpls path-option** command. LSP paths are always pinned until the next reoptimization.

Finally, you can manually force an immediate reoptimization. See *MPLS Global Configuration Tasks* in *Configuring MPLS* in the *JunosE BGP and MPLS Configuration Guide* for information about configuring reoptimization.

Methods for Configuring RSVP-TE Tunnels

You can use either of the following methods to configure RSVP-TE tunnels:

- Configure individual tunnels with the **interface tunnel mpls:** *tunnelName* command.
- Configure multiple tunnels with the same set of parameters with the **mpls tunnels profile** command.

Related Documentation

- Tracking Resources for MPLS Traffic Engineering Overview on page 258
- LDP and RSVP-TE Interface Profile Configuration Tasks on page 284

Tracking Resources for MPLS Traffic Engineering Overview

MPLS traffic engineering uses *admission control* to keep track of resource information. Admission control has an accounting feature that ensures that requests are not accepted when the router does not have sufficient resources to accommodate them.

Currently, bandwidth (BW) and bandwidth-related information are the only resources tracked and used for traffic engineering. Admission control determines whether a setup request can be honored for an MPLS LSP with traffic parameters.

Admission control provides bandwidth information to the IGP protocols, ISIS and OSPF. As new LSPs are created, the available bandwidth decreases. The IGPs can subsequently advertise this information and use it for SPF calculations to determine paths that satisfy the traffic requirements. You can configure readvertisement to occur periodically or when the change crosses some threshold.

Starting Admission Control

Admission control operates on a router-wide basis rather than a per-virtual-router basis. Admission control of resources begins when either of the following occurs:

- You configure resource-related information about an interface, including bandwidth (either total bandwidth or MPLS reservable bandwidth), flooding frequency, flooding threshold, administrative weight, or attribute flags.
- MPLS begins to use admission control services; for example, by attempting to set up a constraint-based LSP.

Admission Control Interface Table

Configuring bandwidth on an interface creates an entry for the interface in the admission control interface table. Each entry in the table stores the following information per interface:

- Maximum (physical or line-rate) bandwidth
- Maximum reservable bandwidth
- The following information per IP class (currently a single, default class)
  - Total available (unreserved) bandwidth
  - Available bandwidth at each MPLS priority level
- Resource flooding threshold and period

The resource flooding threshold and period together control the flooding of the resource information by the IGP protocols, IS-IS and OSPF.

Configuring Traffic-Engineering Resources

You can configure the following resource-related information about an MPLS interface (at either the major interface or subinterface level):

- Bandwidth—Total bandwidth that can be reserved on the interface
- Flooding thresholds—Sets of absolute percentages of total reservable bandwidth that trigger the new bandwidth value to be flooded throughout the network; flooding is triggered when bandwidth increases past any up threshold value or decreases past any down threshold value
- Flooding frequency—Periodicity with which the bandwidth value is flooded, apart from any flooding due to value changes
- Administrative weight—Weight assigned to the interface that supersedes any assigned by the IGP
- Attribute flags—32-bit value that assigns the interface to a resource class and enables a tunnel to discriminate among interfaces by matching against tunnel affinity bits

LSP Preemption

You can develop a preemption strategy whereby a new LSP can claim resources from an existing LSP. Each tunnel can be configured with a *setup* priority and a *hold* priority. Priority levels range from 0 (highest priority) through 7 (lowest priority).

If traffic engineering admission control determines that there are insufficient resources to accept a request to set up a new LSP, the setup priority is evaluated against the hold priority of existing LSPs. An LSP with a hold priority lower than the setup priority of the new LSP can be preempted. The existing LSP is terminated to make room (free resources) for the new LSP. You must assign priorities according to network policies to prevent resource poaching and LSP thrashing.

Related Documentation

- LDP and RSVP-TE Interface Profile Configuration Tasks on page 284
- MPLS Traffic Engineering Overview on page 256

Topology-Driven LSPs Overview

Topology-driven LSPs are implemented for best-effort, hop-by-hop routing. In topology-driven LSP mode, LDP automatically sets up LSPs for IGP, direct, and static routes, subject to filtering by access-lists. JunosE supports downstream-unsolicited LDP using the platform label space.

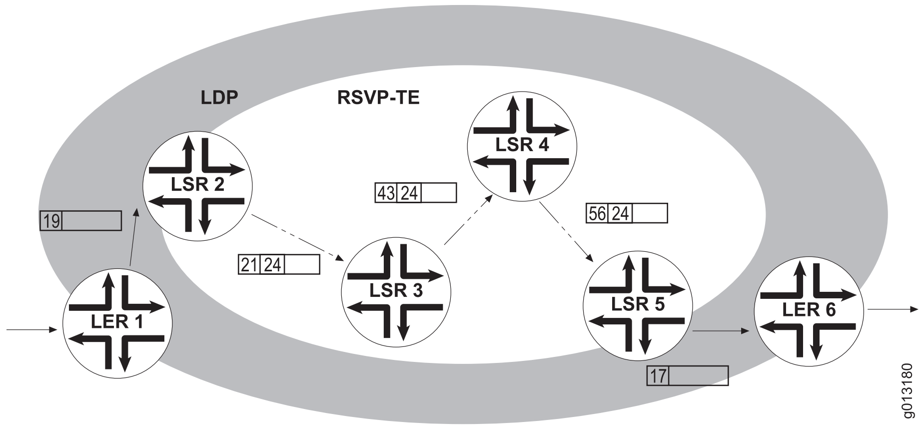
If you use the topology-driven LSP mode to forward plain IP packets, use the **ldp ip-forwarding** command to place LSPs into the IP routing table for forwarding plain IP traffic.

You can use the **mpls ldp advertise-labels** command to limit the number of routes for which labels are advertised. In most cases, you can issue **mpls ldp advertise-labels host-only**.

LDP over RSVP-TE

If you are running RSVP-TE in the core, LDP can tunnel through the core by stacking an LDP LSP over an RSVP-TE LSP, as shown in Figure 58 on page 260. With LDP over RSVP-TE, LDP establishes targeted sessions among the LDP routers at the edge of the RSVP core. From the perspective of the LDP LSP, the RSVP-TE core is a single hop.

Figure 58: LDP Tunneled Through an RSVP-TE Core



In the network topology illustrated in Figure 58 on page 260, the RSVP-TE LSP consists of LSR 2, LSR 3, LSR 4, and LSR 5. The LDP LSP consists of LER 1, LSR 2, LSR 5, and LER 6. The RSVP-TE tunnel appears to LDP as a single hop.

The initial LDP label 19 is switched with label 24 at LSR 2. Because this is the entrance to the RSVP-TE tunnel, label 21 is pushed onto the stack. Label 21 is switched with label 43 at LSR 3. Label 43 is switched with label 56 at LSR 4. LSR 5 pops both labels, pushes label 17, and forwards the packet to LER 6.

On the LDP routers that are on the edge of the core, you must configure a list of peer addresses. The LDP router sends targeted hello messages to those addresses in order to establish targeted sessions across the RSVP-TE domain. The list includes other LDP routers on the edge of the core; for example, in Figure 58 on page 260, you include the address of LSR 5 in the list configured on LSR 2.

Related Documentation

- MPLS Label Distribution Methodology on page 231
- MPLS Label Switching and Packet Forwarding Overview on page 222
- MPLS Label Distribution Protocols Overview on page 242
- IP Data Packet Mapping onto MPLS LSPs Overview on page 233

LDP Graceful Restart Overview

The graceful restart mechanism minimizes the negative effect on MPLS forwarding across an LSR restart. You can configure the neighbors of the LSR to wait for the LSR to restart (helper mode). When the LSR restarts, the neighbors mark their current label mapping entries from the LSR as stale. If the LSR recovers within the proper interval, the entries are no longer marked as stale and are used as they were before the LDP connection failed. If the LSR does not recover in time, the entries are deleted.

LDP graceful restart supports only the downstream-unsolicited mode of label distribution. Successful operation of LDP graceful restart requires that stateful SRP switchover (high availability) be configured on the router. Although you can configure LDP graceful restart if stateful SRP switchover is not configured on the router, the graceful restart capability will not function.

You can configure an LSR to restart itself gracefully and to support graceful restart in its neighbors (helper mode), or helper mode alone. In either case, the LSR includes the fault tolerant (FT) session TLV in the LDP initialization messages it sends at session startup. The TLV includes values for the reconnect timeout and the recovery time. When both graceful restart and the helper mode are disabled, the LSR does not include the TLV in its LDP initialization messages.

The configurable reconnect time specifies how long you want the LSR's neighbors to wait for the LSR to resume exchanging LDP messages with the neighbors after the connection failure. The reconnect timeout value is nonzero when graceful restart is enabled.

When you disable graceful restart but enable helper mode, the reconnect timeout is set to zero to announce to the neighbors that the LSR does not preserve MPLS forwarding state across the restart. The presence of the TLV indicates to the neighbors that the LSR supports them if they gracefully restart. That is, the LSR in this case waits for a gracefully restarting neighbor to resume sending messages.

Table 54 on page 261 summarizes the states possible for LDP graceful restart.

Table 54: Summary of LDP Graceful Restart States

Graceful restart enabled?	Helper mode enabled?	FT TLV sent to neighbor?	Reconnect timeout value sent in TLV
Yes	Yes	Yes	Nonzero
No	Yes	Yes	Zero
No	No	No	–

The recovery time specifies how long the LSR retains its MPLS forwarding state across the restart. When the LSR restarts, it marks the forwarding state entries as stale. The forwarding state holding timer begins counting down from the configured recovery time value. If the timer expires before the restart completes, the LSR deletes all stale entries. When the LSR sends new LDP initialization messages to its neighbors, the messages contain the current value of the timer.

NOTE: An LDP router processes LDP initialization messages received from peers with protocol data units (PDU) lengths in the range of 2095 to 4096. The router responds to the initialization message with the same PDU length that it received from the peer, causing the peer to accept the LDP session.

When the LSR restarts, if a neighbor of the LSR has previously received the FT session TLV from the LSR with a nonzero reconnect timeout value, the neighbor retains the label mapping information that it has previously received from the LSR and marks that information as stale. Alternatively, if the neighbor received an FT session TLV with a timeout value of zero (indicating that only helper mode is enabled) or no TLV at all (indicating that both graceful restart and helper mode are disabled), it deletes the label mapping information.

Also when the LSR restarts, the neighbor sets its neighbor liveness timer to the lesser of the two values, the reconnect timeout value and its own configurable neighbor liveness timer value. If the neighbor liveness timer expires, the neighbor deletes all the stale mappings from the LSR. The configurable value represents the maximum time that the neighbor waits for the restarting LSR to reestablish the LDP session. This enables the neighbor to avoid having to wait an unreasonably long time set by the reconnect timeout value from the restarting LSR.

If the recovery time value in the FT session TLV is zero when a neighbor receives the new LDP initialization message, the neighbor deletes all the stale mappings from the LSR.

If the recovery time value is nonzero, the neighbor starts a neighbor recovery timer set to the lesser of the two values, the recovery time value and its own configurable maximum recovery timeout value. The neighbor also cancels its neighbor liveness timer because the LDP session has been reestablished; it is now waiting on the successful completion of the restart.

The restarting LSR and its neighbors then exchange label mapping information. When a neighbor receives a label–to–FEC binding that matches a stale entry, it removes the staleness marker from the entry. If instead the neighbor receives a new label for the same FEC that is in a stale entry, the neighbor updates the entry with the new label and removes the staleness marker from the entry.

The neighbor deletes any stale entries that remain when the neighbor recovery timer expires.

Dynamic exchange of the graceful restart capability is not supported. In some circumstances, such as when a standby SRP module is removed, an LSR that has communicated to neighbors that it supports graceful restart might subsequently be unable to do so. In such cases, the neighbors receive no indication of that change in support unless you bounce the LDP sessions, for example by issuing the **clear mpls ldp neighbor** command.

Related Documentation

- Configuring LDP Graceful Restart on page 294

LDP-IGP Synchronization Overview

LDP is often used to establish MPLS LSPs throughout a complete network domain using an IGP such as OSPFv2 or IS-IS. In such a network, all links in the domain have IGP adjacencies as well as LDP adjacencies. LDP establishes the LSPs on the shortest path to a destination as determined by IP forwarding.

MPLS data packets can be discarded in these networks when the network IGP is operational on a link for which LDP is not fully operational, because there is no coupling between the LDP operational state and the IGP. When LDP is not fully operational, LDP is considered to not be synchronized with the IGP.

This issue is especially significant for applications such as a core network that does not employ BGP. Another example is an MPLS VPN where each given PE router depends on the availability of a complete MPLS forwarding path to the other PE routers for each VPN that it serves. This means that along the shortest path between the PE routers, each link must have an operational hello adjacency and an operational LDP session, and MPLS label bindings must have been exchanged over each session.

When LDP has not completed exchanging label bindings with an IGP next hop, traffic is discarded if the head end of the LSP forwards traffic because the LSP is assumed to be in place. The following are some examples of when this can happen.

- When an LDP hello adjacency or an LDP session with a peer is lost due to some error while the IGP still points to that peer. IP forwarding of traffic continues on the IGP link associated with the LDP peer rather than being shifted to another IGP link with which LDP is synchronized.
- When a new IGP link comes up, causing the next hop to a certain destination to change in the IGP's SPF calculations. Although the IGP might be up on the new link, LDP might not have completed label exchange for all the routes. This condition might be transient or due to a misconfiguration.

The LDP protocol is unable to indicate to a dependent service the availability of an uninterrupted LSP to the desired destination. LDP-IGP synchronization minimizes this disruption due to LDP not being operational on a link for which the IGP is operational. When synchronization is in effect, the IGP advertises the maximum possible cost or metric for that link. If an alternative next hop exists for traffic, the IGP can choose that next hop for forwarding. If LDP is operational for that next hop, then no traffic is discarded.

The IGP does not advertise the original cost or metric for the link until either LDP label exchange has been completed with the peer on the link or a configured amount of time has passed (the holddown period).

With synchronization configured, LDP notifies the IGP to advertise the maximum cost for the link when one of the following triggering events takes place:

- The LDP hello adjacency goes down.
- The LDP session goes down.
- LDP is configured on an interface.

If the holddown timer has been configured, the timer starts when the triggering event takes place. When the timer expires, LDP notifies the IGP to resume advertising the original cost.

If the holddown timer has not been configured, the IGP waits (endlessly) until bindings have been received from downstream routers for all the FECs that have a next hop on that interface. Only after that takes place does LDP notify the IGP to bring down the cost on the interface.

LDP-IGP synchronization is supported only for directly connected peers and links with the platform label space.

Synchronization Behavior During Graceful Restart

LDP-IGP synchronization does not take place while the IGP is in the process of a graceful restart. When the graceful restart completes, links for which synchronization has been configured are advertised with maximum metrics in either of the following cases:

- LDP is not yet operational on the link and no holddown timer has been configured.
- The configured holddown timer has not expired.

During LDP graceful restart, no synchronization operations are done. If the LDP graceful restart is terminated, LDP notifies the IGPs to advertise the links with the maximum metric.

Synchronization Behavior on LAN Interfaces

LDP-IGP synchronization does not take place on LAN interfaces unless the IGP has a point-to-point connection over the LAN configured on the interface. The reason for this is that multiple LDP peers might be connected on such an interface unless a point-to-point connection to a single peer has been configured. Because synchronization raises the cost on the interface high enough to prevent traffic being forwarded to that link, if multiple peers are connected, the cost is raised on all the peers even though LDP might be unsynchronized with only one of the peers. Consequently, traffic is diverted away from all the peers, an undesirable situation.

Synchronization Behavior on IGP Passive Interfaces

On IGP passive interfaces, the link cost is not raised when LDP-IGP synchronization is configured and a triggering event occurs.

Synchronization and TE Metrics

When traffic engineering is configured for an IGP, LDP-IGP synchronization does not affect the traffic engineering metric advertised for the link, regardless of whether the TE metric is explicitly configured or the default value.

Related Documentation

- Configuring LDP-IGP Synchronization on page 295

Use of RSVP-TE Hello Messages to Determine Peer Reachability

RSVP-TE hello messages enable the router to detect when an RSVP-TE peer is no longer reachable. When the router makes this determination, all LSPs that traverse that neighbor are torn down. Hello messages are optional and can be ignored safely by peers that are not configured to use the feature.

When you enable the hello feature on a virtual router or interface configured with RSVP-TE, that RSVP-TE node periodically sends a unicast hello message to each neighbor with which the node has established an LSP. The exchange of hello messages between the peers establishes a hello adjacency. You can configure the hello interval to establish how frequently the node sends hello messages. Hello messages are exchanged when an LSP is set up and are stopped when the last LSP between the two peers goes away.

You can use the hello feature to reduce the impact of RSVP-TE on system resources. Because a hello timeout is treated as a link failure, RSVP-TE can use the hello timeout instead of path and resv timeouts to determine when to bring down an LSP. High RSVP-TE refresh values reduce the amount of control traffic (and CPU cycles) needed by RSVP-TE to refresh LSP state across the network, thus reducing the impact of RSVP-TE on system resources.

Hello Message Objects

Hello messages can contain a hello request object or a hello ack object. These objects provide a way to request an instance value from a peer and to provide an instance value to a peer. Hello requests are sent to establish and confirm an adjacency with a peer. Hello acks are sent in response to hello requests. However, a hello adjacency peer can treat a hello request as an implicit response to its own request, thus reducing the amount of polling that needs to be done for efficient failure detection.

Hello Message Instances

Each object includes a source instance and a destination instance. The sender generates the source instance for its relationship with the receiver. The value of the source instance is unique for each peer. A given source instance value does not change while the two peers are successfully exchanging hello messages.

The destination instance is simply the source instance value that an RSVP-TE node has most recently received from its peer. If the node has never received a hello message from that peer, then it sets the destination instance value to zero.

Hello adjacency peers monitor the source instance sent by their neighbors. When a peer detects that the value has changed or that its neighbor does not properly report the source instance that the peer transmitted, then the peer treats that neighbor as if it has reset. In these cases, the local peer changes the instance value that it advertises to the neighbor.

Sequence of Hello Message Exchange

When a peer receives a hello message with a hello request object, the receiver generates a hello message with a hello ack object. If the receiver has never received a hello from the sender and the source instance is nonzero, then the receiver updates the destination instance that it sends in response with this new value. When the original sender first receives a hello ack from the peer in response to the hello request, the sender updates the destination instance that it sends in the subsequent hello request with the nonzero source instance it receives in the hello ack.

Consider the following example. An LSP has been established between peers A and B. These adjacent peers have not yet exchanged hello messages.

1. Peer A sends a hello request to Peer B. The request object contains the following:
    - Source instance = 5 (generated by Peer A for this adjacency)
    - Destination instance = 0 (because it has never exchanged messages with Peer B)
2. Peer B receives the hello request and sends a hello ack to Peer A. The ack object contains the following:
    - Source instance = 8 (generated by Peer B for this adjacency)
    - Destination instance = 5 (because that is what Peer B detected in the source instance from peer A)
3. Peer A receives the hello ack and sends another hello request to peer B. The request object contains the following:
    - Source instance = 5 (generated by Peer A for this adjacency)
    - Destination instance = 8 (the source instance generated by Peer B for this adjacency)

The two peers continue exchanging hello messages until the LSP is torn down. The following is true for these message exchanges unless a peer resets:

- Peer A always sends source instance= 5 and destination instance= 8 to Peer B.
- Peer B always sends instance= 8 and destination instance= 5 to Peer A.

Determination That a Peer Has Reset

After the initial exchange of hello messages, both peers perform checks on the messages they receive to determine whether the peer has reset.

Behavior of the Requesting Peer

The requesting peer examines the ack messages it receives. It compares the source instance in each subsequent ack message with the previous value. If the value differs or is set to zero, then the requesting peer treats the acknowledging peer as if communication has been lost.

The requesting peer also determines whether the acknowledging peer is reflecting back the requesting peer's source instance. If the acknowledging peer advertises a wrong value in the destination instance field of the ack message, then the requesting peer treats the acknowledging peer as if communication has been lost.

Behavior of the Acknowledging Peer

The acknowledging peer examines the request messages it receives. It compares the source instance in each subsequent request message with the previous value. If the value differs or is set to zero, then the acknowledging peer treats the requesting peer as if communication has been lost.

The acknowledging peer also determines whether the requesting peer is reflecting back the acknowledging peer's source instance. It compares the destination instance value in each request message with the source instance value that it most recently advertised to the requesting peer. If the requesting peer advertises a wrong value in the destination instance field of the request message, then the acknowledging peer treats the requesting peer as if communication has been lost.

Behavior of Both Peers

When no hello messages are received from a peer within the configured hello interval, the peer is treated as if communication has been lost.

When a peer determines that communication has been lost, it can reinitiate the sending of hello messages. In this case, the peer generates a new source instance different than the one it previously used for communication with its peer.

Related Documentation

- Configuring RSVP-TE Hello Messages to Determine Peer Reachability on page 302

RSVP-TE Graceful Restart Overview

RSVP-TE graceful restart enables routers to maintain MPLS forwarding state when a link or node failure occurs. In a link failure, control communication is lost between two nodes, but the nodes do not lose their control or forwarding state.

A node failure occurs when the LSR has a failure in the RSVP-TE control plane, but not in the data plane. The LSR maintains its data forwarding state. Traffic can continue to be forwarded while RSVP-TE restarts and recovers. The graceful restart feature supports the restoration and resynchronization of RSVP-TE states and MPLS forwarding state between the restarting router and its RSVP-TE peers during the graceful restart recovery period.

The RSVP-TE graceful restart feature enables an LSR to gracefully restart, to act as a graceful restart helper node for a neighboring router that is restarting, or both.

Announcement of the Graceful Restart Capability

LSRs use the RSVP-TE hello mechanism to announce their graceful restart capabilities to their peer RSVP-TE routers. Both restarting LSRs and helper LSRs include the restart_cap object in hello requests and hello acks. The restart_cap object specifies both the graceful restart time and the graceful restart recovery time:

- restart time—The sum of how long it takes the sender to restart RSVP-TE after a control plane failure plus how long it takes to reestablish hello communication with the neighboring RSVP-TE routers.
- recovery time—The period within which you want neighboring routers to resynchronize with the sending router's RSVP-TE state and MPLS forwarding state after the peers have re-established hello communication. The restarting LSR advertises the configured or default recovery time only while the graceful restart is in progress. When the LSR is not currently restarting or when it is incapable of preserving its MPLS forwarding state during the restart, the LSR advertises a recovery time of zero.

Both the restarting router and neighboring GR helper routers save the restart and recovery times that they receive from their peers.

Restarting Behavior

When the control plane fails, the LSR stops sending hello messages to its RSVP-TE neighbors. However, as a restarting router the LSR can continue to forward MPLS traffic because it preserves its MPLS forwarding state during the restart. When RSVP-TE comes back up, the restarted router sends the first hello message to its neighbors with a new source instance value to indicate that it had a control plane failure. The destination instance value in the hello message is set to zero. The recovery time included in this hello message is set to zero only if the router was unable to preserve the MPLS forwarding state or to support control state recovery.

When a neighboring router that has been configured as a graceful restart helper determines that the number of continuous missing hellos has reached the configured hello miss limit, it declares the router to be down. The helper router then waits for a period equal to the restart time that it received from the router and stored before the failure. During this period, the helper router preserves the restarting router's RSVP-TE state and MPLS forwarding state for the established LSPs and keeps forwarding MPLS traffic. However, the helper router suspends the refreshing of path and resv state to the restarting router. The helper router keeps sending hello messages to the restarting router with an unchanged source instance value and a destination instance value set to zero. The helper router removes the RSVP-TE state for any LSP that was in the process of being established when the neighbor was declared to be down.

If the helper router does not receive a hello message from the restarting router during the restart period, the helper router immediately exits the recovery procedure and cleans up the states associated with the restarting router. The helper router determines that the failed neighbor has restarted when it finds a new source instance in the neighbor's hello message. When a nonzero recovery time is received in that hello message, the helper router determines that the restarted neighbor supports state recovery. The helper router then starts the recovery procedures. However, if the recovery time specified in the hello message is zero, then the helper router exits the recovery procedure and cleans up the states associated with the restarting router.

Recovery Behavior

In the recovery period, neighboring helper routers and the restarting router resynchronize the RSVP-TE state and MPLS forwarding state. During this period, MPLS traffic continues to be forwarded.

The helper router starts the recovery procedure by marking as stale the RSVP-TE state associated with the restarting router. The upstream helper router then refreshes all the path messages shared with the downstream restarting router. The upstream helper router includes the recovery_label object in the path message to the downstream restarting router for the label binding information that the restarting router specified before the restart. The downstream helper router does not refresh the reservation state control block (RSB) shared with the restarting router until a corresponding path message is received from the restarting router.

During the recovery period, the restarting router checks for the state associated with an incoming path message. If the RSVP-TE state already exists, the restarting router handles the path message as usual. Otherwise, the restarting router examines the path message for the recovery_label object. If the recovery_label object is not found, the restarting router treats the path message as a setup request for a new LSP and handles the path message as usual.

If the recovery_label object is found, the restarting router searches for the outgoing label based on the incoming interface and incoming label that are specified in the recovery_label object. If the restarting router does not find a match for the forwarding entry, the restarting router treats the path message as a setup request for a new LSP. If the restarting router finds a match, it conveys to the downstream neighbors the outgoing label associated with the forwarding entry in the suggested_label object in the path message and it continues normal operations.

The helper router removes the stale flag for the RSVP-TE state when it receives the corresponding state in path or resv messages sent by the restarting router. When the recovery period expires, the helper router deletes any RSVP-TE states that still have a stale flag. Graceful restart is considered to be complete when the recovery period expires or when the last LSP needing recovery is recovered.

Preservation of an Established LSP Label

Labels used for an established LSP are preserved through the graceful restart by means of the recovery_label object and the suggested_label object in the path messages. The recovery_label object conveys the incoming label of the restarting LSR that the restarting LSR passed to the upstream helper before the restart. The suggested_label object includes the outgoing label that the restarting LSR used before the restart. The suggested_label object conveys the outgoing label from the restarting LSR to its downstream neighbor.

Related Documentation

• Configuring RSVP-TE Graceful Restart on page 303

RSVP-TE Hellos Based on Node IDs Overview

For interoperability with routers that cannot support RSVP-TE graceful restart with link-based hellos, you can use the **mpls rsvp signaling node-hello** command to configure the exchange of node-ID–based RSVP-TE hellos (node hellos). E Series routers use node hellos only to support their graceful restart capabilities.

NOTE: Node hellos are not required for RSVP-TE graceful restart support between routers running JunosE Software or for interoperability with routers running Junos OS.

Graceful restart must be enabled for node hellos to advertise graceful restart. Link-based hellos are not required for graceful restart when you have configured node hellos. However, you might still use link-based hellos to monitor RSVP-TE links and detect link failures.

The node hello sessions are established by the exchange of hello messages in which node IDs are used for the source and destination addresses in the hello packets. The sending router uses its local node ID as the source address and the remote node ID of the receiving router as the destination address.

RSVP-TE uses the configured IGP, IS-IS or OSPF, to learn the local and remote node IDs. In IS-IS, the node ID is the TE router ID as defined in the traffic engineering router ID TLV for IPv4 addresses and in the IPv6 TE Router_ID for IPv6 addresses. In OSPF, the node ID is the TE router ID as defined in the router address TLV for IPv4 addresses and in the Router_IPv6_Address for IPv6 addresses. Only one node-based RSVP-TE hello session can be established for each instance of an IGP adjacency with a peer.

When a router receives a hello message where the destination address is set to the receiving router's local node ID, the router verifies that the node ID is the ID that the IGP advertises. This router must then use its local node ID as the source address when it replies to the sending router.

Node-based hellos are an attractive alternative to link-based hellos for graceful restart when you use bidirectional forwarding detection (BFD) for link monitoring and you have configured node-based hellos on all RSVP-TE peers.

Link-based RSVP-TE hellos are used for monitoring RSVP-TE adjacencies with neighboring routers and for providing RSVP-TE graceful restart. However, the BFD protocol is more effective at monitoring RSVP-TE adjacencies than are link-based hellos.

Link-based RSVP-TE hellos for graceful restart are more resource-intensive option than node-based RSVP-TE hellos when your configuration has several interfaces enabled with MPLS RSVP-TE and carrying RSVP-TE data traffic. Link-based hellos generate a volume of network traffic and processing overhead that is directly proportional to the number of interfaces that are carrying active RSVP-TE tunnels.

Node-based hellos require less messaging and processing overhead in these circumstances. Node hellos require only a single hello session between the two node IDs, compared to link-based hellos that have hello sessions between all interface pairs. Less traffic and overhead result in a lesser impact on scaling.

Node-based hellos can therefore be advantageous even when you are interoperating with routers that are running JunosE Software or Junos OS, if you are using BFD to monitor your RSVP-TE links. If you are not using BFD, then you must use link-based hellos for link monitoring, and link-based hellos then become more practical for graceful restart.

Related Documentation

- Configuring RSVP-TE Hellos Based on Node IDs on page 304

BFD Protocol and RSVP-TE Overview

The Bidirectional Forwarding Detection (BFD) protocol uses control packets and shorter detection time limits to more rapidly detect failures in a network. Also, because they are adjustable, you can modify the BFD timers for more or less aggressive failure detection.

Without BFD, RSVP-TE can learn about adjacency failures by either of two methods. If RSVP-TE hellos are configured, then hello message timeouts indicate a failure. If hellos are not configured, then RSVP-TE learns about failures from resv and path messages.

When a BFD session exists between RSVP-TE peers, a peer that goes down is detected quickly, enabling faster rerouting of traffic. Adjacency failure detection by means of hello messages takes place on the order of seconds, whereas BFD fast failure detection can take place on the order of hundreds of milliseconds.

When you issue the **mpls rsvp bfd-liveness-detection** command on an RSVP-TE major interface, BFD liveness detection is established with all BFD-enabled RSVP-TE peers associated with that interface.

When an RSVP-TE session is established with the remote peer—if BFD is enabled and if the BFD session is not already present—then the local peer attempts to create a BFD session to the remote peer. The BFD session is established only if when both of the following are true:

- At least one RSVP-TE LSP exists between (passes through) a pair of directly connected RSVP-TE major interfaces.
- Both interfaces are BFD-enabled.

Consequently, when the last LSP is torn down between the interfaces, the BFD session is no longer required and is brought down as well.

Each adjacent pair of peers negotiates an acceptable transmit interval for BFD packets. The negotiated value can be different on each peer. Each peer then calculates a BFD liveness detection interval. When a peer does not receive a BFD packet within the detection interval, it declares the BFD session to be down and purges all routes learned from the remote peer.

For general information about configuring and monitoring the BFD protocol, see *JunosE IP Services Configuration Guide*.

Related Documentation

- Configuring the BFD Protocol for RSVP-TE on page 304

Tunneling Model for Differentiated Services Overview

The JunosE Software supports both the pipe model and the uniform model for tunneling with the **mpls tunnel-model** command. The router also provides a way to implement the functionality of the short pipe model for IP packets.

Pipe and Short Pipe Models

In the pipe and short pipe models, any traffic conditioning (that is, in a pure JunosE environment, a change in traffic class/color combination) that is applied when traffic goes through the tunnel has no effect on the EXP bits coding in the inner header. In other words, when traffic exits an LSP (when a label is popped) or when traffic enters an LSP, the inner header's EXP bits coding is not changed.

The pipe and short pipe models differ in the header that the tunnel egress uses when it determines the PHB of an incoming packet. With the short pipe model, the tunnel egress uses an inner header that is used for forwarding. With the pipe model, the outermost label is always used. Because of this, you cannot use PHP with the pipe model.

The pipe model is the default JunosE behavior, which you can configure with the **mpls tunnel-model** command. You cannot configure the short pipe model with this command. In fact, on ingress line modules the traffic class/color combination is always determined from the outermost label, so fabric queuing is also based on the outermost label. However, on the egress line module you can achieve the queuing behavior expected with the short pipe model by attaching IP policies to egress interfaces to reset the traffic class/color combinations based on the IP header. However, this method requires that the outgoing packets to be IP. If the outgoing packets are MPLS, then this short pipe model of queuing is not supported.

Uniform Model

The uniform model of tunneling renders MPLS transparent to the differentiated services operation. From the diff-serv perspective, it is as if MPLS is not used. In the uniform model, if traffic conditioning is applied somewhere along the LSP, the EXP bits of the inner header must be changed at the egress when the inner header becomes the outer header (because of the pop of the outer label).

To specify whether MPLS uses the pipe or uniform model of tunneling for differentiated services:

- Issue the **mpls tunnel-model** command.

  host1(config)#**mpls tunnel-model uniform**

For detailed information about QoS, see the *JunosE Quality of Service Configuration Guide*.

Related Documentation

- Configuring MPLS and Differentiated Services on page 308
- Configuring the Tunneling Model for Differentiated Services on page 309
- *mpls tunnel-model*

EXP Bits for Differentiated Services Overview

MPLS matches on the EXP bits for incoming traffic to set the traffic class/color combination, and sets the EXP bits for outgoing traffic based on the traffic class/color combination.

Incoming Traffic

For incoming MPLS traffic, the traffic class/color combination is set according to the EXP bits in the outermost label, either per the policy attached to the label or per the per-VR rules. The policy has precedence over the per-VR rules. Therefore, fabric queuing is always based on the outer label's EXP bits.

If the traffic is label-switched through the router, the EXP bits value associated with the incoming label that is used for switching—which can be either an outermost label or an inner label after popping one or more outer labels—is passed onto the egress line module. This behavior enables the EXP bits value to be copied to outgoing labels, used to reset the traffic class/color combination on the egress module, or both.

Outgoing Traffic

Outgoing traffic is queued according to traffic class/color combinations. The applied combination can be the same as was set on the ingress line module, or it can be reset on the egress line module by egress IP policy.

Figure 59 on page 274 illustrates how the initial value of the EXP bits is set for the first label pushed. Figure 60 on page 275 illustrates how the EXP bits can be changed for all labels, including the first label, by attached policies or per-VR EXP rules. The following section describes in detail how the EXP bits value is set for outgoing traffic.

Setting the EXP Bits for Outgoing Traffic

Different types of packets distributed into LSPs by the router have different default settings for the EXP bits. For IP packets, the EXP bits value is set to match the IP precedence value from the TOS field of the packet header. For non-IP packets, such as Martini or VPLS packets, the EXP bits value is set to 000. You can use the **mpls copy-upc-to-exp** command to free the EXP bits value in IP packets from being tied to the IP precedence value. Instead, this command sets the EXP bits value to match the user packet class (UPC) value.

The IP precedence value can be copied back into the IP precedence field of the IP packet header at the LSP endpoint on the ingress line module. This action takes place only if the IP header is exposed after popping the MPLS labels and if the uniform tunnel model is employed. The remaining bits of the TOS field are not touched.

In contrast, when you issue **mpls copy-upc-to-exp** command, the EXP bits value is not copied to the UPC field at the LSP endpoint, because the UPC value might have been set by a lower layer policy for a different purpose.

NOTE: For control traffic originated from this router, if an attached per-LSP policy has rules to modify the EXP bits, or if per-VR EXP rules are configured, the EXP bits value copied from the IP precedence value might be overwritten incorrectly because the default traffic class/color combination for control traffic is best-effort/green. You can avoid this situation by establishing an outgoing IP policy that sets the traffic class/color combination for control traffic so that the policy or rules have the correct traffic class/color to work with.

If per-LSP policies are used or per-VR rules are configured, by default all labels pushed by the router for the same packet have the same EXP bits value. That value is determined by the policies or rules.

You can use the **mpls preserve-vpn-exp** command to specify that the EXP bits value for the VPN or Martini or VPLS label pushed by the router cannot be modified by either policy for outer labels or by per-VR rules. This capability is useful if you want the inner labels to have a different value for the EXP bits than do the outer labels. For example, in a VPN you might want the inner label's EXP bits value to be the copied IP precedence value. You might want the base label's EXP bits value set according to the mapping of EXP bits to traffic class/color combination that is defined in your network.

Figure 59: Flow for Initial Setting of EXP Bits for the First Label Pushed

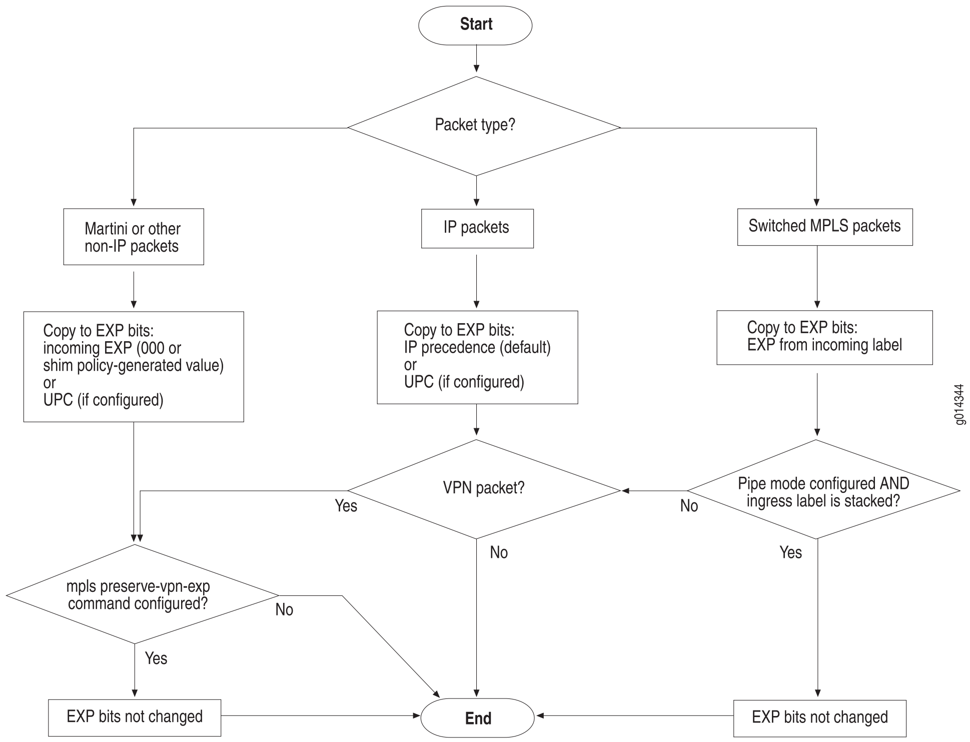
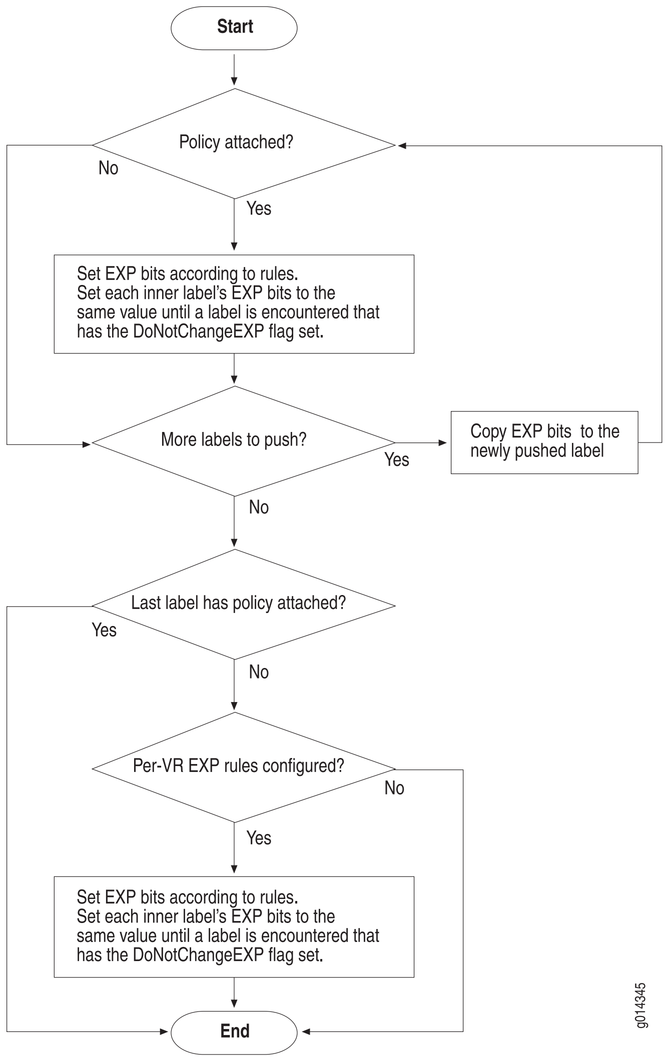


Figure 59 on page 274 shows how packet type and configuration determine how the EXP bits are set for the first label pushed.

Figure 60: Flow for Setting EXP Bits for All Pushed Labels



Related Documentation

- Configuring EXP Bits for Differentiated Services on page 309
- Configuring MPLS and Differentiated Services on page 308

Point-to-Multipoint LSPs Overview

A point-to-multipoint MPLS LSP is an RSVP-TE LSP with a single ingress LSR and one or more egress LSRs. You can use point-to-multipoint LSPs to avoid unnecessary duplication of packets at the ingress router by allowing non-ingress LSRs to replicate the incoming data on one or more outgoing interfaces. Point-to-multipoint LSPs for multicast VPNs are supported for intra-autonomous system (AS) environments (within an AS), but are not supported for inter-AS environments (between ASs).

Although you can use point-to-point LSPs to provide point-to-multipoint services, this type of configuration can cause data replication at the ingress LSR or duplicate traffic within the network. You can use the traffic engineering (TE) capability of LSPs to achieve consistent QoS control and efficient use of network resources, and create point-to-multipoint LSPs to deliver data from one ingress LSR to multiple egress LSRs. The flow of traffic in a point-to-multipoint LSP is not restricted to the paths that are followed for multicast or shortest path routing; instead, you can explicitly configure the values to determine the path. Packet replication takes place only when packets are forwarded to two or more different destinations requiring different network paths.

A point-to-multipoint TE tunnel is composed of multiple point-to-multipoint LSPs. To scale to a large number of nodes or branches in a point-to-multipoint LSP, each LSP is uniquely identified by a point-to-multipoint ID, which is unique for the entire LSP, regardless of the number of branches or leaves it contains. A point-to-multipoint LSP is composed of multiple source-to-leaf sub-LSPs. These sub-LSPs are formed between the ingress and egress LSRs to form the point-to-multipoint LSP.

Point-to-multipoint LSPs can be signaled using one or more path messages. If a path message signals only one sub-LSP, it targets only one leaf in the point-to-multipoint tunnel. Because a single path message might not be large enough to contain all the sub-LSPs in the tunnel and also because you can create path messages specific to a sub-LSP in the tunnel, you can use multiple path messages. However, if you want to minimize the number of control messages required to configure a point-to-multipoint tunnel, you need to use a single path message to signal multiple sub-LSPs.

The following are some of the benefits of using point-to-multipoint LSPs:

- A point-to-multipoint LSP allows you to use MPLS TE for point-to-multipoint data distribution. This functionality provides better control over the path chosen to transmit traffic than that provided by IP multicast.
- You can add and remove branch LSPs from a main point-to-multipoint LSP without disrupting traffic. The unaffected parts of the point-to-multipoint LSP continue to function normally.
- You can enable link protection on a point-to-multipoint LSP. Link protection can provide a bypass LSP for each of the branch LSPs that make up the point-to-multipoint LSP. If any of the primary paths fail, traffic can be quickly switched to the bypass path.
- You can configure branch LSPs statically, dynamically, or as a combination of static and dynamic LSPs.
- You can enable graceful restart on point-to-multipoint LSPs.

Using E Series Routers as Egress LSRs

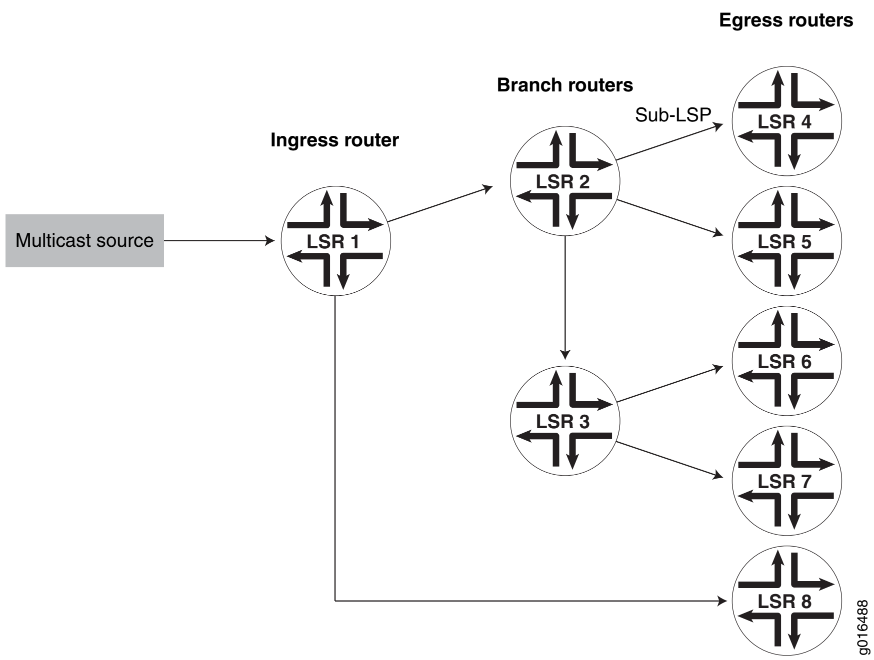
You can use E Series routers as egress LSRs in a point-to-multipoint LSP. To create a point-to-multipoint LSP and to use E Series routers as egress LSRs, no special configuration is required. The configuration that you made for point-to-point LSPs, which enables MPLS RSVP-TE on the interface that must signal an LSP in that virtual router context, is sufficient.

Figure 61 on page 277 shows a point-to-multipoint LSP with multiple egress LSRs. The multicast source sends a packet to the ingress router, LSR 1, which in turn sends the packet on the point-to-multipoint LSP to the branch router, LSR 2. The branch router, LSR 2, is connected to another branch router, LSR 3. Here, LSR 3 is not directly connected to the ingress router, LSR 1, but only to the branch router, LSR 2. These branch routers, in turn, replicate the packet and forward it to E Series routers, LSRs 4 through 7, configured as egress LSRs.

The configuration shown in Figure 61 on page 277 is an example of an LSP that contains segments that run from ingress LSR to one or more branch and egress LSRs. For example, sub-LSPs exist between LSR 1 and LSR 2, and between LSR 2 and LSR 4. The sub-LSP between LSR 2 and LSR 4 is an egress sub-LSP that transmits the replicated packet from branch router, LSR 2, to egress E Series router, LSR 4. Egress LSRs can also be directly connected to the ingress LSR. In this figure, the connection between LSR 8 and LSR 1 is an example of this type.

NOTE: You cannot use E Series routers as core or ingress LSRs. You need to use Juniper Networks routers running Junos OS to function as core or ingress LSRs in the point-to-multipoint LSP.

Figure 61: Simple MPLS Domain



Use the **show mpls rsvp tunnels p2mp role tail** command to view the status and configuration information for point-to-multipoint egress tunnels.

Related Documentation

- Monitoring Status and Configuration for MPLS Tunnels on page 375
- Configuring Point-to-Multipoint LSPs on page 322
- *show mpls tunnels*

CHAPTER 4

Configuring MPLS

This chapter describes how to configure Multiprotocol Label Switching (MPLS) on the router, and contains the following sections:

- Classifying Traffic for Differentiated Services on page 313
- Example Traffic Class Configuration for Differentiated Services on page 318
- Configuring Point-to-Multipoint LSPs on page 322

Basic MPLS Configuration Tasks

Configuring an MPLS network includes a number of tasks:

To configure an MPLS network:

- Configure settings common to all MPLS usage on a given LSR.

  See "MPLS Global Configuration Tasks" on page 281.

- (Optional) Configure LDP or RSVP-TE interface profiles.

  See "LDP and RSVP-TE Interface Profile Configuration Tasks" on page 284.

- Configure each interface on an LSR that uses MPLS.

  See "MPLS Interface Configuration Tasks" on page 285.

- Configure MPLS tunnels or topology-driven LSPs.

  See "MPLS Tunnel Configuration Tasks" on page 287.

- (Optional) Configure a profile that contains settings to be used by multiple MPLS tunnels.

  See "MPLS Tunnel Profile Configuration Tasks" on page 289.

Many users find it convenient to configure MPLS by completing the tasks in each set of tasks before moving to the next set. However, you do not have to complete the tasks in the listed order. For example, you might perform all the pure MPLS tasks relevant to your network and then perform all the relevant LDP or RSVP-TE tasks.

The type of network you want to implement determines which sets of tasks you must complete, as indicated in Table 55 on page 280.

Table 55: Configuration Tasks by Type of Network

Task Set	Traffic Engineering Network	Topology-Driven Network (Best-Effort, Hop-by-Hop, LDP)
Global	Yes	Yes
Interface Profile	Optional	Optional
Interface	Yes	Yes
Tunnel	Yes	No
Tunnel Profile	Yes	No

In addition to the basic configuration tasks, you might need to perform other LDP or RSVP-TE configuration tasks.

To configure LDP and RSVP-TE, depending on your network topology and needs:

- Configure LDP features depending on your network design.

  See "Additional LDP Configuration Tasks" on page 292.

- Configure RSVP-TE features depending on your network design.

  See "Additional RSVP-TE Configuration Tasks" on page 298.

MPLS Global Configuration Tasks

Complete these tasks to configure a virtual router as an LSR. You perform these commands in Global Configuration mode. The following sequence is arbitrary; you can perform these tasks in any order.

Your choice of label distribution protocol determines whether the LDP or RSVP-TE global tasks are appropriate for your network design.

MPLS global configuration tasks include the following sets of tasks:

MPLS Global Tasks

In a typical network, you perform only the first task. You might also perform the optional configuration tasks, but typically do not need to do so.

1. Enable MPLS on a virtual router.

   host1(config)#mpls

2. (Optional) Configure the time-to-live field placed in the MPLS header when a label is first added to an IP packet.

   host1(config)#mpls ip propagate-ttl forwarded

3. (Optional) Configure the tunneling model for differentiated services. See "Configuring MPLS and Differentiated Services" on page 308 for more information and command descriptions.

   host1(config)#mpls tunnel-model uniform

4. (Optional) Specify whether to use the TOS value (the default condition) or the UPC value of the packet as the value of the EXP bits when the router acts as an LER.

   host1(config)#mpls copy-upc-to-exp

5. (Optional) Specify whether the EXP bits for VPN MPLS labels can be modified by EXP bit mapping or by policy for differentiated services.

   host1(config)#mpls preserve-vpn-exp

6. (Optional) Specify whether to create dynamic IP interfaces on top of MPLS major interfaces and optionally what profile to use for them.

   host1(config)#**mpls create-dynamic-interfaces ip on-major-interfaces profile v4intf**

LDP Global Tasks

Typically, you do not configure anything for LDP at the global level, but you can perform the following optional tasks.

1. (Optional) Enable LDP and topology-driven LSP. Any LDP-related command creates LDP implicitly, negating the need to issue this command.

   host1(config)#**mpls ldp**

2. (Optional) Configure the redistribution of IGP routes to LDP.

   host1(config)#**mpls ldp redistribute ospf route-map boston5**

3. (Optional) Configure a global LDP profile that specifies how long an LSR maintains link hello records before another link hello is sent, the interval between link hellos, or both.

   host1(config)#**mpls ldp interface profile ldp1**
   host1(config-ldp)#**hello hold-time 55**
   host1(config-ldp)#**hello interval 10**

4. (Optional) Configure lists of peer addresses that targeted hello messages are sent to or accepted from.

   host1(config)#**mpls ldp targeted-hello send list 10.21.5.87**
   host1(config)#**mpls ldp targeted-hello receive list 192.168.45.25**

   NOTE: The **mpls ldp targeted-hello receive list** command is unnecessary if you configure the **mpls ldp targeted-hello send list** command.

5. (Optional) Configure the hold time and interval values for targeted hello messages used in LDP extended discovery.

   host1(config)#**mpls ldp targeted-hello holdtime 90**
   host1(config)#**mpls ldp targeted-hello interval 30**

6. (Optional) Configure LDP session retry values.

   host1(config)#**mpls ldp session retry-time 2**
   host1(config)#**mpls ldp session retries 1800**

7. (Optional) Configure the period that LDP negotiates with its peer for which the LDP session is maintained in the absence of any LDP messages.

   host1(config)#**mpls ldp session holdtime 1800**

8. (Optional) Configure the interval at which LDP sends session keepalive messages.

   host1(config)#**mpls ldp session keepalive-time 180**

9. (Optional) Specify an IP address to be advertised to peers as the transport address in discovery hello messages.

host1(config)#**mpls ldp discovery transport-address 192.168.34.2**

10. (Optional) Configure independent control as the method of label distribution that LDP uses.

    host1(config)#**mpls ldp independent-control**

11. (Optional) Configure LDP to advertise the explicit null label or a non-null label for the egress router to achieve ultimate hop popping.

    host1(config)#**mpls ldp egress-label explicit-null**

For topology-driven LSPs, perform the following LDP configuration tasks.

1. (Optional) Configure the LSR to create topology-driven LSPs. Enabling LDP automatically creates topology-driven LSPs.

   host1(config)#**mpls topology-driven-lsp**

2. (Optional) Specify filters for the routes and peers to which the labels are advertised.

   host1(config)#**mpls ldp advertise-labels host-only**

3. (Optional) Specify the LSPs to be put into the IP routing table for forwarding plain IP traffic.

   NOTE: **This step is not optional if you are using a topology-driven network to forward plain IP packets.**

   host1(config)#**ldp ip-forwarding host-only**

4. (Optional) Establish a policy governing the distribution of incoming LDP labels.

   host1(config)#**mpls ldp advertise-labels for boston1**

5. (Optional) Remove and then reestablish existing LDP LSPs and to restart topology-driven LDP. Use this command when you have modified or created policies or access lists (with the **mpls ldp-ip-forwarding** and **mpls ldp advertise-labels** commands) and want them to be applied to LDP LSPs that are already in an up state.

   host1#**clear mpls ldp**

RSVP-TE Global Tasks

Typically, you do not configure anything for RSVP-TE at the global level, but you can perform the following optional tasks.

1. (Optional) Enable RSVP-TE. Any RSVP-TE–related command creates RSVP-TE implicitly, negating the need to issue this command.

   host1(config)#**mpls rsvp**

2. (Optional) Configure a global RSVP-TE profile that specifies the timeout period in milliseconds between generation of RSVP refresh messages, the number of refresh messages that can be lost before the PATH or RESV state is ended, or both.

   host1(config)#**mpls rsvp interface profile rsvp4**
   host1(config-rsvp)#**refresh-period 60000**

host1(config-rsvp)#**cleanup-timeout-factor 9**

3. (Optional) Configure retry timer options globally (to apply to all tunnels) to set up an LSP after a setup failure (a failure *other* than one due to no available route). Specify the number of attempts to be made to set up an RSVP-TE tunnel or the interval in seconds between attempts.

   host1(config)#**mpls lsp retries 35**
   host1(config)#**mpls lsp retry-time 55**

4. (Optional) Configure retry timer options globally (to apply to all tunnels) to set up an LSP after a failure due to no available route. Specify the number of attempts to be made to set up an RSVP-TE tunnel or the interval in seconds between attempts.

   host1(config)#**mpls lsp no-route retries 3200**
   host1(config)#**mpls lsp no-route retry-time 45**

5. (Optional) Configure the interval at which the bandwidth values are flooded.

   host1(config)#**mpls traffic-eng link-management timers periodic-flooding 10**

6. (Optional) Configure reoptimization—the frequency at which MPLS searches for better paths for existing tunnels.

   NOTE: Low timer values lead to frequent reoptimization of LSPs, which is undesirable for the following reasons:

   - Frequent changes to the LSPs increases packet loss.
   - Frequent reoptimization increases the load on the router, especially when the router acts as the LSP head end. The load is particularly noticeable in a scaled network, resulting in high CPU utilization on the router.

   host1(config)#**mpls reoptimize timers frequency 180**

   You can also force an immediate search for better paths for all existing LSPs.

   host1#**mpls reoptimize**

7. (Optional) Enable refresh reduction and message bundling.

   host1(config)#**mpls rsvp refresh-reduction**
   host1(config)#**mpls rsvp message-bundling**

8. (Optional) Configure the egress router to advertise the explicit null label.

   host1(config)#**mpls rsvp egress-label explicit-null**

LDP and RSVP-TE Interface Profile Configuration Tasks

The interface profile configuration tasks are optional tasks you may need to perform to configure your network's label distribution options.

LDP and RSVP-TE interface profile configuration tasks include the following sets of tasks:

- LDP Interface Profile Configuration Tasks and Commands on page 285
- RSVP-TE Interface Profile Configuration Tasks and Commands on page 285

LDP Interface Profile Configuration Tasks and Commands

Creating or accessing an LDP interface profile places the CLI in LDP Configuration mode.

1. Access LDP profile configuration mode.

   host1(config)#**mpls ldp interface profile ldp5**

2. Configure LDP interface profile settings, changing the values from the implicit default values.

   host1(config-ldp)#**hello hold-time 30**
   host1(config-ldp)#**hello interval 10**

RSVP-TE Interface Profile Configuration Tasks and Commands

Creating or accessing an RSVP-TE interface profile places the CLI in RSVP Configuration mode. When you have completed the interface profile configuration tasks, you may want to proceed to the section "MPLS Interface Configuration Tasks" on page 285.

1. Access the desired profile configuration mode.

   host1(config)#**mpls rsvp interface profile rsvp4**

2. Configure interface profile settings to define the RSVP tunnel timeout period by specifying the timeout period in milliseconds between generation of RSVP refresh messages, the number of refresh messages that can be lost before the PATH or RESV state is ended, or both.

   host1(config-rsvp)#**refresh-period 60000**
   host1(config-rsvp)#**cleanup-timeout-factor 9**

See "MPLS Overview" on page 213, "RSVP-TE Messages and Sessions" on page 244 for more information about the RSVP tunnel timeout period.

MPLS Interface Configuration Tasks

These tasks are performed at the major interface over which you want to run MPLS. Creating or accessing an interface places the CLI in Interface Configuration mode. You can then configure MPLS options on that interface. The following sequence is arbitrary; you can perform these tasks in any order.

NOTE: Loop detection is always enabled in the JunosE MPLS implementation.

Your choice of label distribution protocol determines whether the LDP or RSVP-TE interface configuration tasks are appropriate for your network design.

MPLS interface configuration tasks include the following sets of tasks:

- MPLS Interface Tasks on page 286
- LDP Interface Tasks on page 286
- RSVP-TE Interface Tasks on page 286

MPLS Interface Tasks

To configure MPLS on the interface:

1. Enable MPLS on the interface.

   host1(config-if)#mpls

   or

   host1(config-if)#no mpls disable

2. (Optional) Configure the interface label space with the VPI and VCI ranges.

   host1(config-if)#mpls atm vpi range 10 200
   host1(config-if)#mpls atm vci range 33 4000

   Only ATM AAL5 interfaces support the interface label space. For MPLS interfaces using the interface label space, you must configure the both the VPI and VCI range. If you do not, the interface will remain operationally down.

3. (Optional) Specify an interface for signaling for an MPLS major interface in the interface label space.

   host1(config-if)#mpls signaling-interface atm 4/0.5

LDP Interface Tasks

To configure LDP on the interface:

1. Start LDP on the interface.

   - Using the default values (an implicit *default* profile):

     host1(config-if)#mpls ldp

   - Using a previously created profile:

     host1(config-if)#mpls ldp profile ldp5

2. (Optional) Suppress transmission of link hello messages to all LSRs.

   host1(config-if)#mpls ldp link-hello disable

RSVP-TE Interface Tasks

To configure RSVP-TE on the interface:

1. Start RSVP-TE on the interface.

   - Using the default values (an implicit *default* profile):

     host1(config-if)#mpls rsvp

- Using a previously created profile:

  host1(config-if)#**mpls rsvp profile rsvp4**

To disable RSVP-TE on the interface:

host1(config-if)#**mpls rsvp disable**

2. (Optional) Configure total bandwidth available on the interface.

   host1(config-if)#**bandwidth 262144**

3. (Optional) Configure total bandwidth reservable for MPLS on the interface.

   host1(config-if)#**mpls bandwidth 4096**

4. (Optional) Specify thresholds that trigger bandwidth flooding when crossed by an increase or decrease in the total reservable bandwidth.

   host1(config-if)#**mpls traffic-eng flood thresholds up 15**
   host1(config-if)#**mpls traffic-eng flood thresholds down 15**

5. (Optional) Specify the resource attributes for the interface so that tunnels can discriminate among interfaces.

   host1(config-if)#**mpls traffic-eng attribute-flags 0x000001f9**

6. (Optional) Configure an administrative weight for the interface that overrides the weight assigned by the IGP.

   host1(config-if)#**mpls traffic-eng administrative-weight 25**

MPLS Tunnel Configuration Tasks

Complete the following tasks to configure a tunnel interface. Configure the tunnel endpoint last; anything configured after the tunnel endpoint does not take effect until the tunnel is brought up the next time. You can perform all other tasks in any order.

NOTE: **Tunnel configuration tasks are relevant only for traffic engineering networks.**

1. Create the MPLS tunnel interface.

   host1(config)#**interface tunnel mpls:boston**

2. (Optional) Configure the LSP to announce its endpoint to an IGP (sometimes referred to as *registering the endpoint*).

   host1(config-if)#**tunnel mpls autoroute announce isis**

3. (Optional) Specify a tunnel metric to be used by an IGP in its SPF calculation.

   host1(config-if)#**tunnel mpls autoroute metric absolute 100**

4. (Optional) Configure the path options used for the tunnel.

   host1(config-if)#**tunnel mpls path-option 3 dynamic isis**

5. (Optional) Configure the bandwidth required for the tunnel.

host1(config-if)#tunnel mpls bandwidth 1240

6. (Optional) Configure preemption hold or setup priority.

   host1(config-if)#tunnel mpls traffic-eng priority 4 4

7. (Optional) Configure resource class affinity.

   host1(config-if)#tunnel mpls traffic-eng affinity 0x000C3000 0xFFFFFFFC

   This example of a masked affinity matches only links configured with attribute flags from 0x000C3000 to 0x000C3003.

8. (Optional) Configure retry timer options to apply to a specific tunnel to set up an LSP after a route or setup failure.

   host1(config-if)#tunnel mpls no-route retries 100
   host1(config-if)#tunnel mpls no-route retry-time 45
   host1(config-if)#tunnel mpls retries 250
   host1(config-if)#tunnel mpls retry-time 65

9. (Optional) Associate a text description with the tunnel.

   host1(config-if)#tunnel mpls description southshore

10. Configure the tunnel endpoint.

   host1(config-if)#tunnel destination 10.12.21.5

Related Documentation

- MPLS Global Configuration Tasks on page 281
- MPLS Interface Configuration Tasks on page 285
- MPLS Tunnel Profile Configuration Tasks on page 289
- Additional LDP Configuration Tasks on page 292
- Additional RSVP-TE Configuration Tasks on page 298

- *interface tunnel*
- *tunnel destination*
- *tunnel mpls affinity*
- *tunnel mpls autoroute announce*
- *tunnel mpls autoroute metric*
- *tunnel mpls bandwidth*
- *tunnel mpls description*
- *tunnel mpls no-route retries*
- *tunnel mpls no-route retry-time*
- *tunnel mpls path-option*
- *tunnel mpls priority*
- *tunnel mpls retries*
- *tunnel mpls retry-time*

MPLS Tunnel Profile Configuration Tasks

If you anticipate having multiple tunnels to share the same configuration, you can reduce your configuration time by using tunnel profiles to configure your tunnels.

In the profile, configure the tunnel endpoint last; anything configured after the tunnel endpoint does not take effect until the tunnel is brought up the next time. You can perform all other tasks in any order.

NOTE: **Tunnel profile configuration tasks are relevant only for traffic engineering networks.**

To configure a tunnel profile:

1. Create an MPLS tunnel profile and enter Tunnel Profile Configuration mode.

   host1(config)#**mpls tunnels profile Lisbon**

2. (Optional) Configure the LSP to announce its endpoint to an IGP.

   host1(config-tunnelprofile)#**tunnel mpls autoroute announce isis**

3. (Optional) Specify a tunnel metric to be used by an IGP in its SPF calculation.

   host1(config-tunnelprofile)#**tunnel mpls autoroute metric absolute 100**

4. (Optional) Configure the path options used for the tunnel.

   host1(config-tunnelprofile)#**tunnel mpls path-option 3 dynamic isis**

5. (Optional) Configure the bandwidth required for the tunnel.

   host1(config-tunnelprofile)#**tunnel mpls bandwidth 1240**

6. (Optional) Configure preemption hold or setup priority.

   host1(config-tunnelprofile)#**tunnel mpls priority 4 4**

7. (Optional) Configure resource class affinity.

   host1(config-tunnelprofile)#**tunnel mpls affinity 0x1100 mask 0xFFFF**

8. (Optional) Configure retry timers options to apply to a specific tunnel to set up an LSP after a route or setup failure.

   host1(config-tunnelprofile)#**tunnel mpls no-route retries 100**
   host1(config-tunnelprofile)#**tunnel mpls no-route retry-time 45**
   host1(config-tunnelprofile)#**tunnel mpls retries 250**
   host1(config-tunnelprofile)#**tunnel mpls retry-time 65**

9. (Optional) Associate a text description with the tunnel.

   host1(config-tunnelprofile)#**tunnel mpls description southshore**

10. Configure the tunnel endpoint.

    - For static tunnels

      host1(config-tunnelprofile)#**tunnel destination 10.1.2.5 10.1.2.6**

All tunnels to the specified destination(s) are configured with the profile settings.

- For dynamic tunnels

  host1(config-tunnelprofile)#tunnel destination isis-level-2 access-list madrid3

When an endpoint is dynamically learned from the specified routing protocol, MPLS searches its tunnel profiles for a match. The dynamic tunnel is established using the settings from the first matching profile.

Related Documentation

- MPLS Global Configuration Tasks on page 281
- MPLS Interface Configuration Tasks on page 285
- MPLS Tunnel Configuration Tasks on page 287
- Additional LDP Configuration Tasks on page 292
- Additional RSVP-TE Configuration Tasks on page 298

- *mpls tunnels profile*
- *tunnel destination*
- *tunnel mpls affinity*
- *tunnel mpls autoroute announce*
- *tunnel mpls autoroute metric*
- *tunnel mpls bandwidth*
- *tunnel mpls description*
- *tunnel mpls no-route retries*
- *tunnel mpls no-route retry-time*
- *tunnel mpls path-option*
- *tunnel mpls priority*
- *tunnel mpls retries*
- *tunnel mpls retry-time*

Configuring Explicit Routing for MPLS

When you configure explicit routing rather than hop-by-hop routing for MPLS, the route the LSP takes is defined by the ingress node. The path consists of a series of hops defined by the ingress LSR. Each hop can be a traditional interface, an autonomous system, or an LSP.

MPLS explicit routing configuration tasks include the following sets of tasks:

- Defining Configured Explicit Paths on page 291
- Specifying Configured Explicit Paths on a Tunnel on page 291
- Configuring Dynamic Explicit Paths on a Tunnel on page 291

Defining Configured Explicit Paths

You can create explicit routing paths *manually* by configuring an explicit path with a name and a series of addresses (hops) from ingress to egress.

To manually configure explicit routing:

1. Define an explicit path and access Explicit Path Configuration mode.

   host1(config)#**mpls explicit-path name xyz**
   host1(config-expl-path)#

2. Do one of the following to configure the hops in the LSP:

   - Set the next hop (if need be) at a particular index in the explicit path.

     host1(config-expl-path)#**index 5 next-address 172.18.100.5**

   - Add the next hop (if need be) after a particular index in the explicit path.

     host1(config-expl-path)#**append-after 5 next-address 192.168.47.22**

3. Configure a next hop at the end of the MPLS explicit path.

   host1(config-expl-path)#**next-address 10.10.9.2**

4. Enable the explicit path.

   host1(config)#**mpls explicit-path name xyz**

   NOTE: To prevent a partially configured explicit path from being used, do not enable it until you have finished configuring or modifying the path.

5. (Optional) List the currently configured explicit path.

   host1(config-expl-path)#**list 5**

Specifying Configured Explicit Paths on a Tunnel

After you have defined a configured explicit path, you can configure the path on a tunnel.

To configure explicit routing on a tunnel:

1. Create an MPLS tunnel.

   host1(config)#**interface tunnel mpls:1**

2. Set the path option.

   host1(config-if)#**tunnel mpls path-option 1 explicit name xyz**

Configuring Dynamic Explicit Paths on a Tunnel

You can create explicit routing paths *dynamically* with a routing protocol. IS-IS and OSPF both currently support explicit routing.

To configure dynamic explicit routing:

1. Create an MPLS tunnel.

   host1(config)#**interface tunnel mpls:bilbao5**

2. Set the path option.

   host1(config-if)#**tunnel mpls path-option 2 dynamic isis**

NOTE: If you configure the MPLS RSVP-TE tunnels to dynamically calculate the explicit path for the tunnels and if a router link-state advertisement (LSA) sends a metric of 65535 for one of the neighbors that is being considered while the Constrained Shortest Path First (CSPF) computation is performed, the metric that is sent in the router LSA is used in the CSPF calculation. Processing the router LSA by the CSPF algorithm ensures that the MPLS RSVP-TE tunnels are not retained in the down state and the tunnels come up properly.

Additional LDP Configuration Tasks

Several of the LDP configuration tasks are optional, and depend on your network topology and needs.

Tasks to configure LDP settings include:

- Configure LDP FEC deaggregation depending on your network design.

  See "Configuring LDP FEC Deaggregation" on page 292.

- Configure the LDP graceful restart mechanism depending on your network design.

  See "Configuring LDP Graceful Restart" on page 294.

- Configure LDP autoconfiguration depending on your network design.

  See "Configuring LDP Autoconfiguration" on page 295.

- Configure LDP-IGP synchronization depending on your network design.

  See "Configuring LDP-IGP Synchronization" on page 295.

- Configure LDP MD5 authentication depending on your network design.

  See "Configuring LDP MD5 Authentication" on page 296.

- Create a filter that determines whether and where LDP labels are distributed depending on your network design.

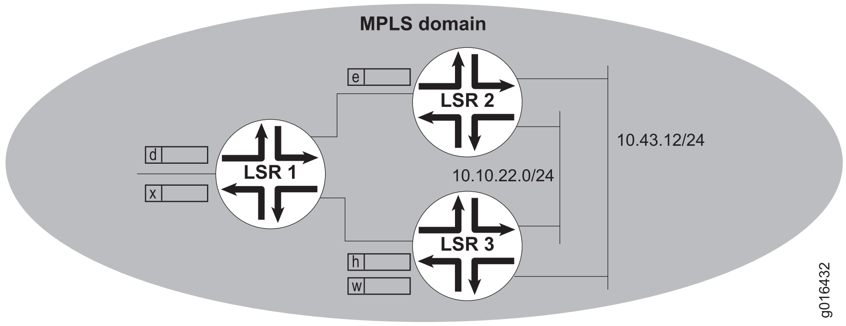
  See "Controlling LDP Label Distribution" on page 297.

Configuring LDP FEC Deaggregation

Beginning with JunosE Release 8.1.0, LDP routers running JunosE Software employ LDP FEC aggregation by default. FEC aggregation means that when an LDP egress router advertises multiple prefixes, all the prefixes are members of the same FEC. Only a single label is advertised for this FEC. LDP maintains this aggregation as the advertisement traverses the network, if possible.

Consider the topology shown in Figure 62 on page 293.

Figure 62: FEC Aggregation and Equal-Cost Paths



In this example, LSR 2 uses FEC aggregation, but LSR 3 does not. Consequently, LSR 2 advertises the single label *e*, mapped to a FEC that includes both prefixes, 10.10.22.0/24 and 10.43.12.0/24.

In contrast, LSR 3 has two FECs, one for 10.10.22.0/24 and one for 10.43.12.0/24. A separate label is bound to each FEC. LSR 3 advertises label *h* for one FEC and label *w* for the other FEC.

LSR 2 and LSR 3 are downstream routers for LSR 1. LSR 1 does not aggregate. Instead, LSR 1 advertises label *d* for 10.10.22.0/24 and label *x* for 10.43.12.0/24.

To configure MPLS LDP FEC deaggregation to bind each prefix on the current virtual router to a separate label:

- Issue the **mpls ldp deaggregate** command:

  host1(config)#**mpls ldp deaggregate**

If you configure MPLS LDP FEC deaggregation to bind a separate label to each prefix on a virtual router, the default behavior is for the LDP egress router to advertise the implicit null label in the label mapping message that it sends to its upstream neighbor. If multiple labels are configured for the LDP egress router and FEC aggregation is configured, when you modify the label advertisement method to be FEC deaggregation, the egress router advertises the implicit null label and does not use separate labels with each prefix. With FEC deaggregation configured, the egress router's upstream neighbor performs a penultimate hop pop (PHP) and the implicit null label never appears in the encapsulation. If you configure MPLS LDP FEC deaggregation, the default advertised label is label 3 (implicit null label). In such a scenario, the penultimate-hop router removes the label and sends the packet to the egress router.

Related Documentation

- Basic MPLS Configuration Tasks on page 280
- Additional LDP Configuration Tasks on page 292
- *mpls ldp deaggregate*

Configuring LDP Graceful Restart

The graceful restart mechanism minimizes the negative effect on MPLS forwarding across an LSR restart by enabling neighbors to wait for the LSR to restart and causing the LSR itself to restart gracefully.

NOTE: Successful operation of LDP graceful restart requires that stateful SRP switchover (high availability) be configured on the router. Although you can configure LDP graceful restart if stateful SRP switchover is not configured on the router, the graceful restart capability will not function.

To configure LDP graceful restart:

1. Enable LDP graceful restart and graceful restart helper mode.

   host1(config)#**mpls ldp graceful-restart**

2. (Optional) Specify the length of time you want the neighbors to wait for the gracefully restarting router to resume sending LDP messages to neighbors after the LDP connection between them fails.

   host1(config)#**mpls ldp graceful-restart reconnect-time 130**

3. (Optional) Specify the length of time the router retains its MPLS forwarding state across a restart.

   host1(config)#**mpls ldp graceful-restart recovery-time 150**

4. (Optional) Specify the maximum length of time that the router waits for a neighbor to complete a graceful LDP restart after the LDP session is reestablished.

   host1(config)#**mpls ldp graceful-restart timers max-recovery 150**

5. (Optional) Specify the length of time the router waits for a neighbor to reestablish the LDP session.

   host1(config)#**mpls ldp graceful-restart timers neighbor-liveness 150**

NOTE: For information about configuring hold timers for LDP graceful restart in scaled environments, see the *Configuring Hold Timers for Successful Graceful Restart in Scaled Scenarios* section in the *JunosE BGP and MPLS Configuration Guide*

Related Documentation

- Basic MPLS Configuration Tasks on page 280
- Additional LDP Configuration Tasks on page 292
- *mpls ldp graceful-restart*
- *mpls ldp graceful-restart reconnect-time*
- *mpls ldp graceful-restart recovery-time*

- *mpls ldp graceful-restart timers max-recovery*
- *mpls ldp graceful-restart timers neighbor-liveness*

Configuring LDP Autoconfiguration

LDP autoconfiguration enables you to ensure that LDP is configured on all interfaces running the IGP (IS-IS or OSPFv2). Using this command prevents you from having to configure LDP individually on each interface in the IGP. You can prevent LDP from being enabled on selected interfaces by issuing the **no mpls ldp autoconfig** command on the interface.

When autoconfiguration is enabled for IS-IS, you can specify whether LDP is automatically configured on all IS-IS interfaces in the virtual router or just the interfaces in a particular IS-IS level. When autoconfiguration is enabled for OSPF, you can specify whether LDP is automatically configured on all OSPF interfaces in the virtual router or just the interfaces in a particular OSPF area.

To configure LDP autoconfiguration to ensure that LDP is configured on all interfaces running the IGP:

- Specify whether LDP is created automatically on the current interface or all interfaces:
  - Create LDP on all interfaces in the IGP router context

    host1(config)#**router ospf 1**
    host1(config-router)#**mpls ldp autoconfig area 1**
  - Create LDP on the current interface

    host1(config)#**interface atm 2/0**
    host1(config-if)#**mpls ldp isis autoconfig**

Related Documentation

- Basic MPLS Configuration Tasks on page 280
- Additional LDP Configuration Tasks on page 292
- *mpls ldp autoconfig*

Configuring LDP-IGP Synchronization

MPLS data packets can be discarded in a network when the network IGP is operational on a link for which LDP is not fully operational, because there is no coupling between the LDP operational state and the IGP. When LDP is not fully operational, LDP is considered to not be synchronized with the IGP.

To configure LDP-IGP synchronization:

1. Specify whether LDP is synchronized with the IGP on the current interface or all interfaces.
   - Synchronize LDP with the IGP on the current interface:

host1(config)#**interface atm 2/0**
host1(config-if)#**mpls ldp isis sync**

- Synchronize LDP with the IGP on all interfaces in the IGP router context:

  host1(config)#**router ospf 1**
  host1(config-router)#**mpls ldp ospf sync**

2. Specify how long the IGP waits to synchronize with LDP.

   host1(config)#**mpls ldp igp sync holddown 15**

Related Documentation

- Basic MPLS Configuration Tasks on page 280
- Additional LDP Configuration Tasks on page 292

- *mpls ldp igp sync holddown*
- *mpls ldp sync*

Configuring LDP MD5 Authentication

LDP MD5 authentication provides protection against spoofed TCP segments that can be introduced into the connection streams for LDP sessions. Authentication is configurable for both directly connected and targeted peers.

You configure a shared secret (password) on potential LDP peers. Any given pair of peers must share the same password. When a peer sends a TCP segment to an LSR, it uses the password and the segment to compute an MD5 digest that it sends along with the segment.

When the LSR receives the segment, the LSR calculates its own version of the digest using its instance of the password and the segment. The LSR validates the segment if the local digest matches the received digest. If the comparison fails—for example, if the password is not configured the same on both peers—the LSR drops the segment and does not send a response to the peer.

You can optionally enable a strict authentication mode that allows only peers configured with passwords to establish sessions. In this mode, LDP hello messages from peers that have no password are ignored. If you do not configure strict authentication, then peers that do not have configured passwords can establish connections with each other.

If you configure LDP MD5 authentication or change the authentication password for a peer while it is in an established LDP session, MPLS restarts that session.

To configure LDP MD5 authentication:

1. Set the password for an LDP peer.

   host1(config)#**mpls ldp neighbor 10.3.5.1 password rop23ers**

2. (Optional) Set strict LDP authentication mode so that only peers with passwords can establish LDP sessions.

   host1(config)#**mpls ldp strict-security**

Related Documentation

- Basic MPLS Configuration Tasks on page 280
- Additional LDP Configuration Tasks on page 292
- *mpls ldp neighbor password*
- *mpls ldp strict-security*

Controlling LDP Label Distribution

By default, LDP advertises label mappings for all IGP prefixes to all LDP peers. In this case, mappings are not advertised for interface addresses. You can alternatively specify that LDP labels be distributed for a particular interface itself, in addition to the subnet that the interface is on. This behavior enables LSPs to be set up to the LSR configured with the interface address.

When the LSR learns an IGP route and tries to decide whether to advertise a label for the destination to a particular LDP neighbor, it attempts to match the destination against a route access list specified by the **mpls ldp advertise-labels** command, in the order in which the commands were issued. The first match determines the action taken, and no further matching is attempted for that destination. If the destination matches, labels are advertised to peers subject to any specified neighbor address list. If either access list is not matched, the labels are not advertised.

To create a filter that determines whether and where incoming (locally assigned) labels are distributed:

- Issue the mpls ldp advertise-labels command one or more times:

  ```
  host1(config)#mpls ldp advertise-labels for net25 to euro3
  ```

When you do not specify a *toAccessList*, the action is taken for all peers.

Consider the following example configuration.

```
host1(config)#mpls ldp advertise-labels for net25 to euro3
host1(config)#mpls ldp advertise-labels for boston1
```

In this example, suppose the LSR receives a label for destination 10.10.11.12. If net25 specifies 10.10.11.12, then the access list action—permit or deny—is taken with the destination. If the action is permit, the peer that the label is advertised to is subject to the access list euro3. If net25 does not include 10.10.11.12, the LSR attempts to match it against boston1. If 10.10.11.12 is present in that access list, the specified action is taken for all peers. If boston1 does not include the destination, the label is not advertised to any peer.

Related Documentation

- Basic MPLS Configuration Tasks on page 280
- Additional LDP Configuration Tasks on page 292
- *mpls ldp advertise-labels*

Additional RSVP-TE Configuration Tasks

All of the following RSVP-TE configuration tasks are optional, depending on your network topology and needs.

Tasks to configure RSVP-TE settings based on your network topology include:

- Configure RSVP MD5 authentication to provide hop-by-hop security.

  See "Configuring RSVP MD5 Authentication" on page 298.
- Configure fast reroute extensions to RSVP-TE to create a bypass tunnel.

  See "Configuring RSVP-TE Fast Rerouting with RSVP-TE Bypass Tunnels" on page 299.
- Configure RSVP-TE peers to exchange hello messages and establish a hello adjacency.

  See "Configuring RSVP-TE Hello Messages to Determine Peer Reachability" on page 302.
- Configure RSVP-TE graceful restart to enable routers to maintain MPLS forwarding state when a link or node failure occurs.

  See "Configuring RSVP-TE Graceful Restart" on page 303.
- Configure the exchange of RSVP-TE node hellos on all RSVP-TE interfaces.

  See "Configuring RSVP-TE Hellos Based on Node IDs" on page 304.
- Configure the BFD Protocol for RSVP-TE.

  See "Configuring the BFD Protocol for RSVP-TE" on page 304.

Configuring RSVP MD5 Authentication

RSVP MD5 authentication provides hop-by-hop security against message spoofing and replay attacks. When authentication is configured, RSVP embeds an integrity object within secure cleartext RSVP messages sent between peers. The integrity object includes a key ID unique to the sender, a message sequence number, and keyed message digest. These attributes enable verification of both packet content and sender.

For all potential RSVP peers, you configure the same key on the MPLS neighbor major interfaces, and then enable RSVP authentication on each of these interfaces. When you enable RSVP authentication on an interface, RSVP creates a security association that includes the key, key ID, hash algorithm, and other associated attributes. Each sender and receiver pair maintains the security association for their shared key.

NOTE: **You must enable authentication on both ends of an RSVP interface to protect the link. Failure to do so can prevent tunnels through the interface from coming up.**

Thereafter, RSVP messages sent by a router through the secured interface include an integrity object that contains a key ID for the security association and an MD5 message digest of the message contents. To protect against message replay attacks, the sending interface also places a sequence number in the integrity object. Each sequence number is a unique, monotonically increasing number.

The secured interface expects each received RSVP message to include an integrity object. The interface drops all RSVP messages that do not contain the object.

The receiver uses the key ID and the sender's address to determine the relevant security association. The key ID is extracted from the received integrity object. The address of the sending interface is extracted from the rsvp_hop object, if present, or from the packet header if the message does not include the rsvp_hop object. The receiver then recomputes the message digest using the association key and algorithm and compares it to the digest received from the peer.

If the digests match, RSVP checks the received sequence number. Every message received from a sender after the first authenticated message must have a sequence number greater than the number from a previously authenticated message from that sender. Messages with invalid sequence numbers are discarded.

If the sequence number is valid, then the RSVP message is authenticated and forwarded for normal RSVP processing. Unauthenticated messages are discarded.

To configure RSVP-TE MD5 authentication:

1. Assign a key to the interface for MD5 authentication between RSVP peers.

   host1(config-if)#**mpls rsvp authentication key 34udR973j**

2. Enable MD5 authentication on the RSVP-TE interface.

   host1(config-if)#**mpls rsvp authentication**

To clear the security association on a receiving peer for the specified sending peer:

- Issue the **clear mpls rsvp authentication** command:

  host1#**clear mpls rsvp authentication 10.3.5.1**

Related Documentation

- Basic MPLS Configuration Tasks on page 280
- Additional RSVP-TE Configuration Tasks on page 298

- *clear rsvp authentication*
- *mpls rsvp authentication*
- *mpls rsvp authentication key*

Configuring RSVP-TE Fast Rerouting with RSVP-TE Bypass Tunnels

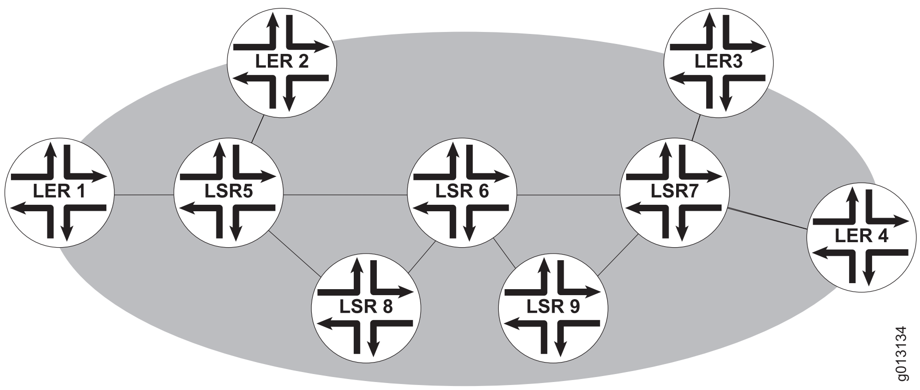
The fast reroute extensions to RSVP-TE enable you to create a single LSP, known as a bypass tunnel, to back up a set of LSPs by bypassing specific links in the LSP. In the event of a failure in any link of the protected RSVP-TE LSP (the primary LSP), MPLS redirects traffic to the associated bypass tunnel in tens of milliseconds.

You must statically configure the bypass tunnel for each link that you want to protect on each router in the LSP. The bypass tunnel must intersect the protected LSP at two locations. The start of the bypass tunnels is the point of local repair (PLR), which is the head end of the protected link. The bypass tunnel terminates downstream of the PLR on the node that represents the end of the bypassed link on the primary LSP.

Each bypass tunnel provides 1:*N* local protection; that is, each bypass tunnel can protect one or more links depending on where you have configured it. The protected primary LSPs are stacked over the bypass tunnel to redirect their traffic around the failure.

The bypass tunnel naturally protects all LSPs that share the bypassed link (the LSP segment from the PLR to the downstream node) and that have requested protection. Consider the network shown in Figure 63 on page 300.

Figure 63: Bypass Tunnel



Suppose you have configured both LER 1 and LER 2 to request bypass protection, and have configured the following two bypass tunnels:

LSR 5 –> LSR 8 –> LSR 6
LSR 6 –> LSR 9 –> LSR 7

If link LSR 5 –> LSR 6 fails, RSVP-TE redirects traffic through LSR 5 –> LSR 8 –> LSR 6. If link LSR 6 –> LSR 7 fails, RSVP-TE redirects traffic through LSR 6 –> LSR 9 –> LSR 7. These two bypass tunnels therefore protect all LSPs routed from LER 1 or LER 2 through LSRs 5, 6, and 7. Notice in Figure 63 on page 300 that if both protected links fail, traffic is still safely redirected through LSR 5 –> LSR 8 –> LSR 6 –> LSR 9 –> LSR 7.

If you want to protect an LSP that traverses *N* nodes against a failure in any link, then you must configure *N*-1 bypass tunnels. As shown in Figure 63 on page 300, each of those bypass tunnels in turn can protect multiple tunnels.

On detecting the link failure, the PLR redirects traffic arriving on all of the protected primary tunnels to the bypass tunnel that protects the failed link. An additional label representing the bypass tunnel is stacked on the redirected packets. This label is popped either at the router that is the remote end of the protected link or at the penultimate hop. The merge point therefore sees traffic with the original label representing the primary tunnel.

When the ingress router learns by RSVP-TE signaling that local protection (a bypass tunnel) is in use, it attempts to find a new optimal path for the tunnel, based on the configured path options. The ingress router sets up the new tunnel before it tears down the old tunnel with the failed link, and switches its traffic to the new tunnel.

You can use the **tunnel mpls path-option** command to configure path options on the bypass tunnel. However, the link being protected by the bypass tunnel must not be in the path if you specify an explicit path.

Configuration Example

The following steps show a partial configuration using the topology in Figure 63 on page 300:

1. On LSR 5, create a bypass tunnel to protect the link between LSR 5 and LSR 6. If you configure path options, you can specify the explicit path for the bypass tunnel either statically or to be dynamically calculated by the IGP traffic engineering mechanism.

   host1(config)#**interface tunnel mpls:bypass56**
   host1(config-if)#**tunnel mpls path-option 1 explicit name bypass**
   host1(config-if)#**tunnel destination 172.20.1.1**
   host1(config-if)#**exit**

2. On LSR 5, enable the explicit path, if configured.

   host1(config)#**mpls explicit-path name bypass enable**
   host1(config-expl-path)#**next-address 10.10.9.2**
   host1(config-expl-path)#**exit**

3. On LSR 5, assign the bypass tunnel to the interface being protected.

   host1(config)#**interface atm 4/0.1**
   host1(config-if)#**mpls backup-path bypass56**

4. On LER 1 (the tunnel ingress), specify that local protection is required for the primary tunnel.

   host1(config)#**interface tunnel mpls:primary1**
   host1(config-if)#**tunnel mpls fast-reroute**

Fast Reroute over SONET/SDH

If you are using MPLS fast reroute over a SONET/SDH interface, reduce the times that the router uses to convert a defect to an alarm. Use the **path trigger delay** command to reduce the time for the path layer and the **trigger delay** command to reduce the time for the section and line layers. Use the following guidelines to set the times:

- Specify a value of 0 milliseconds if the interface does not use APS/MSP or if you want MPLS to have priority over APS/MSP.
- Specify a value of at least 100 milliseconds if this interface uses APS/MSP and if you want APS/MSP to have priority over MPLS.

For more information about these commands, see *JunosE Physical Layer Configuration Guide*.

Related Documentation

- Basic MPLS Configuration Tasks on page 280
- Additional RSVP-TE Configuration Tasks on page 298
- *mpls backup-path*
- *tunnel mpls fast-reroute*

Configuring RSVP-TE Hello Messages to Determine Peer Reachability

The RSVP-TE hello feature enables RSVP-TE peers to exchange hello messages and establish a hello adjacency. The peers use the adjacency to verify reachability. When a peer is no longer reachable, the LSPs that traverse the neighbor are torn down.

High refresh values reduce the amount of control traffic (and CPU cycles) needed by RSVP-TE to refresh LSP state across the network, reducing the impact of RSVP-TE on system resources:

- A hello refresh interval of 1000 milliseconds (a rate of one hello every second) is appropriate for a small configuration—tens of interfaces—without causing performance degradation.
- For larger configurations, the default hello refresh interval of 10,000 milliseconds (a rate of one hello every 10 seconds) is more appropriate and typically does not cause performance degradation.

To configure the RSVP-TE hello feature on all RSVP-TE interfaces in the VR:

1. Issue the **mpls rsvp signaling hello** command.

   host1:vr5(config)#**mpls rsvp signaling hello**
2. (Optional) Configure the refresh interval.

   host1(config-if)#**mpls rsvp signaling hello refresh interval 5000**
3. (Optional) Configure the refresh misses.

   host1(config-if)#**mpls rsvp signaling hello refresh misses 5**

To configure the RSVP-TE hello feature on a RSVP-TE specific interface:

1. Access the interface.

   host1(config)#**interface fastEthernet 4/3**
2. Issue the **mpls rsvp signaling hello** command.

   host1(config-if)#**mpls rsvp signaling hello**
3. (Optional) Configure the refresh interval.

   host1(config-if)#**mpls rsvp signaling hello refresh interval 5000**
4. (Optional) Configure the refresh misses.

   host1(config-if)#**mpls rsvp signaling hello refresh misses 5**

NOTE: Issuing the **refresh interval** or the **refresh misses** keywords only configures the refresh values; this action has no effect on enabling or disabling hellos.

Related Documentation

- Basic MPLS Configuration Tasks on page 280
- Additional RSVP-TE Configuration Tasks on page 298
- *mpls rsvp signaling hello*

Configuring RSVP-TE Graceful Restart

Configure RSVP-TE graceful restart to enable routers to maintain MPLS forwarding state when a link or node failure occurs. Because forwarding state is maintained, traffic can continue to be forwarded while RSVP-TE restarts and recovers. The RSVP-TE graceful restart feature enables an LSR to gracefully restart, to act as a graceful restart helper node for a neighboring router that is restarting, or both.

To configure RSVP-TE graceful restart on the current VR so that the LSR acts as both a graceful restart restarting node and a graceful restart helper node:

1. Enable RSVP-TE graceful restart on the current virtual router.

   host1(config)#**mpls rsvp signaling hello graceful-restart**

2. (Optional) Configure the recovery time—the time within which you want neighboring routers to resynchronize RSVP-TE state and MPLS forwarding state after a graceful restart.

   host1(config)#**mpls rsvp signaling hello graceful-restart recovery-time 140000**

3. (Optional) Configure the restart time—the time within which the sender gracefully restarts RSVP-TE and reestablishes hello communication with RSVP-TE neighbors.

   host1(config)#**mpls rsvp signaling hello graceful-restart refresh restart-time 150000**

To configure RSVP-TE graceful restart on the current VR so that the LSR acts as only a graceful restart helper node for neighbors that support RSVP-TE graceful restart:

- Issue the **mpls rsvp signaling hello graceful-restart mode help-neighbor** command:

  host1(config)#**mpls rsvp signaling hello graceful-restart mode help-neighbor**

Related Documentation

- Basic MPLS Configuration Tasks on page 280
- Additional RSVP-TE Configuration Tasks on page 298
- *mpls rsvp signaling hello graceful-restart*
- *mpls rsvp signaling hello graceful-restart recovery-time*
- *mpls rsvp signaling hello graceful-restart restart-time*

Configuring RSVP-TE Hellos Based on Node IDs

You can configure the exchange of node-ID–based RSVP-TE hellos (node hellos) for interoperability with routers that cannot support RSVP-TE graceful restart with link-based hellos. E Series routers use node hellos only to support their graceful restart capabilities.

NOTE: **Graceful restart must be enabled on the VR so that the node hellos can advertise the graceful restart capabilities. Link-based hellos are not required for graceful restart when you have configured node hellos. However, you might still use link-based hellos to monitor RSVP-TE links and detect link failures.**

To configure the exchange of RSVP-TE node hellos on all RSVP-TE interfaces in the VR:

1. Enable RSVP-TE graceful restart.

   host1:vr5(config)#**mpls rsvp signaling hello graceful-restart**

2. Enable node hellos.

   host1:vr5(config)#**mpls rsvp signaling node-hello**

3. (Optional) Configure the refresh interval.

   host1(config-if)#**mpls rsvp signaling node-hello refresh interval 12000**

4. (Optional) Configure the refresh misses.

   host1(config-if)#**mpls rsvp signaling node-hello refresh misses 5**

   NOTE: Issuing the **refresh interval** or the **refresh misses** keywords only configures the refresh values; this action has no effect on enabling or disabling RSVP-TE node hellos.

Related Documentation

- Basic MPLS Configuration Tasks on page 280
- Additional RSVP-TE Configuration Tasks on page 298

- *mpls rsvp signaling node-hello*

Configuring the BFD Protocol for RSVP-TE

Configure the Bidirectional Forwarding Detection (BFD) protocol for RSVP-TE to more rapidly detect failures in a network and enable faster rerouting around the failures. You can modify the BFD timers for more or less aggressive failure detection.

When configured, BFD liveness detection is established with all BFD-enabled RSVP-TE peers associated with that RSVP-TE major interface.

NOTE: Before the router can use the **mpls rsvp bfd-liveness-detection** command, you must specify a BFD license key. To view an already configured license, use the **show license bfd** command.

To enable BFD (bidirectional forwarding detection) on an RSVP-TE major interface:

1. Access the interface.

   host1(config)#**interface fastEthernet 4/3**

2. Enable BFD liveness detection.

   host1(config-if)#**mpls rsvp bfd-liveness-detection**

The peers in an RSVP-TE adjacency use the BFD timer values to negotiate the actual transmit intervals for BFD packets.

- Use the **minimum-transmit-interval** keyword to specify the interval at which the local peer proposes to transmit BFD control packets to the remote peer.

  host1(config-if)#**mpls rsvp bfd-liveness-detection minimum-transmit-interval 400**

- Use the **minimum-receive-interval** keyword to specify the minimum interval at which the local peer must receive BFD control packets from the remote peer.

  host1(config-if)#**mpls rsvp bfd-liveness-detection minimum-receive-interval 400**

- Use the **minimum-interval** keyword to specify the same value for both the transmit and receive intervals. Configuring a minimum interval has the same effect as configuring the minimum receive interval and the minimum transmit interval to the same value.

  host1(config-if)#**mpls rsvp bfd-liveness-detection minimum-interval 400**

- Use the **multiplier** keyword to specify the detection multiplier value, which roughly equivalent to the number of packets that can be missed before the BFD session is declared to be down. The calculated BFD liveness detection interval can be different on each peer.

  host1(config-if)#**mpls rsvp bfd-liveness-detection multiplier 15**

  NOTE: You can change the BFD liveness detection parameters at any time without stopping or restarting the existing session; BFD automatically adjusts to the new parameter value. However, no changes to BFD parameters take place until the values resynchronize with each peer.

For details on liveness detection negotiation, see *Negotiation of the BFD Liveness Detection Interval* in the *JunosE IP Services Configuration Guide*.

Related Documentation

- Basic MPLS Configuration Tasks on page 280
- Additional RSVP-TE Configuration Tasks on page 298
- *mpls rsvp bfd-liveness-detection*

Configuring IGPs and MPLS

You can use the **tunnel mpls autoroute announce** command to configure a tunnel to announce its endpoint to IS-IS or OSPF so that the IGP can then use the LSP as a shortcut to a destination based on the LSP's metric.

If no tunnels are registered, the IGP calculates the shortest path to a destination by using the shortest path first (SPF) algorithm. The results are represented by the destination node, next-hop address, and output interface, where the output interface is a physical interface.

If you configure an LSP to be announced to the IGP with a certain metric, the LSP appears as a logical interface directly connected to the LSP endpoint. The IGP can consider the LSP as a potential output interface for the LSP endpoint and for destinations beyond the endpoint. In this case, the SPF computation results are represented by the destination node and the output LSP, effectively using the LSP as a shortcut through the network to the destination.

By default, IS-IS and OSPF always use the MPLS tunnel to reach the tunnel endpoint. Best paths determined by SPF calculations are not considered. You can enable the consideration of best paths by issuing the IS-IS or OSPF **mpls spf-use-any-best-path** command. This command causes the IGP to evaluate the LSP as it does any other path. The IGP then either forwards traffic along the best path (which might be the MPLS tunnel), or load-balances between the MPLS tunnel and another path.

The default behavior applies only to reaching the tunnel endpoint itself. For prefixes downstream of the tunnel endpoint, the value of the tunnel metric always determines whether the IGP uses the LSP or the native path, or load-balances between the native path and one or more LSPs.

The tunnel metric can be absolute or relative. An *absolute* metric indicates there is no relationship to the underlying IGP cost. A *relative* metric is added to or subtracted from the underlying IGP shortest path cost.

Example 1 The following commands announce the tunnel to OSPF and specify a relative metric of -2:

```
host1(config-if)#tunnel mpls autoroute announce ospf
host1(config-if)#tunnel mpls autoroute metric relative -2
```

By default, the LSP is preferred to reach the tunnel endpoint. OSPF will treat this LSP as having a metric of 2 less than the shortest path metric it has calculated. The LSP is therefore also preferred over other paths to prefixes beyond the tunnel endpoint.

Example 2 The following commands announce the tunnel to OSPF, specify an absolute metric of 25, and configure OSPF to enable the consideration of SPF best paths:

```
host1(config-if)#tunnel mpls autoroute announce ospf
host1(config-if)#tunnel mpls autoroute metric absolute 25
...
host1(config)#router ospf 1
host1(config-router)#mpls spf-use-any-best-path
```


OSPF uses this metric in its SPF calculations for traffic to the tunnel endpoint as well as beyond the endpoint. Traffic is routed through this LSP only when the other calculated paths have higher metrics.

Configuring the IGPs for Traffic Engineering

For both IGPs, you must issue two commands to enable the IGP to support traffic engineering.

- IS-IS—Enable the flooding of MPLS traffic-engineering link information into the specified IS-IS level with the **mpls traffic-eng** command. You must also specify a stable router interface with the **mpls traffic-eng router-id** command.

  MPLS traffic engineering also requires that IS-IS generate the new-style TLVs that enable wider metrics. Use the **metric-style wide** command to generate the new-style TLVs. If you are using some IS-IS routers that still cannot interpret the new-style TLVs, use the **metric-style transition** command.

- OSPF—Enable OSPF areas for traffic engineering with the **mpls traffic-eng area** command. OSPF generates opaque LSAs—also known as type-10 opaque link area link states—to flood the traffic-engineering information to the specified area. OSPF builds a traffic-engineering database that it uses in the calculation of shortest path to destinations that satisfy specified traffic-engineering constraints. As with IS-IS, you must also specify a stable router interface with the **mpls traffic-eng router-id** command.

  To enable a multicast network and MPLS traffic engineering (TE) network to interoperate on a router running OSPF, use the **mpls traffic-eng multicast-intact** command.

When you configure a node as the downstream endpoint of an LSP, you must provide a stable interface as the router ID for the endpoint. Typically you select a loopback interface because of its inherent stability. Use the **mpls traffic-eng router-id** command to designate the router as traffic engineering capable and to specify the router ID. For all tunnels that end at this node, set the tunnel destination to the destination node's traffic-engineering router identifier, because the traffic-engineering topology database at the tunnel ingress uses that for its path calculation.

You can use the **show isis mpls** and **show isis database** commands to display information about IS-IS traffic engineering:

For OSPF, you can use the **show ip ospf database opaque-area** command to display information about traffic-engineering opaque LSAs.

See *JunosE IP, IPv6, and IGP Configuration Guide* for more information about enabling IS-IS and OSPF to support traffic engineering and monitoring IS-IS and OSPF traffic engineering.

For information about BGP and MPLS, see "Configuring BGP-MPLS Applications" on page 389.

Related Documentation

- Basic MPLS Configuration Tasks on page 280
- *metric-style narrow*
- *metric-style transition*
- *metric-style wide*
- *mpls spf-use-any-best-path*
- *mpls traffic-eng*
- *mpls traffic-eng area*
- *mpls traffic-eng router-id*
- *show ip ospf database*
- *show isis database*
- *show isis mpls adjacency-log*
- *show isis mpls advertisements*
- *show isis mpls tunnel*
- *tunnel mpls autoroute announce*
- *tunnel mpls autoroute metric*

Configuring MPLS and Differentiated Services

TIP: **Before you read this section, we recommend you be thoroughly familiar with the concepts of the JunosE QoS application.**

MPLS employs several strategies to manage different kinds of data streams based on service plans and priority:

- Different conceptual models of diff-serv tunneling that either conceal intermediate LSP nodes from diff-serv operations or render the MPLS network transparent to the diff-serv operations
- Different strategies to set the EXP bits in the shim header to modify or maintain the traffic class/color combination of traffic
- Mapping of traffic behavior aggregates to corresponding per-hop behaviors so that traffic can be differentially switched to the appropriate LSPs to meet your network objectives

To configure MPLS for differentiated services:

- Configure MPLS to use the pipe or uniform model of tunneling for differentiated services.

  See "Configuring the Tunneling Model for Differentiated Services" on page 309.
- Configure EXP bits for differentiated services.

See "Configuring EXP Bits for Differentiated Services" on page 309.

- Configure differentiated services in a sample topology.

  See "Example Differentiated Services Application and Configuration" on page 310.

- Classify traffic In a differentiated services domain.

  See "Classifying Traffic for Differentiated Services" on page 313.

Configuring the Tunneling Model for Differentiated Services

The JunosE Software supports both the pipe model and the uniform model for tunneling with the **mpls tunnel-model** command. The router also provides a way to implement the functionality of the short pipe model for IP packets.

To specify whether MPLS uses the pipe or uniform model of tunneling for differentiated services:

- Issue the **mpls tunnel-model** command.

  host1(config)#**mpls tunnel-model uniform**

Related Documentation

- Configuring MPLS and Differentiated Services on page 308
- Example Differentiated Services Application and Configuration on page 310

- *mpls tunnel-model*

Configuring EXP Bits for Differentiated Services

To set the initial value of the EXP bits to the UPC value associated with the packets:

- Issue the **mpls copy-upc-to-exp** command.

  host1(config)#**mpls copy-upc-to-exp**

To prevent the value of the EXP bits for a VPN/VC label from being modified by a per-LSP policy applied for the outer labels or by per-VR traffic class/color rules:

- Issue the **mpls preserve-vpn-exp** command.

  host1(config)#**mpls preserve-vpn-exp**

Related Documentation

- EXP Bits for Differentiated Services Overview on page 272
- Configuring MPLS and Differentiated Services on page 308
- Example Differentiated Services Application and Configuration on page 310

- *mpls copy-upc-to-exp*
- *mpls preserve-vpn-exp*

Example Differentiated Services Application and Configuration

Figure 64 on page 311 shows an example topology where a service provider offers the following differentiated services to its customers over its MPLS network:

- QoS Internet service—The CE router is managed by the provider and sets the IP precedence to predefined values. IP policy on the PE router sets the traffic-class/color combination according to the incoming well-defined IP precedence value. The policy also sets the UPC value to the incoming well-defined IP precedence value.
- Plain Internet service—IP policy on the PE router leaves the traffic-class/color combination as the default value, best-effort/green. The policy sets the UPC to 0.
- QoS VPN service—For CE-to-PE traffic, the VPN EXP is copied from the IP precedence value when the PE router pushes VPN stacked labels.

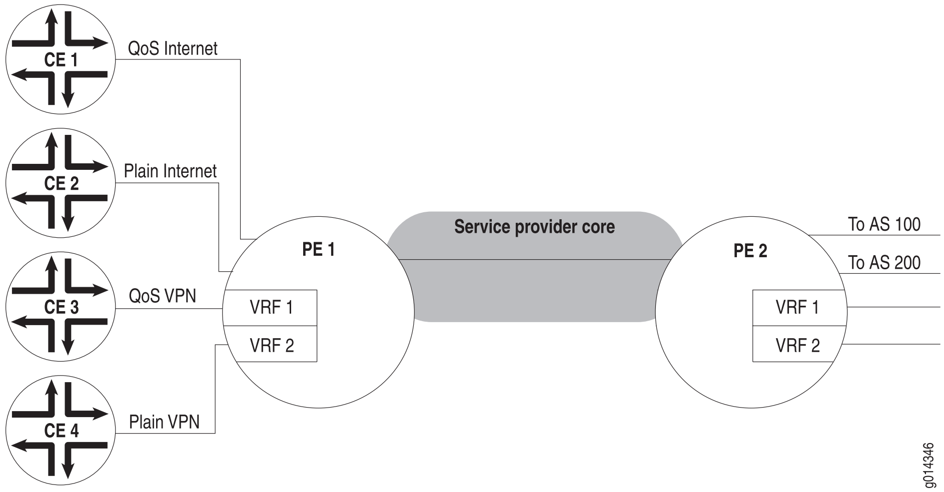
  For PE-to-CE traffic, IP policy on the PE router resets the traffic-class/color combination according to the received, well-defined IP precedence value, so that egress queuing is based on the IP precedence value. This action takes place on the egress line module.
- Plain VPN service—For CE-to-PE traffic, the VPN EXP bits are set to 000 when the PE router pushes VPN stacked labels.

  For PE-to-CE traffic, IP policy on the PE router resets the traffic-class/color combination to the default value, best-effort/green, so that packets are queued as best-effort. The IP precedence value is left unchanged.

In this example, the provider also offers an inter-AS VPN service. The provider's own protocol traffic, for example, BGP signaling traffic such as update messages, is also labeled, with the EXP bits set to the same value as the IP precedence.

The egress queuing of traffic as it leaves the provider is always based on either the VPN EXP bits as received on the core side in inter-AS case, or the IP precedence value in all other cases. It is acceptable that fabric queuing is based on the incoming base label's EXP.

Figure 64: Differentiated Services over an MPLS Network



Differentiated Services Configuration Example

To configure the differentiated services described in this example:

1. Create and attach an IP input policy for the QoS Internet service to CE interfaces on the PE router for incoming traffic.

```
host1(config)#ip classifier-list prec0 ip any any precedence 0
host1(config)#ip classifier-list prec1 ip any any precedence 1
host1(config)#ip policy-list qos-service
host1(config-policy-list)#classifier-group prec0
host1(config-policy-list-classifier-group)#user-packet-class 0
host1(config-policy-list-classifier-group)#traffic-class class0
host1(config-policy-list-classifier-group)#color green
host1(config-policy-list)#classifier-group prec1
host1(config-policy-list-classifier-group)#user-packet-class 1
host1(config-policy-list-classifier-group)#traffic-class class1
host1(config-policy-list-classifier-group)#color green
host1(config)#interface atm 3/0.1
host1(config-subif)#ip policy input qos-service
```

2. Create and attach an IP input policy for the plain Internet service to CE interfaces on the PE router for incoming traffic. All traffic is treated as best effort, so no classifier group is necessary.

```
host1(config)#ip policy-list plain-service
host1(config-policy-list-classifier-group)#user-packet-class 0
host1(config-policy-list-classifier-group)#traffic-class best-effort
host1(config-policy-list-classifier-group)#color green


host1(config)#interface atm 5/0.1
host1(config-subif)#ip policy input plain-service
```

3. Attach an IP output policy for the QoS VPN service to CE interfaces on the PE router for outgoing traffic. The same qos-service policy that is attached to the input in Step 1 can be used on the output, even though the UPC setting is not needed.

```
host1(config)#Interface atm 3/0.1
host1(config-subif)#Ip policy output qos-service
```

Attach an IP output policy for the plain VPN service to CE interfaces on the PE router for outgoing traffic. The same plain-service policy that is attached to the input in Step 2 can be used on output, although the UPC setting is not needed.

```
host1(config)#Interface atm 5/0.1
host1(config-subif)#Ip policy output plain-service
```

4. For traffic toward the core, configure per-VR rules or per-LSP policies to set the base EXP bits value according to the traffic-class/color combination. Issue the **mpls copy-upc-to-exp** command to set the VPN EXP bits value to the UPC value. The UPC value is the same as the IP precedence value for the QoS service case; for all other cases the value is 000. Configure the **mpls preserve-vpn-exp** command so that VPN EXP bits are not subject to policy or to per-VR EXP rules.

```
host1(config)#mpls match traffic-class ... color ... set exp-bits ...
host1(config)#mpls copy-upc-to-exp
host1(config)#mpls preserve-vpn-exp
```

You must attach a policy to the core-side IP interface to set the UPC value of the control traffic appropriately so that the EXP bits value is copied from the UPC when this traffic goes out as MPLS packets.

```
host1(config)#ip classier-list control-traffic-prec0 ...
host1(config)#ip classier-list control-traffic-prec1 ...
host1(config)#ip policy-list core-ip-policy
host1(config-policy-list)#classifier-group control-traffic-prec0
host1(config-policy-list-classifier-group)#user-packet-class prec0
host1(config-policy-list-classifier-group)#traffic-class class0
host1(config-policy-list-classifier-group)#color green
host1(config-policy-list)#classifier-group control-traffic-prec1
host1(config-policy-list-classifier-group)#-packet-class prec1
host1(config-policy-list-classifier-group)#traffic-class class1
host1(config-policy-list-classifier-group)#color green

host1(config)#interface pos 0/0
host1(config-subif)ip policy output core-ip-policy
```

5. For traffic from the core, configure per-VR rules or per-LSP policies to set the traffic-class/color combination—and therefore shape the egress traffic queue—according to the value of the EXP bits in the base label. This action causes

```
host1(config)#mpls match exp-bits <value> set traffic-class <className> color
```

Related Documentation

- Configuring MPLS and Differentiated Services on page 308
- Configuring EXP Bits for Differentiated Services on page 309
- Classifying Traffic for Differentiated Services on page 313

Classifying Traffic for Differentiated Services

In a differentiated services domain, traffic is classified into a behavior aggregate (BA), based on the type of diff-serv behavior for the traffic. At each node, traffic belonging to a particular BA is mapped to the corresponding per-hop behavior (PHB), which provides the scheduling behavior and drop probability required by the traffic.

MPLS uses the EXP bits in the shim header to support differentiated services. The JunosE Software supports both statically configured and signaled mapping between the EXP bits and the PHB of traffic.

In a signaled environment, you can configure on the ingress node the set of PHBs that a tunnel supports, and then the set of PHBs is signaled end to end.

To support differentiated services, MPLS employs two types of LSPs: E-LSPs and L-LSPs. The two types differ in how their PHB is determined. In the JunosE Software, the PHB is a combination of traffic class (also called per-hop scheduling class, or PSC) and drop precedence (color).

- E-LSPs (EXP-inferred-PSC LSP) can transport as many as eight BAs. For E-LSPs, the traffic's PHB is learned from the MPLS shim header.
- L-LSPs (Label-only-inferred-PSC LSP) transport a single PSC. The PHB is determined from a combination of the packet's label, which indicates the traffic class, and the EXP field of the shim header, which indicates the drop precedence.
    - Table 56 on page 313 indicates how the PSC (column 1) is combined with the EXP field (column 2) to determine the PHB for incoming traffic on L-LSPs.

Table 56: Incoming L-LSP PHB Determination

PSC	+	EXP Field	=	PHB
BE		000		BE
CSn		000		CSn
AFn		001		AFn1
AFn		010		AFn2
AFn		011		AFn3
EF		000		EF

For nonstandard PHBs (any that are not listed in Table 56 on page 313), the JunosE Software uses mapping similar to AFn mapping; EXP 001 is mapped to color green, EXP 010 is mapped to yellow, and EXP 011 is mapped to red.

Table 57 on page 314 presents three examples that indicate how the PSC and the EXP field are combined to determine the PHB for traffic on incoming L-LSPs.

Table 57: Examples of Incoming L-LSP PHB Determination

PSC	+	EXP Field	=	PHB
AF2		010		AF22
AF3		010		AF32
AF3		011		AF33

- For outgoing L-LSPs, the EXP is determined by the PHB. Table 58 on page 314 indicates the PHB-to-EXP mapping for outgoing traffic on L-LSPs.

Table 58: Outgoing L-LSP PHB Determination

PHB	=	EXP Field
BE		000
CSn		000
AFn1		001
AFn2		010
AFn3		011
EF		000

For nonstandard PHBs, the mapping is similar to AFn mapping. Red color maps to 011, yellow maps to 010, and green maps to 001.

Tasks to perform static configuration and signaled mapping between the EXP bits and the PHB of traffic include the following sets of tasks:

- Configuring Static EXP-to-PHB Mapping on page 314
- Signaled Mapping for RSVP-TE Tunnels on page 315
- Preference of per-VR Versus per-LSP Behavior on page 317

Configuring Static EXP-to-PHB Mapping

You can configure static EXP-to-PHB mapping at the per-VR level, only for LSPs that do not have specific policies attached (by either per-LSP configured mapping or signaled mapping). The configured mapping applies regardless of label distribution protocol, BGP. LDP, or RSVP-TE.

The PHB of incoming packets is determined from the EXP bits by the match values set with the **mpls match exp-bits** command.

The EXP bits of outgoing packets are determined from the PHB by the match values set with the mpls match traffic-class command.

To configure static EXP-to-PHB mapping:

1. Set a combination of traffic class and color for incoming traffic that matches the specified EXP bits value in the shim header.

   host1(config)#**mpls match exp-bits 1 set traffic-class bronze color red**

   You can repeat the command to support the eight possible EXP bit values.

2. Set the EXP bits in the shim header of outgoing traffic that matches a particular combination of traffic class and color.

   host1(config)#**mpls match traffic-class gold color green set exp-bits 7**

   You can repeat the command to support up to 24 combinations: eight traffic classes supported on the router times three colors.

Signaled Mapping for RSVP-TE Tunnels

For signaled mapping between EXP and PHB, policies apply the EXP bits matching and setting on a per-LSP basis rather than a per-VR basis. Signaled mapping applies only when RSVP-TE is the label distribution protocol.

When traffic is mapped onto the ingress router of the LSP, the EXP bits are set according to a policy attached to the LSP. The policy corresponds to the EXP-to-PHB mapping defined for the LSP. Typically, the policy sets the EXP bits differently according to classifier lists that match on internal class/color information or on a user packet class associated with a packet.

For transit routers and egress routers along the path of the LSP, the incoming EXP bits are matched to determine the traffic class and drop preference (color red, yellow, or green). This matching is accomplished by means of a policy corresponding to the signaled EXP-to-PHB mapping that is created and attached when the LSP is established.

EXP bits are not normally changed on transit routers, but when traffic is sent out of an LSP on a transit router, the bits can be changed by the policy. Normally, however, the net effect is that the EXP-bits remain the same through the mapping sequence of EXP bits to an internal traffic class/color combination back to EXP bits, unless the traffic class/color combination is also modified by other factors.

Because the policy (which maps the EXP bits to an internal traffic class/color combination and vice versa) attached to an LSP is created according to the PHB-ID–to–EXP mapping signaled by RSVP-TE, you must configure on each router a mapping association between PHB IDs and the internal traffic class/color combinations.

The JunosE Software automatically generates and attaches policies when tunnels are established.

Figure 65 on page 316 shows the mapping associations between PHB IDs, EXP bits, and traffic class (TC)/color combination in an E-LSP case.

- Mapping association between PHB ID and EXP bits is configured on ingress routers using the **tunnel mpls diff-serv phb-id** command.

- Mapping association between PHB ID and traffic class/color combination is configured on all routers using the **mpls diff-serv phb-id traffic-class** command.
- Mapping association between EXP bits and traffic class/color combination is done automatically by the JunosE Software at the appropriate routers along the path.

Figure 65: Associations Between PHB ID, EXP Bits, and Traffic Classes/Colors

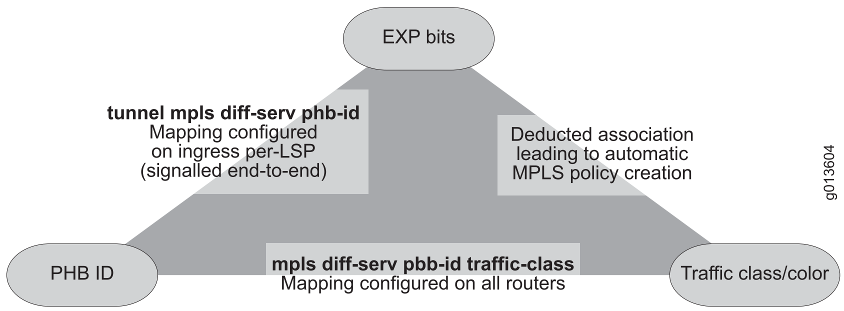
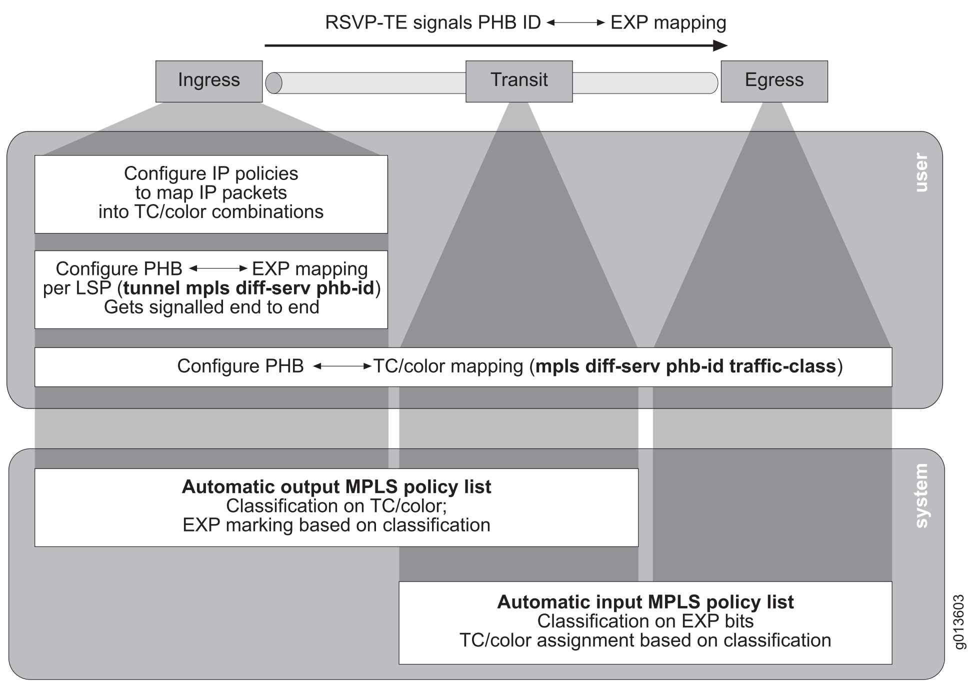


Figure 66 on page 316 shows the operations performed at ingress, transit, and egress systems during signaled mapping sessions.

Figure 66: Signaled Mapping



To define a policy rule that sets the EXP bits in packets to which the policy is applied:

- Issue the **mark-exp** command.

  host1(config-policy-list)#mark-exp 5 classifier-group claclEXP precedence 32

To create or modify an MPLS classifier control list to match on traffic class/color combination or EXP bits:

- Issue the **mpls classifier-list**command.

  host1(config)#mpls classifier-list be-green traffic-class best-effort color yellow

To map the specified PHB ID to the internal traffic class/color combination:

- Issue the **mpls diff-serv phb-id traffic-class** command.

  host1(config)#mpls diff-serv phb-id standard 45 traffic-class gold color green

To create or modify an MPLS policy:

- Issue the **mpls policy-list** command.

  host1(config)#mpls policy-list mpls-exp-setting

To enable collection of policy statistics for a tunnel or LSP. Collection is disabled by default.

- Issue the **mpls policy-statistics** command.

  host1#mpls policy-statistics boston2dc

Policy statistics are displayed when you issue the **show mpls forwarding** or **show mpls tunnel** command, if a policy is attached and policy statistics are enabled.

To specify the traffic class for which LSP-level queues are created and the scheduler profile to be used with the queues:

- Issue the **mpls traffic-class** command.

  host1(config)#mpls traffic-class af1 scheduler-profile af1-scheduler-profile

These classes originate from E-LSPs and L-LSPs (classes derived from the signaled PHB-ID) or regular LSPs (classes configured with the **mpls traffic-class** command)

To specify the PHB supported by a signaled tunnel:

- Issue the **tunnel mpls diff-serv phb-id** command.

  host1(config-if)#tunnel mpls diff-serv phb-id standard 35 exp-bits 5

For E-LSPs, you also use this command to map the PHB to the specified exp-bits *bitValue*. You can repeat the command for up to eight PHB mappings.

For L-LSPs, do not use the **exp-bits** keyword. If you repeat the command, the most recent command overwrites the previous command.

Preference of per-VR Versus per-LSP Behavior

MPLS always prefers the per-LSP method of matching and setting EXP bits by means of applied policies over the per-VR method.

Per-VR matching of EXP bits is not performed on the LSP when an input policy (matching on incoming EXP bits) is attached to the ingress segment of the LSP.

Similarly, per-VR setting of EXP bits is not performed on the LSP when an output policy (setting the outgoing EXP bits) is attached to the egress segment of the LSP.

See the *JunosE Policy Management Configuration Guide* for more information about defining policies.

Example Traffic Class Configuration for Differentiated Services

The commands in this example illustrate a partial network configuration that supports four differentiated service classes on a particular tunnel: a best-effort class, two assured forwarding classes, and an expedited forwarding class. Table 59 on page 318 presents the mapping between EXP bits, PHB, PHB ID, and traffic class/color combination.

Table 59: Differentiated Services Mapping

EXP	PHB	PHB ID	6-bit PHB ID	Traffic Class/Color
000	BE	0x0000	00	best-effort/green
001	AF11	0x2800	10	af1/green
010	AF12	0x3000	12	af1/yellow
011	AF13	0x3800	14	af1/red
100	AF21	0x4800	18	af2/green
101	AF22	0x5000	20	af2/yellow
110	AF23	0x5800	22	af2/red
111	EF	0xb800	46	ef/green

NOTE: This example includes both MPLS and policy configuration commands, and assumes that you are thoroughly familiar with the information and commands presented in the *JunosE Policy Management Configuration Guide*.

The four traffic classes are configured to allocate fabric resources and allow global synchronization of the three segments of the data path through an E Series router: ingress, fabric, and egress. The JunosE Software automatically creates the best-effort traffic class, with a default weight of eight. You must define the remaining three classes, af1, af2, and ef. In this example, the af1 class has twice as much fabric bandwidth as the best-effort class, and the af2 class has twice as much fabric bandwidth as the af1 class. The expedited forwarding traffic (the ef class) requires strict-priority queuing.

```
host1(config)#traffic-class af1
host1(config-traffic-class)#fabric-weight 16
host1(config)#traffic-class af2
```

```
host1(config-traffic-class)#fabric-weight 32
host1(config)#traffic-class ef
host1(config-traffic-class)#fabric-strict-priority
```

Define two scheduler profiles for the af1 and af2 classes on the egress line modules:

```
host1(config)#scheduler-profile af1-scheduler-profile
host1(config-scheduler-profile)#weight 16
host1(config)#scheduler-profile af2-scheduler-profile
host1(config-scheduler-profile)#weight 32
```

Create queue profiles to define how queues are instantiated to implement the corresponding traffic classes and PHBs. The JunosE Software automatically creates the best-effort queue profiles.

```
host1(config)#queue-profile af1-queues
[Queue configuration omitted]
host1(config)#queue-profile af2-queues
[Queue configuration omitted]
host1(config)#queue-profile ef-queues
[Queue configuration omitted]
```

The scheduler and queue profiles are referenced in QoS profiles. For example, you can create a QoS profile for port-based per-class queuing or for LSP-level per-class queuing (configuration omitted).

You must map the PHB IDs to the appropriate traffic class/color combinations:

```
host1(config)#mpls diff-serv phb-id standard 0 traffic-class best-effort color green
host1(config)#mpls diff-serv phb-id standard 10 traffic-class af1 color green
host1(config)#mpls diff-serv phb-id standard 12 traffic-class af1 color yellow
host1(config)#mpls diff-serv phb-id standard 14 traffic-class af1 color red
host1(config)#mpls diff-serv phb-id standard 18 traffic-class af2 color green
host1(config)#mpls diff-serv phb-id standard 20 traffic-class af2 color yellow
host1(config)#mpls diff-serv phb-id standard 22 traffic-class af2 color red
host1(config)#mpls diff-serv phb-id standard 46 traffic-class ef color green
```

Configuration on the Ingress Router

You must access the tunnel interface to map the PHB IDs to the EXP bits. The E Series router signals this mapping to all routers on the tunnel. You can establish different PHB-ID–to–EXP mappings for different tunnels.

```
host1(config)#interface tunnel mpls:example
```

PHB-ID–to–EXP mapping for the best-effort traffic class:

```
host1(config-if)#tunnel mpls diff-serv phb-id standard 0x0000 exp-bits 0
```

PHB-ID–to–EXP mapping for the af1 traffic class:

```
host1(config-if)#tunnel mpls diff-serv phb-id standard 10 exp-bits 1
host1(config-if)#tunnel mpls diff-serv phb-id standard 12 exp-bits 2
host1(config-if)#tunnel mpls diff-serv phb-id standard 14 exp-bits 3
```

PHB-ID–to–EXP mapping for the af2 traffic class:

```
host1(config-if)#tunnel mpls diff-serv phb-id standard 18 exp-bits 4
```

```
host1(config-if)#tunnel mpls diff-serv phb-id standard 20 exp-bits 5
host1(config-if)#tunnel mpls diff-serv phb-id standard 22 exp-bits 6
```

PHB-ID–to–EXP mapping for the ef traffic class:

```
host1(config-if)#tunnel mpls diff-serv phb-id standard 46 exp-bits 7
```

Define classifier control lists to classify the incoming packets into classifier groups. Although not shown here, for each CLACL you must define the rules that will select the appropriate incoming packets: be, af1, af2, or ef.

```
host1(config)#classifier-list be-packets
host1(config)#classifier-list af1-packets
host1(config)#classifier-list af2-packets
host1(config)#classifier-list ef-packets
```

Define a policy that maps the selected packets into traffic classes. For the assured forwarding classes, this example uses rate limit profiles to set the colors.

```
host1(config)#policy-list classify-packets
host1(config-policy-list)#traffic-class best-effort classifier-group bf-packets
host1(config-policy-list)#traffic-class ef classifier-group ef-packets
host1(config-policy-list)#traffic-class af1 classifier-group af1-packets
host1(config-policy-list)#traffic-class af2 classifier-group af2-packets
host1(config-policy-list)#rate-limit-profile af1-profile classifier-group af1-packets
host1(config-policy-list)#rate-limit-profile af2-profile classifier-group af2-packets
host1(config)#rate-limit-profile af1-profile
host1(config-rate-limit-profile)#committed-rate 6000000
host1(config-rate-limit-profile)#committed-burst 1000000
host1(config-rate-limit-profile)#peak-rate 8000000
host1(config-rate-limit-profile)#peak-burst 1000000
host1(config)#rate-limit-profile af2-profile
host1(config-rate-limit-profile)#committed-rate 8000000
host1(config-rate-limit-profile)#committed-burst 1500000
host1(config-rate-limit-profile)#peak-rate 12000000
host1(config-rate-limit-profile)#peak-burst 1000000
```

You attach the policy to the ingress interface of the ingress router. As packets arrive, they are classified with the internal traffic class/color combination and forwarded into the appropriate queues in the fabric. When the packets are sent into the tunnel out of the ingress router, the EXP bits are set according to the router-generated policy (in this example called mpls-exp-setting) that the JunosE Software automatically attached to the tunnel.

Configuration on the Ingress and Transit Routers

When the tunnel is established, the JunosE Software automatically creates an output policy to map traffic-class/color combinations to EXP bits and attaches the policy to the outgoing segment of the tunnel. The JunosE Software generates classifier list and policy list names, and creates the EXP-setting policy as if the following commands were entered:

NOTE: You do not actually issue these commands; they represent the behavior automatically performed by the router.


```
host1(config)#mpls classifier-list be-green traffic-class best-effort color green
host1(config)#mpls classifier-list ef-green traffic-class ef color green
host1(config)#mpls classifier-list af1-green traffic-class af1 color green
host1(config)#mpls classifier-list af1-yellow traffic-class af1 color yellow
host1(config)#mpls classifier-list af1-red traffic-class af1 color red
host1(config)#mpls classifier-list af2-green traffic-class af2 color green
host1(config)#mpls classifier-list af2-yellow traffic-class af2 color yellow
host1(config)#mpls classifier-list af2-red traffic-class af2 color red
host1(config)#mpls policy-list mpls-exp-setting
host1(config-policy-list)#mark 0 classifier-group be-green
host1(config-policy-list)#mark 1 classifier-group af1-green
host1(config-policy-list)#mark 2 classifier-group af1-yellow
host1(config-policy-list)#mark 3 classifier-group af1-red
host1(config-policy-list)#mark 4 classifier-group af2-green
host1(config-policy-list)#mark 5 classifier-group af2-yellow
host1(config-policy-list)#mark 6 classifier-group af2-red
host1(config-policy-list)#mark 7 classifier-group ef-green
```

NOTE: For a topology-driven LSP, you have to configure and apply the classifier list and policy list manually.

Configuration on the Transit and Egress Routers

When the tunnel is established, the JunosE Software automatically creates an input policy to match the EXP bits and map them to the traffic-class/color combinations and attaches the policy to the incoming segment of the tunnel. The JunosE Software generates classifier list and policy list names, and creates the policy as if the following commands were entered:

NOTE: You do not actually issue these commands; they represent the behavior automatically performed by the router.

```
host1(config)#mpls classifier-list bf-packets exp 0
host1(config)#mpls classifier-list af11-packets exp 1
host1(config)#mpls classifier-list af12-packets exp 2
host1(config)#mpls classifier-list af13-packets exp 3
host1(config)#mpls classifier-list af21-packets exp 4
host1(config)#mpls classifier-list af22-packets exp 5
host1(config)#mpls classifier-list af22-packets exp 6
host1(config)#mpls classifier-list ef-packets exp 7
host1(config)#mpls policy-list mpls-exp-matching
host1(config-policy-list)#traffic-class best-effort classifier-group bf-packets
host1(config-policy-list)#traffic-class af1 classifier-group af11-packets
host1(config-policy-list)#traffic-class af1 classifier-group af12-packets
host1(config-policy-list)#traffic-class af1 classifier-group af13-packets
host1(config-policy-list)#traffic-class af2 classifier-group af21-packets
host1(config-policy-list)#traffic-class af2 classifier-group af22-packets
host1(config-policy-list)#traffic-class af2 classifier-group af23-packets
host1(config-policy-list)#traffic-class ef classifier-group ef-packets
host1(config-policy-list)#color green classifier-group af11-packets
host1(config-policy-list)#color green classifier-group af21-packets
```

```
host1(config-policy-list)#color yellow classifier-group af12-packets
host1(config-policy-list)#color yellow classifier-group af22-packets
host1(config-policy-list)#color red classifier-group af13-packets
host1(config-policy-list)#color red classifier-group af23-packets
```

NOTE: For a topology-driven LSP, you must configure and apply the classifier list and policy list manually.

The packets are forwarded to the appropriate fabric queue according to the traffic class/color combination. On a transit router, when the packet is forwarded out of the tunnel, the router-generated output policy then sets the EXP bits back according to the traffic class/color combination. Typically, the effect of the EXP bits to traffic class/color combination to EXP bits is no change.

On an egress router, where the tunnel terminates, no router-generated output policy is attached, and the packets pass out of the router subject to any manually configured IP policy management applied to their traffic class/color combination.

See the *JunosE Policy Management Configuration Guide* for more information about defining policies.

Related Documentation

- Configuring MPLS and Differentiated Services on page 308
- Configuring EXP Bits for Differentiated Services on page 309
- Example Differentiated Services Application and Configuration on page 310
- Classifying Traffic for Differentiated Services on page 313

- *mark-exp*
- *mpls classifier-list*
- *mpls diff-serv phb-id traffic-class*
- *mpls match exp-bits*
- *mpls match traffic-class*
- *mpls policy-list*
- *mpls policy-statistics*
- *mpls traffic-class*
- *tunnel mpls diff-serv phb-id*

Configuring Point-to-Multipoint LSPs

To set up a point-to-multipoint LSP, you configure the primary LSP from the ingress router and the branch LSPs that carry traffic to the egress routers. The configuration of the primary point-to-multipoint LSP is similar to a signaled LSP. In addition to the conventional LSP configuration, you specify a path name on the primary LSP and this same path name on each branch LSP. By default, the branch LSPs are dynamically signaled by means of CSPF and require no configuration. You can alternatively configure the branch LSPs as a static path.

Because E Series routers can only function as egress routers, you must use M-series routers running Junos OS as ingress routers to enable point-to-multipoint LSPs to send packets through the network to the endpoints connected to the egress routers.

Observe the following guidelines to deliver multicast data using point-to-multipoint LSPs on the egress E Series routers:

- The IP interface on which the packet arrives must be an IGMP-owned interface. An IGMP-owned interface refers to an interface in which IGMP is the only multicast protocol enabled.
- The actual route to the source must be through an IGMP-owned interface.

The configuration of an E Series router as an egress router depends on the type of label advertised for the LSR that is the egress router for the prefix. Penultimate hop popping (PHP) is the default when RSVP-TE or LDP is the signaling protocol.

If the egress router advertises an implicit null label to achieve PHP on an upstream neighbor, enable IP IGMP on the static physical interface on which the unlabeled multicast packet is received by the router by using the **ip igmp** interface configuration command.

```
host1(config-if)#ip igmp
```

If the egress router advertises an explicit null or non-null label to its upstream neighbor to include a label in all MPLS packets, complete the following steps to configure an E Series router as an egress router:

1. Create a profile by using the **profile** command. Add commands to enable IP IGMP and IP processing on the loopback interface in the profile.

   ```
   host1(config)#profile mplsdynip
   host1(config-profile)#ip igmp
   host1(config-profile)#ip unnumbered loopback 0
   ```

2. Create a dynamic IPv4 interface on top of MPLS major interfaces and specify the profile that you created in the previous step to set attributes for this interface.

   ```
   host1(config)#mpls create-dynamic-interfaces ip on-major-interfaces profile dynip
   ```

For all types of labels that are advertised for the LSR that is the egress router for the prefix, you must complete the following steps, in addition to those detailed earlier, on the E Series router:

1. Enable IGMP on the interface which owns the route to the source by using the **ip igmp** command. Because the route to the source can change dynamically, we recommend that you enable IGMP on all interfaces of the router or, at least, on all interfaces that might be the next hop interface to the source.

2. Disable the multicast reverse path forwarding (RPF) check policy for all the streams that will be delivered on the point-to-multipoint LSP by using the **ip multicast-routing disable-rpf-check** command. For more information, see *Enabling and Disabling RPF Checks* in the *JunosE Multicast Routing Configuration Guide*.

Related Documentation

- Point-to-Multipoint LSPs Overview on page 275
- *show mpls tunnels*

CHAPTER 5

Monitoring MPLS

This chapter describes the commands you can use to monitor and troubleshoot Multiprotocol Label Switching (MPLS) on E Series routers.

NOTE: The E120 and E320 Broadband Services Routers output for **monitor** and **show** commands is identical to output from other E Series routers, except that the E120 and E320 router output also includes information about the adapter identifier in the interface specifier (*slot/adapter/port*).

This chapter contains the following sections:

Setting the Baseline for MPLS Statistics

You can use the **baseline mpls** commands to set a statistics baseline for MPLS operations. The router implements the baseline by setting the statistics to zero and then subtracting this baseline when you retrieve baseline-relative statistics.

Use the **delta** keyword with the **show mpls** commands to display baselined statistics.

Tasks to set a baseline for MPLS statistics are:

Setting a Baseline for MPLS Major Interface Statistics

To set a statistics baseline for MPLS major interfaces:

- Issue the **baseline mpls interface** command for a specific MPLS major interface.

  host1#**baseline mpls interface boston5**

There is no **no** version.

The following statistics are maintained for each MPLS major interface:

• receive packets and octets	• transmit packets and octets
• receive discarded packets	• transmit discarded packets
• receive error packets	• transmit error packets
• failed label lookups	

Enabling and Setting a Baseline for MPLS Forwarding Table Statistics

To enable and set a statistics baseline for MPLS forwarding table entries:

1. Issue the **mpls statistics label** command to enable the statistics for a specific MPLS in label.

   host1#**mpls statistics label 123**

2. Issue the **baseline mpls label** command for a specific MPLS in label.

   host1#**baseline mpls label 123**

By default, statistics are enabled for incoming labels and RSVP-TE or LDP outgoing labels, but not for others such as BGP outgoing labels. Statistics are not stored in NVS.

When enabled, the following statistics are maintained for each forwarding table entry:

• receive packets and octets	• receive discarded packets	• receive error packets

There is no **no** version for the **baseline mpls label** command. However, you can disable the forwarding table statistics.

To disable the statistics for a specific MPLS in label:

- Issue the **no mpls statistics label** command.

  host1#**no mpls statistics label 123**

Enabling and Setting a Baseline for MPLS Next-Hop Table Statistics

To enable and set a statistics baseline for MPLS next-hop table entries:

1. Issue the **mpls statistics next-hop** command to enable the statistics for a specific MPLS next hop.

   host1#**mpls statistics next-hop 1046**

2. Issue the **baseline mpls next-hop** command for a specific MPLS next hop.

host1#**baseline mpls next-hop 1046**

By default, statistics are enabled for next hops depending on the protocol that created the MPLS next hop. Statistics are not stored in NVS.

When enabled, the following statistics are maintained for each next-hop table entry:

- out packets and bytes
- out discarded packets
- out error packets

There is no **no** version for the **baseline mpls next-hop** command. However, you can disable the next-hop table statistics.

To disable the statistics for a specific MPLS next hop:

- Issue the **no mpls statistics next-hop** command.

  host1#**no mpls statistics next-hop 1046**

Setting a Baseline for MPLS Tunnel Statistics

To set a statistics baseline for MPLS tunnel statistics:

- Issue the **baseline mpls tunnel** command.

  host1#**baseline mpls tunnel tunnel5**

There is no **no** version.

Enabling Statistics Collection for Policies Attached to MPLS Tunnels

To enable collection of the following statistics for each policy attached to a tunnel:

- Issue the **mpls statistics policy** command.

  host1#**mpls statistics policy tunnel5**

Statistics are not stored in NVS. When enabled, the following statistics are maintained for each policy:

- packets and bytes
- classifier group
- EXP bits value

To disable collection of policy statistics for a specific MPLS tunnel:

- Issue the **no mpls statistics policy** command.

  host1#**no mpls statistics policy tunnel5**

Clearing and Re-Creating Dynamic Interfaces from MPLS Major Interfaces

To remove and re-create dynamic IPv4 interfaces and dynamic IPv6 interfaces from all MPLS major interfaces or a specific MPLS major interface:

- Issue the **clear mpls dynamic-interfaces on-major-interfaces** command:

  host1#**clear mpls dynamic-interfaces on-major-interfaces**

You can use this command to reapply policies related to dynamic IPv4 and IPv6 interfaces on top of MPLS major interfaces.

There is no **no** version.

Related Documentation

- *clear mpls dynamic-interfaces on-major-interfaces*

Clearing and Refreshing IPv4 Dynamic Routes in the Tunnel Routing Table

To clear and then refresh a specified IPv4 dynamic route or all IPv4 dynamic routes from the tunnel routing table of the virtual router or a specified VRF:

- Issue the **clear ip tunnel-routes** command.

  host1(config)#**clear ip tunnel-routes ***

There is no **no** version. This command takes effect immediately.

Related Documentation

- *clear ip tunnel-routes*

Clearing and Refreshing IPv6 Dynamic Routes in the Tunnel Routing Table

To clear and then refresh a specified IPv6 dynamic route or all IPv6 dynamic routes from the tunnel routing table of the virtual router or a specified VRF.

- Issue the **clear ipv6 tunnel-routes** command.

  host1(config)#**clear ipv6 tunnel-routes ***

There is no **no** version. This command takes effect immediately.

Related Documentation

- *clear ipv6 tunnel-routes*

Tracing Paths Through the MPLS User Plane

Purpose To trace paths through the MPLS user plane.

Action To trace the path that packets follow enroute to the destination IP address 10.90.101.9:

```
host1:edge1#traceroute 10.90.101.9
Tracing route to 10.90.101.9, TTL = 32, timeout = 2 sec.
(Press ^c to stop.)
1  3ms  2ms  2ms      10.90.101.4    mplsLabel1=4009
mplsExpBits1=0
2  2ms  2ms  2ms      10.90.101.7    mplsLabel1=7004
mplsExpBits1=0
3  2ms  2ms  2ms      10.90.101.9
```

ICMP extensions enable LSRs to append MPLS header information (the label stack) to ICMP destination unreachable and time exceeded messages. This sample output shows the label and EXP bits used to switch the ICMP packets.

Related Documentation

- *Determining Reachability of IP Destinations in the Network*
- *traceroute*

Monitoring ATM VCs and VPI/VCI Ranges Used for MPLS

Purpose

Display information about ATM VCs used as MPLS LSPs and VPI-VCI ranges reserved for MPLS when you use the interface label space for MPLS labels.

Action

To display all VCs and reserved VC ranges on the router:

```
host1#show atm vc
                                                Cate Rx/Tx Rx/Tx Rx/Tx Sta
Interface     VPI VCI  VCD  Type Encap gory Peak   Avg Burst tus
------------ --- ---- ---- ---- ----- ---- ----- ----- ----- ---
ATM 3/0.2      0 101  4375 PVC  AUTO  CBR   1000     0     0 UP
ATM 3/0.3      0 102  4376 PVC  AUTO  CBR   1000     0     0 DOWN
...
ATM 3/0.8099   1 8099 8099 PVC  SNAP  UBR      0     0     0 UP
ATM 3/0.8100   1 8100 8100 PVC  SNAP  UBR      0     0     0 DOWN
8000 circuit(s) found

Reserved VCC ranges:

          Start Start End End
Interface  VPI   VCI  VPI VCI
--------- ----- ----- --- ---
ATM 2/0       2   100   2 102
ATM 2/0       3   300   3 303
2 reservation(s) found
```

To display a summary of all reserved VC ranges on the router, including those reserved for MPLS use:

```
host1#show atm vc reserved
Reserved VCC ranges:

          Start Start End End
Interface  VPI   VCI  VPI VCI
--------- ----- ----- --- ---
ATM 2/0       2   100   2 102
ATM 2/0       3   300   3 303
2 reservation(s) found
```

Meaning

Table 60 on page 330 lists the **show atm vc** command output fields.

Table 60: show atm vc Output Fields

Field Name	Field Description
Interface	Interface type and number
VPI	Virtual path identifier
VCI	Virtual channel identifier
VCD	Virtual circuit descriptor

Table 60: show atm vc Output Fields *(continued)*

Field Name	Field Description
Type	Type of circuit: PVC
Encap	Encapsulation method: AUTO, AAL5, MUX, SNAP, ILMI, F4-OAM
Category	Service type configured on the VC: UBR, UBR-PCR, NRT-VBR, RT-VBR, CBR
Rx/Tx Peak	Peak rate in Kbps
Rx/Tx Avg	Average rate in Kbps
Rx/Tx Burst	Maximum number of cells that can be burst at the peak cell rate
Status	State of the virtual circuit: Up, Down
Start VPI	Starting virtual path identifier (inclusive) of the reserved VC range
Start VCI	Starting virtual circuit identifier (inclusive) of the reserved VC range
End VPI	Ending virtual path identifier (inclusive) of the reserved VC range
End VCI	Ending virtual circuit identifier (inclusive) of the reserved VC range

Related Documentation

- *show atm vc*

Monitoring Global Call Admission Control Configuration

Purpose Display global call admission control (CAC) configuration.

Action To display CAC configuration:

```
host1#show cac
resource info flood interval 180
```

Related Documentation

- *show cac*

Monitoring Interfaces Configured with Traffic Engineering Bandwidth Accounting

Purpose Display interfaces on which traffic engineering bandwidth accounting is configured.

Action To display information about CAC interfaces:

```
host1#show cac interface
atm2/0
  bandwidth 10 kbps
  IP/MPLS reserveable bw 10 kbps
```

```
current total available bw 10 kbps
MPLS TE flooding threshold:
  up   15 30 45 60 75 80 85 90 95 96 97 98 99 100
  down 100 99 98 97 96 95 90 85 80 75 60 45 30 15
MPLS TE administrative weight 0
MPLS TE attribute flags 0
Available BW at 8 priority levels:
    0     10 kbps
    1     10 kbps
    2     10 kbps
    3     10 kbps
    4     10 kbps
    5     10 kbps
    6     10 kbps
    7     10 kbps
```

Meaning Table 61 on page 332 lists the **show cac interfacevc** command output fields.

Table 61: show cac interface Output Fields

Field Name	Field Description
bandwidth	Maximum physical bandwidth in Kbps; line rate
IP/MPLS reserveable bw	Total bandwidth in Kbps that can be reserved for MPLS; includes bandwidth that is already reserved as well as bandwidth not yet reserved
current total available bw	Total bandwidth in Kbps that is available to be reserved
MPLS TE flooding threshold up/down	Absolute percentages of total reservable bandwidth that trigger the flooding of the new bandwidth value throughout the network; flooding is triggered when bandwidth increases past any of the up threshold values and when bandwidth decreases past any of the down threshold values
MPLS TE administrative weight	Weight assigned to the interface that supersedes a weight assigned by the IGP
MPLS TE attribute flags	32-bit value that assigns the interface to a resource class and enables a tunnel to discriminate among interfaces by matching against tunnel affinity bits
Available BW at 8 priority levels	Bandwidth in Kbps that is available at each priority level in the range 0–7

Related Documentation

- *show cac interface*

Monitoring Virtual Router Configuration

Purpose Display the configuration of all virtual routers or a specific virtual router.

Action To display VR configuration:

host1#**show configuration virtual-router euro7**

Related Documentation

- *show configuration*

Monitoring IP and IPv6 Tunnel Routing Tables

Purpose Display the current state of the IPv4 or IPv6 tunnel routing table.

Action To display information about all IP tunnel routes:

```
host1:vr2# show ip tunnel-route all
Protocol/Route type codes:
  I1- ISIS level 1, I2- ISIS level2,
  I- route type intra, IA- route type inter, E- route type external,
  i- metric type internal, e- metric type external,
  O- OSPF, E1- external type 1, E2- external type2,
  N1- NSSA external type1, N2- NSSA external type2
  L- MPLS label, V- VRF, *- via indirect next-hop
  Prefix/Length      Type       Next Hop      Dst/Met                  Interface
------------------ ---------- --------------- ---------- ----------------------------------------
200.200.200.1/32   Ldp        111.111.1.1[L18 110/2      ATM5/1.1
                              ]
                   Rsvp       200.200.200.1[L 110/2      ATM5/1.1
                              25]
```

To display information about all IPv6 tunnel routes:

```
host1:pe1:pe11# show ipv6 tunnel-route all
Protocol/Route type codes:
  O- OSPF, E1- external type 1, E2- external type2,
  N1- SSA external type1, N2- NSSA external type2
  L- MPLS label, V- VRF, *- via indirect next-hop
         Prefix/Length             Type      Dst/Met                  Interface
-------------------------------- --------- ---------- ----------------------------------------
::21.21.21.0/126                 BgpTunnel 200/0      [L20,L26] ATM5/0.10
                                 BgpTunnel 200/0      [L20,L34] ATM5/0.10
2::2/128                         BgpTunnel 200/0      [L20,L26] ATM5/0.10
                                 BgpTunnel 200/0      [L20,L34] ATM5/0.10
```

To display detailed information about all IP tunnel routes beginning with address 200.200.200.1/32:

```
host1:vr2# show ip tunnel-route 200.200.200.1/32 detail
Protocol/Route type codes:
  I1- ISIS level 1, I2- ISIS level2,
  I- route type intra, IA- route type inter, E- route type external,
  i- metric type internal, e- metric type external,
  O- OSPF, E1- external type 1, E2- external type2,
  N1- NSSA external type1, N2- NSSA external type2
  L- MPLS label, V- VRF, *- via indirect next-hop
200.200.200.1/32 Type: Ldp Distance: 110 Metric: 2 Tag: 0 Class: 0
  MPLS next-hop: 3, label 18 on ATM5/1.1 (ip19000003.mpls.ip), nbr 111.111.1.1
```


To display detailed information about all IPv6 tunnel routes beginning with address ::21.21.21.0/126:

```
host1:pe1:pe11# show ipv6 tunnel-route ::21.21.21.0/126 detail all
Protocol/Route type codes:
  O- OSPF, E1- external type 1, E2- external type2,
  N1- SSA external type1, N2- NSSA external type2
  L- MPLS label, V- VRF, *- via indirect next-hop
::21.21.21.0/126 Type: BgpTunnel Distance: 200 Metric: 0 Class: 0
  MPLS next-hop: 18, label 20, VPN traffic, resolved by MPLS next-hop 13
  MPLS next-hop: 13, resolved by MPLS next-hop 34, peer ::ffff:2.2.2.2
  MPLS next-hop: 34, ECMP next-hop, leg count 2
  MPLS next-hop: 17, resolved by MPLS next-hop 16, peer 2.2.2.2
  MPLS next-hop: 16, primary(in use): label 26 on ATM5/0.10, secondary: resolved by MPLS next-hop 0
```

Meaning Table 62 on page 334 lists the **show ip tunnel-route** command and **show ipv6 tunnel route** command output fields.

Table 62: show ip tunnel route and show ipv6 tunnel-route Output Fields

Field Name	Field Description
Prefix	IPv4 or IPv6 address prefix of network destination
Length	Network mask length for prefix
Type	Type of route; protocol
Next Hop	IP address of the next hop to the route, whether it is a local interface or another router; not displayed for IPv6 tunnel routing table
Dst or Distance	Administrative distance for the route
Met or Metric	Number of hops; metric
Interface	Interface type and interface specifier
Tag	Numeric tag that identifies route
Class	Attribute of a route applied only as a result of **set route-class** clause in a table map

Related Documentation

- *show ip tunnel-route*
- *show ipv6 tunnel-route*

Monitoring LDP

Purpose Display information about LDP.

Action To display LDP information:

```
host1#show ldp
LDP
  LSR ID is 80.0.0.2
  FEC Deaggregation is off
  Egress label: implicit-null
  Label distribution control mode: ordered control
  LDP session retry  0 times at interval 10
  LDP session hold time: 180
  LDP session keepalive interval: 20
  LDP targeted-hello hold time: 45
  LDP targeted-hello interval: 15
  Topology Driven LSP enabled
  LSPs used for IP forwarding
      for host addresses only
```

NOTE: The **mpls** keyword is optional and is provided for compatibility with non–E Series implementations.

Meaning Table 63 on page 335 lists the **show ldp** command output fields.

Table 63: show ldp Output Fields

Field Name	Field Description
LSR ID	IP address of label-switched router
FEC Deaggregation	State of FEC deaggregation, on or off
Egress label	Type of label advertised for the LSR that is the egress router for the prefix, implicit null, explicit null, or a non-null label
Label distribution control mode	Label distribution control mode used by LDP for label distribution, independent control, or ordered control
LDP session retry	Configured values for the number of LDP session retry attempts and the retry interval
LDP session hold time	Configured value for the LDP session hold time
LDP session keepalive interval	Interval at which LDP sends session keepalive messages, in seconds
LDP targeted-hello hold time	LDP targeted-hello hold time, in seconds
LDP targeted-hello interval	LDP targeted-hello interval, in seconds
Topology Driven LSP	Status of topology-driven LSP, enabled or disabled

Table 63: show ldp Output Fields *(continued)*

Field Name	Field Description
LSPs used for IP forwarding	LSPs are placed in the IP routing table for forwarding plain IP traffic; displayed only when the **mpls ldp ip-forwarding** command has been configured. Indicates whether the LSPs that are used for IP forwarding are host only, subject to a specified access list, or subject to a specified prefix list.
LDP proto stats	LDP protocol statistics
totalPeersDiscovered	Number of LDP peers discovered
totalAdjacenciesEstablished	Number of LDP adjacencies established
totalSessionsEstablished	Number of LDP sessions established
totalFECElements	Number of FEC elements
totalFECs	Number of FECs
totalInLabels	Number of in labels (sent to upstream neighbor)
totalOutLabels	Number of out labels (received from downstream neighbor)
totalCrLSPSetup	Number of constraint-based routed LSPs set up
totalCrLSPDeleted	Number of constraint-based routed LSPs deleted

Related Documentation

- *show ldp*

Monitoring MPLS Label Bindings

Purpose Use to display label bindings from the MPLS label information base.

Action To display MPLS label bindings:

host1#**show mpls binding**

```
   Frame Relay over MPLS  vc-id 50001 group-id 2
    In    26 neighbor 10.9.1.3
    Out   27 neighbor 10.9.1.3
   VLAN over MPLS  vc-id 240001 group-id 2
    In    22 neighbor 10.9.1.3
    Out   25 neighbor 10.9.1.3

  10.1.1.1/32
    In    10001 neighbor 10.3.11.2
    Out   20001 neighbor 10.3.11.2

  10.2.2.2/32
    In    10002 neighbor 10.4.12.2    stale
    Out   20002 neighbor 10.4.12.2    stale
```

```
10.3.3.3/32
  In     10005 neighbor 10.4.12.2    stale
  Out    20003 neighbor 10.4.12.2    stale

10.4.12.0/30
  In     10003 neighbor 10.5.5.2
  Out    20004 neighbor 10.5.5.2

10.4.23.0/30
  In     10004 neighbor 10.5.5.2
  Out    20005 neighbor 10.5.5.2
```

NOTE: The **ldp** keyword and the **mpls** keyword display the same information.

Meaning Table 64 on page 337 lists the **show ldp binding** command and **show mpls binding** command output fields.

Table 64: show ldp binding and show mpls binding Output Fields

Field Name	Field Description
In	Label sent to upstream neighbor for displayed route
Out	Label received from downstream neighbor for displayed route
neighbor	IP address of neighbor to which the label is sent or received
stale	Label that indicates neighbor has restarted

Related Documentation

- *show ldp binding*
- *show mpls binding*

Monitoring LDP Graceful Restart

Purpose Display information about LDP graceful restart.

Action To display information about LDP graceful restart:

```
host1#show ldp graceful-restart
LDP Graceful Restart is enabled
Helper Mode is enabled
Reconnect Time: 220 sec
Recovery Time: 240 sec
Max Recovery Time: 260 sec
Neighbor Liveness Timer: 280 sec
  Peer 80.0.1.1:0, State: operational, Restarter Mode: disabled, Helper Mode:
enabled
  Peer 80.0.3.3:0, State: operational, Restarter Mode: disabled, Helper Mode:
enabled
```

NOTE: The **mpls** keyword is optional and is provided for compatibility with non–E Series implementations.

Meaning Table 65 on page 338 lists the **show ldp graceful restart** command output fields.

Table 65: show ldp graceful restart Output Fields

Field Name	Field Description
LDP Graceful Restart	State of graceful restart, enabled or disabled
Helper Mode	State of graceful restart helper mode, enabled or disabled
Reconnect Time	Locally configured value for reconnect time, in seconds
Recovery Time	Locally configured value for recovery time, in seconds
Max Recovery Time	Locally configured value for max-recovery timer, in seconds
Neighbor Liveness Timer	Locally configured value for neighbor-liveness timer, in seconds
Peer	Address, state, and LDP graceful restart state for neighbor

Related Documentation

- *show ldp graceful-restart*

Monitoring Interfaces That are Synchronizing with LDP

Purpose Display information about interfaces that are synchronizing with LDP or the specified interface that is synchronizing with LDP.

Action To display information about interfaces synchronizing with LDP:

```
host1#show ldp igp-sync
Atm 0/0:
        LDP configured;  SYNC enabled.
        SYNC status: sync achieved; peer reachable.
        IGP holddown time: infinite.
        Peer LDP Ident: 10.130.0.1:0
        IGP enabled: OSPF 1
```

Meaning Table 66 on page 338 lists the **show ldp igp-sync** command output fields.

Table 66: show ldp igp-sync Output Fields

Field Name	Field Description
LDP	State of LDP, configured, auto-configured, or not configured

Table 66: show ldp igp-sync Output Fields *(continued)*

Field Name	Field Description
SYNC status	State of synchronization, enabled or disabled
IGP holddown time	Value of IGP holddown time, infinite or number of milliseconds
Peer LDP Ident	IP address of LDP peer
IGP enabled	IGP protocol

Related Documentation

- *show ldp igp-sync*

Monitoring LDP Interfaces

Purpose Display information about all LDP interfaces or the specified LDP interface.

Action To display information about all LDP interfaces:

```
host1#show ldp interface
Interface ATM6/0.120
    Interface address: 192.168.12.1/28
    Enabled with profile 'default'
    Configured hold time: 15
    Hello interval: 5
    Hold Time: 1
    242 hello received, 242 hello sent, 0 hello rejected
    1 adjacency created, 0 adjacency deleted,
  Number of adjacencies = 1
    Link hello adjacency: Address: 10.10.12.2, Transport address: 80.0.2.2,
      Up for 00:20:09, Remaining hold time: 11 sec
```

To display brief information about LDP interfaces:

```
host1#show ldp interface brief
Interface              IP-Address         Protocol
ATM6/1.1               192.168.100.21/30  enabled
ATM6/1.3               192.168.100.17/30  enabled
ATM6/1.5               192.168.100.13/30  enabled
ATM6/0.7               172.16.100.1/30    enabled
ATM6/0.8               172.16.100.22/30   enabled
ATM6/0.9               172.16.100.14/30   enabled
```

NOTE: The **mpls** keyword is optional and is provided for compatibility with non–E Series implementations.

Meaning Table 67 on page 340 lists the **show ldp interface** command output fields.

Table 67: show ldp interface Output Fields

Field Name	Field Description
Interface	Identifier of the interface
autoconfigured	LDP has been autoconfigured on the interface
Interface address	IP address of the interface, with address mask
Enabled with profile	Name of profile with which interface was enabled
Configured hold time	Configured period for which a sending LSR maintains a record of link hello messages from the receiving LSR without receipt of another link hello message from that LSR, in seconds
Hello interval	Negotiated interval between link-hello packets, in seconds
Hold time	Lowest configured hold time among all neighbors on the same subnet, used as the effective hold time, in seconds
Number of adjacencies	Number of LDP adjacencies for the interface
Link hello adjacency	Address and transport address of the link hello adjacency; time adjacency has been up in *hh:mm:ss*; remaining hold time for the adjacency in seconds
label alloc	Number of labels allocated and advertised to this peer
label learned	Number of labels received from this peer
accum label alloc	Cumulative total number of labels allocated and advertised to this peer
accum label learned	Cumulative total number of labels received from this peer
last restart time	Time in *hh:mm:ss* since session last restarted
notf	Number of notification messages received or received bad or sent
msg	Number of messages received or received bad or sent
mapping	Number of label mapping messages received or received bad or sent
request	Number of label request messages received or received bad or sent
abort	Number of label abort messages received or received bad or sent
release	Number of label release messages received or received bad or sent

Table 67: show ldp interface Output Fields *(continued)*

Field Name	Field Description
withdraw	Number of label withdraw messages received or received bad or sent
addr	Number of address messages received or received bad or sent
addr withdraw	Number of address withdraw messages received or received bad or sent
msgId	Number of message ID messages received or sent
unknown msg type err	Number of unknown message type errors received
hello recv	Number of hello messages received
hello sent	Number of hello messages sent
bad hello recv	Number of hello messages received bad
adj setup time	Time in *hh*:*mm*:*ss* since adjacency set up
last hello recv time	Time in *hh*:*mm*:*ss* since last hello message received
last hello sent time	Time in *hh*:*mm*:*ss* since last hello message sent
remaining hold time	Time in *hh*:*mm*:*ss* remaining of the hold time
IP-Address	IP address of the interface
Protocol	Administrative state of LDP, enabled or disabled

Related Documentation

- *show ldp interface*

Monitoring LDP Neighbors

Purpose Display LDP neighbor information.

Action To display information about LDP neighbor 10.3.5.1:

```
host1#show ldp neighbor 10.3.5.1
LDP Neighbor: 10.0.2.2
  LSR: Remote 10.0.2.2:0, local 10.0.1.1:0
  Transport address: remote 10.0.2.2, local 10.0.1.1
  State: Operational
  LDP advertisement: Unsolicited
  Up for 00:20:03

  Number of next-hop addresses received = 3
    10.0.2.2  100.6.12.2  100.6.23.2
```



```
  Number of adjacencies = 1
    Link Hello adjacency: address 10.6.12.2, transport 10.0.2.2,
      Up for 00:20:09, remaining hold time: 11 sec
```

To display brief information about all LDP neighbors:

host1#**show ldp neighbor brief**

```
Neighbor                 Transport Address   State
10.0.2.2                10.0.1.1->80.0.2.2     Operational
```

To display information about graceful restart for neighbors:

host1#**show ldp neighbor graceful-restart**

```
LDP Neighbor: 10.0.1.1
  Graceful Restart is disabled
  Helper Mode is enabled
  Reconnect Time: 0 msec
  Recovery Time: 0 msec
  State: operational

LDP neighbor 10.0.2.2
  Graceful Restart is enabled
  Helper Mode is enabled
  Reconnect Time: 220000 msec
  Recovery Time: 0 msec
  State: operational
```

To display information about LDP statistics for the session with each LDP neighbor:

host1#**show ldp neighbor statistics**

```
LDP Neighbor: 10.0.2.2
  Message type      Received   Sent
----------------   --------   ----
Initialization            1      1
Keepalive                85     85
Notification              0      0
Address                   1      1
Address withdraw          0      0
Label mapping             5      5
Label request             0      0
Label withdraw            2      2
Label release             2      2
```

NOTE: The **mpls** keyword is optional and is provided for compatibility with non–E Series implementations.

NOTE: If a password is configured for a peer, you can view the password with the **show configuration** command. This command displays the passwords in cleartext unless the **service password-encryption** command has been issued, in which case the passwords are displayed in encrypted format.

Meaning Table 68 on page 343 lists the **show ldp neighbor** command output fields.

Table 68: show ldp neighbor Output Fields

Field Name	Field Description
LDP neighbor	IP address of LDP peer
LSR	IP address of remote and local peers; the number following the colon is the platform label space ID, and is always 0
Transport address	Transport remote and local address for the TCP session
State	State of the session, nonexistent (session connection not established) or operational (has received keepalive message)
LDP advertisement	Mode of label distribution, downstream-unsolicited or downstream-on-demand
Up	Time that the adjacency has been up, in *hh:mm:ss* format
Graceful Restart	State of graceful restart, enabled or disabled
Helper Mode	State of graceful restart helper mode, enabled or disabled
Reconnect Time	Value for reconnect time received from peer in FT TLV, in milliseconds
Recovery Time	Value for recovery time received from peer in FT TLV, in milliseconds
State	Status of the neighbor’s graceful restart, one of the following:
nonexistent	LDP session is not established
restarting	Neighbor LSR is restarting
recovering	Session with neighbor LSR has reestablished after the neighbor restarted; label bindings are being exchanged
operational	LDP session is up
Neighbor	IP address of LDP peer
Initialization	Number of initialization messages received and sent
Keepalive	Number of keepalive messages received and sent
Notification	Number of notification messages received and sent
Address	Number of address messages received and sent
Address withdraw	Number of address withdraw messages received and sent
Label mapping	Number of label mapping messages received and sent

Table 68: show ldp neighbor Output Fields *(continued)*

Field Name	Field Description
Label request	Number of label request messages received and sent
Label withdraw	Number of label withdraw messages received and sent
Label release	Number of label release messages received and sent

Related Documentation

- *show ldp neighbor*

Monitoring LDP Profiles

Purpose Display a specific LDP profile, or all LDP profiles.

Action To display the default LDP profile:

```
host1:pe2#show ldp profile default
ldp profile default: used by 2 interfaces
   session retry:  10 times at interval 10
```

NOTE: The **mpls** keyword is optional and is provided for compatibility with non–E Series implementations.

Meaning Table 69 on page 344 lists the **show ldp profile** command output fields.

Table 69: show ldp profile Output Fields

Field Name	Field Description
profile	Number of interfaces that use the profile
session retry	Number of attempts that will be made to set up an MPLS LDP session

Related Documentation

- *show ldp profile*

Monitoring LDP Statistics

Purpose Display statistics for LDP on the current virtual router.

Action To display all LDP statistics:

```
host1#show ldp statistics
```

```
  Message type     Received   Sent
----------------   --------   -----
Hello                 25733   25735
Initialization            2       2
Keepalive              9646    9646
Notification              0       0
Address                   2       2
Address withdraw          0       0
Label mapping             8       8
Label request             0       0
Label withdraw            0       0
Label release             0       0
Label abort               0       0
All UDP               25733   25735
All TCP                9654    9654

     Event type         Total
---------------------   -----
Sessions opened             2
Sessions closed             0
Topology changes            5
No router id                0
No address                  0
No interface                0
No session                  0
No adjacency                0
Unknown version             0
Malformed PDU               0
Malformed message           0
Unknown message type        0
Inappropriate message       0
Malformed tlv               0
Bad TLV value               0
Missing TLV                 0
PDU too large               0
PDU too small               0
No Memory                   0
```

NOTE: The **mpls** keyword is optional and is provided for compatibility with non–E Series implementations.

Meaning Table 70 on page 345 lists the **show ldp statistics** command output fields.

Table 70: show ldp statistics Output Fields

Field Name	Field Description
Hello	Number of hello messages received and sent
Initialization	Number of initialization messages received and sent
Keepalive	Number of keepalive messages received and sent
Notification	Number of notification messages received and sent

Table 70: show ldp statistics Output Fields *(continued)*

Field Name	Field Description
Address	Number of address messages received and sent
Address withdraw	Number of address withdraw messages received and sent
Label mapping	Number of label mapping messages received and sent
Label request	Number of label request messages received and sent
Label withdraw	Number of label withdraw messages received and sent
Label release	Number of label release messages received and sent
Label abort	Number of label abort messages received and sent
All UDP	Number of UDP messages received and sent
All TCP	Number of TCP messages received and sent
Sessions opened	Number of session opened events
Sessions closed	Number of session closed events
Topology changes	Number of topology change events
No router id	Number of no router ID events
No address	Number of no address events
No interface	Number of no interface events
No session	Number of no session events
No adjacency	Number of no adjacency events
Unknown version	Number of unknown version events
Malformed PDU	Number of malformed PDU events
Malformed message	Number of malformed message events
Unknown message type	Number of unknown message type events
Inappropriate message	Number of inappropriate message events
Malformed tlv	Number of inappropriate message events
Bad TLV value	Number of bad TLV value events

Table 70: show ldp statistics Output Fields *(continued)*

Field Name	Field Description
Missing TLV	Number of missing TLV events
PDU too large	Number of PDU too large events
PDU too small	Number of PDU too small events
No Memory	Number of no memory events

Related Documentation

- *show ldp statistics*

Monitoring LDP Targeted Hello Receive and Send Lists

Purpose Display LDP targeted hello receive or send list, or both.

Action To display both the LDP targeted hello receive and send lists:

```
host1#show ldp targeted session
Mpls Target Session Status:

 D = Dynamically, S = Statically, A = Access List Configured

    Targeted session sent to 10.9.1.3 is up  Used By: D
      indirect nexthop index 3, resolved
    Targeted session sent to 10.9.1.6 is up  Used By: S
      indirect nexthop index 206, resolved
```

Meaning Table 71 on page 347 lists the **show ldp targeted session** command output fields.

Table 71: show ldp targeted session Output Fields

Field Name	Field Description
D	Targeted session created by layer 2 over MPLS connection; see *Layer 2 Services over MPLS Overview* in the *JunosE BGP and MPLS Configuration Guide* for more information about layer 2 over MPLS
S	Targeted session statically created by user
A	Targeted session created by access list
Used By	Letter representing source of targeted session

Related Documentation

- *show ldp targeted session*

Monitoring MPLS Status and Configuration

Purpose Display status and configuration information about MPLS.

Action To display information about MPLS Status and configuration:

```
host1#show mpls
MPLS administratively enabled
  Current state is Config incomplete
  LSR ID is 10.2.2.2
  Re-optimization timer is 3600
  Label range 3000 ~ 4000
  retry forever at interval 30 during LSP setup if there is route
  retry forever at interval 30 during LSP setup if there is no route
Loop Detect enabled
```

Additional detail is shown when LDP is enabled:

```
LDP
  LSR ID is 80.0.0.2
  FEC Deaggregation is off
  Egress label: implicit-null
  Label distribution control mode: ordered control
  LDP session retry  0 times at interval 10
  LDP session hold time: 180
  LDP session keepalive interval: 20
  LDP targeted-hello hold time: 45
  LDP targeted-hello interval: 15
  Topology Driven LSP enabled
  LSPs used for IP forwarding
      for host addresses only
```

Additional detail is shown when RSVP-TE is enabled:

```
RSVP is enabled
  LSRID 10.1.1.1
  Re-optimization timer is 3600
  Tunnel retry forever at interval 5 if route is available
  Tunnel retry forever at interval 5 if no route is available
  Refresh reduction is OFF
  Message bundling is OFF
  Egress label is non-null
  Hellos are on with an interval of 10000 and miss limit of 4
  Graceful restart is ON
    Restart time 60000 milliseconds
    Recovery time 120000 milliseconds
```

- Additional detail shown when RSVP-TE graceful restart helper mode is enabled:

```
RSVP is enabled
...
  Graceful restart is ON (helper mode)
```

Meaning Table 72 on page 348 lists the **show mpls** command output fields.

Table 72: show mpls Output Fields

Field Name	Field Description
MPLS	Status of MPLS, administratively enabled or disabled, and configuration status
LSR ID	IP address of label-switched router

Table 72: show mpls Output Fields *(continued)*

Field Name	Field Description
Re-optimization timer	Frequency at which LSPs are checked for better paths
Label range	Range of platform label space
retry	Retry behavior to be performed during LSP setup
Loop Detect	Status of loop detection, enabled or disabled
LDP	This field and the following fields are displayed only when LDP is enabled.
LSR ID	IP address of label-switched router
FEC Deaggregation	State of FEC deaggregation, on or off
Egress label	Type of label advertised for the LSR that is the egress router for the prefix, explicit-null or a non-null label
Label distribution control mode	Label distribution control mode used by LDP for label distribution, independent control or ordered control
LDP session retry	Interval in seconds between attempts to set up an MPLS LDP session
LDP session hold time	Period in seconds for which an LSR maintains the session with its LDP peer without receipt of any LDP message from that peer
LDP session keepalive interval	Interval at which LDP sends session keepalive messages, in seconds
LDP targeted hello hold time	LDP targeted-hello hold time, in seconds
LDP targeted-hello interval	LDP targeted-hello interval, in seconds
Topology Driven LSP	Status of topology-driven LSP, enabled or disabled
LSPs used for IP forwarding	LSPs are placed in the IP routing table for forwarding plain IP traffic; displayed only when the **mpls ldp ip-forwarding** command has been configured. Indicates whether the LSPs that are used for IP forwarding are host only, subject to a specified access list, or subject to a specified prefix list.
RSVP is enabled	This field and the following fields are displayed only when RSVP-TE is enabled.
LSRID	IP address of label-switched router
Re-optimization timer	Frequency at which LSPs are checked for better paths

Table 72: show mpls Output Fields *(continued)*

Field Name	Field Description
Tunnel retry	Retry behavior to be performed during LSP setup
Refresh reduction	State of RSVP-TE summary refresh reduction, OFF or ON
Message bundling	State of RSVP-TE summary refresh message bundling, OFF or ON
Egress label	Type of label advertised for the LSR that is the egress router for the prefix, explicit-null or a non-null label
Hellos	State of RSVP-TE hello feature, including the hello refresh interval and hello miss limit
Graceful restart	State of RSVP-TE graceful restart, OFF or ON
helper mode	Graceful restart helper mode is enabled when this field is displayed
Restart time	Graceful restart time, in milliseconds
Recovery time	Graceful restart recovery time, in milliseconds

Related Documentation

- *show mpls*

Monitoring MPLS Explicit Paths

Purpose Display MPLS explicit paths.

Action To display information about all MPLS explicit paths:

```
host1:pe2#show mpls explicit-paths
  path name/identifier rx1-path enabled
    1: next-address 70.70.70.2
    2: next-address 30.30.30.1
  not referenced by any options
  path name/identifier rx1-path2 enabled
    1: next-address 60.60.60.2
    2: next-address 40.40.40.1
  not referenced by any options
```

To display information about the MPLS explicit path named rx1-path2:

```
host1:pe2#show mpls explicit-paths name rx1-path2
  path name/identifier rx1-path2 enabled
    1: next-address 60.60.60.2
    2: next-address 40.40.40.1
  not referenced by any options
```

Meaning Table 73 on page 351 lists the **show mpls explicit-paths** command output fields.

Table 73: show mpls explicit-paths Output Fields

Field Name	Field Description
path name/identifier	Name or identifier of explicit path and status, enabled or disabled, followed by list of path links and the IP address for each link's next address

Related Documentation

- *show mpls explicit-paths*

Monitoring RSVP-TE Status and Configuration

Purpose Display status and configuration information about RSVP-TE.

Action To display information about RSVP-TE status and configuration settings:

```
host1#show mpls rsvp
RSVP is enabled
  LSRID 10.1.1.1
  Re-optimization timer is 3600
  Tunnel retry forever at interval 5 if route is available
  Tunnel retry forever at interval 5 if no route is available
  Refresh reduction is OFF
  Message bundling is OFF
  Egress label is non-null
  Hellos are on with an interval of 10000 and miss limit of 4
  Graceful restart is ON
    Restart time 60000 milliseconds
    Recovery time 120000 milliseconds
```

Additional detail shown when RSVP-TE graceful restart helper mode is enabled:

```
RSVP is enabled
...
  Graceful restart is ON (helper mode)
```

Meaning Table 74 on page 351 lists the **show mpls rsvp** command output fields.

Table 74: show mpls rsvp Output Fields

Field Name	Field Description
RSVP is enabled	RSVP-TE is enabled for MPLS tunnels
LSRID	IP address of label-switched router
Re-optimization timer	Frequency at which LSPs are checked for better paths
Tunnel retry	Retry behavior to be performed during LSP setup
Refresh reduction	State of RSVP-TE summary refresh reduction, OFF or ON
Message bundling	State of RSVP-TE summary refresh message bundling, OFF or ON

Table 74: show mpls rsvp Output Fields *(continued)*

Field Name	Field Description
Egress label	Type of label advertised for the LSR that is the egress router for the prefix, explicit-null or a non-null label
Hellos	State of RSVP-TE hello feature, including the hello refresh interval and hello miss limit
Graceful restart	State of RSVP-TE graceful restart, OFF or ON
helper mode	Graceful restart helper mode is enabled when this field is displayed
Restart time	Graceful restart time, in milliseconds
Recovery time	Graceful restart recovery time, in milliseconds

Related Documentation

- *show mpls rsvp*

Monitoring the RSVP-TE Bypass Tunnels

Purpose Display information about the backup status of primary LSPs protected with bypass tunnels.

Action To display the backup status of protected primary LSPs on a core router:

```
host1(config-if)# show mpls fast-reroute database
                            OutIntf                                Backup
Role        Name            / Label       BackupIntf / Label       Status
----    ---------------   -------------   ----------------------   -----------
Core    LSP 10.1.1.1:6    ATM4/0.2 / 21   tun mpls:bypass23 / 21   Established
Core    LSP 10.1.1.1:7    ATM4/0.2 / 26   tun mpls:bypass23 / 26   Established
Example on a tunnel ingress router
                 OutIntf                                Backup
Role    Name     / Label       BackupIntf / Label       Status
----    ----   -------------   ----------------------   -----------
Head    1      ATM4/0.1 / 27   tun mpls:bypass12 / 27   Established
Head    p2     ATM4/0.1 / 21   tun mpls:bypass12 / 21   Established
```

Meaning Table 75 on page 352 lists the **show mpls fast-reroute** command output fields.

Table 75: show mpls fast-reroute Output Fields

Field Name	Field Description
Role	Role of the router in the LSP: core, head, or tail
Name	Name of the primary LSP
OutIntf / Label	Interface type and specifier of the outgoing interface, and the label associated with that interface

Table 75: show mpls fast-reroute Output Fields *(continued)*

Field Name	Field Description
BackupIntf / Label	Interface type and specifier of the backup interface, and the label associated with that interface
Backup Status	Status of backup protection (bypass) for the LSP

Related Documentation

- *show mpls fast-reroute database*

Monitoring MPLS Labels Used for Forwarding

Purpose Display information for labels being used for forwarding.

Action To display information about MPLS labels used for forwarding:

```
host1:vr2#show mpls forwarding
In label: 28
  Label space: platform label space
  Owner: ldp
  Spoof check: router pe1
  Action:
    MPLS next-hop: 1, lookup on inner header/label
  Statistics:
    0 in pkts
    0 in Octets
    0 in errors
    0 in discard pkts
```

To display summary information about MPLS labels used for forwarding:

```
host1:vr2# show mpls forwarding brief
Platform label space

In Label  Owner                              Action
-------- -------- --------------------------------------
28       ldp      lookup on inner header/label
29       ldp      swap to 20 on ATM2/0.10, nbr 10.10.10.2
30       ldp      lookup on inner header/label
31       ldp      swap to 22 on ATM2/0.10, nbr 10.10.10.2
32       ldp      swap to 23 on ATM2/0.10, nbr 10.10.10.2
```

Meaning Table 76 on page 353 lists the **show mpls forwarding** command output fields.

Table 76: show mpls forwarding Output Fields

Field Name	Field Description
In label	Label sent to upstream neighbor for route
Out label	Label received from downstream neighbor for route
Label space	Label space in which the label is assigned

Table 76: show mpls forwarding Output Fields *(continued)*

Field Name	Field Description
Owner	Signaling protocol that placed the label in the forwarding table: BGP, LDP, or RSVP-TE
Spoof check	Type and location of spoof checking performed on the MPLS packet, router or interface
Action	Action taken for MPLS packets arriving with that label
in pkts	Number of packets sent with the label
in Octets	Number of octets sent with the label
in errors	Number of packets that are dropped for some reason before being sent
in discardPkts	Number of packets that are discarded due to lack of buffer space before being sent

Related Documentation

- *show mpls forwarding*

Monitoring MPLS Interfaces

Purpose Display status and configuration information about MPLS interfaces.

Action To display information about all MPLS interfaces:

```
host1:pe1#show mpls interface
MPLS major interface ATM2/0.10
  ATM circuit type is 1483 LLC encapsulation
  Administrative state is enabled
  Operational state is up
  Operational MTU is 9180
  Received:
    0 packets
    0 bytes
    0 errors
    0 discards
    0 failed label lookups
  Sent:
    0 packets
    0 bytes
    0 errors
    0 discards

LDP information:
    10.1.1.2/24
    enabled with profile 'default'
     0 hello recv, 1 hello sent, 0 hello rej
     0 adj setup, 0 adj deleted,
```

```
  RSVP
    Enabled with profile default
    Authentication is disabled
    Authentication key: <none>
    Hellos are on with an interval of 10000 and miss limit of 4
    Hello settings are not inherited

MPLS minor interface pe1-to-pe2 (transmit)
  Stacked on MPLS major ATM2/0.10
  Operational state is up
  Sent:
    0 packets
    0 bytes

  queue 0: traffic class best-effort, bound to atm-vc ATM2/0.10
    Queue length 0 bytes
    Forwarded packets 0, bytes 0
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0

MPLS minor interface lsp-02020202-1-4 (receive)
  Stacked on MPLS major ATM2/0.10
  Operational state is up
  Statistics not enabled for this interface
```

- The following excerpt shows the output for MPLS interface atm 5/1.1 when RSVP-TE is enabled and RSVP-TE authentication is enabled:

```
host1:pe2#show mpls interface atm 5/1.1
Interface ATM5/1.1 Up
 RSVP enabled with profile default
   Authentication: enabled
   Authentication Key: <a password has been configured>
 LDP not configured
 IP interfaces on this MPLS interface:
    192.168.100.21/30
MPLS Statistics:
  Rcvd: 0 failed lbl lookup, 0 octets, 0 hcOctets
        0 pkts, 0 hcPkts, 0 errors, 0 discards
  Sent: 0 octets, 0 hcOctets, 0 pkts
        0 hcPkts, 0 errors, 0 discards
...
```

To display information about MPLS interface atm 2/0.60:

```
host1:pe2#show mpls interface atm 2/0.60
Interface atm2/0.60 Up
 RSVP not configured
 LDP enabled with profile default
 IP interfaces on this MPLS interface:
    60.60.60.1/16  Session to 4.4.4.4 is operational (active)
Session statistics:
          12 label alloc, 12 label learned,
          12 accum label alloc, 12 accum label learned,
          last restart time = 00:04:44
          Rcvd: 0 notf, 29 msg, 12 mapping, 0 request
                0 abort, 0 release, 0 withdraw, 1 addr
                0 addr withdraw, 29 msgId
                0 bad mapping, 0 bad request, 0 bad abort, 0 bad release
                0 bad withdraw, 0 bad addr, 0 bad addr withdraw
                0 unknown msg type err
```

```
                last info err code = 0x00000000, 0 loop detected
          Sent: 0 notf, 29 msg, 12 mapping, 0 request
                0 abort, 0 release, 0 withdraw, 1 addr
                0 addr withdraw, 29 msgId
        Adjacency statistics:
          58 hello recv, 57 hello sent, 0 bad hello recv
          adj setup time = 00:04:44
          last hello recv time = 00:00:05, last hello sent time = 00:00:05
MPLS Statistics:
   Rcvd: 0 failed lbl lookup, 0 octets, 0 hcOctets
         0 pkts, 0 hcPkts, 0 errors, 0 discards
   Sent: 0 octets, 0 hcOctets, 0 pkts
         0 hcPkts, 0 errors, 0 discards
   1 adjacency, 1 session, 3 accum adjacency, 3 accum session
   14058 hello recv, 14063 hello sent, 0 hello rej
   3 adj setup, 2 adj deleted
```

The following excerpt shows LDP and RSVP-TE information for MPLS interface atm 6/1.1:

```
host1:vr2#show mpls interface atm 6/1.1
...
MPLS major interface ATM6/1.1
  ATM circuit type is 1483 LLC encapsulation
  Administrative state is enabled
  Operational state is up
  Operational MTU is 9180
  Received:
    0 packets
    0 bytes
    0 errors
    0 discards
    0 failed label lookups
  Sent:
    0 packets
    0 bytes
    0 errors
    0 discards
LDP information:
    10.1.1.1/24
    enabled with profile 'default'
     0 hello recv, 2 hello sent, 0 hello rej
     0 adj setup, 0 adj deleted,

  RSVP
    Enabled with profile default
    Authentication is disabled
    Authentication key: <none>
    Hellos are on with an interval of 10000 and miss limit of 4
    Hello settings are not inherited
```

To display detailed information about MPLS interfaces:

```
host1:pe1#show mpls interface detail
MPLS major interface ATM2/0.10
  ATM circuit type is 1483 LLC encapsulation
  Administrative state is enabled
  Operational state is up
  Operational MTU is 9180
  MPLS major interface UID is 0x19000001
  Lower interface UID is 0x0b000031
  Uses platform label space
  Peer IPv4 interface is ATM2/0.10 (UID 0x000000be)
```



```
  No peer IPv6 interface
  Upper IPv4 interface is ip19000001.mpls.ip (UID 0x000000bf, FEC index 0x0000003f)

  No upper IPv4 VPN interface
  No upper IPv6 interface
  No upper IPv6 VPN interface
  Condensed location is 0x00020000
  Received:
    0 packets
    0 bytes
    0 errors
    0 discards
    0 failed label lookups
  Sent:
    0 packets
    0 bytes
    0 errors
    0 discards
  RSVP
    Enabled with profile default
    Authentication is disabled
    Authentication key: <none>

MPLS minor interface pe1-to-pe2 (transmit)
  Stacked on MPLS major ATM2/0.10
  Operational state is up
  MPLS minor interface UID is 0x1a000001
  Lower MPLS major interface UID is 0x19000001
  Sent:
    0 packets
    0 bytes

  queue 0: traffic class best-effort, bound to atm-vc ATM2/0.10
    Queue length 0 bytes
    Forwarded packets 0, bytes 0
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0
MPLS minor interface lsp-02020202-1-4 (receive)
  Stacked on MPLS major ATM2/0.10
  Operational state is up
  MPLS minor interface UID is 0x1a000004
  Lower MPLS major interface UID is 0x19000001
  Statistics not enabled for this interface
```

The following excerpt shows detailed RSVP-TE information for the MPLS interface:

```
host1:vr2#show mpls interface detail
MPLS major interface fastEthernet6/0
RSVP
    Enabled with profile default
    Authentication is disabled
    Authentication key: <none>
    Hellos are enabled
    Hellos interval is 10000 milliseconds
    Hellos miss limit is 4
    Hello settings are not inherited
```

To display summary information about MPLS interfaces:

```
host1:pe1#show mpls interface brief

MPLS major interfaces
```



```
             Admin    Oper
Interface    state    state
---------   -------   -----
ATM2/0.10   enabled   up

MPLS shim interfaces

            Remote-PE   Virtual     Load
               or       Circuit   Balancing   Admin   Oper
Interface   LSP-name      ID        Group     state   state
---------   ---------   -------   ---------   -----   -----

MPLS minor interfaces

                          Lower      Oper
   Interface            MplsMajor   state   Direction
----------------        ---------   -----   ---------
pe1-to-pe2              ATM2/0.10   up      transmit
lsp-02020202-1-4        ATM2/0.10   up      receive
ERX-01-0c-d7:pe1#
```

Meaning Table 77 on page 358 lists the **show mpls interface** command output fields.

Table 77: show mpls interface Output Fields

Field Name	Field Description
Interface	Specifier and status of each interface
RSVP	Status of RSVP, configured or not, and profile used
LDP	Status of LDP, configured or not configured, and profile used
IP interfaces on this MPLS interface	IP address of IP interfaces and session status
Condensed location	Internal, platform-dependent, 32-bit representation of the interface location, used by Juniper Networks Customer support for troubleshooting.
label alloc	Number of labels allocated and advertised to this peer
label learned	Number of labels received from this peer
accum label alloc	Cumulative total number of labels allocated and advertised to this peer
accum label learned	Cumulative total number of labels received from this peer
notf	Number of notification messages received or received bad or sent
mapping	Number of label mapping messages received or received bad or sent
msg	Number of messages sent or received
request	Number of label request messages received or received bad or sent

Table 77: show mpls interface Output Fields *(continued)*

Field Name	Field Description
abort	Number of label request abort messages for downstream on demand received or received bad or sent
release	Number of label release messages received or received bad or sent
withdraw	Number of label withdraw messages received or received bad or sent
addr	Number of address messages received or received bad or sent
addr withdraw	Number of address withdraw messages received or received bad or sent
msgId	Number of message IDs received or sent
unknown message type err	Number or unknown message type errors received
last info error code	Last received notification code
loop detected	Loop detected; for downstream on demand
hello recv	Number of hello messages received
hello sent	Number of hello messages sent
bad hello recv	Number of hello messages received bad
adj setup time	Time in *hh:mm:ss* since adjacency set up
last hello recv time	Time in *hh:mm:ss* since last hello message received
last hello sent time	Time in *hh:mm:ss* since last hello message sent
failed lbl lookup	Number of packets received whose labels are not recognized
octets	Number of octets received or sent
hcoctets	Number of high-capacity (64-bit) octets received or sent
pkts	Number of packets received or sent
hcpkts	Number of high-capacity (64-bit) packets received or sent
errors	Number of packets that are dropped for some reason at receipt or before being sent
discards	Number of packets that are discarded due to lack of buffer space at receipt or before being sent

Table 77: show mpls interface Output Fields *(continued)*

Field Name	Field Description
adjacency	Number of adjacencies currently established
session	Number of sessions currently established
accum adjacency	Cumulative total number of adjacencies established since interface is up
accum session	Cumulative total number of sessions established since interface is up
hello recv	Number of hello messages received
hello sent	Number of hello messages sent
hello rej	Number of hello messages rejected
adj setup	Number of adjacencies set up
adj deleted	Number of adjacencies deleted

Related Documentation

- *show mpls interface*

Monitoring MPLS Minor Interfaces

Purpose Display status and configuration information about MPLS minor interfaces.

The **show mpls interface minor** command displays the same information.

Action To display information about MPLS minor interfaces:

```
host1:pe1#show mpls minor-interface
MPLS minor interface pe1-to-pe2 (transmit)
  Stacked on MPLS major ATM2/0.10
  Operational state is up
  Sent:
    0 packets
    0 bytes

  queue 0: traffic class best-effort, bound to atm-vc ATM2/0.10
    Queue length 0 bytes
    Forwarded packets 0, bytes 0
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0

MPLS minor interface lsp-02020202-1-4 (receive)
  Stacked on MPLS major ATM2/0.10
  Operational state is up
  Statistics not enabled for this interface
```


To display detailed information about MPLS minor interfaces:

```
host1:pe1#show mpls minor-interface detail
MPLS minor interface pe1-to-pe2 (transmit)
  Stacked on MPLS major ATM2/0.10
  Operational state is up
  MPLS minor interface UID is 0x1a000001
  Lower MPLS major interface UID is 0x19000001
  Sent:
    0 packets
    0 bytes

  queue 0: traffic class best-effort, bound to atm-vc ATM2/0.10
    Queue length 0 bytes
    Forwarded packets 0, bytes 0
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0

MPLS minor interface lsp-02020202-1-4 (receive)
  Stacked on MPLS major ATM2/0.10
  Operational state is up
  MPLS minor interface UID is 0x1a000004
  Lower MPLS major interface UID is 0x19000001
  Statistics not enabled for this interface
```

To display summary information about MPLS minor interfaces:

```
host1:pe1#show mpls minor-interface brief
                     Lower      Oper
   Interface        MplsMajor   state   Direction
----------------    ---------   -----   ---------
pe1-to-pe2          ATM2/0.10   up      transmit
lsp-02020202-1-4    ATM2/0.10   up      receive
ERX-01-0c-d7:pe1#
```

Meaning Table 78 on page 361 lists the **show mpls minor-interface** command output fields.

Table 78: show mpls minor-interface Output Fields

Field Name	Field Description
Interface	Specifier and status of each interface

Related Documentation

- *show mpls minor-interface*

Monitoring MPLS Next Hops

Purpose Display MPLS next hops and any available next-hop statistics.

Next hops can be pointed to by MPLS forwarding entries on an LSR, IP or IPv6 routes on an LER, and VPLS bridge groups.

Action To display MPLS next hops:

```
host1:vr2#show mpls next-hop
MPLS next-hop: index 1, lookup on inner header/label
   Statistics are not collected for MPLS switch-context next-hops

MPLS next-hop: index 2, lookup in router pe1
   Statistics are not collected for MPLS switch-context next-hops

MPLS next-hop: index 22, ECMP next-hop, leg count 2
   MPLS next-hop: index 20, label 36 on FastEthernet1/1.120, neighbor 10.120.120.1

   MPLS next-hop: index 21, label 36 on ATM2/1.20, neighbor 10.20.20.1
   Statistics are not collected for MPLS ECMP next-hops

MPLS next-hop: index 24, label 17, resolved by MPLS nextHop index 10
   MPLS next-hop: index 10, resolved by MPLS nextHop index 14, peer address
10.1.1.1
      MPLS next-hop: index 14, ECMP next-hop, leg count 2
         MPLS next-hop: index 12, label 32 on FastEthernet1/1.120, neighbor
10.120.120.1
         MPLS next-hop: index 13, label 32 on ATM2/1.20, neighbor 10.20.20.1
   Sent:
     0 packets
     0 bytes
     0 errors
     0 discards

MPLS next-hop: index 25, label 18, resolved by MPLS nextHop index 10
   MPLS next-hop: index 10, resolved by MPLS nextHop index 14, peer address
10.1.1.1
      MPLS next-hop: index 14, ECMP next-hop, leg count 2
         MPLS next-hop: index 12, label 32 on FastEthernet1/1.120, neighbor
10.120.120.1
         MPLS next-hop: index 13, label 32 on ATM2/1.20, neighbor 10.20.20.1
   Sent:
     0 packets
     0 bytes
     0 errors
     0 discards
```

When one MPLS next-hop points to another MPLS next hop, the second next hop is displayed indented to the first one.

Meaning Table 79 on page 362 lists the **show mpls next-hop** command output fields.

Table 79: show mpls next-hop Output Fields

Field Name	Field Description
index	Next-hop index
label	Label for next hop

Related Documentation

- *show mpls next-hop*

Monitoring the Configured Mapping between PHB IDs and Traffic Class/Color Combinations

Purpose Display the configured mapping between PHB IDs and traffic class/color combinations.

PHB IDs used for L-LSPs do not have color.

Action To display the mapping between PHB IDs and traffic class/color combinations:

```
host1#show mpls phb-id
Mpls PHB-ID traffic-class/color mappings:

--------   ------   --   -------------   -----------   -----   ------
standard   phb-id   0    traffic-class   best-effort   color   green
private    phb-id   0    traffic-class   best-effort   color   yellow
private    phb-id   1    traffic-class   1             color   red
private    phb-id   2    traffic-class   2             color   green
standard   phb-id   1    traffic-class   1             color   yellow
standard   phb-id   2    traffic-class   2             color   red
standard   phb-id   3    traffic-class   3             color   green
standard   phb-id   4    traffic-class   4             color   green
standard   phb-id   6    traffic-class   6             color   n/a
standard   phb-id   7    traffic-class   7             color   n/a
standard   phb-id   10   traffic-class   5             color   n/a
```

Meaning Table 80 on page 363 lists the **show mpls phb-id** command output fields.

Table 80: show mpls phb-id Output Fields

Field Name	Field Description
phb-id	Per-hop behavior ID for which a traffic class/color combination is displayed
traffic-class	Traffic class associated with traffic
color	Color (drop precedence) associated with traffic, green, yellow, or red

Related Documentation

- *show mpls phb-id*

Monitoring RSVP-TE Profiles and MPLS Tunnel Profiles

Purpose Display a specific RSVP-TE or tunnel profile, or all RSVP-TE or tunnel profiles.

Action To display the default RSVP-TE profile:

```
host1:pe2#show mpls rsvp profile default
RSVP profile default: used by 0 interfaces
    refresh period: 30000 ms
    timeout factor: 3
```

To display all MPLS tunnel profiles:

```
host1#show mpls tunnels profile
MPLS Tunnel Profile tunnelProfile
  LSP setup using rsvp-te
  tunnel not announced to any IGP
  (Global) Retry forever
       at (Global) interval 5 during Lsp setup if there is route
  (Global) Retry forever
       at (Global) interval 5 during Lsp setup if there is no route
  metric is relative 0
  path option 2
    path to be dynamically calculated by isis
  destinations include: 1.1.1.1 2.2.2.2 3.3.3.3
    ISIS Level 2 routers
    OSPF border routers
```

Meaning Table 81 on page 364 lists the **show mpls profile** command output fields.

Table 81: show mpls profile Output Fields

Field Name	Field Description
profile	Number of interfaces that use the profile
refresh-period	Timeout period in seconds between generation of refresh messages
timeout factor	Number of refresh messages that can be lost before the session is ended

Related Documentation

- *show mpls profile*

Monitoring RSVP Path State Control Blocks, Reservation State Control Blocks, or Sessions

Purpose Display RSVP path state control blocks, reservation state control blocks, or session information about the virtual router to assist in debugging. A session can be any of the following:

- ingress session—Originating on the router
- egress session—Terminating on the router
- transit session—Travelling through the router

Action To display path state control blocks for an ingress session:

```
host1#show mpls rsvp psb
  PSB: Sender 223.10.1.1 LSPId 1 timeout -- InLabel --
       PHopIntf
       IncomingIntf
       OutgoingIntf ATM2/0.1
       PHopAddr 0.0.0.0 m_ipNextHopAddr 221.1.1.1
       NextHop 221.1.1.1/255.255.255.255 (strict)
       LabelRange --
```

```
       SenderTSpec CType IntServ Controlled Load
                   Token Bucket Rate 0
                   Token Bucket Size 0
                   Peak Data Rate 0
                   Min Policed Unit 0
                   Max Packet Size 0
       Flags : InUse
               RroRequired
               PathRefreshSent
```

To display reservation state control blocks for an ingress session:

```
host1#show mpls rsvp rsb
  RSB: Timeout 157500 label 1/33
       Flags : InUse
               StyleConverted
```

To display RSVP-TE session information:

```
host1:two#show mpls rsvp sessions
Destination 222.9.3.1 TunnelId 1 Extended Tunnel Id 223.10.1.1
  PSB: Sender 223.10.1.1 LSPId 1 timeout 157500 InLabel 17
       Associated Minor Interface: Tunnel 223.10.1.1:1
       PHopIntf ATM2/0.1
       IncomingIntf ATM2/1.1
       OutgoingIntf ATM2/0.3
       PHopAddr 221.1.1.2 m_ipNextHopAddr 122.1.1.1
       NextHop 122.1.1.1/255.255.255.255 (strict)
       LabelRange (generic) min 0 max 1048575
       SenderTSpec CType IntServ Controlled Load
                   Token Bucket Rate 0
                   Token Bucket Size 0
                   Peak Data Rate 0
                   Min Policed Unit 0
                   Max Packet Size 0
       RRO IPv4 hop 221.1.1.2 (strict)
       ADSPEC --
       IN ERO IPv4 hop 221.1.1.1 (strict)
              IPv4 hop 122.1.1.1 (strict)
       OUT ERO IPv4 hop 122.1.1.1 (strict)
       SES ATTR Setup Pri 4, Hold Pri 4, name --
                Flags : IngressReRoute
       TTC --
       Policy Object --
       Unknown Objects --
       Flags : InUse
               PathRefreshSent

  RSB: Timeout 157500 label 16
       Associated Minor Interface: Tunnel 223.10.1.1:1
       FlowSpec CType IntServ Controlled Load
                Token Bucket Rate 0
                Token Bucket Size 0
                Peak Data Rate 0
                Min Policed Unit 0
                Max Packet Size 0
       RRO IPv4 hop 122.1.1.1 (strict)
       Policy Object --
       Unknown Objects --
       Flags : InUse
               StyleConverted
               ResvRefrSent
```

Meaning Table 82 on page 366 lists the **show mpls rsvp** command output fields.

Table 82: show mpls rsvp Output Fields

Field Name	Field Description
PSB	Path state control block
RSB	Reservation state control block
Sender	IP address of PSB or RSB sender
LSPId	ID of LSP
timeout	Period of time in milliseconds before PSB/RSB times out if no refresh arrives.
InLabel	Incoming label information
Associated tunnel	Tunnel identifier for minor interface for which the RSVP information is displayed
PHopIntf	Penultimate hop interface
IncomingIntf	Incoming interface
OutgoingIntf	Outgoing interface
PHopAddr	Penultimate hop address
m_ipNextHopAddr	Next hop address
NextHop	Type of next hop (loose, strict, session)
LabelRange	RSVP session label range
SenderTSpec	Traffic parameters for the sender
Token Bucket Rate	Sender's description of generated traffic, in kbps
Token Bucket Size	Sender's description of generated traffic, in kbps
Peak Data Rate	Lender's peak traffic generation rate
Min Policed Unit	Minimum packet size generated by sender
Max Packet Size	Maximum packet size generated by sender
RRO	Record route object
ADSPEC	Indicates presence of this QoS object

Table 82: show mpls rsvp Output Fields *(continued)*

Field Name	Field Description
IN ERO	Incoming explicit route object
OUT ERO	Outgoing explicit route object
SES ATTR	RSVP session attributes
Setup Pri	Setup priority of tunnel
Hold Pri	Hold priority of tunnel
name	Name of the tunnel
Flags	One or more of the IngressReRoute (the ingress router can reroute the LSP), Local Protection (routers can use local repair mechanism to fix the LSP; this fix might violate the explicit route object associated with the LSP), and MergingPermitted (LSPs can be merged) flags
TTC	Indicates presence of the traffic trunk classifier object
Policy Object	Indicates presence of the policy object
Unknown Objects	Indicates presence objects not defined by the RSVP specification
PSB Flag InUse	PSB in use
PSB Flag Deleted	PSB deleted
PSB Flag NonRsvp	Non-RSVP hop present
PSB Flag RouteChangeNotify	Route change notification received
PSB Flag EroChanged	Explicit route object changed
PSB Flag NextHopChanged	Next hop has changed
PSB Flag RtNextHopChanged	Routing table next hop changed
PSB Flag EgressStatusChanged	PSB egress status has changed
PSB Flag QosChanged	QoS characteristics have changed
PSB Flag LabelChanged	Label has changed

Table 82: show mpls rsvp Output Fields *(continued)*

Field Name	Field Description
PSB Flag ResvRefreshNeeded	Reservation refresh needed
PSB Flag PathRefreshNeeded	Path refresh needed
PSB Flag RroRequired	Record route object required
PSB Flag Egress	Session is egress
PSB Flag PathRefreshSent	Path refresh sent
PSB Flag EgressFilterFF	Egress filter reservation style Fixed Filter
PSB Flag QosCorrectionNeeded	QoS correction needed
PSB Flag IsPathTrigger	Has path refresh been triggered
RSB Flag InUse	RSB in use
RSB Flag Deleted	RSB deleted
RSB Flag RcvdAck	Acknowledgment received
RSB Flag StyleConverted	Reservation style converted to shared explicit
RSB Flag IsPathTrigger	Reservation refresh triggered
Destination	RSVP session destination address
TunnelId	Number representing the RSVP session tunnel ID
Extended Tunnel Id	IP address representing the RSVP session extended tunnel ID

Related Documentation

- *show mpls rsvp*

Monitoring RSVP MD5 Authentication

Purpose Display information about RSVP MD5 authentication.

Action To display information about RSVP MD5 authentication:

```
host1#show mpls rsvp authentication
Mpls interface FastEthernet2/4
  RSVP Authentication Secure Association with peer 10.2.2.2
    Receive Sequence Number 4592798942692985943
```



```
  RSVP Authentication Secure Association with peer 10.3.3.3
    Receive Sequence Number 4592798942692912623

Mpls interface ATM6/0.2
  RSVP Authentication Secure Association with peer 102.2.2.2
    Receive Sequence Number 4592798942692985934

Mpls interface ATM6/0.3
  RSVP Authentication Secure Association with peer 10.2.2.2
    Receive Sequence Number 4592798942692985956
```

Meaning Table 83 on page 369 lists the **show mpls rsvp authentication** command output fields.

Table 83: show mpls rsvp authentication Output Fields

Field Name	Field Description
RSVP Authentication Secure Association with peer	IP address of a peer with which the router has a security association
Receive Sequence Number	Sequence number of first authenticated packet from peer; subsequent packets from the peer must be greater than this base number

Related Documentation

- *show mpls rsvp authentication*

Monitoring RSVP-TE Interfaces Where BFD is Enabled

Purpose Display information about RSVP-TE major interfaces on which BFD is enabled.

For point-to-point interfaces, this command displays a single secure association with the single peer at the remote end. For multiaccess type interfaces (Ethernet), this command displays, if present, multiple secure associations from the various RSVP speakers.

Action To display information about RSVP-TE major interfaces on which BFD is enabled:

```
host1#show mpls rsvp bfd interfaces
                 Bfd Enabled RSVP interfaces
                 ---------------------------
            Minimum       Minimum       Minimum
Interface   Interval   Rx-Interval   Tx-Interval   Multiplier
---------   --------   -----------   -----------   ----------
ATM2/0.1    300        300           300           3
```

Meaning Table 84 on page 369 lists the **show mpls rsvp bfd interfaces** command output fields.

Table 84: show mpls rsvp bfd interfaces Output Fields

Field Name	Field Description
Interface	RSVP-TE major interface on which BFD is enabled

Table 84: show mpls rsvp bfd interfaces Output Fields *(continued)*

Field Name	Field Description
Minimum Interval	Minimum interval in milliseconds, used when the minimum receive interval and minimum transmit intervals have the same value
Minimum Rx-Interval	Minimum receive interval in milliseconds; minimum interval at which the local peer must receive BFD control packets from the remote peer
Minimum Tx-Interval	Minimum transmit interval in milliseconds; interval at which the local peer proposes to transmit BFD control packets to the remote peer
Multiplier	Detection multiplier value; roughly equivalent to the number of packets that can be missed before the BFD session is declared to be down

Related Documentation

- *show mpls rsvp bfd interfaces*

Monitoring RSVP-TE Interface Counters

Purpose Display various counters for a particular RSVP-TE interface or all RSVP-TE interfaces.

Action To display counters for RSVP-TE interface atm 6/0.1:

```
host1#show mpls rsvp counters atm 6/0.1
Interface ATM6/0.1
  Path Sent           247          Path Rcvd           0
  Path Error Sent     0            Path Error Rcvd     0
  Path Tear Sent      0            Path Tear Rcvd      0
  Resv Sent           0            Resv Rcvd           245
  Resv Error Sent     0            Resv Error Rcvd     0
  Resv Tear Sent      0            Resv Tear Rcvd      0
  Resv Conf Sent      0            Resv Conf Rcvd      0
  SRefresh Sent       0            SRefresh Rcvd       0
  Ack Sent            0            Ack Rcvd            0
  Nack Objects Sent   0            Nack Objects Rcvd   0
  Msg Bundles Sent    0            Msg Bundles Rcvd    0
  Error Msgs Rcvd     0            Misordered Messages 0
  Send Failures       0            Msgs not acked      0
  Path Triggers       1            Resv Triggers       0
  Forwarded Pkts      0
  Hello Sent          7097         Hello Rcvd          7097
  Hello Ack Sent      0            Hello Ack Rcvd      7097
  Hello Discarded     0            Hello Ack Discarded 0
  Hello Suppressed    0
```

Meaning Table 85 on page 371 lists the **show mpls rsvp counters** command output fields.

Table 85: show mpls rsvp counters Output Fields

Field Name	Field Description
Path Sent	Number of path messages sent on the interface
Path Rcvd	Number of path messages received on the interface
Path Error Sent	Number of patherror messages sent on the interface
Path Error Rcvd	Number of patherror messages received on the interface
Path Tear Sent	Number of pathtear messages sent on the interface
Path Tear Rcvd	Number of pathtear messages received on the interface
Resv Sent	Number of resv messages sent on the interface
Resv Rcvd	Number of resv messages received on the interface
Resv Error Sent	Number of resverr messages sent on the interface
Resv Error Rcvd	Number of resverr messages received on the interface
Resv Tear Sent	Number of resvtear messages sent on the interface
Resv Tear Rcvd	Number of resvtear messages received on the interface
Resv Conf Sent	Number of resvconf messages sent on the interface
Resv Conf Rcvd	Number of resvconf messages received on the interface
Srefresh Conf Sent	Number of srefresh messages sent on the interface
Srefresh Conf Rcvd	Number of srefresh messages received on the interface
Ack Conf Sent	Number of resvconf messages sent on the interface
Ack Conf Rcvd	Number of resvconf messages received on the interface
Nack Objects Sent	Number of nack objects sent on the interface
Nack Objects Rcvd	Number of nack objects received on the interface
Msg Bundles Objects Sent	Number of message bundles sent on the interface
Msg Bundles Rcvd	Number of message bundles received on the interface
Error Msgs Rcvd	Number of error messages received on the interface
Misordered Messages	Number of misordered messages received on the interface

Table 85: show mpls rsvp counters Output Fields *(continued)*

Field Name	Field Description
Send Failures	Number of failures to successfully send messages on the interface
Path Triggers	Number of locally triggered path messages
Resv Triggers	Number of locally triggered resv messages
Forwarded Pkts	RSVP control packets that are forwarded through the router
Hello Sent	Number of hello messages sent
Hello Rcvd	Number of hello messages received
Hello Ack Sent	Number of acknowledgments sent in response to hello requests received
Hello Ack Rcvd	Number of acknowledgments received in response to hello requests sent
Hello Discarded	Number of hello messages discarded
Hello Ack Discarded	Number of hello ack messages discarded
Hello Suppressed	Number of hello messages suppressed; message generation is suppressed when a hello request object is received from the destination node within the hello interval

Related Documentation

- *show mpls rsvp counters*

Monitoring RSVP-TE Graceful Restart

Purpose Display information about the state of RSVP-TE graceful restart.

Action To display information about RSVP-TE graceful restart:

```
host1#show mpls rsvp hello graceful restart
Graceful restart is ON
    Warning: Graceful restart is NOT active
    Warning: Hellos not configured on all interfaces
    Restart time 60000 milliseconds
    Recovery time 120000 milliseconds
```

Meaning Table 86 on page 373 lists the **show mpls rsvp hello graceful restart** command output fields.

Table 86: show mpls rsvp hello graceful restart Output Fields

Field Name	Field Description
Graceful-restart	State of graceful restart, ON or Off
Warning	State of graceful restart attributes
Restart time	Graceful restart time, in milliseconds
Recovery time	Graceful restart recovery time, in milliseconds

Related Documentation

- *show mpls rsvp hello graceful restart*

Monitoring RSVP-TE Hello Adjacency Instances

Purpose Display summary or detailed information about RSVP-TE hello adjacency instances.

Action To display summary information about RSVP-TE hello adjacency instances:

```
host1#show mpls rsvp hello instance
  Up - neighbor is up
  GR - graceful restart is in progress
  Peer
Address     Interface    Interval    Miss Limit  State
--------    ---------    --------    ----------  ------
10.1.1.2    ATM6/1.1     10000       4           Up
10.3.1.2    ATM6/0.3     10000       4           GR
```

The following output shows that the two peers are identified as RSVP-TE node hellos peers:

```
host1#show mpls rsvp hello instance
  Up - neighbor is up
  GR - graceful restart is in progress
  Peer
Address Interface Interval Miss Limit State
-------- --------- -------- ---------- ------
10.1.1.2 <any>     10000     4          Up
10.3.1.2 <any>     10000     4          GR
11.2.3.1 Atm3/1.3 10000     4          GR
```

To display detailed information about RSVP-TE hello adjacency instances:

```
host1#show mpls rsvp hello instance detail
Neighbor 10.1.1.2 on interface ATM6/1.1
  Local Address 10.1.1.1
  Restart Time: 60000 msecs
  Recovery Time: 120000 msecs
  State: Up
  SrcInstance 0x4379F084
  DstInstance 0x4379EA19
  Interval            : 10000
  Miss Limit          : 4
  Hellos Sent         : 0
```

```
Hellos Received     : 382
Hellos Suppressed   : 381
Hello Acks Sent     : 382
Hello Acks Received : 0
```

Meaning Table 87 on page 374 lists the **show mpls rsvp hello instance** command output fields.

Table 87: show mpls rsvp hello instance Output Fields

Field Name	Field Description
Peer Address	Address of the peer in the RSVP-TE hello adjacency
Interface	Specifier and status of each interface
any	Identifies an RSVP-TE node hello peer
Interval	Interval at which hellos are sent to the neighbor, in milliseconds
Miss Limit	Number of hello messages from the neighbor that can be missed before the adjacency is considered to be down
State	State of the adjacency with the neighbor
AdjLost	Adjacency has been lost. Hellos were received from the peer but have timed out. The peer is known to be capable of graceful restart, so the router is waiting for hellos to resume from the peer. The router is actively sending hellos to the peer. The router declares the peer to be dead if it does not receive hellos from the peer during the peer's advertised recovery period. The router declares the peer to be gracefully restarting if hellos are seen from the peer and its sequence number has changed. The router declares the peer to be up if hellos are seen from the peer and its sequence number has not changed.
Dead	Hellos were received from the peer but have timed out. The router is not in graceful restart helper mode. The router changes the local hello sequence number and does not send hellos to the peer. The router transitions to Down if new control traffic needs to be sent to the peer or if the peer starts sending control traffic.
Down	No hellos have been received from the peer. The router is actively sending hellos to the peer.
GR	Graceful restart in progress. The hello sequence number from the peer changed. The router is in graceful restart helper mode and actively sending hellos to the peer.
Up	Hellos were received from the peer. The router is actively sending hellos to the peer.
Neighbor	IP address of remote hello adjacency peer

Table 87: show mpls rsvp hello instance Output Fields *(continued)*

Field Name	Field Description
Local Address	IP address of the local hello adjacency peer
Restart Time	Graceful restart time, in milliseconds
Recovery Time	Graceful restart recovery time, in milliseconds
SrcInstance	Nonzero 32-bit value that represents the sender's hello instance. The value is maintained on a per-neighbor basis. This instance value changes only when the sending peer resets, when the sender's router reboots, or when communication is lost between the hello adjacency peers.
DstInstance	32-bit SrcInstance value most recently received from hello adjacency peer; value is zero when no instance has been received from that peer
Hellos Sent	Number of hello messages sent
Hellos Received	Number of hello messages received
Hellos Suppressed	Number of hello messages suppressed; message generation is suppressed when a hello request object is received from the destination node within the hello interval
Hellos Acks Sent	Number of acknowledgments sent in response to hello requests received
Hellos Acks Received	Number of acknowledgments received in response to hello requests sent

Related Documentation

- *show mpls rsvp hello instance*

Monitoring Status and Configuration for MPLS Tunnels

Purpose Display status and configuration for all tunnels or for a specific tunnel in the current router context.

Action To display the status and configuration for all tunnels:

host1#**show mpls tunnels**

```
Tunnel 1 to   0.0.0.0
  State:  Enabled with Incomplete Config
  tunnel not announced to any IGP
  (Global) Retry forever
       at (Global) interval 5 during Lsp setup if there is route
  (Global) Retry forever
       at (Global) interval 5 during Lsp setup if there is no route
  metric is relative 0
```

```
Tunnel 2 to   222.9.1.3
  State:  Up
  Out label 24 on ATM3/0.1 nbr 10.10.11.5
    14 pkts, 0 hcPkts, 2156 octets
    0 hcOctets, 0 errors, 0 discardPkts
  tunnel not announced to any IGP
  (Global) Retry forever
       at (Global) interval 5 during Lsp setup if there is route
  (Global) Retry forever
       at (Global) interval 5 during Lsp setup if there is no route
  metric is relative 0
phb-id 2
path option 2
    option is currently used - path is calculated  by isis
        10.10.11.5
        10.10.12.3
        222.9.1.3
    next reoptimization in 1687 seconds
  stacked labels:
FastEthernet2/4.1     222.9.1.3          R0        Out 18 on  tun mpls:1

Tunnel tail-de090106-1-18a for  222.9.1.2
  State:  Up
  In  label 20 on ATM3/0.1
    0 pkts, 0 hcPkts, 0 octets
0 hcOctets, 0 errors, 0 discardPkts
```

In the output for Tunnel 2 shown here, the line **phb-id 2** indicates that the tunnel is an L-LSP with PHB-ID 2. You can then display the output of the **show mpls phb-id** command to determine the corresponding PSC.

A result of Incomplete Configuration in the display indicates either no tunnel endpoint or no label distribution protocol.

To display a summary of all MPLS tunnels for the current router context:

```
host1:pe2#show mpls tunnels brief
name/id          destination  metric  state/label/intf
vpnEgressLabel3  0.0.0.0      R0      Incoming 1048573 on stack
vpnEgressLabel4  0.0.0.0      R0      Incoming 1048572 on stack
pe2-to-pe1       1.1.1.1      R0      Outgoing 300 on atm2/0.60
                 2.2.2.2      R0      Incoming 3000 on atm2/0.70
```

Meaning Table 88 on page 376 lists the **show mpls tunnels** command output fields.

Table 88: show mpls tunnels Output Fields

Field Name	Field Description
Label	Label prepended to packets before being sent across tunnel
State	Status of tunnel: Establishing, Traffic Engineering Negotiation, Up, Down, Enabled with Incomplete Config, Disabled with Incomplete Config, Disabled, or Releasing
on	Location of tunnel

Table 88: show mpls tunnels Output Fields *(continued)*

Field Name	Field Description
tunnel is announced to	Protocols to which the tunnel is announced
metric	Metric type, relative or absolute
phb-id	PHB ID supported by this tunnel; for E-LSPs an additional exp-bits entry is displayed after the phb-id entry
pkts	Number of packets sent across tunnel
hcPkts	Number of high-capacity (64-bit) packets sent across tunnel
octets	Number of octets sent across tunnel
hcOctets	Number of high-capacity (64-bit) octets sent across tunnel
errors	Number of packets that are dropped for some reason before being sent
discardPkts	Number of packets that are discarded due to lack of buffer space before being sent
name/id	Tunnel identifier
destination	Tunnel destination; router ID of egress router
metric	Value of tunnel metric and whether the metric is relative (R) or absolute (A)
state/label/intf	Functional state of tunnel, label for the tunnel, and interface where tunnel resides

Related Documentation

- *show mpls tunnels*

Verifying and Troubleshooting MPLS Connectivity

In IP networks, you can use the **ping** and **traceroute** commands to verify network connectivity and find broken links or loops. In an MPLS-enabled network, you can use the **mpls ping** and **trace mpls** commands to detect plane failures in different types of MPLS applications and network topologies.

Tasks to verify and troubleshoot connectivity in IP and MPLS-enabled networks are:

- Sending an MPLS Echo Request Packet to an IP or IPv6 Address on page 378
- Tracing the Path of an MPLS Echo Request Packet to an IP or IPv6 Address on page 378
- Sending an MPLS Echo Request Packet to a Martini Circuit on page 378

Sending an MPLS Echo Request Packet to an IP or IPv6 Address

To send an MPLS echo request packet to the specified IP or IPv6 address:

- Issue the **ping mpls** command.

  host1:pe1#**ping mpls ip 10.2.2.2/32**

Tracing the Path of an MPLS Echo Request Packet to an IP or IPv6 Address

To send MPLS echo request packets to discover and examine the path MPLS packets follow to the specified IP or IPv6 address:

- Issue the **trace mpls ip** command.

  host1:pe1#**trace mpls ip 192.168.25.1/32**

Sending an MPLS Echo Request Packet to a Martini Circuit

To send an MPLS echo request packet to the specified layer 2 cross-connect virtual (Martini) circuit:

- Issue the **ping mpls l2transport** command.

  host1:pe1#**ping mpls l2transport FastEthernet1/0.1 detail**

Tracing the Path of an MPLS Echo Request Packet to a Martini Circuit

To send MPLS echo request packets to discover and examine the path MPLS packets follow to the specified layer 2 cross-connect virtual (Martini pseudowire) circuit:

- Issue the **trace mpls l2transport** command.

  host1:pe1#**trace mpls l2transport FastEthernet1/0.1 detail**

Sending an MPLS Echo Request Packet to an L3VPN IP or IPv6 Prefix

To send an MPLS echo request packet to the specified L3VPN IP or IPv6 prefix:

- Issue the **ping mpls l3vpn** command.:

  host1:pe1#**ping mpls l3vpn vrf pe11 10.32.45.21/32**

Tracing the Path of an MPLS Echo Request Packet to an L3VPN IP or IPv6 Prefix

To send MPLS echo request packets to discover and examine the path MPLS packets follow to the L3VPN IP or IPv6 prefix:

- Issue the **trace mpls l3vpn** command.

  host1:pe1#**trace mpls l3vpn vrf pe11 10.2.3.21/32**

Sending an MPLS Echo Request Packet to an RSVP-TE Tunnel

To send an MPLS echo request packet to the specified RSVP-TE tunnel:

- Issue the **ping mpls rsvp** command.

  host1:pe1#**ping mpls rsvp tunnel west1**

Tracing the Path of an MPLS Echo Request Packet to an RSVP-TE Tunnel

To send MPLS echo request packets to discover and examine the path MPLS packets follow to the specified RSVP-TE tunnel:

- Issue the **trace mpls rsvp tunnel** command.

  host1:pe1:pe11#**trace mpls rsvp tunnel west1 detail**

Sending an MPLS Echo Request Packet to a VPLS Instance

To send an MPLS echo request packet to the specified VPLS instance:

- Issue the **ping mpls vpls** command.

  host1:pe1#**ping mpls vpls vrf pe11 vplsA remote-site-id 2**

Tracing the Path of an MPLS Echo Request Packet to a VPLS Instance

To send MPLS echo request packets to discover and examine the path MPLS packets follow to the specified VPLS instance:

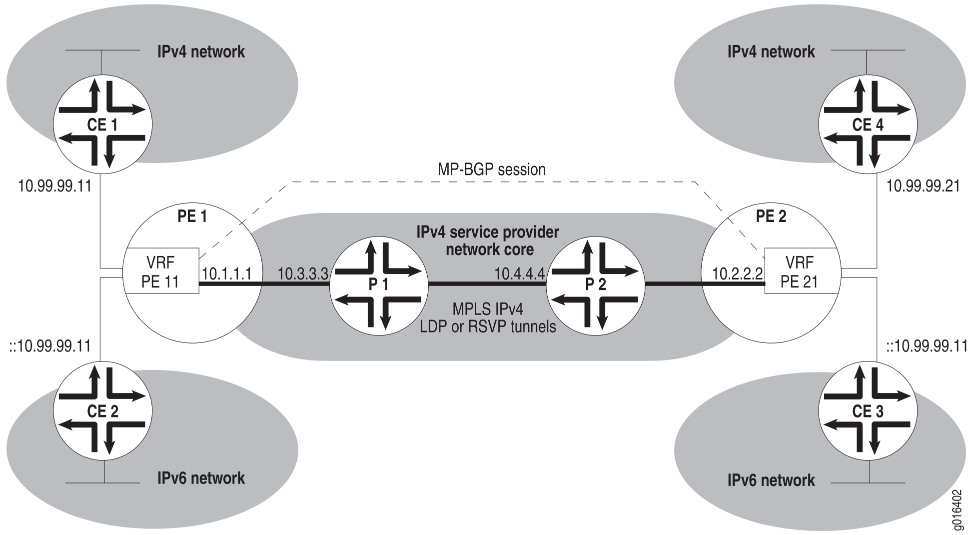
- Issue the **trace mpls vpls** command.

  host1:pe1#**trace mpls vpls vrf pe11 vplsA remote-site-id 2 detail**

Packet Flow Examples for Verifying MPLS Connectivity

Figure 67 on page 380 shows a sample IPv4/IPv6 L3VPN topology with LDP or RSVP-TE base tunnels. Two base tunnels (one in each direction) are present between 10.1.1.1 and 10.2.2.2. The packet flow examples that follow refer to this sample topology.

Figure 67: Sample MPLS L3VPN Topology



Packet Flow Examples for MPLS LSPs to an IP Prefix

Use the **ping mpls ip** and **trace mpls ip** commands for MPLS LSPs that are configured to use LDP; labeled BGP; or a combination of LDP, BGP, and RSVP-TE (as for inter-AS and carrier-of-carriers topologies). When you specify a VRF name, the LSP to the specified prefix must originate from the VRF because the ping is generated from the specified VRF.

Packet Flow Example for the ping mpls Command

The following example illustrates the packet flow that results when you issue the **ping mpls ip** command from router PE 1 (10.1.1.1) to router PE 2 (10.2.2.2) over an LDP base tunnel.

```
host1:pe1#ping mpls ip 10.2.2.2/32
```


1. PE 1 sends an MPLS echo request UDP packet that contains an LDP IPv4 sub-TLV. The packet is sent as a labeled packet over the target LSP. The packet has the following attributes:

Source address	10.1.1.1
Destination address	127.0.0.0/8
UDP port	3503
TTL	255
IPv4 prefix in the TLV	10.2.2.2/32
Sender's handle	Randomly generated 32-bit number used to match the reply
Sequence number	Integer that is incremented for each echo request packet

2. Router P 1 label-switches the packet to P 2.
3. Router P 2 label-switches the packet to PE 2 (assuming PHP is not configured).
4. Router PE 2 pops the label and determines that the destination address is in the 127.0.0.0/8 subnet. PE 2 sends the packet up to the control plane. The MPLS ping application on the control plane then creates an MPLS echo reply to the received echo request. The echo reply packet has a return code of 3, which means that the replying router is an egress for the FEC at stack depth. The echo reply packet includes the Interface and Label Stack TLV to indicate both the interface on which the request packet was received and the incoming label stack. The MPLS echo reply packet is sent back as a (labeled) UDP packet with the following attributes:

Source address	10.2.2.2
Destination address	10.1.1.1
UDP port	3503

5. When the MPLS echo reply reaches router PE 1, the router matches the sender's handle and the sequence number to the echo request packet that PE 1 sent out. If the values match, the CLI displays an exclamation point (!).

The following sample output represents what you might see when you issue the **ping mpls ip** and **ping mpls ip detail** commands for the topology shown in Figure 67 on page 380.

```
host1:pe1#ping mpls ip 10.2.2.2/32
Sending 5 UDP echo requests for LDP IPv4 prefix, timeout = 2 sec
!!!!!
Success rate = 100% (5/5), round-trip min/avg/max = 4294967295/4/0 ms
```

```
host1:pe1#ping mpls ip 10.2.2.2/32 detail
Sending 5 UDP echo requests for LDP IPv4 prefix, timeout = 2 sec
  MplsNextHopIndex 32 handle 8073311
    '!' - success, 'Q' - request not transmitted,
```



```
    '.' - timeout, 'U' - unreachable,
    'R' - downstream router but not destination
    'M' - malformed request, 'N' - downstream router has no mapping

Sending MPLS ping echo request, handle 8073311 seq 21241
! 10.2.2.2 Replying router is an egress for the FEC at stack depth/0 seq 21241
Sending MPLS ping echo request, handle 8073311 seq 21242
! 10.2.2.2 Replying router is an egress for the FEC at stack depth/0 seq 21242
Sending MPLS ping echo request, handle 8073311 seq 21243
! 10.2.2.2 Replying router is an egress for the FEC at stack depth/0 seq 21243
Sending MPLS ping echo request, handle 8073311 seq 21244
! 10.2.2.2 Replying router is an egress for the FEC at stack depth/0 seq 21244
Sending MPLS ping echo request, handle 8073311 seq 21245
! 10.2.2.2 Replying router is an egress for the FEC at stack depth/0 seq 21245

Success rate = 100% (5/5), round-trip min/avg/max = 4/4/0 ms
```

Packet Flow Example for the trace mpls Command

The following example illustrates the packet flow that results when you issue the **trace mpls ip** command from router PE 1 (10.1.1.1) to router PE 2 (10.2.2.2) over an LDP base tunnel.

host1:pe1#**trace mpls ip 10.2.2.2/32**

1. PE 1 sends an MPLS echo request UDP packet that contains an LDP IPv4 sub-TLV and a Downstream Mapping TLV. The packet has the following attributes:

Source address	10.1.1.1
Destination address	127.0.0.0/8
UDP port	3503
TTL	1
IPv4 prefix in the TLV	10.2.2.2/32
Sender’s handle	Randomly generated 32-bit number used to match the reply
Sequence number	Integer that is incremented for each echo request packet

2. The TTL expires on router P 1. P 1 exceptions the packet up to the control plane. Router P 1 then creates an MPLS echo reply packet in reply to the received MPLS echo request. The MPLS echo reply packet has a return code of 8, which means that the packet would have been label-switched at the outermost label (label-stack depth 1). The Downstream Mapping TLV is set to indicate the path that the packet would have taken from the router. The Interface and Label Stack TLV is included in the echo reply packet. The MPLS echo reply packet is sent back as a labeled UDP packet with the following attributes:

Source address	10.3.3.3
Destination address	10.1.1.1

UDP port	3503

3. When the MPLS echo reply reaches router PE 1, the router matches the sender's handle and the sequence number to the echo request packet that PE 1 sent. The CLI displays the router ID of the router that sent the echo reply. The **detail** version of the command displays the downstream mapping TLV contained in the MPLS echo reply.
4. Steps 1–3 are repeated with a TTL of 2 and the destination address set to router P 2's router ID, 10.4.4.4.
5. Router PE 1 next sends an MPLS echo request with a TTL of 3. This packet's TTL expires on router PE 2. PE 2 exceptions the packet up to the control plane. The MPLS trace application on the control plane then creates an MPLS echo reply to the received echo request. The echo reply packet has a return code of 3, which means that the replying router is an egress for the FEC at stack depth. The echo reply packet includes the Interface and Label Stack TLV to indicate both the interface on which the request packet was received and the incoming label stack. The Downstream Mapping TLV is not included in the echo reply packet.
6. When PE 2's echo reply packet reaches router PE 1, the router matches PE 2's handle and the sequence number to the echo request packet that PE 1 sent. The CLI displays the router ID for PE 2, indicating that PE 2 is the target router.

The following sample output represents what you might see when you issue the **trace mpls ip** command for the topology shown in Figure 67 on page 380.

```
host1:pe2#trace mpls ip 10.1.1.1/32
Tracing LDP IPv4 prefix, timeout = 2 sec, Max TTL 32
  MplsNextHopIndex 60, handle 8073312

1 2ms 10.44.44.44 Label switched at stack-depth/1
2 1ms 10.33.33.33 Label switched at stack-depth/1
3 2ms 10.1.1.1 Replying router is an egress for the FEC at stack depth/0
```

Packet Flows for ping and trace to L3VPN IPv4 Prefixes

This example describes packet flow for an MPLS ping is sent from VRF PE 11 on router PE 1 to the IPv4 prefix 10.99.99.21/32. For validation at the remote end, the source address of the echo request packet must be the same as the update-source address of BGP peer.

host1:pe1#**ping mpls l3vpn vrf pe11 10.99.99.21/32**

1. An MPLS echo request packet containing a single VPN IPv4 sub-TLV is sent from PE 1 with the following attributes:

Source address	10.1.1.1
Destination address	127.0.0.0/8
UDP port	3503
TTL	255
Sender's handle	Randomly generated 32-bit number used to match the reply
Sequence number	Integer that is incremented for each echo request packet

   The VPN IPv4 sub-TLV has the route distinguisher set to that of the VRF and the IPv4 prefix set to 10.99.99.21/32. The packet exits PE 1 with two labels.

2. Router P 1 switches labels based on the outer label of the packet and forwards the packet to P 2.
3. Router P 2 switches labels based on the outer label of the packet and forwards the packet to PE 2.
4. Router PE 2 pops both labels and determines that the destination address is in the 127.0.0.0/8 subnet. PE 2 sends the packet up to the control plane. The MPLS ping application on the control plane then creates an MPLS echo reply to the received echo request. The echo reply packet has a return code of 3, which means that the replying router is an egress for the FEC at stack depth. The echo reply packet includes the Interface and Label Stack TLV to indicate both the interface on which the request packet was received and the incoming label stack. The MPLS echo reply packet is sent back as a (labeled) UDP packet with the following attributes:

Source address	10.2.2.2
Destination address	10.1.1.1
UDP port	3503

5. When the MPLS echo reply reaches router PE 1, the router matches the sender's handle and the sequence number to the echo request packet that PE 1 sent. The CLI displays an exclamation point (!).

Packet flow for an MPLS trace to an L3VPN IPv4 prefix is the same as for an IPv4 prefix except that the echo request packets and echo reply packets contain the VPN IPv4 sub-TLV instead of the LDP IPv4 sub-TLV. The following sample output represents what you might see when you issue the **trace mpls l3vpn** and **trace mpls l3vpn vrf** commands for the topology shown in Figure 67 on page 380.

```
host1:pe1:pe11#ip8:pe1#trace mpls l3vpn 10.99.99.21/32 detail
Tracing VPN IPv4 prefix, timeout = 2 sec, Max TTL 32
  MplsNextHopIndex 73 handle 8073322
```



```
1 0ms 10.33.33.33 Label switched at stack-depth/2
  TLV Interface and Label stack 20 bytes
    Router 10.33.33.33 Intf 10.10.10.2
    [L34 EXP 0 TTL 1] [L68 EXP 0 S TTL 1]
  TLV Downstream mapping 24 bytes
    Router 10.31.31.2 Intf 10.31.31.1 mtu 9180
    [L56 EXP 0 LDP] [L68 EXP 0 S Unknown]
  TLV Downstream mapping 24 bytes
    Router 10.34.34.2 Intf 10.34.34.1 mtu 1500
    [L79 EXP 0 LDP] [L68 EXP 0 S Unknown]
2 2ms 10.55.55.55 Label switched at stack-depth/2
  TLV Interface and Label stack 20 bytes
    Router 10.55.55.55 Intf 10.34.34.2
    [L79 EXP 0 TTL 1] [L68 EXP 0 S TTL 2]
  TLV Downstream mapping 24 bytes
    Router 10.120.120.2 Intf 10.120.120.1 mtu 1500
    [L43 EXP 0 LDP] [L68 EXP 0 S Unknown]
3 3ms 10.2.2.2 Replying router is an egress for the FEC at stack depth
  TLV Pad 20 bytes
  TLV Interface and Label stack 20 bytes
    Router 10.2.2.2 Intf 10.120.120.2
    [L43 EXP 0 TTL 1] [L68 EXP 0 S TTL 3]
```

host1:pe1#**trace mpls l3vpn vrf pe11 10.99.98.21/32 reply pad-tlv exp-bits 5 detail**

```
Tracing VPN IPv4 prefix, timeout = 2 sec, Max TTL 32
  Handle 1921136  MplsNextHopIndex 78 [L68,L34]

1 0ms 10.33.33.33 Label switched at stack-depth/2
  TLV Pad 20 bytes
  TLV Interface and Label stack 20 bytes
    Router 10.33.33.33 Intf 10.10.10.2
    [L34 EXP 5 TTL 1] [L68 EXP 0 S TTL 1]
  TLV Downstream mapping 24 bytes
    Router 10.31.31.2 Intf 10.31.31.1 mtu 9180
    [L56 EXP 5 LDP] [L68 EXP 0 S Unknown]
  TLV Downstream mapping 24 bytes
    Router 10.34.34.2 Intf 10.34.34.1 mtu 1500
    [L79 EXP 5 LDP] [L68 EXP 0 S Unknown]
2 2ms 10.55.55.55 Label switched at stack-depth/2
  TLV Pad 20 bytes
  TLV Interface and Label stack 20 bytes
    Router 10.55.55.55 Intf 10.34.34.2
    [L79 EXP 5 TTL 1] [L68 EXP 0 S TTL 2]
  TLV Downstream mapping 24 bytes
    Router 10.120.120.2 Intf 10.120.120.1 mtu 1500
    [L43 EXP 5 LDP] [L68 EXP 0 S Unknown]
3 3ms 10.2.2.2 Replying router is an egress for the FEC at stack depth
  TLV Pad 20 bytes
  TLV Interface and Label stack 20 bytes
    Router 10.2.2.2 Intf 10.120.120.2
    [L43 EXP 5 TTL 1] [L68 EXP 0 S TTL 3]
```

Inter-AS Topology

When an L3VPN ping or trace is transmitted, the TTL value on the inner (VPN) label is set to 1 by default. This value causes the TTL to expire on the egress PE of the L3VPN LSP and an echo reply can be sent back to the source. However, in an inter-AS topology, this behavior might result in premature termination of the ping or trace. You can use the **bottom-label-ttl** keyword to avoid this problem.

Packet Flows to L3VPN IPv6 Prefixes

Packet flow for an MPLS ping and trace to an L3VPN IPv6 prefix is the same as for an IPv4 prefix except that the echo request packets and echo reply packets contain the VPN IPv6 sub-TLV instead of the VPN IPv4 sub-TLV.

Related Documentation

- Verifying and Troubleshooting MPLS Connectivity on page 377
- *ping mpls ip*
- *ping mpls l2transport*
- *ping mpls l3vpn*
- *ping mpls rsvp tunnel*
- *ping mpls vpls*
- *trace mpls ip*
- *trace mpls l2transport*
- *trace mpls l3vpn*
- *trace mpls rsvp tunnel*
- *trace mpls vpls*

Troubleshooting MTU Problems in Point-to-Point LSPs

In an MPLS-enabled network with point-to-point LSPs, you can use the **data-size** keyword with the **ping mpls** command to detect MTU failures in different types of MPLS applications and network topologies. You can use the **data-size** keyword to troubleshoot MTU problems in point-to-point MPLS LSPs to determine whether MPLS packets with a particular size can be forwarded over an MPLS point-to-point LSP, when the size of the packets exceeds the MTU size at any of the LSRs that are nodes of the LSP. If you specify the packet size for MPLS echo requests, you can determine the exact LSR where the MTU size is exceeded and the MPLS packets are discarded.

Tasks to verify and troubleshoot MTU problems in point-to-point LSPs are:

- Troubleshooting MTU Problems in a Point-to-Point MPLS LSP Associated with an IP or IPv6 Address on page 387
- Troubleshooting MTU Problems in a Point-to-Point MPLS LSP Associated with an L3VPN IP or IPv6 Prefix on page 387
- Troubleshooting MTU Problems in a Point-to-Point MPLS LSP Associated with a Martini Circuit on page 387
- Troubleshooting MTU Problems in a Point-to-Point MPLS LSP Associated with an RSVP-TE Tunnel on page 387
- Troubleshooting MTU Problems in a Point-to-Point MPLS LSP Associated with a VPLS Instance on page 387

Troubleshooting MTU Problems in a Point-to-Point MPLS LSP Associated with an IP or IPv6 Address

To discover the LSR in a point-to-point MPLS LSP, associated with an IP or IPv6 prefix, that causes MPLS packets to be discarded owing to the size of the packet exceeding the MTU size:

- Issue the **trace mpls ip** command with the **data-size** keyword.

  host1:pe1#**trace mpls ip 192.168.25.1/32 data-size 60**

Troubleshooting MTU Problems in a Point-to-Point MPLS LSP Associated with an L3VPN IP or IPv6 Prefix

To discover the LSR in a point-to-point MPLS LSP, associated with an L3VPN IP or IPv6 prefix, that causes MPLS packets to be discarded owing to the size of the packet exceeding the MTU size:

- Issue the **trace mpls l3vpn** command with the **data-size** keyword.

  host1:pe1#**trace mpls l3vpn vrf pe11 10.2.3.21/32 data-size 60**

Troubleshooting MTU Problems in a Point-to-Point MPLS LSP Associated with a Martini Circuit

To discover the LSR in a point-to-point MPLS LSP, associated with a Martini circuit, that causes MPLS packets to be discarded owing to the size of the packet exceeding the MTU size:

- Issue the **trace mpls l2transport** command with the **data-size** keyword.

  host1:pe1#**trace mpls l2transport FastEthernet1/0.1 data-size 60**

Troubleshooting MTU Problems in a Point-to-Point MPLS LSP Associated with an RSVP-TE Tunnel

To discover the LSR in a point-to-point MPLS LSP, associated with an RSVP-TE tunnel, that causes MPLS packets to be discarded owing to the size of the packet exceeding the MTU size:

- Issue the **trace mpls rsvp tunnel** command with the **data-size** keyword.

  host1:pe1#**trace mpls rsvp tunnel west1 data-size 60**

Troubleshooting MTU Problems in a Point-to-Point MPLS LSP Associated with a VPLS Instance

To discover the LSR in a point-to-point MPLS LSP, associated with a VPLS instance, that causes MPLS packets to be discarded owing to the size of the packet exceeding the MTU size:

- Issue the **trace mpls vpls** command with the **data-size** keyword.

  host1:pe1#**trace mpls vpls vplsA sender-site-id 1 remote-site-id 2 data-size 60**

Related Documentation

- *ping mpls ip*
- *ping mpls l2transport*
- *ping mpls l3vpn*
- *ping mpls rsvp tunnel*
- *ping mpls vpls*
- *trace mpls ip*
- *trace mpls l2transport*
- *trace mpls l3vpn*
- *trace mpls rsvp tunnel*
- *trace mpls vpls*

CHAPTER 6

Configuring BGP-MPLS Applications

This chapter contains the following sections:

MBGP Overview

NOTE: Before you read this chapter, we recommend you be thoroughly familiar with both BGP and MPLS. For detailed information about those protocols, see "Configuring BGP Routing" on page 3 and "Configuring MPLS" on page 279.

The BGP multiprotocol extensions (MP-BGP) enable BGP to support IPv4 services such as BGP multicast and BGP/MPLS virtual private networks (VPNs). BGP/MPLS VPNs are sometimes known as RFC 2547bis VPNs. Some of the applications for which you might use BGP/MPLS VPNs are to transport packets across an IP backbone, enable overlapping VPNs, operate inter-AS VPNs, enable multicast across VPNs, and provide carrier-of-carriers VPNs.

Related Documentation

- Understanding MBGP Address Families on page 391
- Equal-Cost Multipath Support Overview on page 392
- BGP/MPLS VPN Components Overview on page 394
- BGP/MPLS VPN Platform Considerations on page 402
- MBGP References on page 402
- IPv6 VPN Overview on page 409

Understanding MBGP Address Families

The BGP multiprotocol extensions specify that BGP can exchange information within different types of *address families*. The JunosE BGP implementation defines the following different types of address families:

- Unicast IPv4—If you do not explicitly specify the address family, the router is configured to exchange unicast IPv4 addresses by default. You can also configure the router to exchange unicast IPv4 routes in a specified VRF.
- Multicast IPv4—If you specify the multicast IPv4 address family, you can use BGP to exchange routing information about how to reach a multicast source instead of a unicast destination. For information about BGP multicasting commands, see "Configuring BGP Routing" on page 3. For a general description of IPv4 multicasting, see *IPv4 Multicast Overview* in the *JunosE Multicast Routing Configuration Guide*.
- VPN IPv4—If you specify the VPN-IPv4 (also known as VPNv4) address family, you can configure the router to provide IPv4 VPN services over an MPLS backbone. These VPNs are often referred to as BGP/MPLS VPNs.
- Unicast IPv6—If you specify the IPv6 unicast address family, you can configure the router to exchange unicast IPv6 routes or unicast IPv6 routes in a specified VRF. For a description of IPv6, see *IPv6 Overview* in the *JunosE IP, IPv6, and IGP Configuration Guide*.

- Multicast IPv6—If you specify the multicast IPv6 address family, you can use BGP to exchange routing information about how to reach an IPv6 multicast source instead of an IPv6 unicast destination. For a general description of IPv6 multicasting, see *IPv6 Multicast Overview* in the *JunosE Multicast Routing Configuration Guide*.
- VPN IPv6—If you specify the VPN-IPv6 address family, you can configure the router to provide IPv6 VPN services over an MPLS backbone. These VPNs are often referred to as BGP/MPLS VPNs.
- L2VPN—If you specify the L2VPN address family, you can configure the PE router for VPLS L2VPNs or VPWS L2VPNs to exchange layer 2 network layer reachability information (NLRI) for all VPLS or VPWS instances. Optionally, you can use the **signaling** keyword with the **address-family** command for the L2VPN address family to specify BGP signaling of L2VPN reachability information. Currently, you can omit the **signaling** keyword with no adverse effects. For a description of VPLS, see "Configuring VPLS" on page 613. For a description of VPWS, see "Configuring VPWS" on page 677.
- Route-target—If you specify the route-target address family, you can configure the router to exchange route-target membership information to limit the number of routes redistributed among members. For a description of route-target filtering, see "Understanding Route-Target Filtering for MBGP VPNs Overview" on page 427.
- VPLS—If you specify the VPLS address family, you can configure the router to exchange layer 2 NLRI for a specified VPLS instance. For a description of VPLS, see "Configuring VPLS" on page 613.
- VPWS—If you specify the VPWS address family, you can configure the PE router to exchange layer 2 NLRI for a specified VPWS instance. For a description of VPWS, see "Configuring VPWS" on page 677.

For information about specifying an address family, see *Multicast VPNs Overview* in the *JunosE Multicast Routing Configuration Guide*.

Related Documentation

- MBGP Overview on page 391
- Understanding VPN-IPv4 Addresses on page 397
- Understanding Route Targets on page 397

- *address-family*

Equal-Cost Multipath Support Overview

Equal-cost multipath (ECMP) is a traffic load-balancing feature that enables traffic to the same destination to be distributed over multiple paths that have the same cost. BGP ECMP support for BGP/MPLS VPNs enables MPLS VPN routes to be included in the list of available equal-cost paths. You can specify that up to 16 equal-cost paths be considered.

The set of ECMP legs in a network can contain MPLS indirect next hops, either as a leg itself or pointed to by a leg. If the path to any of the MPLS indirect next hops fails, then the routing protocol begins recalculating the set of viable routes as soon as it is notified of the failure. When the recalculation has finished, the protocol then updates the routing table with the new routes.

From the time the path fails until the routing table is updated, the traffic flowing over the ECMP leg that has the failed MPLS indirect next hop is lost.

To reduce the amount of lost traffic, the failed path is quickly pruned from the ECMP set as soon as the protocol is notified of the connectivity failure. Traffic for the destination is then forwarded over the remaining equal-cost paths to the destination. When the recalculated set of routes is installed in the routing table, traffic for the destination is forwarded by means of the new route.

ECMP sets can have an MPLS indirect next hop as one of the legs in the following scenarios:

- In a BGP-MPLS VPN where a given VPN prefix is learned from multiple PE routers.
- When multiple RSVP-TE tunnels are created over different paths to the same destination.
- In a network that connects IPv6 islands across an IPv4 core, where a given IPv6 prefix is learned from multiple egress PEs running IPv6.

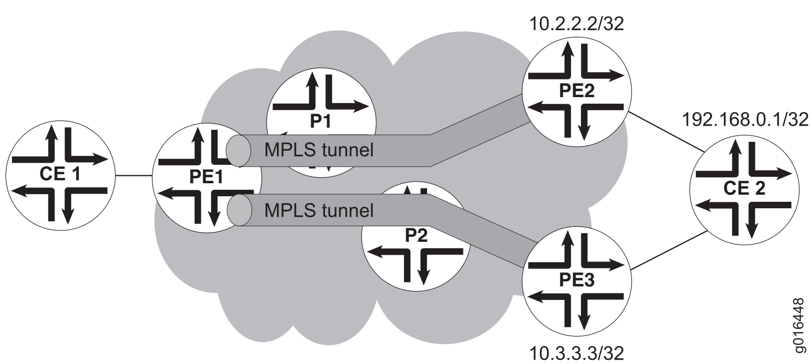
Related Documentation

- MBGP Overview on page 391
- Example: Simple ECMP Scenario for BGP/MPLS VPN on page 393
- Example: Enabling BGP ECMP for BGP/MPLS VPN IBGP on page 454
- Example: Enabling BGP ECMP for BGP/MPLS VPN EBGP on page 455

Example: Simple ECMP Scenario for BGP/MPLS VPN

Consider the simple ECMP scenario for a BGP/MPLS VPN shown in Figure 68 on page 393.

Figure 68: ECMP BGP/MPLS VPN Scenario



With respect to PE 1, this network has an ECMP set of two equal-cost legs for the VPN prefix of CE 2, 192.168.0.1/32:

- PE 1 -> P 1 -> PE 2 -> CE 2

- PE 1 -> P 2 -> PE 3 -> CE 2

The details of these routes are displayed by the following command:

```
host1:pe1:pe1-ce1#show ip route 192.168.0.1 detail
192.168.0.1/32 Type: Bgp Distance: 200 Metric: 0 Tag: 0 Class: 0
  MPLS next-hop: 741, ECMP next-hop, leg count 2
    MPLS next-hop: 389, label 17, VPN traffic, resolved by MPLS next-hop 376
      MPLS next-hop: 376, resolved by MPLS next-hop 385, peer 10.3.3.3
        MPLS next-hop: 385, label 24 on GigabitEthernet1/1/0.2
(ip19000002.mpls.ip [V:pe1]), nbr 10.3.2.2
    MPLS next-hop: 740, label 18, VPN traffic, resolved by MPLS next-hop 729
      MPLS next-hop: 729, resolved by MPLS next-hop 737, peer 10.2.2.2
        MPLS next-hop: 737, label 27 on GigabitEthernet1/1/0.1
(ip19000001.mpls.ip [V:pe1]), nbr 10.3.1.2
```

If the connection to PE 2 fails, BGP marks the MPLS next hop 729 as a failed indirect next hop as soon as BGP is notified of the loss of connectivity. However, some traffic continues to be forwarded to CE 2 through PE 2; this traffic is lost. BGP quickly prunes the failed route from the FIB, stopping this traffic loss, and then recalculates the routes to CE 2. During this period, traffic for CE 2 is forwarded only through PE 3. When the new routes are installed in the FIB, traffic is forwarded to CE 2 by means of the newly installed route.

Related Documentation

- MBGP Overview on page 391
- Equal-Cost Multipath Support Overview on page 392
- Monitoring the VRF Routing Table on page 512
- *show ip route*

BGP/MPLS VPN Components Overview

If you have specified the VPN-IPv4 address family, you can configure virtual private networks across an IP backbone. BGP carries routing information for the network and MPLS labels, whereas MPLS transports the data traffic. Figure 69 on page 395 shows a typical scenario.

The service provider backbone comprises two types of routers:

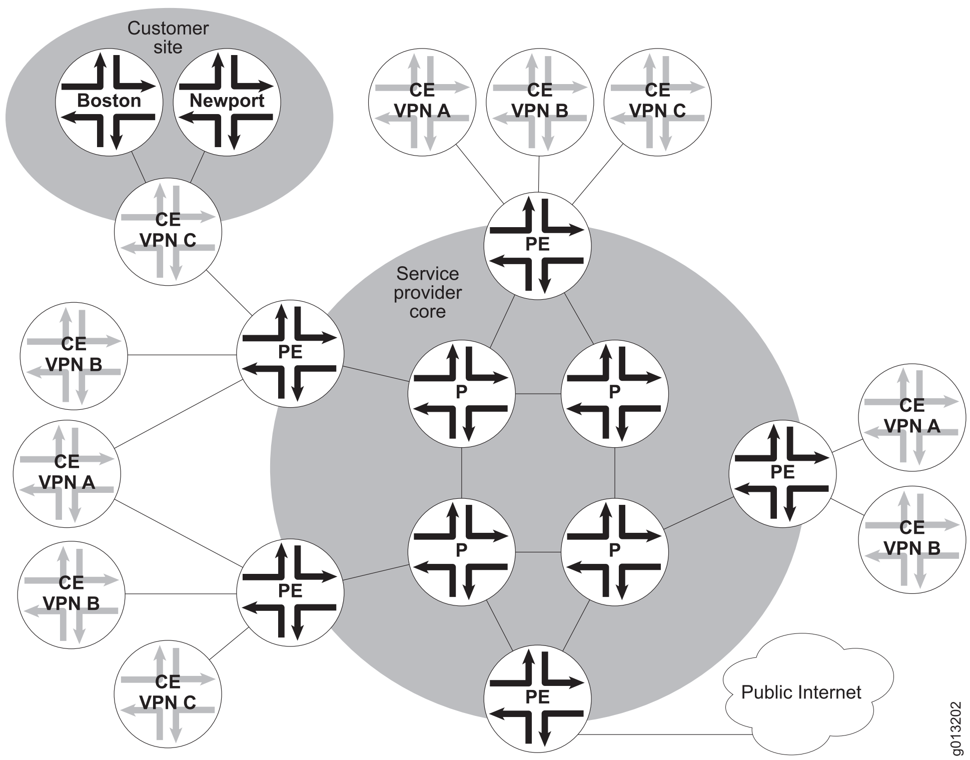
- Provider edge routers (PE routers)
- Provider core routers (P routers)

PE routers are situated at the edge of the service provider core and connect directly to customer sites. These routers must run BGP-4, including the BGP/MPLS VPN extensions. They must also be able to originate and terminate MPLS LSPs. (See "Configuring MPLS" on page 279, for more information.)

P routers connect directly to PE routers or other P routers and do not connect directly to customer sites. These routers must be able to switch MPLS LSPs—that is, they function as MPLS label-switching routers (LSRs) and might function as label edge routers (LERs). Running BGP-4 on the P routers is not necessary to be able to exchange routing information for VPNs. You might run BGP-4 on the core routers for other reasons, such as exchanging routing information for the public Internet or implementing route reflectors. The P routes do not need to contain any information about customer sites.

PE routers communicate with customer sites through a direct connection to a customer edge (CE) device that sits at the edge of the customer site. The CE device can be a single host, a switch, or, most typically, a router. When the CE device is a router, it is a routing peer of all directly connected PE routers, but it is not a routing peer of CE routers at any other site. The link between the CE router and the PE router can employ any type of encapsulation. Using MPLS is not necessary. In Figure 69 on page 395, each PE router connects to multiple CE routers and at least one P router. Although only one customer site is shown, each CE router lies within a customer site.

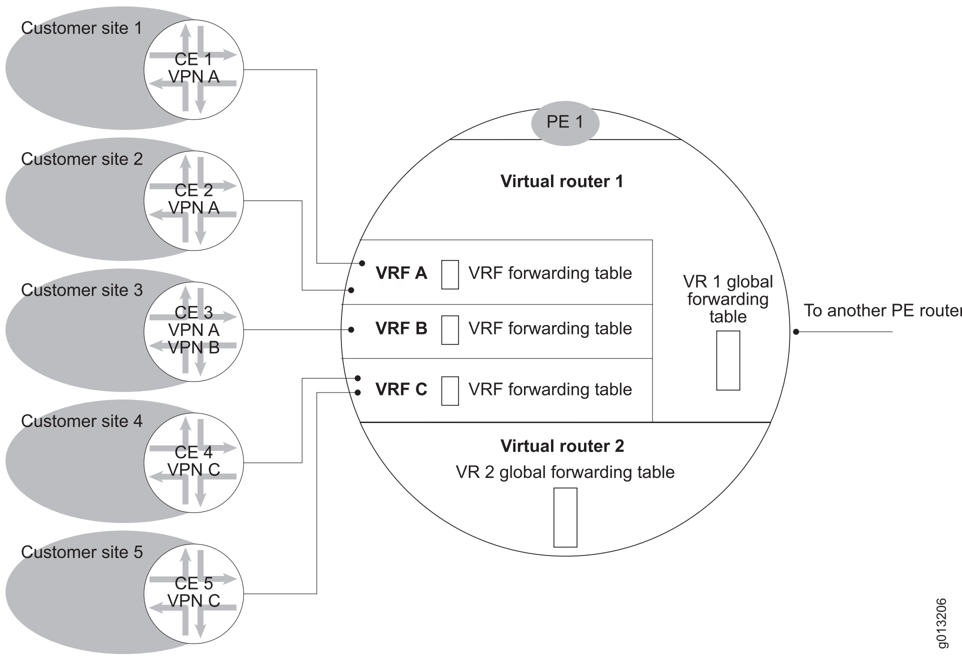
Figure 69: BGP/MPLS VPN Scenario



A customer site is a network that can communicate with other networks in the same VPN. A customer site can belong to more than one VPN. Two sites can exchange IP packets with each other only if they have at least one VPN in common.

Each customer site that is connected to a particular PE router is also associated with a VPN routing and forwarding instance (VRF). As shown in Figure 70 on page 396, each VRF has its own forwarding table distinct from that of other VRFs and from the virtual router's global forwarding table.

Figure 70: BGP/MPLS VPN Components



A given VRF's forwarding table includes only routes to sites that have at least one VPN in common with the site that is associated with the VRF. For example, in Figure 70 on page 396, the forwarding table in VRF B stores routes only to sites that are members of at least one of the VPNs to which Customer Site 3 belongs.

VRFs exist within the context of a virtual router (VR). A given virtual router can have zero or more VRFs, in addition to its global routing table (which is not associated with any VPN, CE router, or customer site). A router can support up to 1000 forwarding tables; that is, up to a combined total of 1000 VRs and VRFs.

You assign one or more interfaces or subinterfaces to a given VRF. If multiple customer sites are members of the same set of VPNs, they can share a VRF—that is, you do not need to create a specific VRF for each customer site. In Figure 70 on page 396, Customer Sites 1 and 2 share VRF A; both sites belong to the same set of VPNs. The router looks up a packet's destination in the VRF associated with the interface on which the packet is received. The VRFs are populated by BGP while it learns routes from the VPN. If a customer site is a member of multiple VPNs, the routes learned from all those VPNs populate the VRF associated with the site.

Related Documentation

- MBGP Overview on page 391
- Understanding MBGP Address Families on page 391
- Understanding VPN-IPv4 Addresses on page 397
- Example: Distribution of Routes and Labels with BGP on page 398
- Packet Transport Across an IP Backbone with MPLS Overview on page 403

Understanding VPN-IPv4 Addresses

Because each VPN has its own private address space, the same IP address might be used in several VPNs. To provide for more than one route to a given IPv4 address (each route unique to a single VPN), BGP/MPLS VPNs use route distinguishers (RDs) followed by an IPv4 address to create unique VPN-IPv4 addresses. A route can have only one RD.

The RD contains no routing information; it simply enables you to create unique VPN-IPv4 address prefixes. You can specify the RD in either of the following ways:

- An autonomous system (AS) number followed by a 32-bit assigned number. If the AS number is from the public address space, it must have been assigned to the service provider by the Internet Assigned Numbers Authority (IANA). The service provider can choose the assigned number. We recommend you do not use numbers from the private AS number space.
- An IP address followed by a 16-bit assigned number. If the IP address is from the public IP address space, it must have been assigned to the service provider by IANA. The assigned number may be chosen by the service provider. Use of numbers from the private IP address space is strongly discouraged.

You can create unique VPN-IPv4 addresses by assigning a unique RD to each VRF in your network. However, the optimal strategy depends on the configuration of your network. For example, if each VRF always belongs to only one VPN, you might use a single RD for all VRFs that belong to a particular VPN.

Related Documentation

- Understanding MBGP Address Families on page 391
- BGP/MPLS VPN Components Overview on page 394
- Example: Distribution of Routes and Labels with BGP on page 398

Understanding Route Targets

A route-target extended community, or route target, is a type of BGP extended community that you use to define VPN membership. The route target appears in a field in the update messages associated with VPN-IPv4.

You create route-target import lists and route-target export lists for each VRF. The route targets that you place in a route target export list are attached to every route advertised to other PE routers. When a PE router receives a route from another PE router, it compares the route targets attached to each route against the route-target import list defined for each of its VRFs. If any route target attached to a route matches the import list for a VRF, then the route is imported to that VRF. If no route target matches the import list, then the route is rejected for that VRF.

Depending on your network configuration, the import and export lists may be identical. Typically, you do the following:

- Allocate one route-target extended-community value per VPN.

- Configure the import list and the export list to include the same information: the set of VPNs comprising the sites associated with the VRF.

For more complicated scenarios—for example, hub-and-spoke VPNs—the route-target import list and the route-target export list might not be identical.

A route-target import list is applied before any inbound routing policy (route map) is applied. If an inbound route map contains a **set extcommunity** clause, the clause replaces all extended communities in the received route. BGP applies the default route-target export list associated with the VRF if the route does not have any route-target extended-community attributes after the inbound policy has been applied. On the other hand, the default export list is not applied if either a valid route-target export list is received or the inbound route map sets one or more route targets.

Related Documentation

- Understanding MBGP Address Families on page 391
- Example: Distribution of Routes and Labels with BGP on page 398
- Understanding Route-Target Filtering for MBGP VPNs Overview on page 427
- Hub-and-Spoke VPNs on page 423

- *set extcommunity*

Example: Distribution of Routes and Labels with BGP

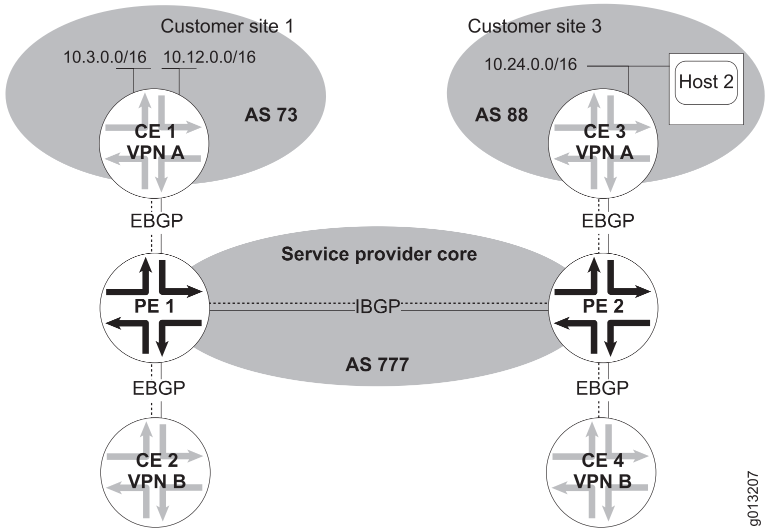
The extensions to BGP include enhancements to update messages that enable them to carry the route distinguishers, route-target extended-community information, and MPLS labels required for BGP/MPLS VPNs.

Consider the simple example shown in Figure 71 on page 399. The customer edge devices are connected with their associated provider edge routers by external BGP sessions (CE 1—;PE 1 and CE 3—PE 2). PE 1 and PE 2 are BGP peers by an internal BGP session across the service provider core in AS 777.

In this example, the PE routers run EBGP to the CE routers to do the following:

- Learn the prefixes of the networks in the local customer site.
- Advertise routes to networks and remote customer sites.

Figure 71: Route and Label Distribution



Rather than running EBGP between the PE routers and the CE routers, you can do either of the following:

- Run an IGP (such as IS-IS, OSPF, or RIP) between the CE router and the PE router.
- Configure static routes on the CE and PE routers (on the CE router this would typically be a default route).

In this example the two customer sites use different AS numbers, which simplifies configuration. Alternatively, the same AS numbers can be used.

Customer site 1 has two networks that need to be reachable from customer site 3—10.3.0.0/16 and 10.12.0.0/16—and uses BGP to announce these prefixes to PE;1. CE 1 uses a standard BGP update message as shown in Figure 72 on page 401 to carry this and additional information. CE 1 is withdrawing prefix 10.1.0.0/16. CE 1 specifies its own address as the next hop; 10.4.1.1 is from the private address space of VPN A.

PE 1 passes the advertisement along the backbone through an IBGP session, but uses MP-BGP rather than standard BGP-4. Consequently, PE 1 uses an extended BGP update message, which is different in format from the standard message, as shown in Figure 72 on page 401.

The extended update uses different attributes for some of the advertised information. For example it carries the advertised prefixes in the MP-Reach-NLRI attribute instead of the NLRI attribute. Similarly, it uses the MP-Unreach-NLRI attribute for withdrawn routes rather than the withdrawn-routes attribute.

PE 1 advertises the customer site addresses by prepending information to the addresses as advertised by CE 1, thus creating *labeled VPN-IPv4 prefixes*. The prepended information consists of a route distinguisher and an MPLS label.

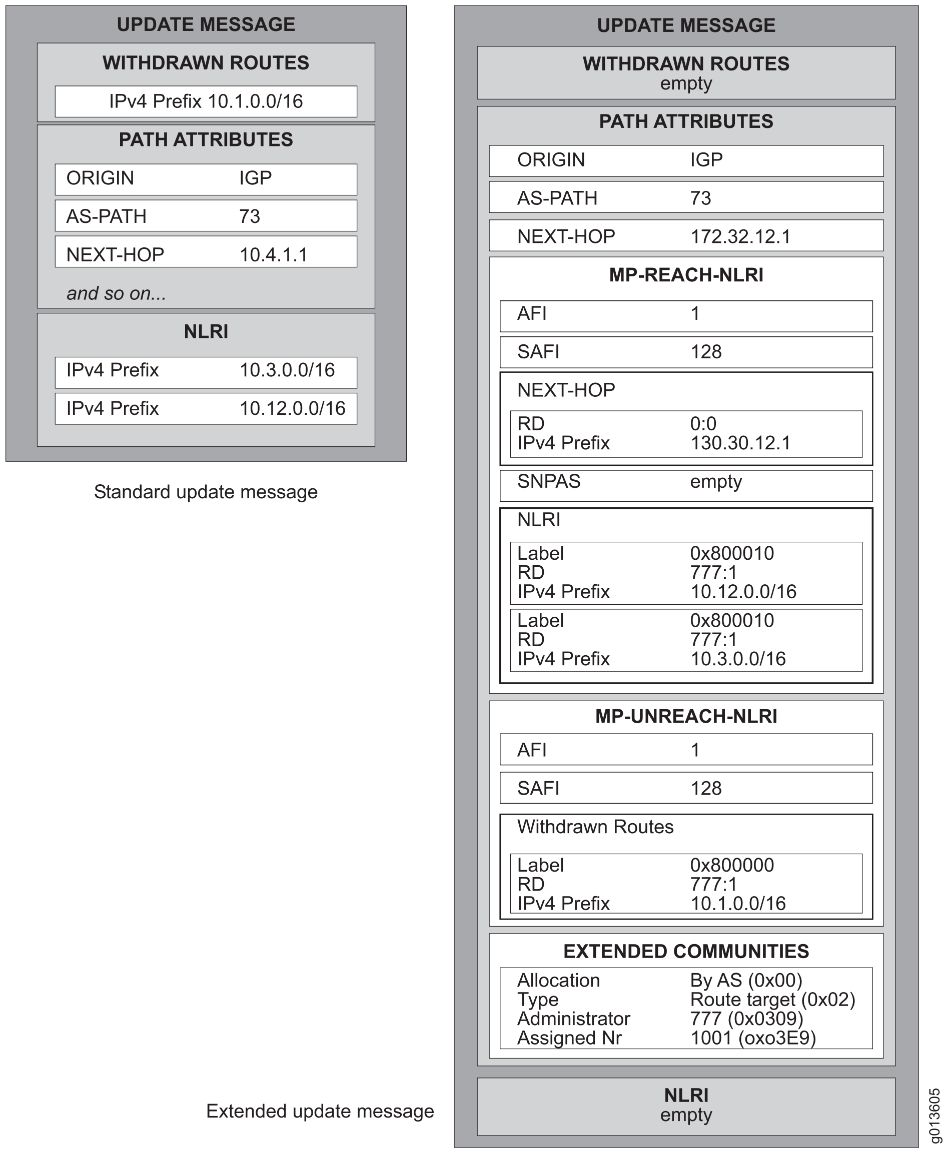
Because the CE router uses IPv4 addresses from the VPN's private address space, these addresses can be duplicated in other VPNs to which PE 1 is attached. PE 1 associates a route distinguisher with each IPv4 address to create a globally unique address. In this example, the RD consists of the AS that PE 1 belongs to and a number that PE 1 assigns. The RD is prepended immediately before the IPv4 address.

PE routers assign MPLS labels to each VRF. In this example, the label for the VRF associated with customer site 1 is 16. The MPLS label is prepended immediately before the route distinguisher.

NOTE: **The explicit null label is prepended only to routes that are being withdrawn in the MP-REACH-NLRI attribute.**

Some non–E Series implementations allocate a separate label for each prefix. By default, the E Series router generates one label for all BGP routes advertised by the VRF, thus reducing the number of stacked labels to be managed. The **ip mpls forwarding-mode label-switched** command enables you to have the router generate a label for each different FEC pointed to by a BGP route in a given VRF. However, some routes always receive a per-VRF label; see “Understanding Labels Creation per FEC” on page 453 for more information.

Figure 72: Standard and Extended BGP Update Messages



Using the **next-hop-self** option on PE 1 causes PE 1 to set the next-hop attribute to its own address, 172.32.12.1. Doing so is necessary because the next hop provided by CE 1 is from VPN A's private address space and has no meaning in the service provider core. In addition, PE 2 must have PE 1's address so that it can establish an LSP back to PE 1. The next-hop address must also be carried in the MP-Reach-NLRI attribute, according to MP-BGP.

The extended update also has the extended-communities attribute, which identifies the VPN to which the routes are advertised. In this example, the route target is 777:1001, identifying VPN A.

Related Documentation

- BGP/MPLS VPN Components Overview on page 394

- Understanding VPN-IPv4 Addresses on page 397
- Understanding Route Targets on page 397

- *ip mpls forwarding-mode label-switched*

BGP/MPLS VPN Platform Considerations

For information about modules that support BGP/MPLS VPNs on the ERX7xx models, ERX14xx models, and the ERX310 Broadband Services Router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support BGP/MPLS VPNs.

For information about modules that support BGP/MPLS VPNs on E120 and E320 Broadband Services Routers:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support BGP/MPLS VPNs.

Related Documentation

- MBGP Overview on page 391
- MBGP References on page 402

MBGP References

For more information about BGP/MPLS VPNs, consult the following resources:

- BGP-MPLS VPN extension for IPv6 VPN—draft-ietf-l3vpn-bgp-ipv6-03.txt (December 2004 expiration)
- Connecting IPv6 Islands across IPv4 Clouds with BGP—draft-ietf-ngtrans-bgp-tunnel-04.txt (July 2002 expiration)
- *JunosE Release Notes*, *Appendix A*, *System Maximums*—Refer to the Release Notes corresponding to your software release for information about maximum values.
- RFC 2545—Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing (March 1999)
- RFC 2858—Multiprotocol Extensions for BGP-4 (June 2000)
- RFC 3107—Carrying Label Information in BGP-4 (May 2001)
- RFC 4364—BGP/MPLS IP Virtual Private Networks (VPNs) (February 2006)
- RFC 4684—Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs) (2006)

For more information about BGP and MPLS, see the *References* sections in "Configuring BGP Routing" on page 3 and in "Configuring MPLS" on page 279.

NOTE: IETF drafts are valid for only 6 months from the date of issuance. They must be considered as works in progress. Please refer to the IETF Website at http://www.ietf.org for the latest drafts.

Related Documentation

- MBGP Overview on page 391
- BGP/MPLS VPN Platform Considerations on page 402

Packet Transport Across an IP Backbone with MPLS Overview

As described in the previous section, PE 1 and PE 2 exchange routing information, including MPLS labels for their customer sites, by means of a BGP session established between them across the service provider core.

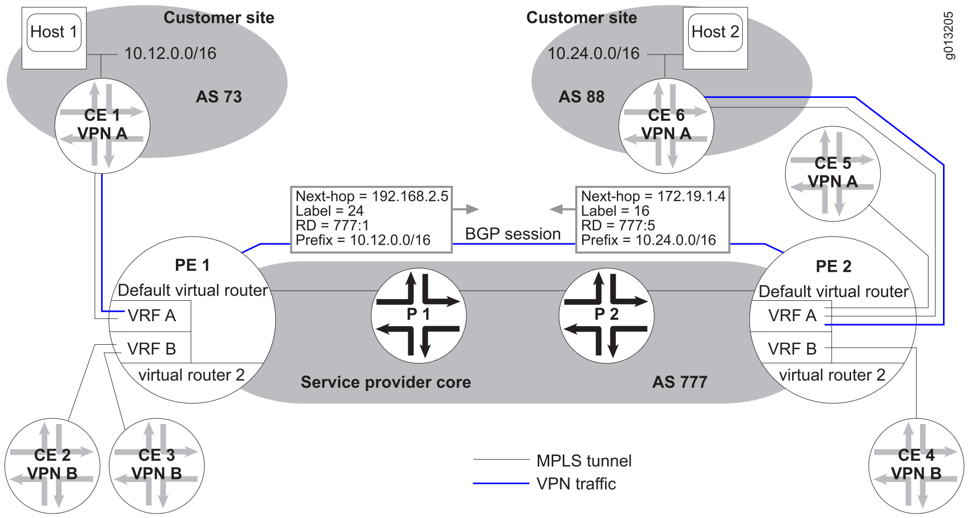
NOTE: To better understand MPLS before you read this section, see "Configuring MPLS" on page 279.

Labels are employed in both the BGP control plane and the MPLS data plane. In the control plane, BGP advertises a route with an in label; this in label is also the label needed when MPLS traffic is received. BGP receives routes with an associated out label; the out label is the label sent with MPLS traffic.

Consider the network shown in Figure 73 on page 404. If you display the in label on PE 1, you see that MP-BGP advertises a labeled VPN-IPv4 prefix of 10.12.0.0/16 with an in label of 24 (and an RD of 777:1, as shown in the illustration).

```
host1:pe1#show ip bgp vpn all field in-label
Prefix         In-label
10.12.0.0/16     24
10.24.0.0/16     none
```

Figure 73: BGP/MPLS VPN Route Exchange



If you display the in label on PE 2, you see that MP-BGP advertises a labeled VPN-IPv4 prefix of 10.24.0.0/16 with an in label of 16 (and an RD of 777:5, as shown in the illustration).

```
host2:pe2#show ip bgp vpn all field in-label
Prefix          In-label
10.12.0.0/16      none
10.24.0.0/16      16
```

On PE 1, you see that MP-BGP receives a labeled VPN-IPv4 prefix of 10.24.0.0/16 with an out label of 16. MP-BGP on PE 2 advertised this label with the prefix. In the data plane, MPLS traffic is sent by PE 1 to PE 2 with this label.

```
host1:pe1#show ip bgp vpn all field out-label
Prefix          Out-label
10.12.0.0/16      none
10.24.0.0/16      16
```

On PE 2, you see that MP-BGP receives a labeled VPN-IPv4 prefix of 10.12.0.0/16 with an out label of 24. MP-BGP on PE 1 advertised this label with the prefix. In the data plane, MPLS traffic is sent by PE 2 to PE 1 with this label.

```
host2:pe2#show ip bgp vpn all field out-label
Prefix          Out-label
10.12.0.0/16      24
10.24.0.0/16      none
```

The data packets are transported within a VPN across the service provider core by MPLS. This transport process requires two layers of MPLS labels, stacked one upon the other.

The inner labels are assigned by each PE router for each VRF. When an MPLS packet arrives at the egress PE router, that egress PE router uses the inner label to determine which VRF the packet is destined for. In the default, per-VRF label allocation mode (described in "Understanding Labels Creation per FEC" on page 453), the egress PE router does an IP lookup in the IP forwarding table of that VRF using the IP destination address in the IP packet that is encapsulated in the MPLS packet. The egress PE router then forwards the IP packet (without the MPLS header) to the appropriate customer site. The inner labels themselves are communicated between PE routers in the MP-BGP extended update messages as described in the previous section.

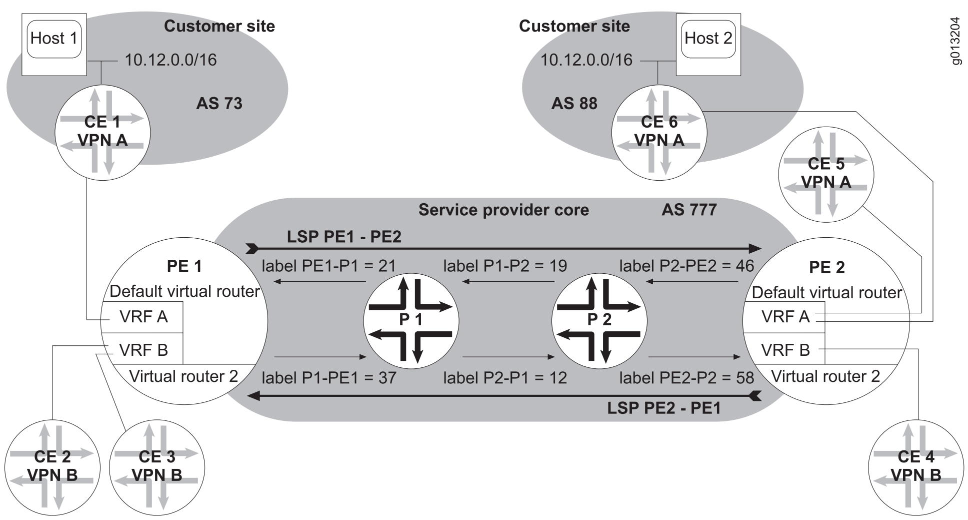
MPLS uses the outer labels to forward data packets from the ingress PE router through a succession of P routers across the core. This succession of P routers constitutes a label-switched path (LSP), also referred to as an MPLS tunnel. The labels are assigned to links in the path.

At each P router, MPLS pops the outer label from a data packet. The label is an index into the P router's forwarding table, from which it determines both the next hop along the LSP and another label. The router pushes the label on to the label stack and forwards the packet to the next P router. The combination of popping one label and pushing another is known as a label swap. At the egress PE router, MPLS pops the outer label, then the inner label. The inner label determines the CE router to which the packet is sent. The P routers never examine the inner MPLS label or the destination IP address encapsulated in the MPLS packet.

In many cases, the PE routers are fully meshed by means of LSPs. You can use tunnel profiles to simplify the LSP configuration process. See "Configuring MPLS" on page 279, for procedures to configure an LSP.

Each LSP is unidirectional for data traffic, so you must establish LSPs in both directions for two-way data transport. Figure 74 on page 405 shows that two LSPs have been created between PE 1 and PE 2. PE 1 and PE 2 have an MP-BGP session as shown previously in Figure 73 on page 404.

Figure 74: LSP Creation for BGP/MPLS VPN



The PE 1–PE 2 LSP carries traffic only from PE 1 to PE 2, using label 21 for the PE 1 to P 1 link, label 19 for the P 1 to P 2 link, and label 46 for the P 2 to PE 2 link. PE 1 can forward data packets along the LSP to PE 2 and its customer sites.

Similarly, the PE 2–PE 1 LSP carries traffic only from PE 2 to PE 1, using label 58 for the PE 2 to P 2 link, label 12 for the P 2 to P 1 link, and label 37 for the P 1 to PE 1 link. PE 2 can forward data packets along the LSP to PE 1 and its customer sites.

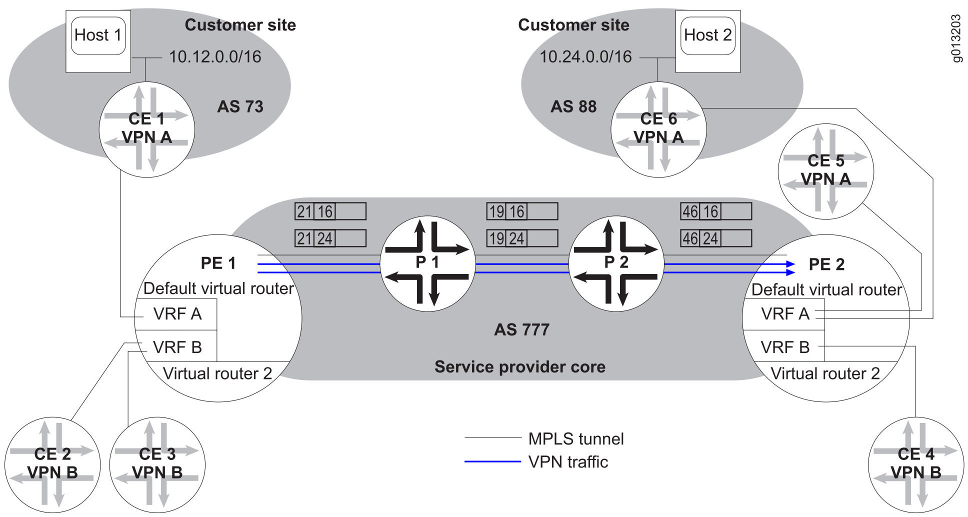
Related Documentation

- BGP/MPLS VPN Components Overview on page 394
- Example: Transporting Packets Across an IP Backbone with MPLS on page 406
- Example: Data Transport Process on page 407
- *show ip bgp*

Example: Transporting Packets Across an IP Backbone with MPLS

The process of data transport is shown in Figure 75 on page 406. PE 1 has already received announcements from PE 2; an LSP has been established between PE 1 and PE 2.

Figure 75: Traffic Across the MPLS Backbone of a BGP/MPLS VPN



Host 1 constructs an IP packet with the address of Host 2 as the final destination, and sends the packet to router CE 1. CE 1 encapsulates the packet appropriately and forwards it to PE 1.

PE 1 receives the packet from CE 1. Based on the interface the packet came in on, PE 1 determines that it must use the forwarding table for VRF A to route the packet. PE 1 looks up the destination address of Host 2 in the forwarding table of VRF A and finds the following instructions:

- Push label 16; that is, prepend it to the data packet. This innermost label identifies the VRF on PE 2, where the final destination and interface lookup takes place. Label 16 was previously allocated by PE 2 and communicated to PE 1 by MP-BGP. VRF A shows this label as part of the NLRI for destination address Host 2.
- Push label 21 and forward the MPLS-encapsulated data packet to router P 1. Label 21 is prepended to label 16; the labels are *stacked*. Label 21 becomes the outermost label and is assigned to the first segment—PE 1–P 1—in the label-switched path from PE 1 to PE 2. The LSP was previously configured.

P 1 receives the data packet from PE 1 and pops label 21. P 1 looks up label 21 in its forwarding table and determines it must push label 19 on the stack, and forwards the data packet to P 2.

P 2 receives the data packet from P 1 and pops label 19. P 2 looks up label 19 in its forwarding table and determines it must push label 46 on the stack, and forwards the data packet to PE 2.

PE 2 receives the data packet from P 2, and looks up label 46. PE 2 determines it is the egress router of the LSP and must pop label 46. Then it proceeds to look up the next label, label 16, and determines that the packet goes to VRF A. Then the IP address is looked up in VRF A to determine the destination and outgoing interface for the packet. PE 2 forwards the packet to CE 6.

CE 6 receives the IP packet from PE 2 and looks up the destination address Host 2. Subsequent forwarding to Host 2 occurs by means of the IGP in the customer site.

The network structure shown in Figure 75 on page 406 consists of two VPNs, A and B. VPN A comprises CE 1, CE 5, and CE 6. VPN B comprises CE 2, CE 3, and CE 4. CE 1 has data traffic destined for both CE 5 and CE 6. Because both of these destination sites are within the same VPN, PE 1 uses the same forwarding table, in VRF A, to do the lookups and MPLS encapsulation. The innermost label determines the destination VRF and is the same for all packets in that VPN, even if they are destined for different CE routers. CE 2 and CE 3 have traffic destined for CE 4. Because these all are in VPN B, PE 1 uses a different forwarding table, in VRF B, for looking up destinations for traffic originating with these sites. However, both VPNs use the same LSP, because both VPNs use the same ingress (PE 1) to and the same egress (PE 2) from the service provider core. Remember that the illustrated LSP carries data traffic only from PE 1 to PE 2. Traffic from PE 2 to PE 1 requires a different LSP.

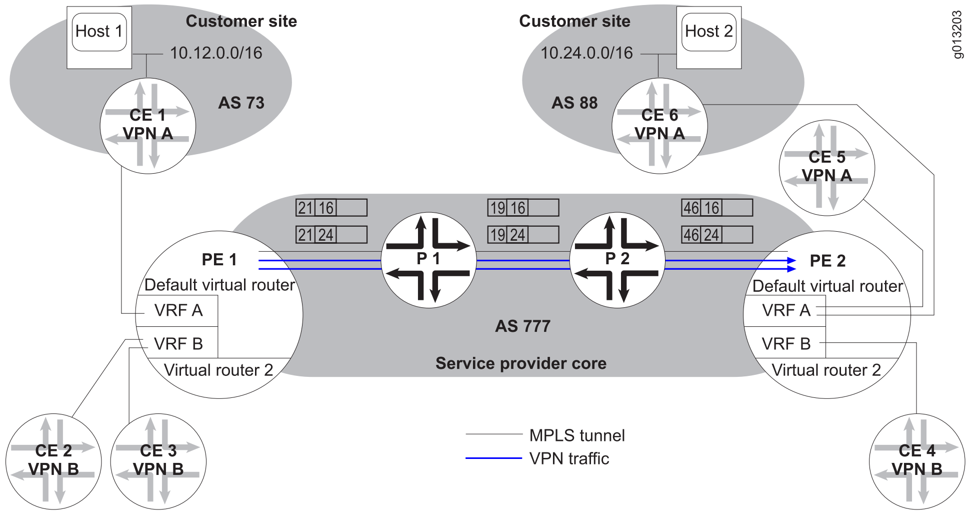
Related Documentation

- Packet Transport Across an IP Backbone with MPLS Overview on page 403

Example: Data Transport Process

The process of data transport is shown in Figure 76 on page 408. PE 1 has already received announcements from PE 2; an LSP has been established between PE 1 and PE 2.

Figure 76: Traffic Across the MPLS Backbone of a BGP/MPLS VPN



Host 1 constructs an IP packet with the address of Host 2 as the final destination, and sends the packet to router CE 1. CE 1 encapsulates the packet appropriately and forwards it to PE 1.

PE 1 receives the packet from CE 1. Based on the interface the packet came in on, PE 1 determines that it must use the forwarding table for VRF A to route the packet. PE 1 looks up the destination address of Host 2 in the forwarding table of VRF A and finds the following instructions:

- Push label 16; that is, prepend it to the data packet. This innermost label identifies the VRF on PE 2, where the final destination and interface lookup takes place. Label 16 was previously allocated by PE 2 and communicated to PE 1 by MP-BGP. VRF A shows this label as part of the NLRI for destination address Host 2.
- Push label 21 and forward the MPLS-encapsulated data packet to router P 1. Label 21 is prepended to label 16; the labels are *stacked*. Label 21 becomes the outermost label and is assigned to the first segment—PE 1–P 1—in the label-switched path from PE 1 to PE 2. The LSP was previously configured.

P 1 receives the data packet from PE 1 and pops label 21. P 1 looks up label 21 in its forwarding table and determines it must push label 19 on the stack, and forwards the data packet to P 2.

P 2 receives the data packet from P 1 and pops label 19. P 2 looks up label 19 in its forwarding table and determines it must push label 46 on the stack, and forwards the data packet to PE 2.

PE 2 receives the data packet from P 2, and looks up label 46. PE 2 determines it is the egress router of the LSP and must pop label 46. Then it proceeds to look up the next label, label 16, and determines that the packet goes to VRF A. Then the IP address is looked up in VRF A to determine the destination and outgoing interface for the packet. PE 2 forwards the packet to CE 6.

CE 6 receives the IP packet from PE 2 and looks up the destination address Host 2. Subsequent forwarding to Host 2 occurs by means of the IGP in the customer site.

The network structure shown in Figure 76 on page 408 consists of two VPNs, A and B. VPN A comprises CE 1, CE 5, and CE 6. VPN B comprises CE 2, CE 3, and CE 4. CE 1 has data traffic destined for both CE 5 and CE 6. Because both of these destination sites are within the same VPN, PE 1 uses the same forwarding table, in VRF A, to do the lookups and MPLS encapsulation. The innermost label determines the destination VRF and is the same for all packets in that VPN, even if they are destined for different CE routers. CE 2 and CE 3 have traffic destined for CE 4. Because these all are in VPN B, PE 1 uses a different forwarding table, in VRF B, for looking up destinations for traffic originating with these sites. However, both VPNs use the same LSP, because both VPNs use the same ingress (PE 1) to and the same egress (PE 2) from the service provider core. Remember that the illustrated LSP carries data traffic only from PE 1 to PE 2. Traffic from PE 2 to PE 1 requires a different LSP.

Related Documentation

- Packet Transport Across an IP Backbone with MPLS Overview on page 403

IPv6 VPN Overview

The JunosE Software supports IPv6 VPNs tunneled over an MPLS IPv4 backbone. A service provider can offer IPv4 VPN services, IPv6 VPN services, or both. MPLS over IPv6 is not currently supported. MPLS base tunnels to IPv6 destinations as tunnel endpoints are not supported, so you cannot establish an MPLS IPv6 backbone.

NOTE: **You must configure an IPv6 interface in the parent VR for IPv6 VPNS to work.**

BGP can negotiate VPNv6 capability without having to negotiate the IPv6 capability. BGP next-hop encoding varies depending on whether the backbone is IPv4 or IPv6. In the JunosE Software implementation for IPv6 VPNs, the BGP next hops in the MP-BGP update message follow the convention for BGP next-hop encoding for IPv4 backbone. If an E Series router receives a BGP next hop that follows the encoding for an MPLS-enabled IPv6 backbone, that BGP next hop is treated as unreachable because currently no MPLS base tunnel to the native IPv6 tunnel endpoint address can exist.

The PE routers have both IPv4 and IPv6 capabilities. They maintain IPv6 VRFs for their IPv6 sites and encapsulate IPv6 traffic in MPLS frames that are then sent into the MPLS core network.

Link-local scope addresses cannot be used for reachability across IPv6 VPN sites and can never be advertised by means of MP-BGP to remote PE routers. Global scope addresses are expected to be used within and across IPv6 VPN Sites.

Creating an address family for a VRF automatically disables both synchronization and automatic summarization for that VRF.

All features previously supported for BGP/MPLS IPv4 VPNs, such as policy-based routing, redistribution to and from other protocols, aggregation, route-flap dampening, and so on are also supported for BGP/MPLS IPv6 VPNs.

Related Documentation

- MBGP Overview on page 391
- BGP/MPLS VPN Components Overview on page 394
- Intra-AS IPv6 VPNs on page 410
- Understanding IPv6 VPN Services Across Multiple Autonomous Systems on page 421

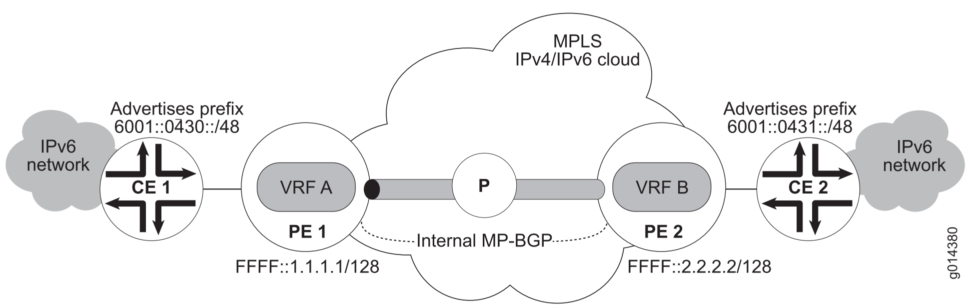
Intra-AS IPv6 VPNs

Understanding Intra-AS IPv6 VPNs

In Figure 77 on page 410, a service provider is offering IPv6 VPN service over an MPLS-enabled IPv4 backbone. The base MPLS tunnels are established in the IPv4 core network with either of the MPLS signaling protocols (LDP or RSVP). The ingress PE router pushes the LSP tunnel label directly onto the label stack of the labeled IPv6 VPN packet. The topmost label imposed corresponds to the LSP that runs from the ingress PE router to the egress PE router. The BGP next-hop field identifies the egress PE router, and therefore the topmost label to be pushed on the stack. The bottom label is the label bound to the IPv6 VPN prefix by means of BGP.

The CE devices can attach to the VRFs on the PE routers using both an IPv6 link and an IPv4 link. In Figure 77 on page 410, the CE devices attach to the VRFs over an IPv4 link, and use MP-BGP to connect to the VRFs on the PE routers. This arrangement enables the PE routers to learn IPv6 routes from MP-BGP running over TCPv4 from the CE devices. You can also configure IPv6 static routes in the VRFs on the PE routers to reach the networks through the CE IPv6 link. Alternatively, you can configure the static routes with any routing protocol that supports IPv6, such as OSPFv3.

Figure 77: IPv6 VPN Services over IPv4 MPLS



The PE routers use an MP-BGP session over TCPv4 to advertise the IPv6 routes from the CE devices to the remote PE routers. The IPv6 routes are advertised as labeled VPNv6 routes with a BGP next hop set to the base tunnel endpoint destination address. The next hop is formatted as an IPv4-mapped IPv6 address.

For IPv6 VPN services over an IPv4 backbone, the BGP next hop in the MP_REACH_NLRI attribute contains a VPN-IPV6 address with the RD set to zero and with the 16-byte IPv6 address encoded as an IPv4-mapped IPv6 address that contains the IPv4 address of the advertising PE router. This IPv4 address must be routable in the service provider's backbone.

BGP Control Plane Behavior Overview

The VPN service in "IPv6 VPN Services over IPv4 MPLS" on page 410 includes both CE 1 (VRF A) and CE 2 (VRF B). The MPLS base tunnels are established to tunnel endpoints PE 1 and PE 2 at their loopback interfaces. The loopback address for PE 1 is FFFF::1.1.1.1/128; for PE 2, it is FFFF::2.2.2.2/128.

The BGP next hop that is advertised in the MP-BGP update includes the following:

- A VPN-IPV6 address with the RD set to zero
- The 16-byte IPv6 address encoded as an IPv4-mapped IPv6 address that contains the IPv4 loopback address of the advertising PE router

The IPv4 IGP, such as OSPF, advertises the reachability of the loopback interfaces on the PE routers. LDP binds label L2 to 1.1.1.1/32 on the P router.

CE–PE Behavior Overview

CE 1 is connected to VRF A in PE 1 through an IPv4 interface. Similarly, CE 2 is connected to VRF B in PE 2 through an IPv4 interface. You can alternatively run OSPF to the CE devices over IPv6 links and redistribute the OSPF IPv6 routes into BGP.

The MP-BGP sessions between the CE devices and the VRFs in the PE routers are established over TCPv4. The AFI value is 2, indicating IPv6; the SAFI value is 1, indicating unicast. CE 1 advertises IPv6 network 6001:0430::/48 to its MP-BGP peer in VRF A. CE 2 advertises 6001:0431::/48 to its MP-BGP peer in VRF B. When it receives the advertised prefix in VRF A, BGP adds 6001:0430::/48 to its BGP VPNv6 RIB with the stacked label L1, which MPLS allocated for this prefix. The default IPv6 VRF label is L1.

PE–PE Behavior Overview

PE 1 advertises the VPNv6 prefixes in the MP_REACH_NLRI attribute of the update messages sent to its MP-IBGP peer, PE 2. The AFI and SAFI values are negotiated for VPNv6. The AFI value is 2 for IPv6, and the SAFI value is 128 for MPLS-labeled VPN-IPv6.

When PE 2 receives the VPNv6 prefix 6001:0430::/48 with label L1, it imports the prefix into VRF B because VRF B's import route target matches the route target received in the MP-BGP update. For all labeled VPNv6 prefixes installed in VRF B that come from the same endpoint on PE 1 (loopback FFFF::1.1.1.1/128), a single dynamic IPv6 interface stacked on top of an MPLS tunnel head is created in VRF B regardless of the number of different stacked labels associated with each VPNv6 prefix. The prefix is then installed in VRF B's routing table as pointing to this dynamic IPv6 interface.

If PE 1 is not running either JunosE or Junos OS, each VPNv6 prefix usually has a different stacked label value sent in the MP-BGP update. If an implementation allocates one VPN interface per received stacked label, this behavior might potentially become a scaling issue if many dynamic IPv6 interfaces are allocated to resolve each VPNv6 prefix in VRF B.

MPLS Data Plane Behavior Overview

When PE 2 receives a data packet from CE 2 destined for the 6001:0430::/48 network, the router detects a native IPv6 packet on its link to CE 2. PE 2 does a lookup in its VRF B IPv6 routing table, prepends labels L2 and L1 to the IPv6 header, and then forwards this packet on its core-facing IPv6 dynamic interface. When the P router receives this packet, it performs a lookup on L2 and label switches the packet toward PE 1. The P router either replaces L2 with another label or pops that label if PE 1 requested PHP.

When PE 1 receives the packet on its core-facing interface, it pops all the labels, and performs a lookup in the IPv6 table of VRF A (which is associated with L1) using the destination address in the IPv6 header. After that, PE 1 forwards the IPv6 packet out to CE 1 on the IPv6 link.

Related Documentation

- IPv6 VPN Overview on page 409

IPv4 VPN Services Across Multiple Autonomous Systems

The following sections describe IPv4 services across multiple autonomous systems:

Understanding IPv4 VPN Services Across Multiple Autonomous Systems

Inter-AS services, sometimes known as interprovider services, support VPNs that cross AS boundaries. VPNs might need to cross AS boundaries because of a customer deployment that involves geographically separated ASs. The VPN sites can be provided by the same service provider or by different service providers as part of a joint VPN service offering. Inter-AS services are also useful to service providers that use confederations of sub-ASs to reduce the IBGP mesh inside the AS.

You can support these inter-AS services in three different ways, known as inter-AS option A, option B, and option C. Option C is preferred to option B; option B is preferred to option A. For inter-AS options B and C, you must explicitly configure MPLS on all the inter-AS links.

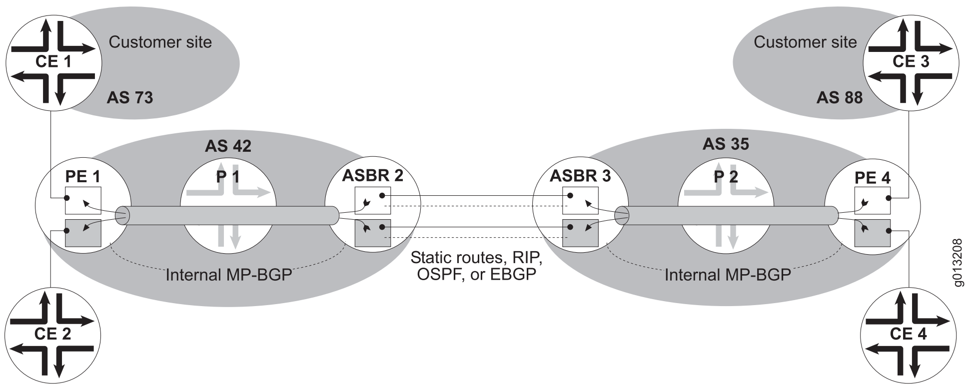
Related Documentation

- BGP/MPLS VPN Components Overview on page 394
- Inter-AS Option A Overview on page 413
- Inter-AS Option B Overview on page 414
- Inter-AS Option C Overview on page 417

Inter-AS Option A Overview

Figure 78 on page 413 illustrates the first method, where you create a VRF for each VPN on each AS boundary router.

Figure 78: Inter-AS Topology with VRFs on Each AS Boundary Router



Within each AS, routes are announced by internal MP-BGP and the data packets are forwarded across an MPLS tunnel. You create a logical connection such as an ATM VC between each pair of VRFs (on separate AS boundary routers); these logical connections can share the same physical connection. The following factors limit the scalability of this method:

- All inter-AS VPN routes (potentially a very large number) must be stored in the BGP RIBs and IP routing tables on the AS boundary routers.
- You must configure VRFs on each AS boundary router.

MPLS tunnels are unidirectional; Figure 78 on page 413 shows only the tunnels established to carry traffic from ASBR 2 to PE 1 and from PE 4 to ASBR 3. Note that ASBR 2 and ASBR 3 are both also PE routers. In that sense, ASBR 2 treats ASBR 3 as a CE router, and ASBR 3 treats ASBR 2 as a CE router.

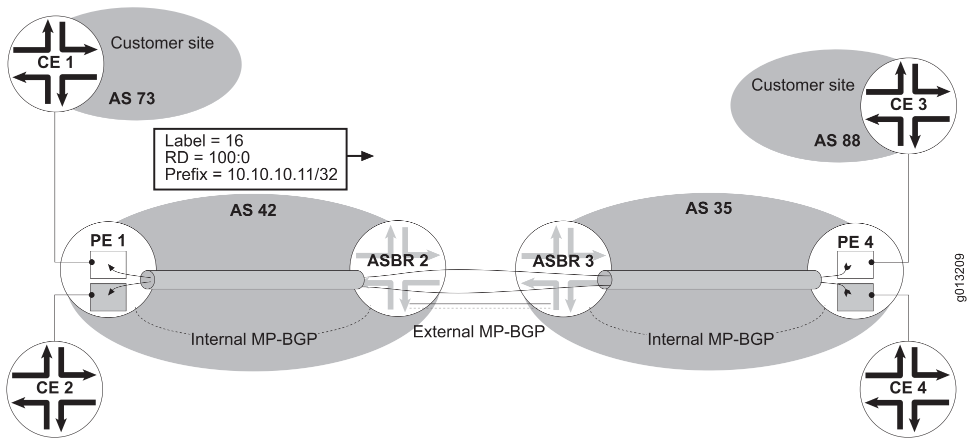
Related Documentation

- BGP/MPLS VPN Components Overview on page 394
- Understanding IPv4 VPN Services Across Multiple Autonomous Systems on page 412
- Inter-AS Option B Overview on page 414
- Inter-AS Option C Overview on page 417

Inter-AS Option B Overview

The second method is known as inter-AS option B or 2547bis option B, after IETF RFC 4364—BGP/MPLS IP Virtual Private Networks (VPNs) (February 2006). This method uses BGP to signal VPN labels between the AS boundary routers (Figure 79 on page 414). The base MPLS tunnels are local to each AS. Stacked tunnels run from end to end between PE routers on the different ASs. This method provides greater scalability, because only the BGP RIBs store all the inter-AS VPN routes.

Figure 79: Inter-AS Topology with End-to-End Stacked MPLS Tunnels



PE 1 assigns labels for routes to the customer sites, and distributes both the label assignments and the VPN-IPv4 routes throughout AS 42 in extended BGP update messages by means of internal MP-BGP. ASBR 2 then distributes the routes to ASBR 3 with external MP-BGP; ASBR 2 specifies itself as the next-hop address and assigns a new label to the route so that ASBR 3 can properly direct traffic. ASBR 3 propagates the routes by internal MP-BGP throughout AS 35, including to PE 4.

Related Documentation

- BGP/MPLS VPN Components Overview on page 394
- Understanding IPv4 VPN Services Across Multiple Autonomous Systems on page 412
- Inter-AS Option A Overview on page 413
- Example: Intra-AS Option B IPv4 VPNs on page 414
- Inter-AS Option C Overview on page 417

Example: Intra-AS Option B IPv4 VPNs

You can use the **show ip bgp vpn all field in-label** and **show ip bgp vpn all field out-label** commands in the context of each VPN element to display the in label and out label associated with the route at that point. Suppose that CE 1 advertises a route to prefix 10.10.10.11/32 to its external BGP peer PE 1 (10.2.2.2) in VRF A. PE 1 associates the label 16 with this route; an extended update message sent to internal MP-BGP peer ASBR 2 carries this information as a labeled VPN-IPv4 prefix (label 16, RD 100:0, IPv4 prefix 10.10.10.11/32).

```
host1:pe1#show ip bgp vpn all field in-label
Prefix              In-label
10.10.10.11/32      16
```

On PE 1, no out label is associated with the IPv4 prefix 10.10.10.11/32.

```
host1:pe1#show ip bgp vpn all field out-label
Prefix              Out-label
10.10.10.11/32      none
```

ASBR 2 receives the labeled VPN-IPv4 prefix and generates a new label, 44, to associate with this VPN-IPv4 prefix instead of label 16 when it sends the prefix to ASBR 3.

```
host1:asbr2#show ip bgp vpn all field out-label
Prefix              Out-label
10.10.10.11/32      16
```

```
host1:asbr2#show ip bgp vpn all field in-label
Prefix              In-label
10.10.10.11/32      44
```

ASBR 2 receives MPLS frames with label 44 (the in label) from ASBR 3 and sends MPLS frames with label 16 (the out label) to PE 1.

The inter-AS next hop shows label 44 as the label advertised to inter-AS peer ASBR 3. Label 44 was generated for the indirect next hop PE router/label pair, 10.2.2.2 (PE 1) and 16. Indirect next hop 1.1.1.1 is for the MP-IBGP peering between PE 1 (loopback address 1.1.1.1) and ASBR 2. Indirect next hop 10.5.5.5 is ASBR 3.

```
host1:asbr2#show ip bgp vpn all next-hops
Indirect next-hop 1.1.1.1
  Resolution in IP route table of VR
       IP indirect next-hop index 10
       Reachable (metric 3)
       Number of direct next-hops is 1
              Direct next-hop ATM6/1.20 (10.20.20.1)
  Resolution in IP tunnel-route table of VR
       MPLS indirect next-hop index 29
       Reachable (metric 3)
       Number of direct next-hops is 1
              Direct next-hop: MPLS next-hop 23
  Reference count is 1

Indirect next-hop 10.5.5.5
   Resolution in IP route table of VR
       IP indirect next-hop index 5
       Reachable (metric 0)
       Number of direct next-hops is 1
              Direct next-hop ATM6/0.21 (10.5.5.5)
  Resolution in IP tunnel-route table of VR
       MPLS indirect next-hop index 14
       Reachable (metric 0)
       Number of direct next-hops is 1
              Direct next-hop ATM6/0.21.mpls
  Reference count is 3
```

```
host1:asbr2#show mpls next-hop 23
MPLS next-hop: 23, label 33 on ATM6/1.20, nbr 10.20.20.1
  Sent:
```



```
    0 packets
    0 bytes
    0 errors
    0 discards
```

host1:asbr2#**show mpls next-hop 29**

```
MPLS next-hop: 29, resolved by MPLS next-hop 23, peer 1.1.1.1
  MPLS next-hop: 23, label 33 on ATM6/1.20, nbr 10.20.20.1
  Statistics collection is disabled
```

host1:asbr2#**show mpls forwarding brief**

```
....
44       bgp      swap to 16, push 34 on ATM6/1.20, nbr 10.20.20.1
```

host1:asbr2#**show mpls forwarding label 44**

```
In label: 44
  Label space: platform label space
  Owner: bgp
  Spoof check: router ASBR2
  Action:
    MPLS next-hop: 30, label 43, resolved by MPLS next-hop 29
      MPLS next-hop: 29, resolved by MPLS next-hop 23, peer 1.1.1.1
        MPLS next-hop: 23, label 34 on ATM6/1.20, nbr 10.20.20.1
  Statistics:
    0 in pkts
    0 in Octets
    0 in errors
    0 in discard pkts
```

ASBR 3 in turn generates a new label, 50, to advertise with the VPN-IPv4 prefix to its internal MP-BGP peer inside its autonomous system, AS 35. Indirect next hop 4.4.4.4 is for the MP-IBGP peering between PE 4 (loopback address 4.4.4.4) and ASBR 3. Indirect next hop 10.5.5.50 is ASBR 2.

host1:asbr3#**show ip bgp vpn all field out-label**

```
Prefix            Out-label
10.10.10.11/32    44
```

host1:asbr3#**show ip bgp vpn all field in-label**

```
Prefix            In-label
10.10.10.11/32    50
```

host1:asbr3#**show ip bgp vpn all next-hops**

```
Indirect next-hop 4.4.4.4
  Resolution in IP route table of VR
       IP indirect next-hop index 11
       Reachable (metric 3)
       Number of direct next-hops is 1
              Direct next-hop ATM4/0.33 (33.33.33.2)
  Resolution in IP tunnel-route table of VR
       MPLS indirect next-hop index 28
       Reachable (metric 3)
       Number of direct next-hops is 1
              Direct next-hop: MPLS next-hop 22
  Reference count is 1

Indirect next-hop 10.5.5.50
  Resolution in IP route table of VR
       IP indirect next-hop index 4
       Reachable (metric 0)
       Number of direct next-hops is 1
              Direct next-hop ATM6/1.21 (10.5.5.50)
  Resolution in IP tunnel-route table of VR
```



```
       MPLS indirect next-hop index 11
       Reachable (metric 0)
       Number of direct next-hops is 1
              Direct next-hop ATM6/1.21.mpls
  Reference count is 3
```

host1:asbr3#**show mpls forwarding brief**

```
...
50        bgp      swap to 44,  on ATM6/1.21
```

In turn, ASBR 3 receives MPLS frames with label 50 (the in label) from PE 4 and sends MPLS frames with label 44 (the out label) to ASBR 2.

PE 4 receives the VPN-IPv4 prefix with label 50:

host1:pe4#**show ip bgp vpn all field out-label**

```
Prefix             Out-label
10.10.10.11/32     50
```

On PE 4, no in label is associated with the IPv4 prefix 10.10.10.11/32.

host1:pe4#**show ip bgp vpn all field in-label**

```
Prefix             In-label
10.10.10.11/32     none
```

The labels that are generated to be sent to the inter-AS BGP peers are generated for each next-hop PE router/received label tuple. Scaling is improved when all routes advertised from a given VRF have the same label; this is the default E Series router behavior. You can disable this behavior by issuing the **ip mpls forwarding-mode label-switched** command for the VRF.

Related Documentation

- BGP/MPLS VPN Components Overview on page 394
- Understanding IPv4 VPN Services Across Multiple Autonomous Systems on page 412
- Inter-AS Option B Overview on page 414
- Understanding Labels Creation per FEC on page 453

- *ip mpls forwarding-mode label-switched*
- *show ip bgp*
- *show mpls forwarding*
- *show mpls next-hop*

Inter-AS Option C Overview

The third method of configuring inter-AS services and inter-AS VPNs is known as inter-AS option C or 2547bis option C. This method is described in RFC 4364—BGP/MPLS IP Virtual Private Networks (VPNs) (February 2006).

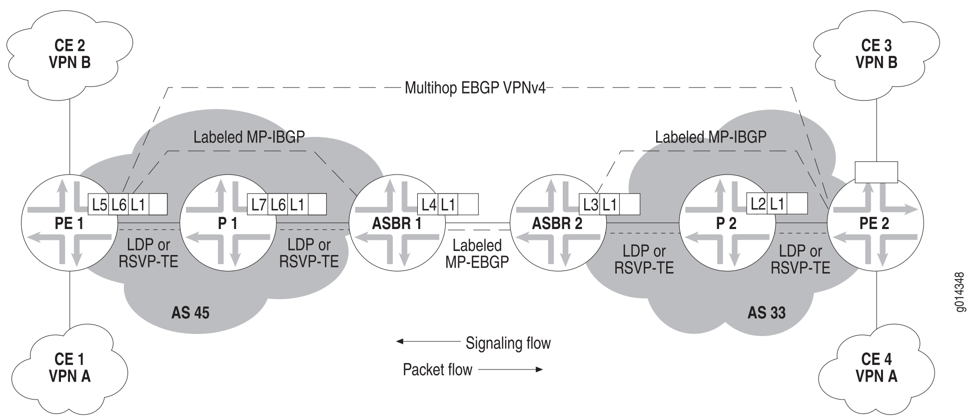
Inter-AS option C, similarly to the carrier-of-carriers configuration, requires a label-switched path from a packet's ingress PE router to its egress PE router. Option C introduces multihop EBGP redistribution of labeled VPN-IPv4 routes between source and destination autonomous systems. Labeled IPv4 routes are redistributed by EBGP between neighboring autonomous systems. Inter-AS option C uses BGP as the label distribution protocol.

In an inter-AS option C network, ASBRs do not maintain or distribute VPN-IPv4 routes. Each ASBR maintains labeled IPv4 /32 routes to the PE routers within its AS. The ASBR distributes these routes to other autonomous systems with EBGP. If transit autonomous systems are included in the topology, their ASBRs must also distribute the labeled /32 routes with EBGP. This configuration creates a label-switched path from the ingress PE router to the egress PE router. This configuration enables the PE routers in different autonomous systems to establish multihop EBGP connections to each other, and to exchange VPN-IPv4 routes over those connections.

Two different configuration scenarios are possible with option C, one employing a two-label stack and the other a three-label stack.

Figure 80 on page 418 illustrates the three-label stack scenario. PHP is not used in this example.

Figure 80: Topology for Three-label Stack Configuration for Inter-AS Option C



In this topology, you can use either LDP or RSVP-TE to establish an LSP between each ASBR router and the PE router in an autonomous system. A labeled MP-IBGP session exists between the ASBR and the PE router in each autonomous system. A labeled MP-EBGP session exists between the two ASBR routers. The ASBR routers advertise the loopback IP addresses of their PE routers and associates the prefixes with labels.

When PE 1 learns the PE 2 loopback address and PE 2 learns the PE 1 loopback address, these PE routers can establish a multihop MP-EBGP session in order to exchange VPN-IPv4 routes. Because VPN-IPv4 routes are only exchanged between end PE routers, no other router on the path from PE 1 to PE 2 needs to keep or install VPN routes in its RIB or FIB.

1. P 2 learns label L2 for the route to the loopback address on PE 2 by means of LDP or RSVP-TE from PE 2.
2. ASBR 2 learns label L3 for the route to the loopback address on PE 2 by means of LDP or RSVP-TE from P 2.

Each ASBR builds its own MPLS forwarding table with the received and advertised routes and labels. ASBR 2 uses its own IP address as the next hop.

3. ASBR 2 uses an MP-EBGP labeled unicast session to advertise label L4 for the route to the loopback address on PE 2 to neighboring ASBR 1.
4. ASBR 1 receives this route and the associated label L4.
5. ASBR 1 assigns label L6 to the route to the loopback address on PE 2 and changes the next-hop address to its own address.
6. ASBR 1 then uses an MP-IBGP session to advertise that address to PE 1. PE 1 therefore has an update with the label information and a next hop to ASBR 1.
7. P 1 learns label L7 for the route to the loopback address on ASBR 1 by means of LDP or RSVP-TE from ASBR 1.
8. PE 1 learns label L5 for the route to the loopback address on ASBR 1 by means of LDP or RSVP-TE from P 1.
9. PE 1 learns label L1 for the VPN-IPv4 route from the multihop EBGP session with PE 2.

Because the routes to the PE routers are unknown to all P routers other than the ASBRs, the ingress PE must push a three-label stack on packets received from the VPN end users. This is illustrated in Figure 80 on page 418 as follows:

1. The first (innermost or bottom) label, L1, is assigned by the egress PE router, PE 2. This label is obtained from the multihop MP-EBGP session. It corresponds to the packet's destination address in a particular VRF at the remote PE router.
2. The middle label, L6, is assigned by ASBR 1. This label is obtained from the MP-IBGP labeled unicast session from the ASBR. It corresponds to the /32 route to the egress PE router, PE 2.
3. The top (outermost) label, L5, is assigned by the ingress PE router's IGP next hop. P 1. This label is obtained from an LDP or RSVP-TE session with the next hop. It corresponds to the /32 route to ASBR 1.

While the packet travels across the VPN from ingress router PE 1, labels are swapped as follows:

1. P 1 swaps outermost label L5 for L7 to get to its next hop, ASBR 1.
2. ASBR 1 pops outermost label L7 and swaps the middle label L6 for L4 to get to ASBR 2.
3. ASBR 2 swaps outer label L4 for L3 to get to its next hop, P 2.
4. P 2 swaps outer label L3 for L2 to get to its next hop, PE 2.
5. PE 2 pops outer label L2 and inner label L1 and then processes the IP data packet.

In contrast to the three-label stack scenario described previously, in a two-label stack scenario, BGP labeled unicast is not used inside the autonomous system. Instead, only LDP is used as the label distribution protocol. A PE router in one AS has a direct LSP to a PE in another AS, achieved by using LDP labels within the AS and BGP labels across the AS boundary.

For a two-label stack scenario to work, you must issue the **mpls ldp redistribute bgp** command on the ASBRs. This command enables the BGP prefixes to be advertised by LDP inside the autonomous systems. For more information on this command, see *mpls ldp redistribute*.

Related Documentation

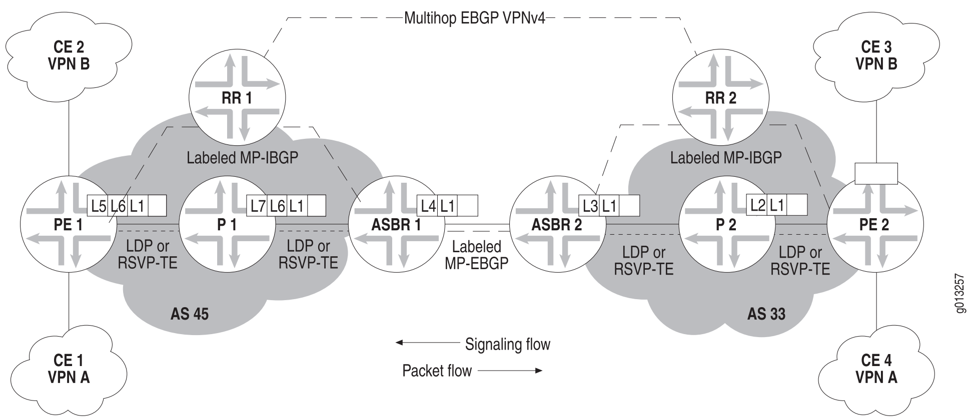
- BGP/MPLS VPN Components Overview on page 394
- Understanding IPv4 VPN Services Across Multiple Autonomous Systems on page 412
- Inter-AS Option A Overview on page 413
- Inter-AS Option B Overview on page 414
- Inter-AS Option C with Route Reflectors Overview on page 420

Inter-AS Option C with Route Reflectors Overview

When the BGP/MPLS VPN peer is a route reflector (Figure 81 on page 420), issue the **neighbor next-hop-unchanged** command to prevent the RR from rewriting the BGP next-hop attribute when the RR advertises routes to external neighbors. Issuing this command causes the VPN RR that is multihop peering with another RR in the AS to send the next hop unchanged for the VPN routes that it advertises.

Issuing this command automatically removes the **neighbor next-hop-self** configuration (enabled or disabled) on the peer or peer group.

Figure 81: Topology for Inter-AS Option C with Route Reflectors



Related Documentation

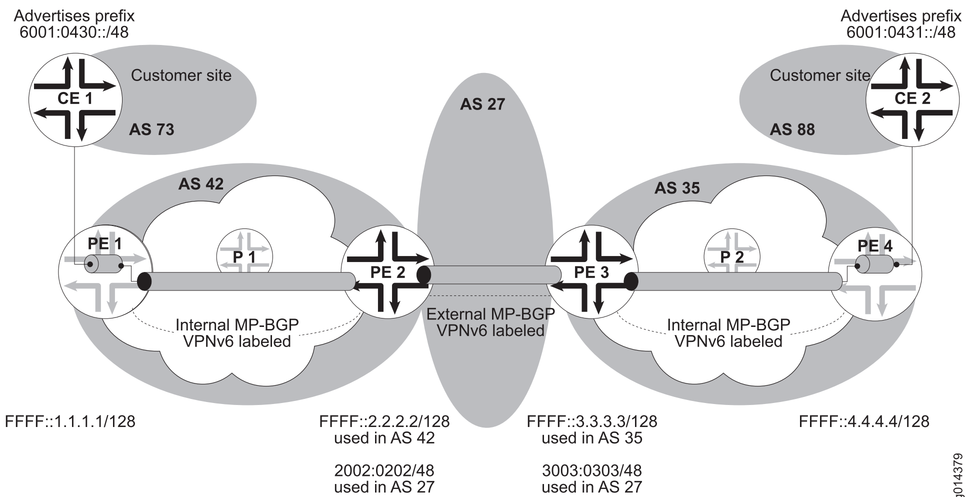
- BGP/MPLS VPN Components Overview on page 394
- Understanding IPv4 VPN Services Across Multiple Autonomous Systems on page 412
- Inter-AS Option C Overview on page 417
- Understanding Route-Target Filtering for MBGP VPNs Overview on page 427
- *neighbor next-hop-self*
- *neighbor next-hop-unchanged*

Understanding IPv6 VPN Services Across Multiple Autonomous Systems

The JunosE Software supports inter-AS services for IPv6 VPNs in addition to IPv4 VPNs. See "Understanding IPv4 VPN Services Across Multiple Autonomous Systems" on page 412 for more information about inter-AS services and IPv4 VPNs.

The JunosE Software currently supports only 2547bis option B for IPv6 VPNs. This method—described in RFC 4364—BGP/MPLS IP Virtual Private Networks (VPNs) (February 2006)—uses BGP to signal VPN labels between the AS boundary routers (Figure 82 on page 421). The base MPLS tunnels are local to each AS. Stacked tunnels run from end to end between PE routers on the different ASs. This method enhances scalability, because only the BGP RIBs store all the inter-AS VPN routes.

Figure 82: Inter-AS IPv6 VPN Services



In Figure 82 on page 421, the base tunnels between the PE routers are established in the IPv4 core networks with LDP or RSVP. The PE routers advertise IPv6 prefixes from the CE devices within their respective ASs as VPNv6 prefixes with MP-IBGP. For example, PE 1 advertises the CE 1 prefix 6001:0430::/48 over to PE 2 in its MP_REACH_NLRI attribute. The next-hop attribute in the update message is the PE 1 loopback address—the IPv4-mapped IPv6 address, FFFF::1.1.1.1/128.

PE 2 advertises 6001:0430::/48 by means of MP-EBGP to PE 3. The prefix is sent as a VPNv6-labeled prefix (2002:0202/48), with the default BGP next hop being the IPv4-mapped IPv6 address of the IPv4 interface going to PE 3.

For inter-AS services, in contrast to intra-AS services, JunosE Software supports both IPv4 backbone and IPv6 backbone types of BGP next-hop encodings. The default BGP next-hop encoding used for IPv6 VPN inter-AS services is the one specified for the IPv4 backbone where IPv4-mapped IPv6 addresses are used. Alternatively, you might also configure the IPv6 backbone type of BGP next-hop encoding by configuring route maps that use native IPv6 addresses for the BGP next hop.

Related Documentation

- MBGP Overview on page 391
- BGP/MPLS VPN Components Overview on page 394
- IPv6 VPN Overview on page 409

VPN Topologies

The following sections explain VPN topologies:

- Full-Mesh VPNs on page 422
- Hub-and-Spoke VPNs on page 423
- VPN Overlap on page 424

Full-Mesh VPNs

In a full-mesh VPN, each site in the VPN can communicate with every other site in that same VPN. For example, in Figure 83 on page 422, each site in VPN A can communicate with all other VPN A sites but not with the sites in VPN B.

Figure 83: Site Connectivity in a Full-Mesh VPN

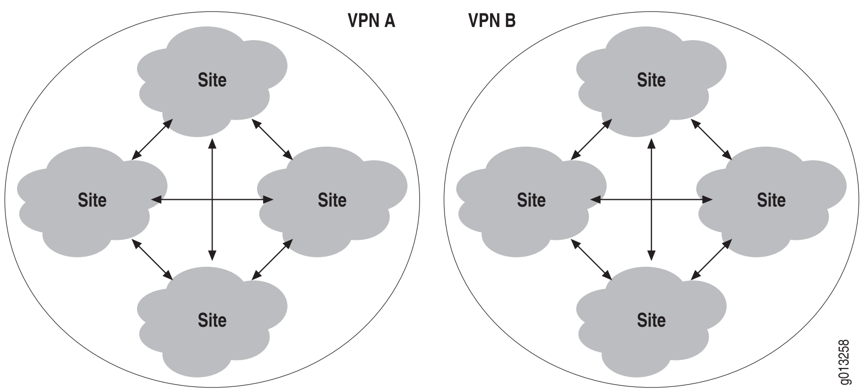
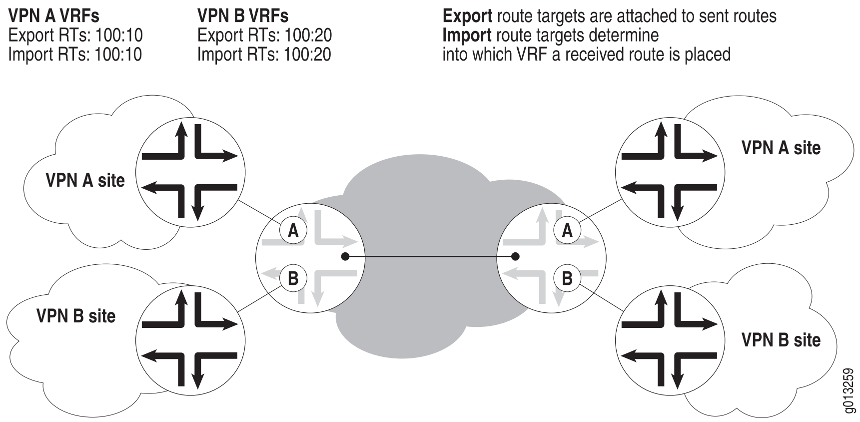


Figure 84 on page 423 illustrates how you can configure the VRF import and export route targets to build a full-mesh VPN. Each VRF in VPN A has the same route target, 100:10, in their import list and export list. Each VPN A VRF accepts only received routes that have this route target attached. Because this route target is attached to each route advertised by VPN A VRFs, every site in VPN A accepts routes only from other sites in VPN A. The same principle applies to VPN B.

Figure 84: Route Target Configuration for a Full-Mesh VPN



Related Documentation

- BGP/MPLS VPN Components Overview on page 394
- Hub-and-Spoke VPNs on page 423
- VPN Overlap on page 424
- Example: Full-Mesh VPNs on page 439

Hub-and-Spoke VPNs

In a hub-and-spoke VPN, the spoke sites in the VPN can communicate only with the hub sites; they cannot communicate with other spoke sites, as shown in Figure 85 on page 423.

Figure 85: Site Connectivity in a Hub-and-Spoke VPN

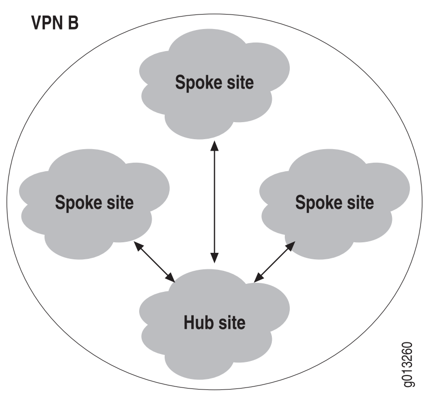
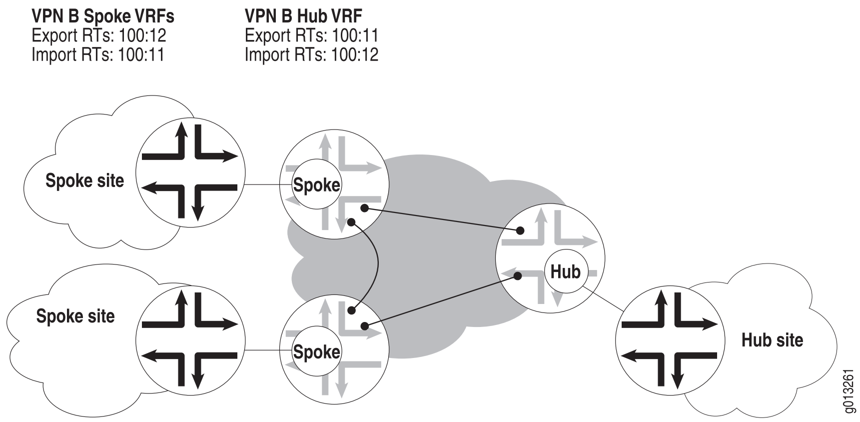


Figure 86 on page 424 shows how to configure the VRF import and export route targets to build a hub-and-spoke VPN. Each spoke VRF has the same export route target, 100:12. The hub VRF has its import route target set to 100:12, so it accepts only routes from the spoke VRFs. Each spoke VRF has the same import route target, 100:11. Every route advertised by any spoke has an attached route target of 100:12. Because that route target does not match the import route target of any spoke, the spokes cannot accept any routes from another spoke. However, the hub VRF has an export route target of 100:11, so routes advertised by the hub do match the import target of each spoke and are accepted by all of the spokes.

Figure 86: Route Target Configuration for a Hub-and-Spoke VPN



Related Documentation

- BGP/MPLS VPN Components Overview on page 394
- Full-Mesh VPNs on page 422
- VPN Overlap on page 424
- Understanding Route-Target Filtering for MBGP VPNs Overview on page 427
- Example: Hub-and-Spoke VPNs on page 441
- Prefix Advertisement with Duplicate AS Numbers Overview on page 463

VPN Overlap

In an overlapping VPN, a site is a member of more than one VPN. For example, in Figure 87 on page 424, the middle site is a member of both VPN A and VPN B. In other words, that site can communicate with all other VPN A sites and all other VPN B sites. An overlapping VPN is often used to provide centralized services. The central site might contain DNS servers or WWW servers or management stations that need to be reachable from multiple VPNs. Overlapping IPv4 and IPv6 VPNs are supported by the same route-target mechanism.

Figure 87: Site Connectivity in an Overlapping VPN

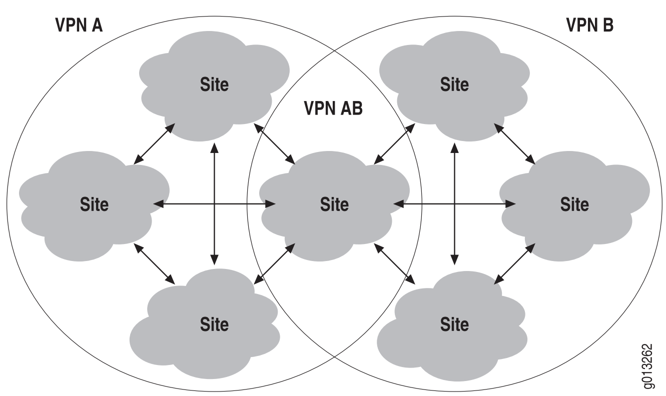
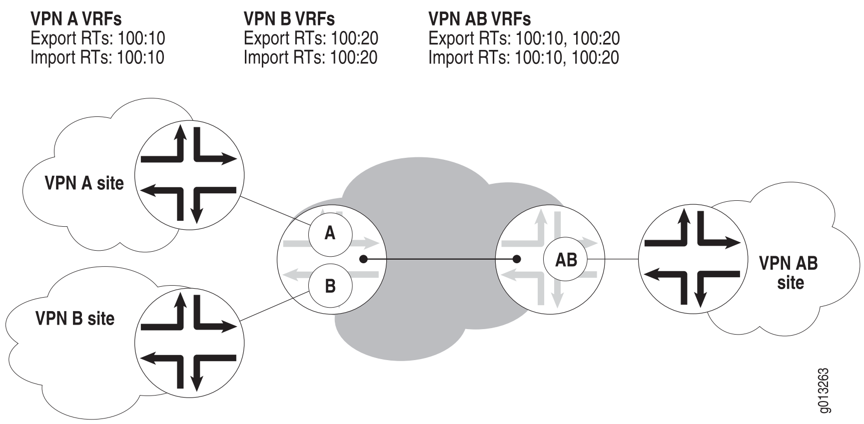


Figure 88 on page 425 shows how to configure the VRF import and export route targets to build an overlapping VPN. In this example, the export and import route targets are different for VPN A and VPN B. Therefore, VPN A does not accept routes from VPN B and VPN B does not accept routes from VPN A.

The import route target list for the overlapping VPN AB includes both 100:10 and 100:20. VPN AB can therefore accept routes advertised by any site in either VPN A or VPN B. Because the VPN AB export route target list also includes both 100:10 and 100:20, every route advertised by VPN AB can be accepted by any site in either VPN A or VPN B.

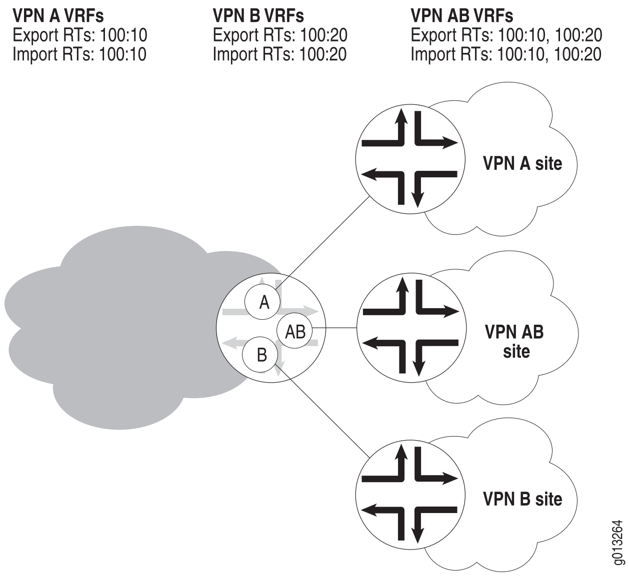
Figure 88: Route Target Configuration for an Overlapping VPN



An interesting special case of an overlapping VPN is when two VRFs on the same PE router belong to the same VPN as shown in Figure 89 on page 426. The configuration of the VRF import and export route targets is the same as for the example in Figure 88 on page 425.

If the export route target of one VRF (for example, the VPN AB VRF) matches the import route target of another VRF (for example, the VPN A VRF), then BGP routes are exported from one VRF to the other VRF; in this case from the VPN AB VRF to the VPN A VRF. Consequently, traffic that arrives in one VRF is forwarded out another VRF without going through the MPLS core network.

Figure 89: Overlapping VPNs on a Single PE



From a given CE router you can ping the local address of any VRF that has a VPN overlapping another VPN to which the CE router belongs.

To achieve this internally, the router obtains the source address as follows:

- If the next-hop interface is in the same VRF and the interface is numbered, the router uses the source address of the interface.
- If the next-hop interface is in the same VRF and the interface is unnumbered, the router uses either the source address of the interface it is pointing to or the router ID of the VRF.
- If the next-hop interface is in a different VRF, the router uses the source address of the VRF. If the router does not have a router ID value, the packet is discarded.

NOTE: The source address of the transmit interface is not used as the source address of the packet.

Related Documentation

- BGP/MPLS VPN Components Overview on page 394
- Full-Mesh VPNs on page 422
- Hub-and-Spoke VPNs on page 423

Route-Target Filtering for MBGP VPNs

The following sections explain route-target filtering for MBGP VPNs:

- Understanding Route-Target Filtering for MBGP VPNs Overview on page 427
- Understanding Route-Target Membership Information Exchange on page 428

Understanding Route-Target Filtering for MBGP VPNs Overview

In typical BGP configurations, you can use cooperative route filtering to reduce the amount of processing required for inbound BGP updates and the amount of BGP control traffic generated by BGP updates. Cooperative route filtering works by having the remote peer install a BGP speaker's inbound route filter as its own outbound route filter. This filtering causes the remote peer to advertise only those routes that the local peer can accept.

For BGP/MPLS VPNs, route-target filtering is a better approach. Route-target filtering controls the distribution of BGP routes based on the VPNS (indicated by the route-target extended communities) to which peer routers belong. PE routers use the MP_REACH_NLRI and MP_UNREACH_NLRI attributes in BGP updates to exchange information about each router's route-target membership.

The PE router subsequently advertises VPN NLRI—the routing information carried in MP-BGP update messages—only to peers that are members of a route target that is associated with the VPN route. The VPN routes flow in the opposite direction to the route-target membership information.

Route-target filtering works across multiple ASs and with asymmetric VPN topologies, such as a hub-and-spoke. Route-target filtering can reduce the size of the BGP routing table in PE routers, as well as the amount of VPN NLRI exchange traffic between routes in the VPN. Route-target filtering also reduces router memory requirements by reducing the amount of routing information stored and propagated. For example, route reflectors scale according to the total number of VPN routes present in their network. With route-target filtering, you can reduce the scaling requirements of the reflectors by restricting the number of VPN routes they must process to only those VPN routes actually used by the route reflector clients.

Applications such as BGP/MPLS VPNs, VPLS L2VPNs, and VPWS L2VPNs all use route targets as part of their route reachability information, and can therefore employ route-target filtering and potentially accrue the benefits of reduced traffic and smaller routing tables.

Related Documentation

- Understanding Route Targets on page 397
- Hub-and-Spoke VPNs on page 423
- Understanding Route-Target Membership Information Exchange on page 428
- Default Route Advertisement Overview on page 431
- Understanding Route Selection When Route-Target Filtering Is Enabled on page 432
- Configuring Route-Target Filtering on page 433

Understanding Route-Target Membership Information Exchange

BGP peers exchange route-target membership information in the following sequence:

1. When the BGP peers negotiate the BGP multiprotocol extensions capability during the establishment of a BGP session, they indicate support for the route-target address family by including the (AFI, SAFI) value pair for the route-target membership NLRI (RT-MEM-NLRI) attribute. This pair has an AFI value of 1 and a SAFI value of 132.
2. If the capability is successfully negotiated, BGP speaker Router A expresses its interest in a VPN route target by advertising to its peers the RT-MEM-NLRI attribute that contains the particular route target. This attribute is represented as a prefix in the following format:

   AS number:route-target extended community/prefix length

   - *AS number*—Number of the originating AS
   - *route-target extended community*—Two-part number identifying the route target extended community. Consists of *number1:number2*, where:
     - *number1*—Autonomous system (AS) number or an IP address
     - *number2*—Unique integer; 32 bits if *number1* is an AS number; 16 bits if *number1* is an IP address
   - *prefix length*—Length of the prefix. A prefix less than 32 or greater than 96 is invalid. However, the prefix for the Default-RT-MEM-NLRI attribute is an exception to this rule. For the Default-RT-MEM-NLRI attribute, 0 is a valid prefix length.

   For example, 100:100:53/36 is a valid RT--MEM-NLRI.
3. Remote peers of Router A use the route-target membership advertised by Router A to filter their VPN routes that are outbound to Router A. A peer advertises a VPN route to Router A only when one of the following conditions is true
   - Router A advertised a default route-target membership.
   - Router A advertised membership in any of the route targets associated with the VPN route.
4. Router A then receives and processes the RT-MEM-NLRI attributes sent by its peers to determine which VPN routes it advertises to the peers.

BGP speakers advertise and withdraw the RT-MEM-NLRI attribute in MP-BGP update messages. BGP speakers ignore RT-MEM-NLRI attributes received from peers that have not successfully negotiated this capability with the speaker.

If dynamic negotiation for the route-refresh capability is enabled, BGP negotiates the route-refresh capability for the RT-MEM-AFI-SAFI address family when a peer is activated in that family. As a consequence, you can use the **clear ip bgp soft** command to refresh the RT-MEM-NLRI routes in the BGP speaker's Adj-RIBs-Out table.

The usefulness of BGP VPN route-target filtering depends on the sparseness of route target membership among the VPN sites. In configurations where VPNs are members of many route target communities—that is, route target membership is dense—the amount of VPN NLRI exchange traffic is about the same regardless of whether route-target filtering is configured.

Related Documentation

- Understanding Route-Target Filtering for MBGP VPNs Overview on page 427
- Understanding RT-MEM-NLRI Routing Updates Exchange on page 429
- Default Route Advertisement Overview on page 431
- Configuring Route-Target Filtering on page 433

- *clear ip bgp*

Understanding RT-MEM-NLRI Routing Updates Exchange

RT-MEM-NLRI routing updates are processed in the following sequence:

1. During the initial RT-MEM-NLRI route exchange that takes place when a session with a peer is being brought up, BGP sends an End-of-RIB marker for RT-MEM-AFI-SAFI that signals it has finished advertising route-target membership information.
2. Remote peers interpret the End-of-RIB marker for RT-MEM-AFI-SAFI to mean that the BGP speaker has advertised all of its route target-memberships. If the BGP speaker does not receive an End-of-RIB marker for RT-MEM-AFI-SAFI from a remote BGP peer in the context of the route-target address family, by default the local BGP speaker waits for 60 seconds before timing out.
3. The BGP speaker then starts advertising its VPN routes. The routes are passed through the outbound route-target membership filters for that peer.
4. When a BGP speaker receives a RT-MEM-NLRI update message, it re-evaluates the advertisement status of VPN routes that match the corresponding route target in the peer's Adj-RIBS-Out table. This can result in an incremental update that advertises or withdraws some routes for the VPN.

You can use the **bgp wait-on-end-of-rib** command to specify how long BGP waits for the End-ofRIB marker from route-target peers.

When the route-refresh capability has been negotiated for the route-target address family. BGP handles route-refresh messages for the RT-MEM-AFI-SAFI by resending all RT-MEM-NLRI routes to the remote peer

You can use the **neighbor maximum-prefix** command to specify the maximum number of prefixes that the speaker can receive from a BGP peer. To prevent a peer from continually flapping, when it goes to state idle because the maximum number of prefixes has been reached, the peer stays in state idle until you use the **clear ip bgp** command to issue a hard clear.

You can specify the **strict** keyword to force BGP to check the maximum prefix against all received routes. The accepted and received routes will likely differ when you have configured inbound soft reconfiguration and route filters for incoming traffic.

Route-target filtering generally follows the standard BGP rules for route advertisement to determine when to advertise RT-MEM-NLRI prefixes that have been received from BGP peers. Table 89 on page 430 lists the destinations that the prefixes are advertised to based on their source. In this table, client-to-client reflection is enabled and the source and destination peers are not the same.

Table 89: Route-Target Filtering Advertisement Rules for Routes Received from Peers

Routes Received From	Advertise to IBGP Route Reflector Client?	Advertise to IBGP Route Reflector Nonclient?	Advertise to EBGP Peer?	Advertise to EBGP Confederation Peer?
IBGP route reflector client	Yes	Yes	Yes	Yes
IBGP route reflector nonclient	Yes	No	Yes	Yes
EBGP peer	Yes	Yes	Yes	Yes
EBGP confederation peer	Yes	Yes	Yes	Yes

Advertising to IBGP clients varies from the standard advertisement rules in terms of path attribute modifications. When locally originated RT-MEM-NLRI routes are advertised to IBGP route reflector clients, BGP does the following:

- Sets the originator ID as the router ID of the advertising router.
- Sets the next hop as the local address of the session.

This behavior is useful when the route reflector does not advertise the Default-RT-MEM-NLRI route.

When locally originated RT-MEM-NLRI routes are advertised to IBGP route reflector nonclients, the route from the client is advertised to the nonclient peer when the best path route is advertised by a nonclient but an alternative route from a client exists. This behavior signals the client's interest in the route target routes that were not selected as the best path.

You cannot filter RT-MEM-NLRI routes with inbound policies or outbound policies, because policy items cannot currently match a RT-MEM-NLRI prefix (origin AS number:route target). However, you can filter route-target filtering routes with policies that include items that match on other BGP attributes, such as the extended community attached to the route-target filtering route.

Related Documentation

- Understanding Route-Target Membership Information Exchange on page 428
- Understanding the Conditions for Advertising RT-MEM-NLRI Routes on page 431
- Configuring Route-Target Filtering on page 433

- *bgp wait-on-end-of-rib*

- *clear ip bgp*
- *neighbor maximum-prefix*

Understanding the Conditions for Advertising RT-MEM-NLRI Routes

The following conditions must be met for routes in the route-target address family to be advertised to a BGP peer:

1. The BGP peers have successfully negotiated the route-target address family.
2. The import route-target list for the IPv4 VRF is not empty or is transitioning to empty.

In a VRF, a RT-MEM-NLRI attribute represented by (origin AS number:route target) is advertised for every route target added to the VRF's route target import list when the preceding conditions have been met.

A withdrawal for the RT-MEM-NLRI attribute is generated when the route target is removed from this VRF's import list.

Related Documentation

- Understanding Route-Target Membership Information Exchange on page 428
- Understanding RT-MEM-NLRI Routing Updates Exchange on page 429
- Configuring Route-Target Filtering on page 433

Default Route Advertisement Overview

You can configure BGP to send a default route to indicate that the speaker accepts routes for any VPNs associated with any route target. For example, this might be desirable for a route reflector advertising to one of its PE router clients, or when a VPN provider is migrating the network to route-target filtering but one or more PEs in the provider's network do not support this feature.

When you configure the default route, the RT-MEM-NLRI attribute contains 0:0:0/0 as the Default-RT-MEM-NLRI. This 4-byte prefix contains only the local (origin) AS number field, set to zero.

By default, BGP does not generate or advertise the Default-RT-MEM-NLRI route. You can use the **default-information originate** command to generate the Default-RT-MEM-NLRI route and send it to all peers. You can use the **neighbor default-originate** command generate the route and send it to a particular peer group. The configuration must be the same for all members of the peer group.

Use the **route-map** keyword to specify outbound route maps to apply to the default route. The route map can modify the attributes of the default route.

This command takes effect immediately. However, if the contents of the route map specified with this command change, the new route map may or may not take effect immediately. If the **disable-dynamic-redistribute** command has been configured, you must issue the **clear ip bgp redistribution** command to apply the changed route map.

Outbound policy configured for the neighbor (using the **neighbor route-map out** command) is applied to default routes that are advertised because of the **default-information originate** command.

Policy specified by a route map with the **default-information originate** command is applied at the same time as the policy for redistributed routes, before any outbound policy for peers.

A BGP speaker sends the Default-RT-MEM-NLRI route only to the peers with which it has negotiated the route-target filtering capability. Any other peers are considered to be unaware of this capability and have no use for that route.

Related Documentation

- Understanding Route-Target Filtering for MBGP VPNs Overview on page 427
- Understanding Route-Target Membership Information Exchange on page 428
- Configuring Route-Target Filtering on page 433

- *clear ip bgp redistribution*
- *default-information originate*
- *disable-dynamic-redistribute*
- *neighbor default-originate*
- *neighbor route-map*

Understanding Route Selection When Route-Target Filtering Is Enabled

When route-target filtering is enabled for a peer, BGP applies outbound filters to initially prevent the speaker from advertising any VPN routes to the peer.

If the BGP speaker subsequently receives a Default-MEM-NLRI route from a peer, BGP applies outbound filters for the peer to prevent route-target filtering from suppressing any VPN routes sent to the peer.

BGP follows the standard route selection process to find the route-target filtering best path for RT-MEM-NLRI routes received from other autonomous systems. The selection is based on the AS path and other MP-NLRI path-attributes attached to the route.

The route-target membership information, which includes the route target and the originator AS number, enables BGP speakers to use the standard path selection rules to remove duplicate, less-preferred paths from the total set of paths to route-target membership peers.

For RT-MEM-NLRI routes that originated within the local AS and are received from an IBGP peer, BGP considers the route-target filtering best path to be the set of all available IBGP paths for the RT-MEM-NLRI prefix. BGP then sets outbound route filters so that VPN routes that match the route target are sent to all IBGP peers that advertised the RT-MEM-NLRI route. This behavior does not affect how the BGP speaker in turn advertises the RT-MEM-NLRI routes.

When BGP selects a RT-MEM-NLRI route from a peer as the best path for the RT-MEM-NLRI prefix, BGP modifies the outbound filters to enable the speaker to advertise to that peer all VPN routes that correspond to that route target. These filters affect the subsequent calculation of the peer's Adj-RIBs-Out entries.

EBGP confederation peers are treated as IBGP peers when the BGP speaker is selecting the route-target filtering best path. When the BGP speaker advertises routes, then the EBGP confederation peers are treated normally, as EBGP peers.

You can control the maximum number of received EBGP best paths that are considered for path selection. The **external-paths** command limits external route target membership, thus controlling the number of EBGP peers that receive the route target VPN routes referenced by the RT-MEM-NLRI route. BGP ignores routes received from the peer after the limit specified with the **external paths** command is reached.

Related Documentation

- Understanding Route-Target Filtering for MBGP VPNs Overview on page 427
- Configuring Route-Target Filtering on page 433

- *external-paths*

Configuring Route-Target Filtering

To configure route-target filtering:

1. Enable the BGP routing process in the specified AS.

   The AS number identifies the PE router to other BGP routers.

   host1(config)#**router bgp 738**

2. Configure the peers for the BGP speaker. Use **neighbor** commands to specify the PE router peers to which BGP advertises routes and to configure any additional BGP attributes.

   host1(config-router)#**neighbor 10.2.2.2 remote-as 45**
   host1(config-router)#**neighbor 10.2.2.2 update-source loopback 0**
   host1(config-router)#**neighbor 10.2.2.2 next-hop-self**

3. Create the route-target address family to configure the router to use BGP signaling to exchange the RT-MEM-NLRI attribute with peer routers.

   Optionally, you can use the **signaling** keyword with the **address-family** command when you configure the route-target address family to specify BGP signaling of reachability information. Currently, you can omit the **signaling** keyword with no adverse effects.

   host1(config-router)#**address-family route-target signaling**

4. Activate the neighbors that routes of the route-target address family are exchanged with for this BGP session. The neighbors must first be created in the default IPv4 unicast address family.

   host1(config-router-af)#**neighbor 10.2.2.2 activate**
   host1(config-router-af)#**neighbor 10.2.2.2 next-hop-self**

5. (Optional) Configure BGP to send a Default-MEM-NLRI route for all peers in the address family or for a specific peer or peer group in the address family.

   host1(config-router-af)#**default-information originate**
   or
   host1(config-router-af)#**neighbor 10.2.2.2 default-originate**

6. Set the maximum number of received external BGP paths that can be accepted for route-target signaling.

   host1(config-router-af)#**external-paths 2**

7. Configure any additional address family parameters desired for the session.

Related Documentation

- Understanding Route-Target Filtering for MBGP VPNs Overview on page 427
- Understanding Route Selection When Route-Target Filtering Is Enabled on page 432

- *address-family*
- *default-information originate*
- *external-paths*
- *neighbor activate*
- *neighbor default-originate*
- *neighbor next-hop-self*
- *neighbor remote-as*
- *neighbor update-source*
- *router bgp*

Configuring BGP VPN Services

The following tasks explain how to configure VPN services:

- Configuring a VRF to Provide BGP VPN Services on page 434
- Configuring a PE Router to Provide BGP VPN Services on page 436

Configuring a VRF to Provide BGP VPN Services

To configure a VRF to provide BGP VPN services:

1. Create the VRF.

   host1(config)#**virtual-router vr1**
   host1:vr1(config)#**ip vrf vrfA**

2. Assign a route distinguisher to the VRF.

   host1:vr1(config-vrf)#**rd 100:100**

3. Set the route-target import and route-target export lists for the VRF.

   host1:vr1(config-vrf)#**route-target import 100:1**
   host1:vr1(config-vrf)#**route-target export 100:1**

4. (Optional) Set import and export maps for the VRF.

    host1:vr1(config-vrf)#**import map Another-route-map**
    host1:vr1(config-vrf)#**export map A-route-map**
    host1:vr1(config-vrf)#**exit**

5. Assign interfaces for PE-to-CE links to the VRF from outside or inside the VRF context:

    host1:vr1(config)#**interface gigabitEthernet 1/0**
    host1:vr1(config-if)#**ip vrf forwarding vrfA**
    host1:vr1:vrfA(config-if)#**ip address 10.16.2.77 255.255.255.0**
    host1:vr1:vrfA(config-if)#**exit**

    or

    host1:vr1(config)#**virtual-router :vrfA**
    host1:vr1:vrfA(config)#**interface gigabitEthernet 1/0**

    NOTE: You can also use the **ip vrf forwarding** command to specify secondary route lookup at the parent (global) level, in the event the original lookup does not yield any results.

6. Use either of the following methods to establish how the VRF learns routes to customer sites:

    - Create static routes to the customer site in the VRF by one of the following methods:

        host1(config)#**virtual-router vr1**
        host1:vr1(config)#**ip vrf vpnA**
        host1:vr1(config-vrf)#**ip route vrf vrfA 10.3.0.0 255.255.0.0 10.1.1.1**
        host1:vr1(config-vrf)#**ip route vrf vrfA 10.12.0.0 255.255.0.0 10.1.1.1**

        or

        host1(config)#**virtual-router vr1:vrfA**
        host1:vr1:vrfA(config)#**ip route 10.3.0.0 255.255.0.0 10.1.1.1**
        host1:vr1:vrfA(config)#**ip route 10.12.0.0 255.255.0.0 10.1.1.1**

    - Configure an IGP on the VRF to learn routes from the CE router.

        See "Configuring the IGP in the VRF Context" on page 451 for examples.

    - Configure a PE-to-CE EBGP session.

        See "Example: Configuring PE-to-CE BGP Sessions" on page 457 for information about configuring EBGP.

7. (Optional) Configure the router to generate a label for each different FEC pointed to by a BGP route in the VPN.

    host1:vr1(config-vrf)#**ip mpls forwarding-mode label-switched**

8. (Optional) For carrier-of-carriers VPNs, configure carrier-of-carriers mode in the provider carrier's PE router that connects to the customer carrier's network.

    host1:vr1:VrfA(config)#**mpls topology-driven-lsp**

See “Carrier-of-Carriers IPv4 VPNs Overview” on page 485 for information about configuring carrier-of-carriers VPNs. See *Creating Multicast VPNs* in *JunosE Multicast Routing Configuration Guide*.

Related Documentation

- Monitoring the VRF on page 513
- *export map*
- *import map*
- *interface gigabitEthernet*
- *ip address*
- *ip mpls forwarding-mode label-switched*
- *ip route*
- *ip vrf*
- *ip vrf forwarding*
- *mpls topology-driven-lsp*
- *rd*
- *route-target*
- *virtual-router*

Configuring a PE Router to Provide BGP VPN Services

To configure a PE router to provide BGP VPN services:

1. Configure PE-to-PE LSPs.

   See “Configuring MPLS” on page 279, for information about configuring LSPs.
2. Enable BGP routing.

   host1:vr1(config)#**router bgp 100**
3. (Optional) Disable automatic route-target filtering.

   host1:vr1(config-router)#**no bgp default route-target filter**
4. Configure PE-to-PE BGP sessions.

   a. Create the PE-to-PE session.

      host1:vr1(config)#**router bgp 100**
      host1:vr1(config-router)#**neighbor 192.168.1.158 remote-as 100**

   b. Create the VPN-IPv4 address family.

      host1:vr1(config-router)#**address-family vpnv4**

   c. Activate the PE-to-PE session in the VPN-IPv4 address family.

      host1:vr1(config-router-af)#**neighbor 192.168.1.158 activate**
      host1:vr1(config-router-af)#**exit-address-family**

d. (Optional) Enable the BGP speaker to check the reachability of indirect next hops when selecting the best VPN-IPv4 route to a prefix.

host1:pe1(config-router-af)#**check-vpn-next-hops**

5. Configure PE-to-CE BGP sessions.

a. Enable and configure BGP:

host1:vr1(config)#**router bgp 100**

See "Configuring BGP Routing" on page 3, for more information about configuring BGP.

b. Specify the IPv4 unicast address family for each VRF:

host1:vr1(config-router)#**address-family ipv4 unicast vrf vrfA**

c. Configure the method of route advertisement by doing one of the following:

- Use **neighbor** commands to specify peers to which BGP advertises the routes:

  host1:vr1(config-router)#**neighbor 10.12.13.0 remote-as 200**

- Use **network** commands or the **redistribute static** command to make BGP advertise static routes to customers.

  host1:vr1(config-router)#**network 10.3.0.0 mask 255.255.0.0**
  host1:vr1(config-router)#**redistribute static**

- Use **redistribute** commands to make BGP advertise IGP routes to customers.

  host1:vr1(config-router)#**redistribute ospf**

6. (Optional) Configure an AS override.

See "Example: Using a Single AS Number for All CE Sites" on page 460 for examples.

7. (Optional) Force the BGP speaker to accept routes that have the speaker's AS number in its AS path.

host1:vr1(config-router)#**bgp enforce-first-as**

See *Creating Multicast VPNs* in *JunosE Multicast Routing Configuration Guide*.

Related Documentation

- *address-family*
- *bgp enforce-first-as*
- *check-vpn-next-hops*
- *exit-address-family*
- *neighbor activate*
- *neighbor remote-as*
- *network*
- *redistribute*
- *router bgp*

Creating a VRF and Assigning a Route Distinguisher

To create a VRF:

1. Access the desired virtual router context.

   host1(config)#virtual-router vr1

2. Create the VRF(s) for that VR.

   host1:vr1(config)#ip vrf vrfA

The route distinguisher enables you to establish unique VPN-IPv4 addresses to accommodate the possibility that more than one VPN might use the same IP address from their private address spaces.

Related Documentation

- Understanding Route Targets on page 397
- Monitoring the VRF on page 513

- *ip vrf*
- *rd*

Definition of Route Targets for VRFs Overview

BGP uses an extended-community attribute, the *route target*, to filter appropriate VPN routes into the correct VRFs. You configure the *export list* on the VRF to specify export route targets. When BGP advertises a route from this VRF's forwarding table, it associates the list of export route targets with the route and includes this attribute in the update message that advertises the route.

You also configure a route-target *import list* on each VRF to specify import route targets. When a PE router receives a route, BGP compares the route target list associated with the route (and carried in the update message) with the import list associated with each VRF configured in the PE router.

An export list defines a route-target extended community. Routes having any route target in their export list that matches a route target in a VRF's import list are installed in the VRF's forwarding table.

An import list defines a route-target extended community. Only routes that have at least one matching route target in their associated export list can be installed into the VRF's forwarding table.

If the import and export lists are identical, use the **both** keyword to define both lists simultaneously.

You can add only one route target to a list at a time.

For VPN-IPv4 routes received from another PE router, if *any* route target in the export list matches a route target in a VRF's import list, then the route is installed in that VRF's forwarding table.

A route-target export list can be modified on the sending PE router by an export map or outbound routing policy. It can be modified on the receiving PE router by an import map or inbound routing policy.

Related Documentation

- Understanding Route Targets on page 397
- Defining Route Targets for VRFs on page 439
- Disablement of Automatic Route-Target Filtering on page 452

Defining Route Targets for VRFs

To define route targets for VRFs, do the following:

1. Allocate one route-target extended-community value per VPN.
2. Define the route-target import list and a route-target export list to include only the route-target extended-community values for the VPN(s) to which the VRF belongs:

   host1:vr1(config-vrf)#**route-target export 777:100**
   host1:vr1(config-vrf)#**route-target import 777:100**

   If the import and export lists are identical, you can use the **both** keyword to define the lists simultaneously:

   host1:vr1(config-vrf)#**route-target both 777:105**

A route-target export list can be modified on the sending PE router by an export map or outbound routing policy. It can be modified on the receiving PE router by an import map or inbound routing policy.

Related Documentation

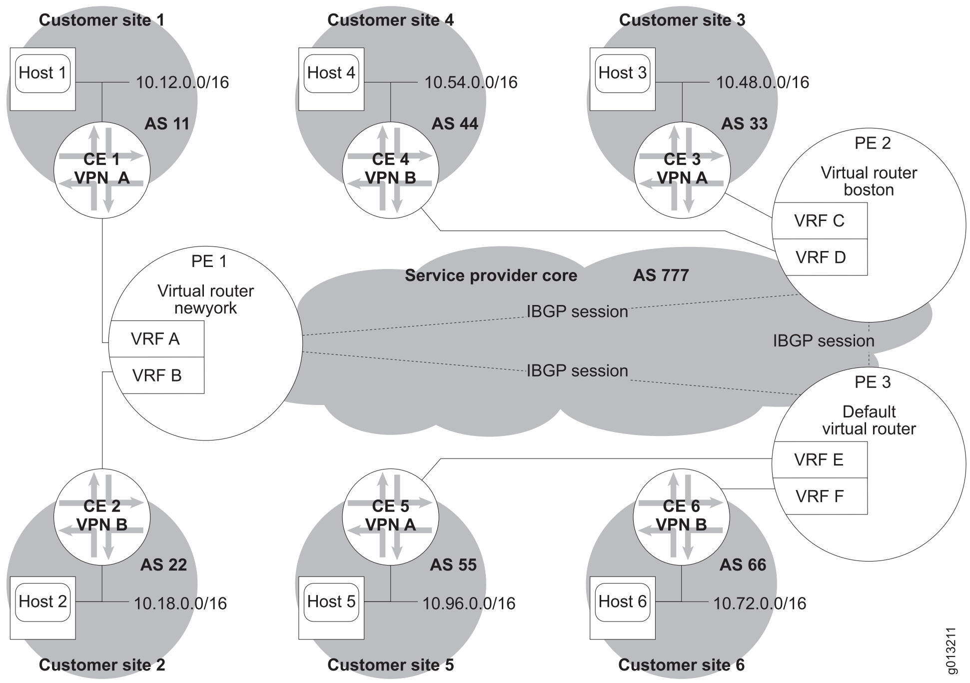
- Definition of Route Targets for VRFs Overview on page 438
- Monitoring the VRF on page 513

- *route-target*

Example: Full-Mesh VPNs

In a fully meshed VPN, each site in the VPN can reach every other site in the VPN. Figure 90 on page 440 illustrates a situation with two fully meshed VPNs, VPN A and VPN B. VPN A includes Customer Sites 1, 3, and 5 through VRFs A, C, and E. VPN B includes Customer Sites 2, 4, and 6 through VRFs B, D, and F.

Figure 90: Fully Meshed VPNs



BGP sessions exist between PE 1 and PE 2, PE 2 and PE 3, and PE 3 and PE 1. The MPLS paths through the service provider core are omitted for clarity.

To configure route targets for this fully meshed scenario, you specify the same route target for the import list and export list on all VRFs in VPN A. The VRFs in VPN B use a different route target, but it is the same for the import list and export list for all.

Route-target configuration on PE 1:

```
host1(config)#virtual-router newyork
host1:newyork(config)#ip vrf vrfA
host1:newyork(config-vrf)#route-target both 777:1
host1:newyork(config-vrf)#exit
host1:newyork(config)#ip vrf vrfB
host1:newyork(config-vrf)#route-target both 777:2
```

Route-target configuration on PE 2:

```
host2(config)#virtual-router boston
host2:boston(config)#ip vrf vrfC
host2:boston(config-vrf)#route-target both 777:1
host2:boston(config-vrf)#exit
host2:boston(config)#ip vrf vrfD
host2:boston(config-vrf)#route-target both 777:2
```

Route-target configuration on PE 3:

```
host3(config)#ip vrf vrfE
host3(config-vrf)#route-target both 777:1
host3(config-vrf)#exit
host3(config)#ip vrf vrfF
```



```
host3(config-vrf)#route-target both 777:2
```

Related Documentation

- Full-Mesh VPNs on page 422
- Monitoring the VRF on page 513
- *ip vrf*
- *route-target*
- *virtual-router*

Example: Hub-and-Spoke VPNs

In one type of a hub-and-spoke design, only the hub site can reach every site in the VPN. All other sites—spokes—can reach only the hub site. (More complex hub-and-spoke designs are possible, but require additional configuration and route targets to achieve.) In Figure 91 on page 442, Customer Site 1 is the hub site for VPN A. As such it can reach both spokes, Customer Sites 2 and 3 through VRF A. Customer Site 2 can reach only the hub, customer 1, through VRF C. Customer Site 3 can reach only the hub, customer 1, through VRF E.

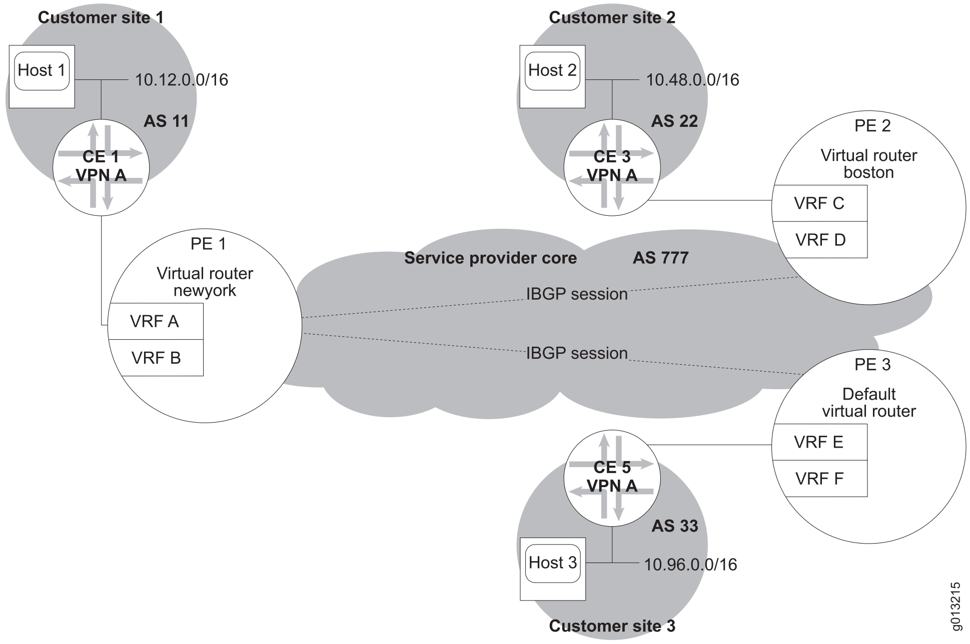
BGP sessions exist between PE 1 and PE 2 and between PE 1 and PE 3. In most situations, BGP itself is fully meshed, but that level of complexity is not necessary for this example. The MPLS paths through the service provider core are omitted for clarity.

To configure route targets for this hub and spoke, you specify different import and export route targets on the hub VRF. On the spoke VRFs, you switch these route targets.

Route-target configuration on PE 1:

```
host1(config)#virtual-router newyork
host1:newyork(config)#ip vrf vrfA
host1:newyork(config-vrf)#route-target export 777:25
host1:newyork(config-vrf)#route-target import 777:50
```


Figure 91: Hub-and-Spoke VPN



Route-target configuration on PE 2:

```
host2(config)#virtual-router boston
host2:boston(config)#ip vrf vrfC
host2:boston(config-vrf)#route-target export 777:50
host2:boston(config-vrf)#route-target import 777:25
```

Route-target configuration on PE 3:

```
host3(config)#ip vrf vrfE
host3(config-vrf)#route-target export 777:50
host3(config-vrf)#route-target import 777:25
```

This configuration ensures that when VRF E on PE 3 receives an update message from PE 1, BGP installs the advertised route only if it has a route target of 25. Routes from PE 2 have a route target of 50, and cannot be installed. Similarly, when VRF C on PE 2 receives an update message from PE 1, BGP installs the advertised route only if it has a route target of 25. Routes from PE 3 have a route target of 50, and cannot be installed. When PE 1 receives updates from either PE 2 or PE 3, the routes have a route target of 50, match VRF A's import list, and are installed in VRF A's forwarding table.

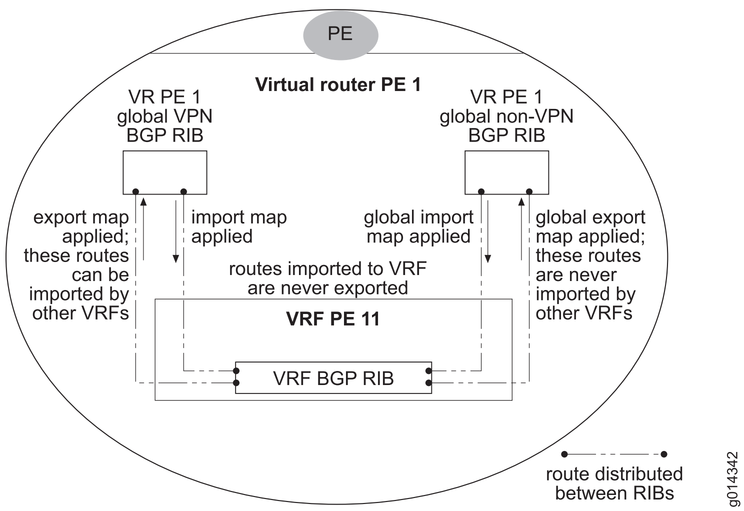
Related Documentation

- Hub-and-Spoke VPNs on page 423
- Monitoring the VRF on page 513
- *ip vrf*
- *route-target*
- *virtual-router*

Understanding Route Distribution for a VRF using Maps

The combination of the route-target export list of VRF A and the route-target import list of VRF B determines whether routes from VRF A are distributed to VRF B. You can provide finer-grained control of route distribution by associating any combination of export, import, global export, and global import maps with VRFs. As shown in Figure 92 on page 443, a route is distributed (leaked) between RIBs and its attributes are changed as specified in the route map when the map returns an accept message. If the map returns a deny message, then the route is not distributed.

Figure 92: Import and Export Maps



Both IPv4 and IPv6 VPNs are supported. You can specify that only IPv4 or only IPv6 routes are imported or exported. By default, the import or export map applies to both kinds of routes. You can configure some maps to apply to IPv4 routes and different maps to apply to IPv6 routes.

When the name or the contents of a route map change, BGP automatically waits for a nonconfigurable hold-down interval of 30 seconds and then re-imports or re-exports the appropriate routes using the modified route map.

Even when suppressed by an aggregate or auto-summary route, the more specific routes are distributed. Aggregation and auto-summarization take place in each VRF independently. For example, a route that is imported into a VRF is only aggregated in that VRF if an aggregate address has been configured in the context of the BGP address family for that VRF.

Routes maintain their type when exported. Private prefixes are exported without being converted into public prefixes. Consequently the prefix of an exported route is the same as the original route. Global export maps are therefore not useful when NAT is enabled.

Subsequent Distribution of Routes

Routes that are imported from the global BGP non-VPN RIB (with a global import map) into a VRF RIB are never exported again. Because these routes are not exported to the global VPN RIB, they are not advertised to other PE routers. These imported routes are never exported to the VRF RIBs of overlapping VPNs.

Routes that are exported from a VRF RIB to the global BGP non-VPN RIB with the global export map are never imported back in to any VRF.

Routes that are imported from the global BGP VPN RIB (with an import map) into a VRF RIB are never exported again.

Routes that are exported from a VRF RIB to the global VPN RIB can be imported into the RIB of other VRFs. This behavior might be seen with overlapping VPNs.

Related Documentation

- Characteristics of Import and Global Import Maps on page 444
- Characteristics of Export and Global Export Maps on page 445
- Assigning a Route Map to the VRF on page 445
- Types of Maps Overview on page 446
- Example: Configuring a Global Import Map for Specific Routes on page 478
- Traffic Flow from the Internet to the VPN Overview on page 483

Characteristics of Import and Global Import Maps

Import maps and global import maps can import both labeled and unlabeled routes. If you want to import only one or the other, you can use a **match mpls-label** command in the global import route map. Furthermore, if BGP imports labeled routes from the global BGP non-VPN RIB into a VRF RIB and then advertises them further upstream as labeled routes, the MPLS cross-connects are correctly created and MPLS forwarding works. The global VPN RIB never contains unlabeled routes, so the issue is moot for import maps.

When a route that was previously imported into the local VRF RIB is modified in the global BGP RIB (VPN or non-VPN) such that it no longer matches the import or global import map, that route is removed from the local VRF RIB.

Imported routes point to the same interface and next hop as the original route. Shared IP interfaces are not created.

Table 90 on page 444 lists additional characteristics of import and global import maps.

Table 90: Characteristics of Import and Global Import Maps

Characteristic	Import	Global Import
Distributes routes from the global BGP VPN RIB local to the VR. This RIB is often referred to as the core VPN RIB.	Yes	–
Distributes routes from the global BGP non-VPN RIB local to the VR. This RIB is often referred to as the core non-VPN RIB or core RIB.	–	Yes
Imports all types of routes (received routes, redistributed routes, network routes, aggregate routes, and auto-summary routes).	Yes	Yes

Table 90: Characteristics of Import and Global Import Maps *(continued)*

Characteristic	Import	Global Import
Imports both best and non-best routes. The best route selection (including the decision to use or not use ECMP) is made in the VRF after the routes are imported.	Yes	Yes

Related Documentation

- Understanding Route Distribution for a VRF using Maps on page 443
- Types of Maps Overview on page 446
- *match mpls-label*

Characteristics of Export and Global Export Maps

Export maps and global export maps can export both labeled and unlabeled routes. If you want to export only one or the other, you can use a **match mpls-label** command in the export or global export route map.

Table 91 on page 445 lists additional characteristics of export and global export maps.

Table 91: Characteristics of Export and Global Export Maps

Characteristic	Export	Global Export
Distributes routes to the global BGP VPN RIB local to the VR. This RIB is often referred to as the core VPN RIB.	Yes	–
Distributes routes to the global BGP non-VPN RIB local to the VR. This RIB is often referred to as the core non-VPN RIB or core RIB.	–	Yes
Exports all types of routes (received routes, redistributed routes, network routes, aggregate routes, and auto-summary routes).	Yes	–
Exports only locally originated routes (all routes other than those that have been received).	–	Yes
Exports both best and non-best routes. The best route selection is made again in the core after the export.	Yes	Yes

Related Documentation

- Understanding Route Distribution for a VRF using Maps on page 443
- Types of Maps Overview on page 446
- *match mpls-label*

Assigning a Route Map to the VRF

For information about creating a route map to be used as an import or export map, see "Configuring BGP Routing" on page 3. The following example shows how to apply the route map *routemap5* to the VRF *vpnA* configured on the virtual router *boston*.

1. Configure the virtual router

   host1(config)#virtual router boston

2. Configure the VRF instance

   host1:boston(config)#ip vrf vpnA

3. Apply the route map to the VRF instance configured on the virtual router

   host1:boston(config-vrf)#import map routemap5

Related Documentation

- Types of Maps Overview on page 446
- Monitoring the VRF on page 513

- *import map*
- *export map*

Types of Maps Overview

VPNv6 routes can be exported from the BGP RIB of an IPv6 VRF to the global IPv6 BGP RIB based on policy by means of a route map and the **global export map** command.

For example, if you have a mixed IPv4 and IPv6 VPN configuration, but want only the IPv6 VPN routes to be exported from the IPv6 VRF into the global IPv6 RIB, you can use a route map that matches on IPv6 access-lists (IPv6 prefix-lists). You can have the route map disallow IPv4 VPN routes by matching on IPv4 access lists that filter out IPv4 prefixes.

VPNv6 routes can be exported from the BGP RIB of an IPv6 VRF to the global IPv6 BGP RIB based on policy by means of a route map and the **global export map** command.

For example, if you have a mixed IPv4 and IPv6 VPN configuration, but want only the IPv6 VPN routes to be exported from the IPv6 VRF into the global IPv6 RIB, you can use a route map that matches on IPv6 access-lists (IPv6 prefix-lists). You can have the route map disallow IPv4 VPN routes by matching on IPv4 access lists that filter out IPv4 prefixes.

Export Maps

You can use an export map to change the attributes of a route when it is exported from a VRF to the global BGP VPN RIB local to the VR. This RIB is often referred to as the core VPN RIB. Export maps can optionally filter routes.

When the VRF route matches the export map, the route is exported and the attributes are changed as specified in the export map.

When the VRF route does not match the export map, the **filter** keyword determines what happens. If the **filter** keyword has been issued, then the route is not exported. If the **filter** keyword has not been issued, then the route is exported but the attributes of the route are not modified (because the export map was not matched).

If you do not configure an export map, then all routes are exported from the VRF to the global BGP VPN RIB. However, routes that are imported into the VRF cannot be exported again.

Global Export Maps

You can use a global export map to change the attributes of a route when it is exported from a VRF to the global BGP non-VPN RIB local to the VR.

If the VRF route matches the export map, then the route is exported and the attributes are changed as specified in the export map. If the VRF route does not match the export map, then the route is not exported. If you do not configure a global export map, then no routes are exported from the VRF to the global BGP non-VPN RIB.

Routes that are imported into the VRF cannot be exported again. As a consequence, VPN routes can be injected only into the global IP routing table on the PE router that is directly connected to the CE router that originates the prefix.

See "Exporting IPv6 VPN Routes Globally into the Global BGP IPv6 RIB" on page 448 for information about global export maps and IPv6 VPNs.

Import Maps

You can use an import map to change the attributes of a route when it is imported from the global BGP VPN RIB to a VRF. You can also use an import map to filter routes. If you associate an import map with a VRF, that VRF then accepts only received routes that pass the import map (and match the import route target list).

Global Import Maps

Global import maps enable BGP routes to be imported from the global BGP non-VPN RIB into the BGP RIB of a VRF based on a configured route map. You can use import maps as an automated mechanism that enables a subset of the Internet to be reachable from a VPN. This feature is intended to provide simplified central access to a limited number of centralized services in the provider network. Use this feature to import only a relatively small number (tens) of routes from the global domain into the VPNs, such as a small number of routes to DNS servers, content servers, management stations, and so on.

If instead you import the full Internet routing table into one or more VPNs, too much memory will be consumed because this action stores multiple copies of the full Internet routing table. To prevent an accidental misconfiguration, you must specify the maximum number of routes to be imported into a VRF when you configure global import. If you must provide access to the full Internet from a VPN, use the **fallback global** command.

Use the **max-routes** keyword to specify the maximum number of routes that you want to be imported into the local RIB. BGP generates a log message when the specified number of routes has been imported; no additional routes are imported.

Related Documentation

- Understanding Route Distribution for a VRF using Maps on page 443
- Characteristics of Import and Global Import Maps on page 444
- Characteristics of Export and Global Export Maps on page 445
- Configuring Secondary Routing Table Lookup on page 449

- *global export map*

Exporting IPv6 VPN Routes Globally into the Global BGP IPv6 RIB

To export only IPv6 VPN routes from the IPv6 VRF into the global IPv6 RIB:

1. Configure an IPv6 access list to export IPv6 VPN prefixes to the global IPv6 RIB.

   host1(config)#**ipv6 access-list everything-v6 permit any any**

2. Configure an IPv4 access list to disallow the export of IPv4 prefixes to the global IPv4 RIB.

   host1(config)#**access-list nothing-v4 deny ip any any**

3. Configure a route map to permit global export of IPv6 VPN routes to the global IPv6 RIB.

   host1(config)#**route-map export-only-v6**
   host1(config-route-map)#**match ip address nothing-v4**
   host1(config-route-map)#**match ipv6 address everything-v6**
   host1(config-route-map)#**set local-preference 444**
   host1(config-route-map)#**exit**
   host1(config)#**ip vrf foo**
   host1(config-route-vrf)#**global export map export-only-v6**

If you need to export both IPv4 and IPv6 VPN routes from the IPv4/IPv6 VRF to the global IPv4 BGP RIB and to the global IPv6 BGP RIB, then configure a route map that permits both IPv4 and IPv6 prefixes.

Related Documentation

- Monitoring the VRF on page 513
- *access-list*
- *global export map*
- *route-map*

Assigning an Interface to a VRF

You must assign an interface or subinterface to a VRF so that when the router receives a packet at this interface, it routes the packet using the VRF's forwarding table rather the global forwarding table. You can assign the interface from outside the context of the VRF or inside the context of the VRF.

To assign an interface to a VRF from outside the VRF context:

1. Select the interface.
2. Specify the VRF to associate with the interface.

   host1:vr1(config)#**interface gigabitEthernet 1/0**
   host1:vr1(config-if)#**ip vrf forwarding vrfA**

3. Assign an IP address to the interface because forwarding the interface from the VR to the VRF removes the existing IP configuration from the interface.

   host1:vr1:vrfA(config-if)#**ip address 10.16.2.77 255.255.255.0**

To assign an interface to a VRF from inside the VRF context:

1. Select the interface.
2. Enter the VRF context.

   host1:vr1(config)#**virtual-router :vrfA**
3. Associate the interface.

   host1:vr1:vrfA(config)#**interface gigabitEthernet 1/0**

   In this case, you do not have to reassign an IP address to the interface because you did not use the **ip vrf forwarding** command.

Related Documentation

- Monitoring the VRF on page 513
- Monitoring the VRF Routing Table on page 512
- *ip vrf forwarding*

Configuring Secondary Routing Table Lookup

You can enable secondary routing table lookup on the virtual router routing table of the parent (global) virtual router. The secondary lookup takes place when the initial route lookup on a VRF is unsuccessful. You can define secondary routing table lookup outside the context of the VRF or inside the context of the VRF.

To configure secondary routing table lookup from outside the VRF context:

1. Select the interface.

   host1:vr1(config)#**interface gigabitEthernet 1/0**
2. Specify a VRF and that you want it to perform secondary routing table lookup.

   host1:vr1(config-if)#**ip vrf forwarding vrfA fallback global**
   host1:vr1:vrfA(config-if)#**ip address 10.12.4.5 255.255.255.0**

To specify from inside the VRF context that an interface use the fallback global routing table lookup:

1. Select the interface.

   host1:vr1(config)#**interface gigabitEthernet 1/0**
2. Enter the VRF context.

   host1:vr1(config-if)#**virtual-router :vrfA**
3. Specify that the VRF perform a secondary routing table lookup.

   host1:vr1:vrfA(config-if)#**ip fallback global**

Related Documentation

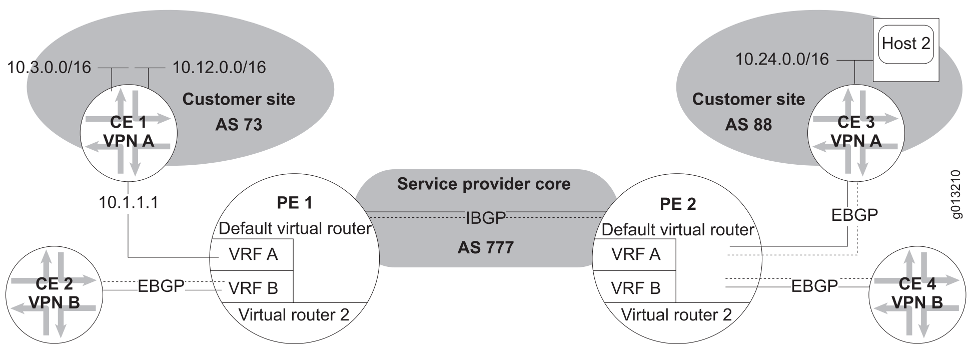
- Example: Configuring a Fallback Global Option on page 477
- Monitoring the VRF on page 513
- Monitoring the VRF Routing Table on page 512
- *ip fallback global*
- *ip vrf forwarding*

Example: Adding Static Routes to a VRF

Consider the network structure shown in Figure 93 on page 450. If no routing protocol—BGP or any other IGP—is running between the PE router and the CE router, you must use the **ip route vrf** command to add a static route in the customer's VRF for each prefix in that customer's site.

Each of these static routes must point to the link connecting the PE router to the CE router. Typically, you redistribute these static routes in the VRF's address family in BGP or use **network** commands to make those prefixes reachable from other CE routers in the same VPN.

Figure 93: Configuring Static Routes



In Figure 93 on page 450, PE 2 has external BGP connections to CE 3 and CE 4. PE 1 has an EBGP connection to CE 2. However, no BGP (or IGP) connection exists between PE 1 and CE 1. The following example shows how to configure static routes on VRF A for both prefixes in CE 1.

```
host1(config)#virtual-router pe1
host1:pe1(config)#ip vrf vpnA
host1:pe1(config-vrf)#ip route vrf vrfA 10.3.0.0 255.255.0.0 10.1.1.1
host1:pe1(config-vrf)#ip route vrf vrfA 10.12.0.0 255.255.0.0 10.1.1.1
```

Related Documentation

- Route Advertisements to Customers Overview on page 458
- Monitoring the VRF Routing Table on page 512
- *ip route*
- *ip vrf*
- *virtual-router*

Configuring the IGP in the VRF Context

If you do not configure static routes on the VRF for each prefix in the associated customer site, then you must configure an IGP on the VRF so that the VRF can learn routes from customer sites.

1. After creating a VRF, you can access it as if it were a virtual router for the purpose of configuring the IGP.
   - If you are in the context of the virtual router that has the VRF, you access the VRF as follows:

     host1(config)#**virtual-router :vrfa**
     host1:default:vrfa(config)#

   - If you are *not* in the context of the virtual router that has the VRF, you access the VRF as follows:

     host1(config)#**virtual-router boston:vrfa**
     host1:boston:vrfa(config)#

2. The following configuration shows how to configure OSPF. You can similarly configure RIP and IS-IS.
   1. Configure the VRF instance.

      host1(config)#**ip vrf vrfa**

   2. Specify the unique two-part route distinguisher for a VRF.

      host1(config-vrf)#**rd 100:5**

   3. Configure the OSPF routing process.

      host1:default:vrfa(config)#**router ospf 100**

   4. Add the route target to the import and export list of the VRF.

      host1(config-vrf)#**route-target both 100:5**
      host1(config-vrf)#**exit**

   5. Access the VRF from the VR context.

      host1(config)#**virtual-router :vrfa**

   6. Redistribute routes sourced from BGP.

      host1:default:vrfa(config-router)#**redistribute bgp**

At this point you proceed with the IGP configuration for the VRF.

Related Documentation

- Monitoring the VRF on page 513
- *virtual-router*

Configuring the IGP Outside the VRF Context

The RIP and OSPF protocols also enable you to specify a VRF and configure the protocol without actually entering the VRF context.

For example, for OSPF you might issue the following command and then complete OSPF configuration tasks for VRF A:

```
host1(config)#router ospf 100 vrf vrfa
```

For RIP, you create the RIP process, specify the address family for the VRF, and specify redistribution of BGP routes for VRF A:

```
host1(config)#router rip 100
host1(config-router)#address-family ipv4 vrf vrfa
host1(config-router-af)#redistribute bgp
```

At this point you proceed with RIP configuration for the VRF. For information about configuring IS-IS, OSPF, or RIP as the IGP, see the cit-junose-ip-ip6-igp;. For information about configuring BGP as the IGP, see "Configuring BGP Routing" on page 3.

Related Documentation

- *virtual-router*

Disablement of Automatic Route-Target Filtering

When BGP receives a VPN-IPv4 or VPN-IPv6 route from another PE router, BGP stores that route in its local routing table only if at least one VRF imports a route target of that route. If no VRF imports any of the route targets of the route, BGP discards the route; this feature is called automatic route-target filtering. The intention is that BGP keeps track of routes only for directly connected VPNs, and discards all other VPN-IPv4 or VPN-IPv6 routes to conserve memory.

If a new VPN is connected to the router (that is, if the import route-target list of a VRF changes), BGP automatically sends a route-refresh message to obtain the routes that it previously discarded.

You can use the **no bgp default route-target filter** command to disable automatic route-target filtering globally for all VRFs. However, automatic route-target filtering is always disabled on route reflectors that have at least one route-reflector client. You cannot enable automatic route-target filtering for such route reflectors.

If route-target filtering is turned off, BGP automatically sends out a route-refresh message over every VPNv4 or VPNv6 unicast session (for which the route-refresh capability was negotiated) to get previously filtered routes. If the route-refresh capability was not negotiated over the session, BGP bounces the session.

Related Documentation

- Definition of Route Targets for VRFs Overview on page 438
- *bgp default route-target filter*

Understanding Labels Creation per FEC

By default, the router minimizes the number of stacked labels to be managed by generating a single label for all BGP routes advertised by a given VRF; this is a per-VRF label. Upon receiving traffic for a per-VRF label, the router performs a label pop and a route lookup to forward the traffic to the next hop.

You can use the **ip mpls forwarding-mode label-switched** command to configure the router to generate a label for each different FEC that a BGP route points to in the VPN; this is a per-FEC label. Issuing this command enables you to avoid a route lookup for traffic destined for CE routers, because in this mode traffic is label switched to the corresponding next hop over that interface; a route lookup is not performed.

The route for which a label is allocated can be an ECMP route; in that case, the label-switched traffic uses ECMP.

For the following types of routes, the router always generates a per-VRF label and forwards traffic after a route lookup (rather than label switching the traffic without a route lookup) regardless of the status of this command:

- Local connected interfaces redistributed into BGP, regardless of the interface type.
- BGP redistributed routes that point to loopback interfaces.

Related Documentation

- Example: Distribution of Routes and Labels with BGP on page 398
- Packet Transport Across an IP Backbone with MPLS Overview on page 403
- Example: Intra-AS Option B IPv4 VPNs on page 414
- Creating Labels per FEC on page 453

- *ip mpls forwarding-mode label-switched*

Creating Labels per FEC

The following configuration is used to configure the router to generate a label for each different FEC that a BGP route points to in the VPN. The following commands configure a router where BGP is running in VRF pe11 and static and connected routes are redistributed into the VRF:

1. Configure the VRF.

   host1(config)#ip vrf pe11

2. Configure the router to generate a label for each different FEC that a BGP route points to in a VPN.

   host1(config-vrf)#ip mpls forwarding-mode label-switched

3. Establish the static routes for the VRF interface.

   host1(config-vrf)#ip route vrf pe11 10.3.4.5 255.255.255.255 fastEthernet 0/1
   host1(config-vrf)#ip route vrf pe11 10.1.1.1 255.255.255.255 loopback 1

host1(config-vrf)#exit

4. Configure the BGP routing process.

   host1(config)#router bgp 100

5. For the IPv4 address family, configure the router to exchange IPv4 addresses in unicast.

   host1(config-router)#address-family ipv4 unicast vrf pe11
   host1(config-router-af)#exit

6. Configure the router to advertise the network routes without waiting for IGP synchronization with BGP.

   host1(config-router)#no synchronization

7. Configure the router to redistribute routes that are established automatically when IP is enabled on an interface.

   host1(config-router)#redistribute connected

8. Disable automatic synchronization.

   host1(config-router)#no auto-summary

9. Configure the router to redistribute static routes.

   host1(config-router)#redistribute static

For each connected route that is redistributed into the VRF and advertised across the BGP/MPLS VPN, the router assigns a per-VRF label rather than a per-FEC label.

The static route 10.1.1.1/32 points to loopback interface 1. BGP therefore advertises this static route with a per-VRF label.

Related Documentation

- Understanding Labels Creation per FEC on page 453
- Monitoring the VRF Routing Table on page 512

- *ip mpls forwarding-mode label-switched*

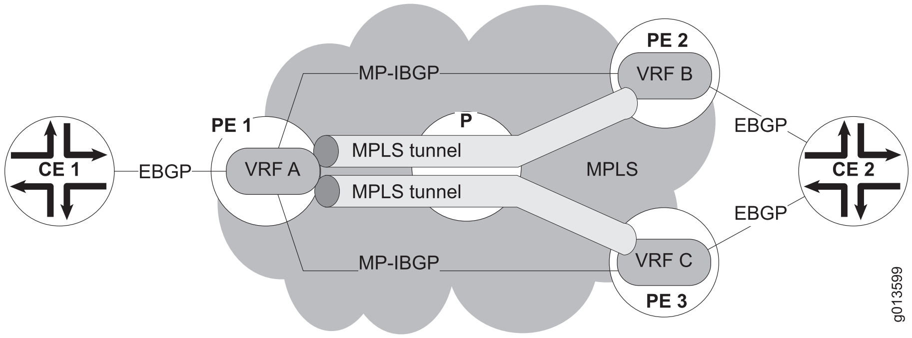
Example: Enabling BGP ECMP for BGP/MPLS VPN IBGP

Enabling ECMP support for BGP/MPLS VPNs allows multiple VPN routes to be included in the list of available equal-cost paths. You can use the **maximum-paths** command with the **ibgp** or **eibgp** keywords to enable ECMP support for BGP/MPLS VPNs.

The **eibgp** keyword specifies that the E Series router consider *both* external BGP (EBGP) and internal BGP (IBGP) paths when determining the number of equal-cost paths to the same destination that BGP can submit to the IP routing table. The **ibgp** keyword specifies that the E Series router consider multiple internal IBGP paths, but not EBGP paths, when determining the number of equal-cost paths.

You can create an ECMP environment in which multiple IBGP paths are selected as multipaths and used for load balancing. In the example shown in Figure 94 on page 455, the E Series router gives equal consideration to IBGP VPN routes learned from multiple remote PE devices when determining load balancing.

Figure 94: BGP/MPLS VPN IBGP Example



The sample BGP/MPLS network connects PE 1, PE 2, and PE 3, which are configured for VPNv4 unicast IBGP peering. CE 1 and CE 2 are configured for EBGP peering with the PE devices. CE 2 is multihomed, connected to both PE 2 and PE 3.

VRF A has two equal-cost paths through the MPLS network to get to CE 2: the IBGP path to PE 2, and the IBGP path to PE 3.

To support BGP/MPLS ECMP, PE 1 is configured with the **maximum-paths ibgp** command under IPv4 unicast VRF A address family. Doing this allows IBGP paths from both PE 2 and PE 3 to be selected as multipaths for use in load balancing.

Traffic from CE 1 to CE 2 that takes an IBGP route from PE 1 to either PE 2 or PE 3 is forwarded as MPLS-encapsulated packets. PE 2 and PE 3 receive the MPLS-encapsulated traffic from PE 1, remove the MPLS encapsulation, and then forward the traffic as IP packets by means of their EBGP route to CE 2.

Related Documentation

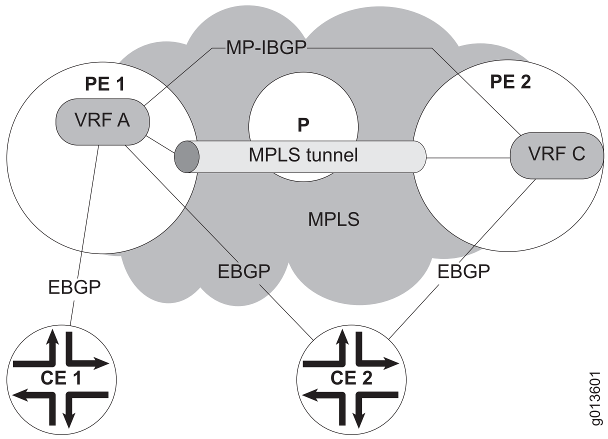
- Equal-Cost Multipath Support Overview on page 392
- *maximum-paths*

Example: Enabling BGP ECMP for BGP/MPLS VPN EBGP

You can create a mixed ECMP environment in which both EBGP and IBGP paths are selected as multipaths and used for load balancing. Doing this enables the E Series router to take into account both EBGP VPN routes learned from a CE router device and IBGP VPN routes learned from a remote PE device when determining load balancing.

In Figure 95 on page 456, a BGP/MPLS network connects PE 1 and PE 2, which are configured for VPNv4 unicast IBGP peering. CE 1 and CE 2 are configured for EBGP peering with the PE devices. CE 2 is multihomed, connected to both PE 1 and PE 2.

Figure 95: BGP/MPLS VPN EIBGP Example



VRF A has two paths to get to CE 2: the IBGP path through the MPLS network, and the EBGP path by means of regular IP.

To support BGP/MPLS ECMP, PE 1 is configured with the **maximum-paths eibgp** command in the IPv4 unicast VRF A address family. Doing this allows both the EBGP paths from CE 2 and the IBGP paths from PE 2 to be selected as multipaths in the VRF A routing information base (RIB) for use in load balancing.

Traffic taking the various routes from CE 1 to CE 2 is treated as follows:

- Traffic from CE 1 to CE 2 that takes the EBGP route from PE 1 is forwarded as IP packets.
- Traffic from CE 1 to CE 2 that takes the IBGP route from PE 1 is forwarded as MPLS-encapsulated packets. PE 2 receives the MPLS-encapsulated traffic from PE 1, removes the encapsulation, and then forwards the traffic as IP packets by means of the EBGP route to CE 2.

Related Documentation

- Equal-Cost Multipath Support Overview on page 392
- *maximum-paths*

VPN Address Exchange Overview

To limit the exchange of routes to those from within the VPN-IPv4 address family, and to set other desired BGP parameters:

1. Specify that the router exchanges addresses within a VPN by choosing the VPN-IPv4 address family.
2. Specify individual neighbors or peer groups to exchange routes with from only within the current (VPN-IPv4) address family.
3. Configure BGP parameters for VPN services.

   See “Configuring BGP Routing” on page 3, for information about configuring BGP sessions. The section “Understanding BGP Command Scope” on page 18 has tables that list BGP commands according to their scope. From Address Family Configuration mode, you can issue the commands in Table 7 on page 19 and Table 9 on page 21.

4. Exit Address Family Configuration mode.

Related Documentation

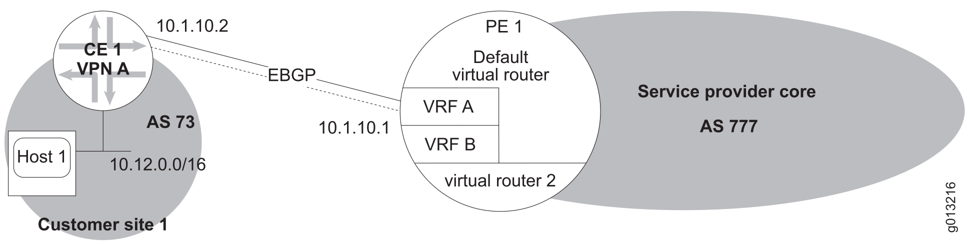
- Understanding MBGP Address Families on page 391
- Understanding VPN-IPv4 Addresses on page 397

Example: Configuring PE-to-CE BGP Sessions

If you have established a BGP session between a PE and a particular CE router, you can configure BGP sessions with all the other customer sites within the VPN so that they can learn the routes to the particular CE router.

Configuring the PE-to-CE external BGP session is a bit different from the usual external BGP session. You must configure the session in the context of the IPV4 unicast address family of the VRF. Consider the topology shown in Figure 96 on page 457.

Figure 96: PE-to-CE Session



You configure the characteristics of VRF A, the global BGP attributes, the address family for the session, and BGP attributes relevant to the VRF or address family.

```
host1(config)#ip vrf vrfa
host1(config-vrf)#rd 777:5
host1(config-vrf)#route-target both 777:5
host1(config-vrf)#exit
host1(config)#interface gigabitEthernet 1/0
host1(config-if)#ip vrf forwarding vrfA
host1(config-if)#ip address 10.1.10.1 255.255.255.0
host1(config-if)#exit
host1(config)#router bgp 777

(Not shown: Configuration of other global BGP attributes)
host1(config-router)#address-family ipv4 unicast vrf vrfA
host1(config-router-af)#neighbor 10.1.10.2 remote-as 73

(Not shown: Configuration of BGP attributes relevant to the VRF or the address family)
```

See "Configuring BGP Routing" on page 3, for more information about configuring BGP.

Related Documentation

- Example: Preventing Routing Loops on page 461
- Monitoring the VRF on page 513

- *address-family*
- *interface gigabitEthernet*
- *ip address*
- *ip vrf*
- *ip vrf forwarding*
- *neighbor remote-as*
- *route-target*
- *router bgp*

Route Advertisements to Customers Overview

If you established static routes on a PE router for each prefix in a particular customer site, you can configure BGP on the PE router to advertise these static routes to customer sites within the VPN with **network** commands.

```
host1:vr1(config-router)#network 10.3.0.0
host1:vr1(config-router)#network 10.12.0.0
```

In this example, both networks end on a classful boundary, eliminating the need to configure a network mask.

Alternatively, you can use the **redistribute** command to advertise the static routes as follows:

```
host1:vr1(config-router)#redistribute static
```

See "Configuring BGP Routing" on page 3, for more information about advertising static routes.

If the PE router learns routes from a CE router by means of an IGP, you can configure BGP to advertise these IGP routes to all customer sites within the VPN with **redistribute** commands. For example, if the PE router learns the routes by means of OSPF, you can issue the following command to inject these routes into BGP for advertisement:

```
host1:vr1(config-router)#redistribute ospf
```

See "Configuring BGP Routing" on page 3, for more information about advertising IGP routes.

Related Documentation

- Example: Adding Static Routes to a VRF on page 450
- *network*
- *redistribute*

Example: Disabling the Default Address Family

PE routers can exchange routes in the IPv4 address family, VPNv4 address family, or both. Issuing the **neighbor remote-as** command automatically activates the IPv4 unicast address family, meaning that the PE router exchanges routes in the IPv4 unicast address family with that peer.

The following commands illustrate how to configure the exchange of routes in both the IPv4 unicast and the VPNv4 unicast address families for a BGP peer:

```
host1:vr1(config)#router bgp 777
host1:vr1(config-router)#neighbor 10.26.5.10 remote-as 100
host1:vr1(config-router)#address-family vpnv4 unicast
host1:vr1(config-router-af)#neighbor 10.26.5.10 activate
host1:vr1(config-router-af)#exit-address-family
```

The **neighbor remote-as** command activated the IPv4 unicast address family for the peer. The **address-family** command entered the context of the VPNv4 unicast family and the **neighbor activate** command activated the address family for the peer.

- Disabling the Exchange of Routes for a Specific Peer on page 459
- Disabling the Exchange of Routes for all Peers on page 459

Disabling the Exchange of Routes for a Specific Peer

The following commands illustrate one way to disable the exchange of routes in the IPv4 unicast address family and enable the exchange of routes in the VPNv4 unicast address family:

```
host1:vr1(config)#router bgp 777
host1:vr1(config-router)#neighbor 10.26.5.10 remote-as 100
host1:vr1(config-router)#address-family ipv4 unicast
host1:vr1(config-router-af)#no neighbor 10.26.5.10 activate
host1:vr1(config-router-af)#exit-address-family
host1:vr1(config-router)#address-family vpnv4 unicast
host1:vr1(config-router-af)#neighbor 10.26.5.10 activate
host1:vr1(config-router-af)#exit-address-family
```

In this case, the **no neighbor activate** command specifically disables the IPv4 unicast address family for that peer alone; no other peers are affected. The VPNv4 unicast address family is activated for the peer as in Example 1.

Disabling the Exchange of Routes for all Peers

The following commands illustrate another way to disable the exchange of routes in the IPv4 unicast address family and enable the exchange of routes in the VPNv4 unicast address family:

```
host1:vr1(config)#router bgp 777
host1:vr1(config-router)#no bgp default ipv4-unicast
host1:vr1(config-router)#neighbor 10.26.5.10 remote-as 100
host1:vr1(config-router)#address-family vpnv4 unicast
host1:vr1(config-router-af)#neighbor 10.26.5.10 activate
host1:vr1(config-router-af)#exit-address-family
```

In this case, the **no bgp default ipv4-unicast** command prevents the automatic enabling of the IPv4 unicast address family for all peers subsequently configured with the **neighbor remote-as** command. Previously configured peers are not affected. The VPNv4 unicast address family is activated for the peer as in Examples 1 and 2.

Related Documentation

- Understanding MBGP Address Families on page 391
- *address-family*
- *bgp default ipv4-unicast*
- *exit-address-family*
- *neighbor activate*
- *neighbor remote-as*
- *router bgp*

Example: Using a Single AS Number for All CE Sites

If you want to use the same AS number for all of your CE sites, you can substitute a PE router's autonomous system number for that of a neighbor by specifying the neighbor's IP address in the **neighbor as-override** command. If you fail to do this, the CE router recognizes its AS in the AS path of received routes and determines it has discovered a routing loop; the routes are rejected.

In the following example, the router's AS number of 777 overrides the neighboring router's AS number of 100.

```
host1:vr1(config)#router bgp 777
host1:vr1(config-router)#neighbor 172.16.20.10 remote-as 100
host1:vr1(config-router)#neighbor 172.16.20.10 update-source loopback0
host1:vr1(config-router)#address-family ipv4 vrf vpn1
host1:vr1(config-router-af)#neighbor 172.25.14.12 remote-as 100
host1:vr1(config-router-af)#neighbor 172.25.14.12 as-override
```

NOTE: To apply the new policy to routes that are already present in the BGP routing table, you must use the **clear ip bgp** command to perform a soft clear or hard clear of the current BGP session.

Behavior is different for outbound policies configured for peer groups for which you have enabled Adj-RIBs-Out. If you change the outbound policy for such a peer group and want to fill the Adj-RIBs-Out table for that peer group with the results of the new policy, you must use the **clear ip bgp peer-group** command to perform a hard clear or outbound soft clear of the peer group. You cannot merely perform a hard clear or outbound soft clear for individual peer group members because that causes BGP to resend only the contents of the Adj-RIBs-Out table.

Related Documentation

- Example: Preventing Routing Loops on page 461

- *address-family*
- *neighbor as-override*
- *neighbor remote-as*
- *neighbor update-source*
- *router bgp*

Example: Preventing Routing Loops

Routing loops can occur when routes learned from a peer are later advertised back to that peer. Normally such routing loops are prevented by the AS-path attribute. However, the AS path cannot prevent routing loops in a network configuration with the following characteristics:

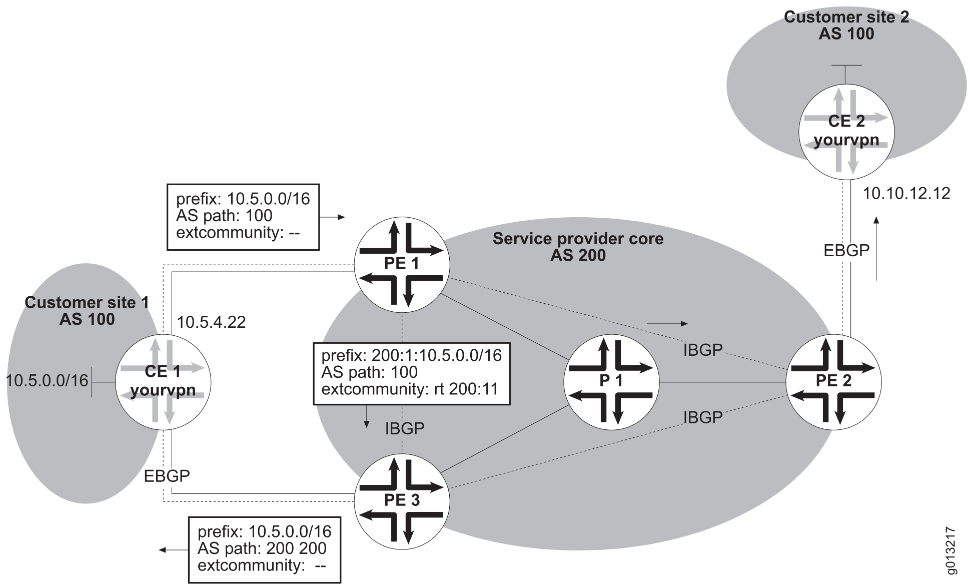
- BGP is running between CE and PE routers.
- You use a single AS number for all customer sites, and have issued the **neighbor as-override** command for the PE routers.
- A CE router is dual-homed to two or more PE routers.

The site-of-origin extended community attribute enables BGP to filter out such routes to prevent routing loops in this network. You can use the **set extcommunity** command to specify a site of origin and then use the **match extcommunity** command and an outbound route map to filter routes; for more information, see *Extended Community Lists* in the cit-junose-ip-services;.

Alternatively, you can use the **neighbor site-of-origin** command alone to achieve the same effect in such a network configuration. If you use this command to configure a site of origin for routes from a peer, then routes advertised to that peer that contain this site of origin are filtered out and not advertised. This behavior is followed regardless of whether the **neighbor send-community extended** command has been issued for the peer.

Consider the network shown in Figure 97 on page 462, which enables PE 3 to advertise back to CE 1 routes that it learned from PE 1 that originated with CE 1. In a typical network configuration, CE 1 rejects these routes because it determines from the AS path that a routing loop exists. In this particular network, the **neighbor as-override** command prevents this method of detection.

Figure 97: Network with Potential Routing Loops



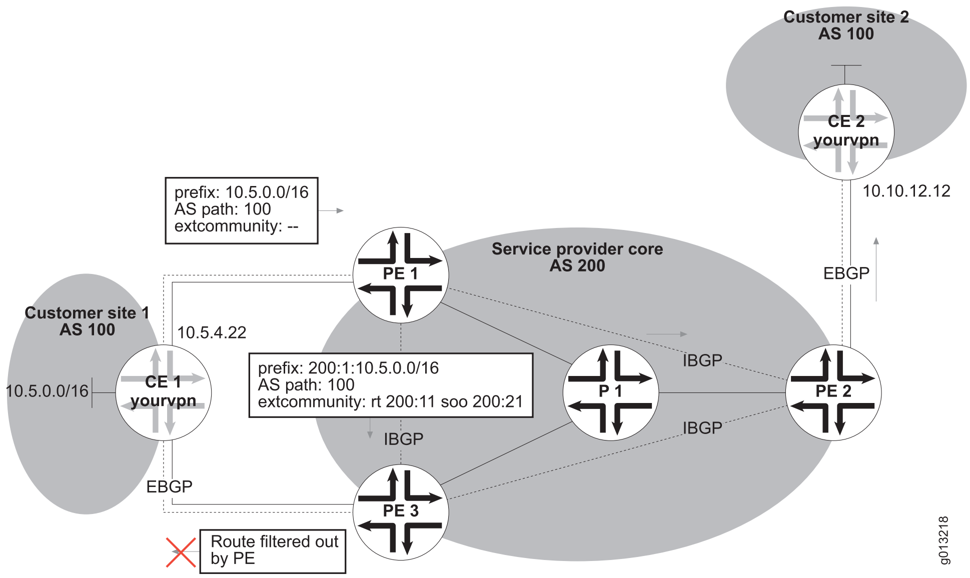
The following commands are relevant to the illustrated network:

```
host1:pe1(config)#ip vrf yourvpn
host1:pe1(config-vrf)#rd 200:1
host1:pe1(config-vrf)#route-target both 200:11
...
host1:pe1(config)#router bgp 200
host1:pe1(config-router)#address-family ipv4 unicast vrf yourvpn
host1:pe1(config-router)#neighbor 10.5.4.22 remote-as 100
host1:pe1(config-router)#neighbor 10.5.4.22 as-override
...
```

Now, suppose instead you assign a unique site of origin to each CE router in the network and configure the BGP session on each PE router with the site of origin. The result of the following (partial) configuration is shown in Figure 98 on page 463.

```
host1:pe1(config)#ip vrf yourvpn
host1:pe1(config-vrf)#rd 200:1
host1:pe1(config-vrf)#route-target both 200:11
...
host1:pe1(config)#router bgp 200
host1:pe1(config-router)#address-family ipv4 unicast vrf yourvpn
host1:pe1(config-router)#neighbor 10.5.4.22 remote-as 100
host1:pe1(config-router)#neighbor 10.5.4.22 as-override
host1:pe1(config-router)#neighbor 10.5.4.22 site-of-origin 200:21
...
```


Figure 98: Preventing Potential Routing Loops in the Network



Related Documentation

- Example: Configuring PE-to-CE BGP Sessions on page 457
- Example: Using a Single AS Number for All CE Sites on page 460
- *match extcommunity*
- *neighbor site-of-origin*
- *set extcommunity*

Prefix Advertisement with Duplicate AS Numbers Overview

When a BGP speaker receives a route that has the speaker's AS number in its AS path, the speaker declares that route to be a loop and discards it. However, in some circumstances, as in the implementation of a hub-and-spoke VPN topology, this is not the desired behavior. You want the BGP speaker (hub) to accept such routes. You can use the **neighbor allowas-in** command to specify the number of times that a route's AS path can contain the BGP speaker's AS number.

New policy values are applied to all routes that are sent (outbound policy) or received (inbound policy) after you issue the command.

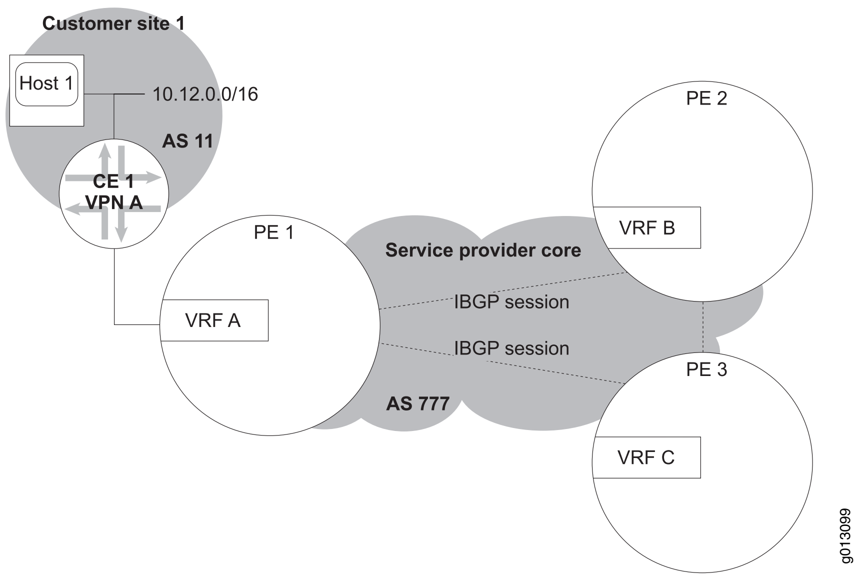
To apply the new policy to routes that are already present in the BGP routing table, you must use the **clear ip bgp** command to perform a soft clear or hard clear of the current BGP session.

Behavior is different for outbound policies configured for peer groups for which you have enabled Adj-RIBs-Out. If you change the outbound policy for such a peer group and want to fill the Adj-RIBs-Out table for that peer group with the results of the new policy, you must use the **clear ip bgp peer-group** command to perform a hard clear or outbound soft clear of the peer group. You cannot merely perform a hard clear or outbound soft clear for individual peer group members because that causes BGP to resend only the contents of the Adj-RIBs-Out table.

The behavior is different within the VPNv4 address family than it is in other address families. For other address families, you must configure the feature on all the peers. In contrast, IBGP peers within the VPNv4 address family always accept routes containing their own AS number by default. Issuing this command in the VRF for such a peer has no effect on the behavior of IBGP peers in this address family. This behavior reduces the provisioning overhead for VPNv4 IBGP peers.

However, you must configure the feature on the peer router at the hub. Consider the hub-and-spoke topology shown in Figure 99 on page 464. PE 1, PE 2, and PE 3 are peers in the VPNv4 address family. Routes received from CE 1 may contain the AS number (777) local to the PE routers. You must issue the **neighbor allowas-in** command for VRF A on PE 1.

Figure 99: Allowing Local AS in VPNv4 Address Family



Related Documentation

- Hub-and-Spoke VPNs on page 423
- *clear ip bgp*
- *neighbor allowas-in*

Route Importation Control Overview

You can control how many routes a PE router can add to a particular VRF's forwarding table by specifying a maximum limit and a warning threshold. When the router attempts to add a route, it compares the limit you configure against a route count it maintains for routes already in the VRF's forwarding table.

With a warning threshold configured, the following behavior takes place when the PE router attempts to add a route:

- When adding the route causes the route count to exceed the warning threshold for the first time, the router adds the route and generates a warning-threshold-exceeded log entry.
- As long as the route count stays above the warning threshold, adding more routes does not generate more warning-threshold-exceeded log entries.
- If the route count fluctuates below and above the warning threshold due to route deletions and additions, an interval of 5 minutes since the last warning-threshold-exceeded log entry must pass before another warning-threshold-exceeded log entry can be generated. This behavior prevents the system log from being flooded with log entries.

With a limit configured, the following behavior takes place when the PE router attempts to add a route:

- When adding the route causes the route count to exceed the limit for the first time, the router rejects the route and generates a limit-exceeded log entry.
- As long as the route count stays at the limit, further attempts to add routes fail, but do not generate any more limit-exceeded log entries.
- If the route count fluctuates below and up to the limit due to route deletions and additions, no further limit-exceeded log entries are generated until a 5-minute interval has passed since the last limit-exceeded log entry. This behavior prevents the system log from being flooded with log entries.

When you issue the command, the router immediately reevaluates the current number of routes against the new limit. If the current number of routes is greater than the maximum configured limit, the router might remove dynamically learned routes in order to enforce the new limit.

Related Documentation

- BGP/MPLS VPN Components Overview on page 394
- Definition of Route Targets for VRFs Overview on page 438

VRF–to–VR Peering Overview

In some circumstances you might want a CE router, which connects to the PE router by means of a VRF, to be able to establish an EBGP peering session directly with the parent VR in which the VRF has been configured. The global instance of BGP for the PE router runs in the parent VR to exchange VPN routes with its peers by means of internal or external MP-BGP. BGP could also be learning IPv4 unicast Internet routes from one or more of its core-facing, internal or external BGP peers.

In the context of the VRF, you can use the **ip route parent-router** command to add a static host route to a stable interface (typically a loopback interface) in the parent VR by way of a hidden VRF-internal interface.

```
host1(config)#virtual-router PE1
host1:PE1(config)#interface loopback 1
host1:PE1(config-if)#ip address 10.20.20.2 255.255.255.255
host1:PE1(config-if)#exit
```



```
host1:PE1(config)#virtual-router :PE11
host1:PE1:PE11(config)#ip route parent-router loopback 1
```

In this example, assume that the global instance of BGP for the PE router runs in the parent VR, PE 1, to exchange VPN routes with its peers by means of internal or external MP-BGP. BGP can also be learning IPv4 unicast Internet routes from one or more of its core-facing, internal or external BGP peers.

By virtue of the static route configured in VRF PE 11, a CE router that connects to that VRF can establish an EBGP session directly to loopback 1 (10.20.20.2) in the parent VR, PE 1. The same PE router can therefore provide both VPN and Internet access to any attached CE routers.

The specified interface must be preexisting and have an alias assigned with the **description** command.

The route points to a next-hop interface that is internal to the VRF and created automatically when the VR comes up. This interface is hidden and cannot be displayed with the **show ip interface** command. You must use the **show ip route** or **show ip static** commands to display the interface.

If the interface in the parent VR goes down or is deleted, the static route added in the VRF will continue to exist.

Related Documentation

- Enabling VRF-to-VR Peering on page 466
- Traffic Flow from the Internet to the VPN Overview on page 483

- *description*
- *interface loopback*
- *ip address*
- *ip route*
- *show ip route*
- *show ip static*
- *virtual-router*

Enabling VRF-to-VR Peering

In this configuration, assume that the global instance of BGP for the PE router runs in the parent VR, PE 1, to exchange VPN routes with its peers by means of internal or external MP-BGP. BGP can also be learning IPv4 unicast Internet routes from one or more of its core-facing, internal or external BGP peers.

By virtue of the static route configured in VRF PE 11, a CE router that connects to that VRF can establish an EBGP session directly to loopback 1 (10.20.20.2) in the parent VR, PE 1. The same PE router can therefore provide both VPN and Internet access to any attached CE routers.

To enable VRF-to-VR peering:

1. Configure the virtual router PE1.

   host1(config)#**virtual-router PE1**

2. Define the loopback interface for the virtual router.

   host1:PE1(config)#**interface loopback 1**

3. Configure the IP address.

   host1:PE1(config-if)#**ip address 10.20.20.2 255.255.255.255**
   host1:PE1(config-if)#**exit**

4. Configure virtual router PE11.

   host1:PE1(config)#**virtual-router :PE11**

5. Configure static routes on PE11.

   host1:PE1:PE11(config)#**ip route parent-router loopback 1**

Related Documentation

- VRF–to–VR Peering Overview on page 465
- *ip route*
- *show ip route*
- *show ip static*

Fast Reconvergence in VPN Networks

The following sections describe fast reconvergence in VPN networks:

Fast Reconvergence in VPN Networks Overview

By default, BGP does not confirm the reachability of the BGP indirect next hop of VPNv4 routes received over an MP-IBGP session until those routes have been imported into a VRF.

To BGP, the next hops of VPNv4 routes that are still in the global VPNv4 table (viewable with the **show ip bgp vpnv4 all** command) are always reachable. As a result, VPNv4 route reflectors that have multiple paths to the same prefix select the best route to reflect without taking into account the reachability of the BGP indirect next hop. Instead, best-path selection is based on weight, local preference, AS-path length, and other attributes.

After the route has been imported into a VRF, the reachability of the BGP indirect next hop is based on the presence of an MPLS tunnel (LDP or RSVP-TE) to the next-hop address and not on the presence of an IP route to the next-hop address.

Disregarding the reachability of the BGP indirect next hop when the router selects the best route to reflect can cause very slow reconvergence (up to 90 seconds) after a topology change in BGP/MPLS VPN networks that match all of the following conditions:

- Have a full mesh of LDP MPLS tunnels
- Have multihomed CE routers
- Use the same RD for multiple VRFs
- Rely on VPNv4 route reflectors as arbiters for selecting the best VPNv4 route from a set of clients

If a PE router fails in such a network, the route reflector must quickly reflect the VPNv4 route from the next-best PE router without having to wait for the BGP session to the failed PE router to time out. Depending on the network topology, you can achieve fast reconvergence by assigning unique RDs to each VRF or by enabling next-hop reachability checking.

Related Documentation

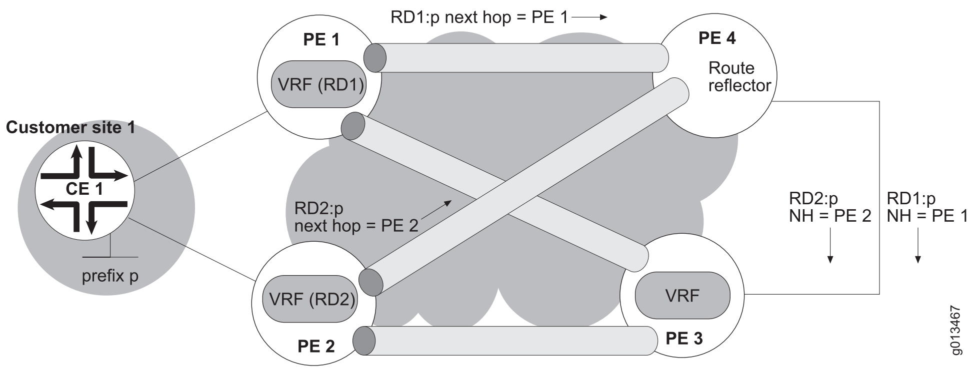
- Understanding Route-Target Filtering for MBGP VPNs Overview on page 427
- Fast Reconvergence with Unique RDs on page 468
- Fast Reconvergence by Means of Reachability Checking on page 469
- *show ip bgp*

Fast Reconvergence with Unique RDs

You can assign a unique RD for the VRFs in each PE router to avoid the slow reconvergence issue. The route reflectors in the network consider advertised routes with different RDs to be different prefixes and therefore reflect both routes.

In Figure 100 on page 468, route reflector PE 4 reflects to PE 3 routes to the CE router through both PE 1 and PE 2. Suppose that the route through PE 1 is better than the route through PE 2. If you have assigned different RDs to the VRFs, then PE 4 reflects both routes to its client, PE 3.

Figure 100: Topology for Fast Reconvergence by Means of Unique VRF RDs, Before Tunnels Go Down



If PE 1 goes down, the MPLS tunnels to it (PE 4–PE 1 and PE 3–PE 1) are dropped immediately. However, because the route reflector does not take into account the reachability of the next hop, it still reflects both the PE 1 route and the PE 2 route.

When PE 3 imports these routes into its VRF, it resolves the routes and discovers that the tunnel to PE 1 is down. PE 3 declares the next hop for the route through PE 1 to be unreachable. It then selects the PE 2 route as the best route and installs it in the VRF's IP routing table.

On the other hand, if the VRFs in PE 1 and PE 2 share the same RD, the route reflector reflects only the best route, in this example the route through PE 1. If PE 1 goes down in this situation, PE 4 still reflects the route through PE 1. When PE 3 resolves the route, it finds that the tunnel is down and declares the next hop to be unreachable. Traffic then suffers a delay due to slow reconvergence.

Assigning a unique RD for each VRF can be useful for other reasons as well:

- PE-to-PE forwarding requires an MPLS tunnel from the ingress PE router to the egress PE router. In some topologies, such as networks with a sparse RSVP-TE mesh where the route reflector is not in the forwarding path, little correlation exists between the presence of an MPLS tunnel or IP connectivity from the route reflector to the egress PE router and the presence of the MPLS tunnel from the ingress PE router to the egress PE router.

  For these networks, relying on the ingress PE router is better than relying on the route reflector to decide which route is best. For this to work properly, the ingress PE router must be able to choose from all available paths, which in turn requires that each VRF have a unique RD.

- If each VRF has a unique RD and the ingress PE router has all feasible paths to choose from, you can configure IBGP multipath and ECMP traffic over multiple PE-to-PE MPLS tunnels. This configuration is not possible if you use the same RD on multiple VRFs, because the ingress PE router in that case picks a single route that resolves to a single MPLS tunnel that is used end-to-end.

Related Documentation

- Understanding VPN-IPv4 Addresses on page 397
- Fast Reconvergence in VPN Networks Overview on page 467
- Fast Reconvergence by Means of Reachability Checking on page 469

Fast Reconvergence by Means of Reachability Checking

You might not want to assign different RDs for each VRF in some circumstances, such as the following:

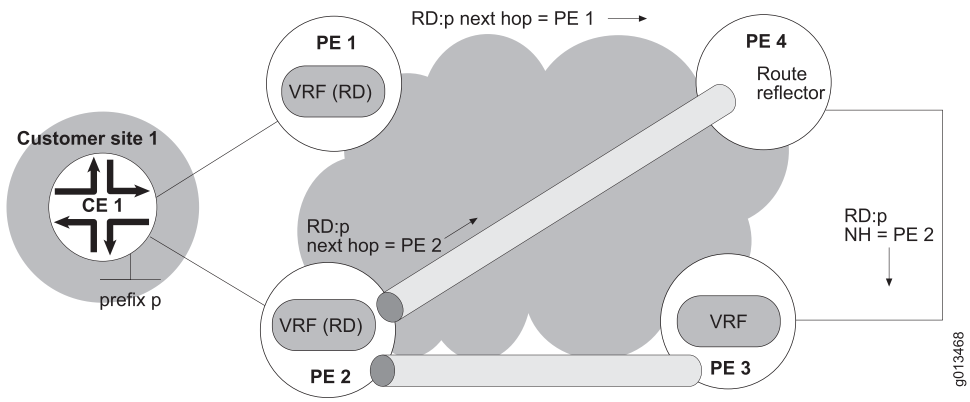
- Allocating a unique RD for each VRF might be an administrative burden.
- If the network is already in operation and configured with the same RD for all VRFs in a given VPN, then changing the RDs might affect service.
- Each route reflector might act as an arbiter for a geographic area and be responsible for maintaining a list of all feasible paths to egress PE routers that can be used to reach

a given prefix. Because the route reflector selects only one best path and reflects that single best path toward its clients and nonclients, the amount of state in the network is reduced. The core of the network and other geographic areas need only the one best route to each prefix in a given remote geographical area.

You can use the **check-vpn-next-hops** command to avoid the slow reconvergence problem without having to configure a unique RD for each VRF. When you issue this command, BGP verifies the reachability of the next hop on VPNv4 routes received from MP-IBGP peers before it imports those routes into a VRF. This behavior enables the VPNv4 route reflectors to take into account the reachability of the next hop when they select the best route to reflect.

Consider a topology similar to that discussed in the previous section. As before, the route through PE 1 is considered to be the best. VRFs share the same RD, but reachability checking has been enabled. In Figure 101 on page 470, PE 1 has already failed, and tunnels PE 3–PE 1 and PE 4–PE 1 have gone down.

Figure 101: Topology for Fast Reconvergence by Means of Reachability Checking, After Tunnels Go Down



When the MPLS tunnel (RSVP-TE or LDP) to the next hop of the best route goes down, the VPNv4 route reflector immediately advertises the next-best route (if any) without waiting for the MP-IBGP session to go down. In this example, that route is through PE 2.

This command is available only in the context of the VPNv4 unicast and VPNv6 unicast address families. The behavior is the same for both address families.

Use the **show ip bgp vpnv4 all summary** or **show bgp ipv6 vpnv6 all summary** command to view the status of next hop reachability checking.

Related Documentation

- Fast Reconvergence in VPN Networks Overview on page 467
- Fast Reconvergence with Unique RDs on page 468
- *check-vpn-next-hops*
- *show bgp ipv6*
- *show ip bgp*

Understanding BGP Routing Rules

Understanding BGP Sending of Labeled and Unlabeled Unicast Routes

You can issue the **neighbor send-label** command to enable BGP to exchange both labeled and unlabeled unicast routes in the same address family (same AFI) over the same BGP peering session. The routes can be IPv4 or IPv6 routes. When you issue the **neighbor send-label** command, JunosE always proposes SAFI 4 and SAFI 1. If this command has not been configured, then JunosE proposes only SAFI 1.

A route is advertised as a labeled route within a given BGP peering session in either of the following cases:

- You issue the **neighbor send-label** command, but no outbound route map has been configured.
- You issue the **neighbor send-label** command and an outbound route map has been configured, and the route map executes a **set mpls-label** clause for the advertised route.

In all other cases, the route is advertised as an unlabeled route.

Related Documentation

- BGP Next-Hop-Self Overview on page 471
- Understanding BGP Processing of Received Routes on page 472
- Understanding BGP Advertising Rules for Labeled and Unlabeled Routes with the Same AFI on page 473

- *neighbor send-label*
- *set mpls-label*

BGP Next-Hop-Self Overview

When a BGP router reports itself as the next hop, whether because of an explicit **neighbor next-hop-self** configuration or implicitly as a result of participating in an EBGP session, BGP allocates a new in label and adds an entry to the MPLS forwarding table, creating a label-to-next-hop mapping.

When a BGP router does not report itself as the next hop, whether because of an explicit **neighbor next-hop-unchanged** configuration or implicitly as a result of a participating in an IBGP session, BGP does not allocate a new in label. Instead, if the route is advertised as a labeled route, BGP uses the existing out label. This feature is used mainly on route reflectors.

The determination to allocate an in label is made only after the outbound route map has been processed. Therefore, the in label allocation and the creation of the label-to-next-hop mapping are performed after the need is apparent, conserving the number of in labels allocated.

Related Documentation

- Understanding BGP Sending of Labeled and Unlabeled Unicast Routes on page 471
- Understanding BGP Processing of Received Routes on page 472
- Understanding BGP Advertising Rules for Labeled and Unlabeled Routes with the Same AFI on page 473

- *neighbor next-hop-self*
- *neighbor next-hop-unchanged*

Understanding BGP Processing of Received Routes

BGP processes received routes differently depending on whether the route is labeled or unlabeled, unicast or VPN.

Labeled Unicast Routes

When BGP receives a labeled route from a directly connected peer, BGP uses the MPLS major interface that is next to the peer IP interface to resolve the route's BGP next hop. If the MPLS major interface exists and is up, then the next hop is reachable.

When the received labeled route is not from a directly connected peer, BGP attempts to resolve the BGP indirect next hop of the route in the IP tunnel routing table. When the BGP indirect next hop is reachable, BGP adds the route to both the IP routing table and to the IP tunnel routing table. The route is added as a U-T (unicast-tunnel-usable) route.

Unlabeled Unicast Routes

When BGP receives an unlabeled route from a directly connected peer, the route's next hop is resolved to the directly connected interface.

When the received unlabeled route is not from a directly connected peer, BGP resolves the BGP indirect next hop of the route in the IP routing table. If the BGP indirect next hop is reachable, BGP adds the route to the IP routing table as a U (unicast) route.

Resolving IPv6 Indirect Next Hops

When the address of the indirect next hop is an IPv4-mapped IPv6 address, BGP resolves the indirect next hop in the IPv4 routing table and IPv4 tunnel routing table. When the indirect next hop is a native IPv6 address, the indirect next hop is resolved in the IPv6 routing table and IPv6 tunnel routing table.

Labeled VPN Routes

In the core VRF, when BGP receives a BGP-labeled VPN route from a multihop VPN peer, it attempts to resolve the BGP indirect next hop in the IP tunnel routing table. If the labeled VPN route is received from a nonmultihop peer, then the BGP indirect next hop is always resolved, because a connected route to that peer exists in the IP tunnel routing table.

Table 92 on page 473 summarizes indirect next hop resolution.

Table 92: Resolution of Indirect Next Hops

Route Type	Table in Which BGP Indirect Next Hop Resolves
Unlabeled unicast	IP routing table
Labeled unicast	IP tunnel routing table, IP routing table
Labeled VPN	IP tunnel routing table

Related Documentation

- Understanding BGP Sending of Labeled and Unlabeled Unicast Routes on page 471
- BGP Next-Hop-Self Overview on page 471
- Understanding BGP Advertising Rules for Labeled and Unlabeled Routes with the Same AFI on page 473

Understanding BGP Advertising Rules for Labeled and Unlabeled Routes with the Same AFI

When BGP receives a route to a prefix with the same AFI in both labeled and unlabeled forms, only one of these routes can be selected as the best route. The action taken after the best route is selected depends on whether the best route is labeled or unlabeled, and on what SAFI was previously negotiated with peers other than the one from which it received the best route. Table 93 on page 473 lists the advertising action taken for the best route, whether labeled or unlabeled.

Table 93: Advertising Action Taken Following Best Route Selection

Best Route	SAFI Negotiated with Peer	Action Taken
Unlabeled	SAFI 1 and SAFI 4 (unlabeled and labeled)	Advertises unlabeled route.
Unlabeled	SAFI 1 (unlabeled)	Advertises unlabeled route.
Unlabeled	SAFI 4 (labeled)	Withdraws labeled route.
Labeled	SAFI 1 and SAFI 4 (unlabeled and labeled)	Advertises labeled route.
Labeled	SAFI 1 (unlabeled)	Withdraws unlabeled route.
Labeled	SAFI 4 (labeled)	Advertises labeled route.

BGP sends a route-refresh message for each SAFI that it has negotiated with a peer. For example, if a speaker has negotiated both SAFI 1 and SAFI 4 with a particular peer, then when you issue the **clear ip bgp neighbor soft-in** command, BGP sends two route-refresh messages to this neighbor, one for each SAFI.

Related Documentation

- Understanding BGP Sending of Labeled and Unlabeled Unicast Routes on page 471
- BGP Next-Hop-Self Overview on page 471
- Understanding BGP Processing of Received Routes on page 472
- *clear ip bgp*

Understanding VPN Communication

Understanding Internet Access and VPNs

Normally, hosts in a VPN cannot communicate with hosts in the Internet because the routing table in a VRF contains only routes to sites in the VPN and not routes to sites in the Internet. The exchange of traffic between a VPN and the Internet requires both of the following:

- Traffic flow from the VPN to the Internet
- Traffic flow from the Internet to the VPN

The most common, and simplest, method for providing Internet access is to configure two separate logical circuits. One logical circuit runs between the CE router and the VRF and is used for VPN traffic. The other logical circuit runs between the CE router and the parent VR of the VRF and is used for Internet traffic. These logical circuits are typically FR circuits, ATM circuits, or VLANs.

The following sections describe alternative methods of providing Internet access for situations in which having two separate logical circuits is not acceptable or desirable.

Related Documentation

- BGP/MPLS VPN Components Overview on page 394
- Traffic Flow from the VPN to the Internet Overview on page 475
- Traffic Flow from the Internet to the VPN Overview on page 483
- Creation of a BGP Session Between the CE Router and the Parent VR Overview on page 479

Traffic Flow from the VPN to the Internet Overview

Traffic from a CE router arrives on a PE interface that exists in the context of a VRF. The PE router then looks up the destination address of the IP packet in the context of the VRF routing table rather than the VR routing table.

Problems

The VRF routing table lookup introduces the following complication.

- The size of the Internet routing table. Placing a full default-free Internet routing table in the VRF routing table is not feasible because it does not scale. The PE router would have to support more than 100,000,000 routes, because the full default-free Internet routing table is currently about 120,000 routes and the router must support up to 1,000 VRFs.

Solutions

The following methods enable advertising of Internet routes to VPN sites and thus enable traffic flow from the VPNs to the Internet:

- Configure default routes instead of a full default-free Internet routing table in the VRF. The default routes must point to a shared IP interface that you create on top of the layer 2 interface that points to the Internet gateway.
- Configure a single full default-free Internet routing table in the context of the parent VR and share this one table among all VRFs with the fallback global feature. Fallback global enables an additional lookup in the IP routing table of the parent VR in the event that the IP route lookup in the child VRF fails.
- When reachability to a small number of networks in the Internet is required, then configure a global import map to import only the specific route to these networks into the VRF.

You can create multiple IP interfaces on top of a single layer 2 interface. One of those interfaces is the primary IP interface for receiving and sending IP packets. The other interfaces are shared IP interfaces that are used only to send traffic.

Related Documentation

- Understanding Internet Access and VPNs on page 474
- Example: Configuring a Default Route to a Shared Interface on page 476
- Example: Configuring a Fallback Global Option on page 477
- Example: Configuring a Global Import Map for Specific Routes on page 478
- Creation of a BGP Session Between the CE Router and the Parent VR Overview on page 479

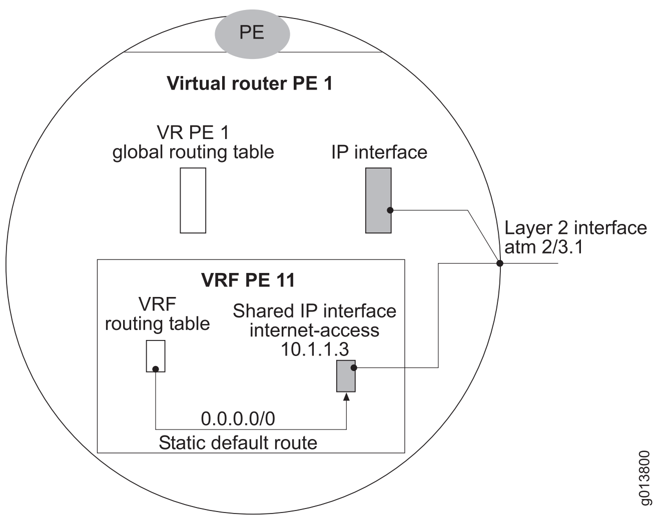
Example: Configuring a Default Route to a Shared Interface

For the first solution you create a default route in the VRF that points to a shared IP interface. You must manually create the shared IP interface on top of the layer 2 interface that points to the Internet gateway. See Figure 102 on page 476.

The main disadvantage of this approach is that if multiple Internet gateways are available, BGP cannot select the egress gateway that is optimal for each destination prefix. Because BGP has only a default route in the VRF, it has to point that single default route to a single uplink interface. All the Internet-bound traffic must flow out of that interface.

You cannot configure traffic for one prefix to flow out of one uplink interface and traffic to another prefix to flow out of another uplink interface. That behavior requires a full default-free Internet routing table in the VRF, which is a complication that you want to avoid.

Figure 102: Static Default Route for Internet Access



The following commands illustrate how to create a shared IP interface in the VRF and point a default route to it:

```
host1(config)#virtual-router pe1:pe11
host1:pe1:pe11(config)#interface ip internet-access
host1:pe1:pe11(config-if)#ip share-interface atm2/1.3
host1:pe1:pe11(config-if)#ip address 10.1.1.3 255.255.255.255
host1:pe1:pe11(config-if)#exit
host1:pe1:pe11(config)#ip route 0.0.0.0 0.0.0.0 ip internet-access
```

See *Shared IP Interfaces* in the *JunosE IP, IPv6, and IGP Configuration Guide*, for information about shared IP interfaces and default routes.

Related Documentation

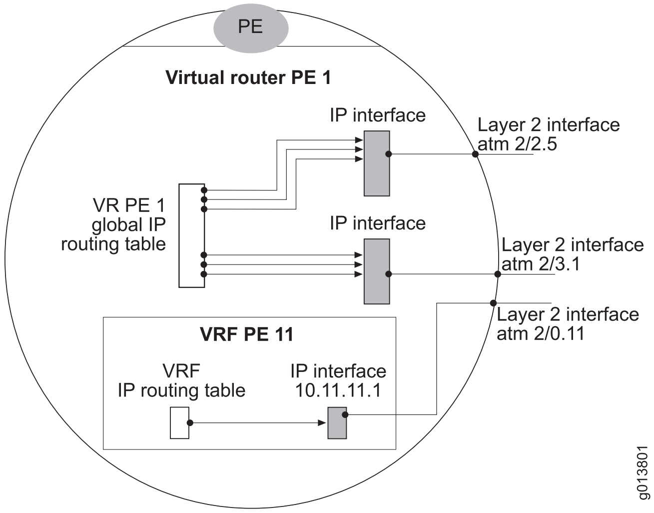
- Traffic Flow from the VPN to the Internet Overview on page 475
- Creation of a BGP Session Between the CE Router and the Parent VR Overview on page 479
- *interface ip*

- *ip address*
- *ip route*
- *ip share-interface*
- *virtual-router*

Example: Configuring a Fallback Global Option

For the second solution you use the fallback global option on the PE–CE IP interface (Figure 103 on page 477). If you have configured this option, the PE router simultaneous performs two different lookups when a packet arrives from the CE router. One lookup is in the IP routing table of the VRF; the other lookup is in the IP routing table of the parent VR.

Figure 103: Fallback Global Option



If BGP finds a route in the VRF context, it uses that route. If BGP does not find a route in the VRF context but does find a route in the VR context, it falls back on the global route in the parent VR. BGP drops the packet if it does not find a route in either context.

To enable fallback global on a PE-CE IP interface:

```
host1:pe1(config)#interface atm2/0.11
host1:pe1(config-if)#ip vrf forwarding pe11 fallback global
host1:pe1:pe11(config-if)#atm pvc 11 0 11 aal5snap
host1:pe1:pe11(config-if)#ip address 10.11.11.1 255.255.255.0
host1:pe1:pe11(config-if)#exit
```

See “Configuring Secondary Routing Table Lookup” on page 449 for more information.

Related Documentation

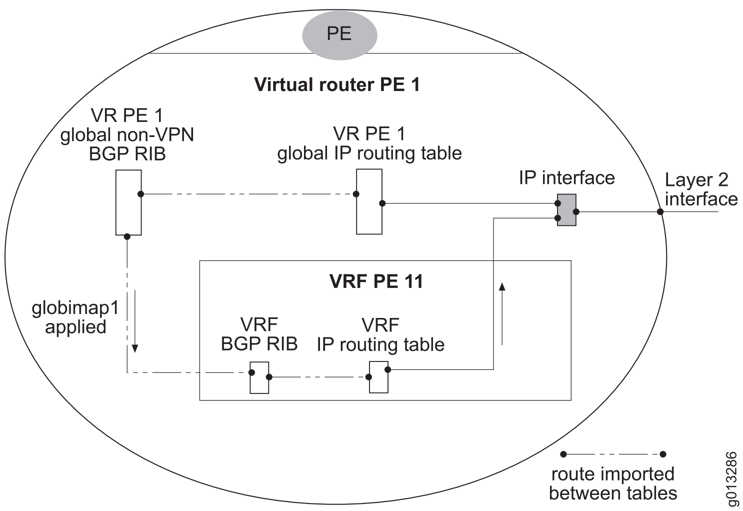
- Traffic Flow from the VPN to the Internet Overview on page 475
- Creation of a BGP Session Between the CE Router and the Parent VR Overview on page 479
- *atm pvc*

- *interface atm*
- *ip address*
- *ip vrf forwarding*

Example: Configuring a Global Import Map for Specific Routes

For the third solution you create a global import map to import only the specific routes needed to reach the desired small number of networks in the Internet. See Figure 104 on page 478.

Figure 104: Global Import Map Applied to Routes Imported from VRF BGP RIB



The global import map enables global BGP routes to be automatically imported into the BGP RIB table in a VRF. The route map determines which routes are imported and which are not. When they are installed in the VRF routing table, the imported routes point to IP interfaces in the parent virtual router.

To configure a route map and global import map for importing specific routes.

```
host1(config)#virtual-router pe1
host1:pe1(config)#prefix-list internet-host permit 10.5.5.5/32
host1:pe1(config)#route-map globimap1
host1:pe1(config-route-map)#match ip address prefix-list internethost
host1:pe1(config-route-map)#exit
host1:pe1(config)#ip vrf pe11
host1:pe1(config-vrf)#rd 100:1
host1:pe1(config-vrf)#route-target both 100:1
host1:pe1(config-vrf)#global import map globimap1
```

Related Documentation

- Traffic Flow from the VPN to the Internet Overview on page 475
- Understanding Route Distribution for a VRF using Maps on page 443
- Monitoring the VRF on page 513

- *global import map*
- *ip vrf*
- *match ip address*
- *rd*
- *route-map*
- *route-target*
- *virtual-router*

Creation of a BGP Session Between the CE Router and the Parent VR Overview

The fallback global option enables traffic that arrives at a VRF from the CE router to be sent out on the uplink determined to be optimal by using the full Internet routing table present in the parent VR.

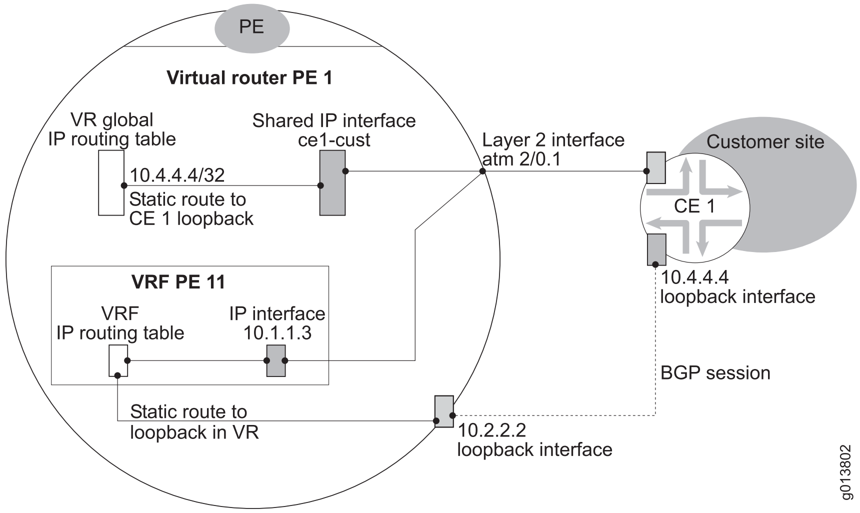
If a CE router is multihomed to multiple PE routers, it must receive a full Internet routing table from each of the PE routers so that the CE router can determine which of the PE routers is optimal for a given Internet prefix.

You can easily create a BGP session from the VRF to the CE router to advertise routes in the VRF to the CE router. However, doing this is insufficient because the VRF does not contain the full Internet routing table, which is present only in the parent VR.

This situation requires a BGP session from the parent VR to the CE router (Figure 105 on page 480). This BGP session in turn requires a route in the VRF to the loopback interface in the parent VR that is used for BGP peering with the CE router. To achieve this configuration, you must do both of the following:

1. In the parent VR, create a shared IP interface for the PE-CE interface and point a static route to the loopback of the CE router to the shared interface.
2. Use a global import map in the VRF to import into the VRF the route to the loopback interface in the parent VR.

Figure 105: BGP Session Between CE Router and Parent VR



You must also configure either fallback global or a default route to a manually created shared interface in the VRF. See "Example: Configuring a Fallback Global Option" on page 477 or "Example: Configuring a Default Route to a Shared Interface" on page 476 for details.

You can use the BGP session between the CE router and the parent VR to enable the CE router to advertise prefixes within the VPN site that can be reachable from the global Internet. An alternative configuration is to use a global export map as described in "Understanding Route Distribution for a VRF using Maps" on page 443.

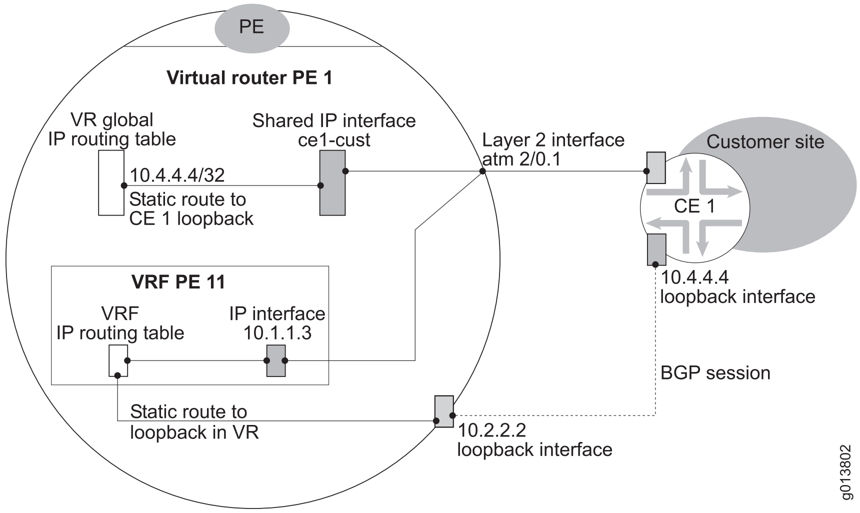
Related Documentation

- Traffic Flow from the VPN to the Internet Overview on page 475
- Example: Creating a BGP Session Between the CE Router and the Parent VR on page 480

Example: Creating a BGP Session Between the CE Router and the Parent VR

The following example creates a BGP session from the VRF to the CE router to advertise routes in the VRF to the CE router.

Figure 106: BGP Session Between CE Router and Parent VR



The following commands configure a shared IP interface in the parent VR and point a static route for the loopback in the CE router to it:

```
host1(config)#virtual-router pe1
host1:pe1(config)#interface ip ce1-cust
host1:pe1(config-if)#ip share-interface atm2/0.1
host1:pe1(config-if)#ip address 10.1.1.3 255.255.255.255
host1:pe1(config-if)#exit
host1:pe1(config)#ip route 10.4.4.4 255.255.255.255 ip ce1-cust
```

The following commands make the loopback in the parent VR reachable from the VRF by means of a global import map:

```
host1(config)#virtual-router pe1
host1:pe1(config)#prefix-list VRloop permit 10.2.2.2/32
host1:pe1(config)#route-map globimaploop
host1:pe1(config-route-map)#match ip address prefix-list VRloop
host1:pe1(config-route-map)#exit
host1:pe1(config)#ip vrf pe11
host1:pe1(config-vrf)#rd 100:1
host1:pe1(config-vrf)#route-target both 100:1
host1:pe1(config-vrf)#global import map globimaploop
```

The following commands create a BGP session between the CE router and the parent VR.

On host 1, VR PE 1:

```
host1(config)#virtual-router pe1
host1:pe1(config)#router bgp 100
host1:pe1(config-router)#neighbor 10.4.4.4 remote-as 200
host1:pe1(config-router)# neighbor 10.4.4.4 ebgp-multihop
host1:pe1(config-router)#neighbor 10.4.4.4 update-source loopback1
host1:pe1(config-router)#exit
```

On host 2, VR CE 1:


```
host2(config)#virtual-router ce1
host2:ce1(config)#interface loopback 1
host2:ce1(config-if)#ip address 10.4.4.4 255.255.255.255
host2:ce1(config-if)#exit
host2:ce1(config)#ip route 10.2.2.2 255.255.255.255 atm2/1.1
host2:ce1(config)#router bgp 200
host2:ce1(config-router)#neighbor 10.2.2.2 remote-as 100
host2:ce1(config-router)#neighbor 10.2.2.2 ebgp-multihop
host2:ce1(config-router)#neighbor 10.2.2.2 update-source loopback1
host2:ce1(config-router)#exit
```

You must also configure either fallback global or a default route to a manually created shared interface in the VRF. See "Example: Configuring a Fallback Global Option" on page 477 or "Example: Configuring a Default Route to a Shared Interface" on page 476 for details.

You can use the BGP session between the CE router and the parent VR to enable the CE router to advertise prefixes within the VPN site that can be reachable from the global Internet. An alternative configuration is to use a global export map as described in "Understanding Route Distribution for a VRF using Maps" on page 443.

Related Documentation

- Creation of a BGP Session Between the CE Router and the Parent VR Overview on page 479
- Monitoring the VRF on page 513

- *global import map*
- *interface ip*
- *interface loopback*
- *ip address*
- *ip route*
- *ip share-interface*
- *ip vrf*
- *match ip address*
- *neighbor ebgp-multihop*
- *neighbor remote-as*
- *neighbor update-source*
- *rd*
- *route-map*
- *route-target*
- *router bgp*
- *virtual-router*

Traffic Flow from the Internet to the VPN Overview

When traffic flows from the Internet to a VPN, the traffic arrives at the PE router on an interface in the global context. BGP performs a lookup in the global IP routing table, which normally does not contain VPN routes. You can use one of the following methods to advertise public VPN routes to the Internet (get the routes into the global routing table) and thus enable traffic flow from the Internet to those VPNS.

- Manually create shared interfaces in the parent VR and manually add static routes to those shared interfaces. See "VRF–to–VR Peering Overview" on page 465 for more information.
- Export VPN routes to the global BGP RIB. See "Understanding Route Distribution for a VRF using Maps" on page 443.

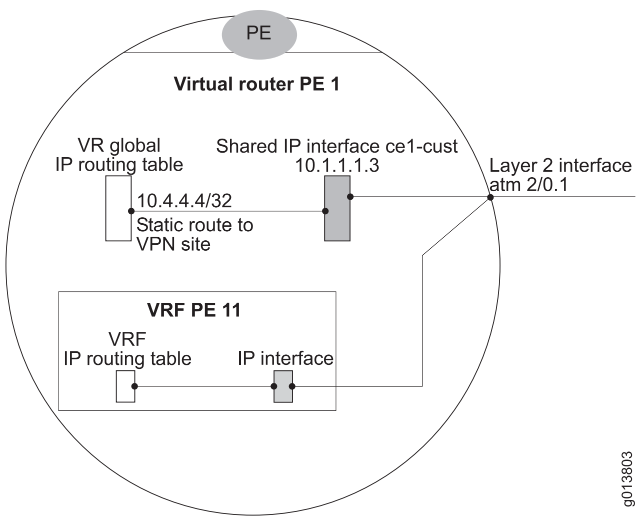
Related Documentation

- Understanding Internet Access and VPNs on page 474
- Example: Adding Static Routes to a Shared IP Interface on page 483
- Example: Exporting VPN Routes to Global BGP RIB Using Global Export Maps on page 484

Example: Adding Static Routes to a Shared IP Interface

You can introduce routes to VPN sites into the global routing table by placing static routes to the VPN sites into the global table. The static routes must point to shared IP interfaces that are shares of the PE-CE interface for each particular VPN site. The static routes must then be injected into BGP (possibly as part of an aggregate) so that they can be reached from the Internet. Figure 107 on page 483 illustrates this approach:

Figure 107: Static Route to Shared IP Interface



The following commands configure the shared interface and a static route:

```
host1(config)#virtual-router pe1
```

```
host1:pe1(config)#interface ip ce1-cust
host1:pe1(config-if)#ip share-interface atm2/0.1
host1:pe1(config-if)# ip address 10.1.1.3 255.255.255.0
host1:pe1(config-if)#exit
host1:pe1(config)#ip route 10.4.4.4 255.255.255.255 ip ce1-cust
```

Related Documentation

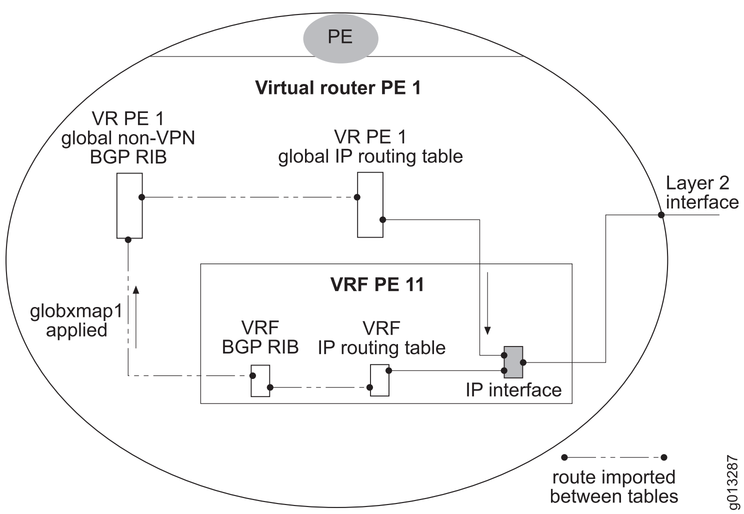
- Traffic Flow from the Internet to the VPN Overview on page 483
- *interface ip*
- *ip address*
- *ip route*
- *ip share-interface*
- *virtual-router*

Example: Exporting VPN Routes to Global BGP RIB Using Global Export Maps

The global export map enables VPN routes to be automatically exported from the BGP RIB table in a VRF to the global BGP RIB table (the BGP RIB table of the parent VR) based on policy. A route map determines which routes are exported and which are not.

When they are installed in the global IP routing table, these exported routes point to the IP interface in the VRF as shown in Figure 108 on page 484. See "Types of Maps Overview" on page 446 for more information.

Figure 108: Global Export Map Applied to Routes Exported from VRF BGP RIB



The following commands configure the route map and global export map:

```
host1(config)#virtual-router pe1
host1:pe1(config)#access-list dot-one permit 0.0.0.1 255.255.255.0
host1:pe1(config)#route-map globxmap1
host1:pe1(config-route-map)#match ip address dot-one
host1:pe1(config-route-map)#set local-pref 200
host1:pe1(config-route-map)#exit
```



```
host1:pe1(config)#ip vrf pe11
host1:pe1(config-vrf)#rd 100:1
host1:pe1(config-vrf)# route-target both 100:1
host1:pe1(config-vrf)#global export map globxmap1
host1:pe1(config-vrf)#exit
```

Related Documentation

- Traffic Flow from the Internet to the VPN Overview on page 483
- Monitoring the VRF on page 513

- *access-list*
- *global export map*
- *ip vrf*
- *match ip address*
- *rd*
- *route-map*
- *route-target*
- *set local-preference*
- *virtual-router*

IPv4 VPNs

The following sections explain IPv4 VPNs:

Carrier-of-Carriers IPv4 VPNs Overview

A carrier-of-carriers VPN is a two-tiered relationship between a provider carrier and a customer carrier. In a carrier-of-carriers VPN, the provider carrier provides a VPN backbone network for the customer carrier (Tier 1). The customer carrier, in turn, provides layer 3 VPN or Internet services to its end customers (Tier 2).

This section provides the background you need to understand carrier-of-carriers VPNs in general, but deals with IPv4 VPNs. For information about carrier-of-carriers IPv6 VPNs, see "Carrier-of-Carriers IPv6 VPNs Overview" on page 492.

The carrier-of-carriers VPN enables the customer carrier to provide the following services for its end customers:

- Traditional IP services—The customer carrier provides Internet connections for its customers and uses the provider carrier's VPN to connect its dispersed networks.

- Layer 3 VPN services—The customer carrier provides VPN services for its customers and uses the provider carrier's VPN for the backbone that connects the customer carrier's VPN sites. This environment is called a hierarchical VPN, because there are multiple tiers of VPNs—the tier-1 backbone VPN of the provider carrier and the tier-2 VPNs of the customer carrier.

In a hierarchical carrier-of-carriers VPN environment, each carrier (or ISP) maintains the internal routes of its customers in VRF tables on its PE routers. Therefore, the customer carrier's internal routes are installed into the VRF routing tables of the provider carrier's PE routers and advertised across the provider carrier's core. Similarly, the internal routes of the customer carrier's customers are installed into the VRF routing tables of the customer carrier's PE routers. The customer carrier's external routing information is exchanged by its PE routers (which connect to the provider carrier's VPN) over their own IBGP session.

NOTE: **To the customer carrier, the router it uses to connect to the provider carrier's VPN is a PE router. However, the provider carrier views this device as a CE router.**

Carrier-of-carriers VPNs provide the following benefits to the customer carriers:

- Reduced VPN administration—The VPN backbone is managed by the provider carrier.
- Reduced routing management—Intersite routing issues are the responsibility of the provider carrier.
- Flexibility—The VPN backbone can be used to deliver both VPN services and Internet connectivity services.

The following benefits are provided to the provider carriers:

- Reduced VPN administration—Provider carriers do not have to maintain separate VPNs for each customer carrier's end customer.
- Reduced router management—Customer carriers manage their own CE routers.
- Scalability—The provider carrier's PE routers do not maintain the end customer's external routes (as required in a traditional networking environment); the carrier-of-carriers network easily scales as the number of external routes and VPNs increases.

The following sections describe the two types of carrier-of-carriers environments.

Related Documentation

- MBGP Overview on page 391
- Customer Carrier as an Internet Service Provider on page 487
- Customer Carrier as a VPN Service Provider on page 489

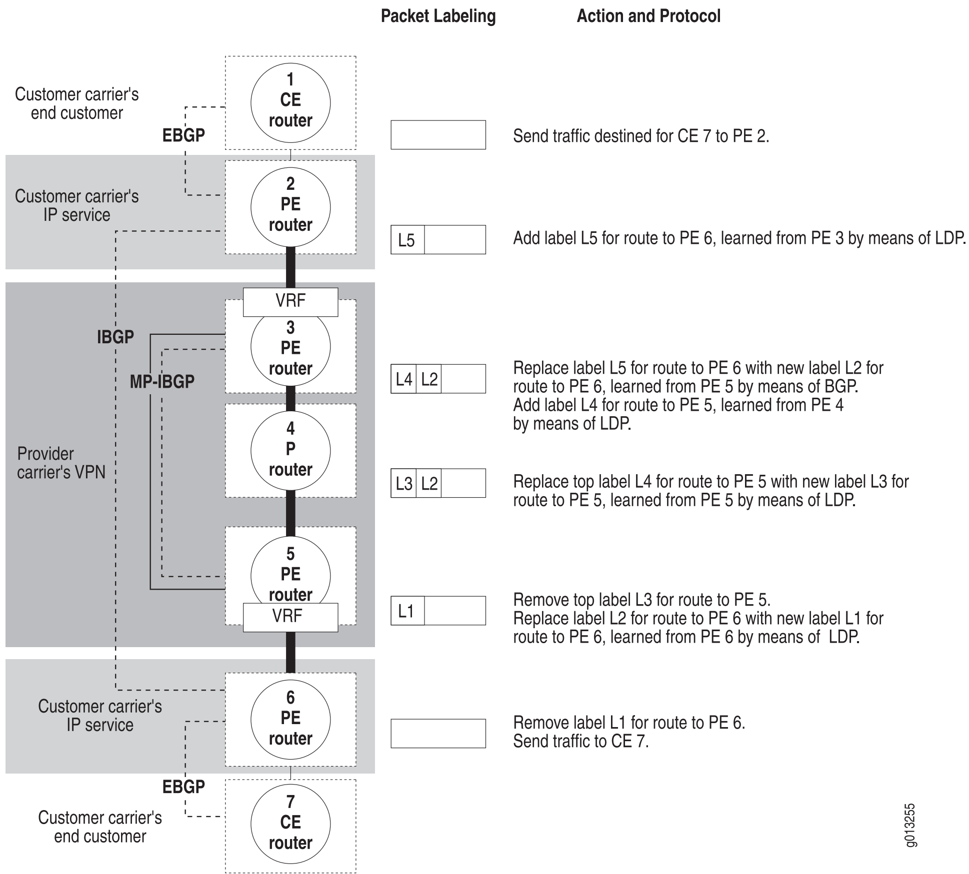
Customer Carrier as an Internet Service Provider

The provider carrier's VPN can function as the backbone for a customer carrier that provides Internet services for its customers at multiple sites. In this type of carrier-of-carriers environment, MPLS label-switched paths are established among the customer carrier's PE routers that connect to the provider carrier at each site. Routes are learned and maintained as follows:

- The customer carrier's internal routes are learned and advertised across the provider carrier's VPN. The customer carrier's external routes are *not* installed in the provider's VPN.
- The customer carrier's PE routers that connect to the provider's VPN use LDP to exchange labels for the internal routes between themselves and the provider carrier's PE router.
- The customer carrier's PE routers that connect to the provider's VPN learn external routes through IBGP sessions among themselves.

Figure 109 on page 488 shows a sample carrier-of-carriers environment in which the customer carrier provides Internet connectivity services to its customers. The figure shows how the labels are added and removed as the traffic traverses the network. The label-signaling protocol is assumed to be LDP.

Figure 109: Carrier-of-Carriers Internet Service



Related Documentation

- Carrier-of-Carriers IPv4 VPNs Overview on page 485
- Configuring Customer Carrier as an Internet Service Provider on page 488

Configuring Customer Carrier as an Internet Service Provider

You must complete the following configuration process when the customer carrier provides Internet connectivity for its customers.

On the provider carrier's PE router:

1. Basic MPLS Configuration Tasks on page 280
2. Configure BGP.
3. Configure an IGP.
4. Configure LDP.
5. Configure VRF.
6. Enable carrier-of-carriers support on the VRF; use the **mpls topology-driven-lsp** command in the context of the VRF virtual router to enable MPLS support.

7. Enable LDP on the interface in the VRF that connects to the customer carrier's PE router.
8. Use the **show ip bgp vpnv4 vrf** *vrfname* **summary** command to verify that carrier-of-carriers support is enabled.

On the customer carrier's PE router that connects to the provider carrier's PE router:

1. Configure MPLS.
2. Configure BGP.
3. Configure an IGP.
4. Configure LDP—Enable carrier-of-carriers support on the VR; use the **mpls topology-driven-lsp** command in the context of the VRF virtual router to enable LDP support.
5. Enable LDP on the interface in the VR that connects to the provider carrier's PE router.

Related Documentation

- Customer Carrier as an Internet Service Provider on page 487
- *mpls topology-driven-lsp*

Customer Carrier as a VPN Service Provider

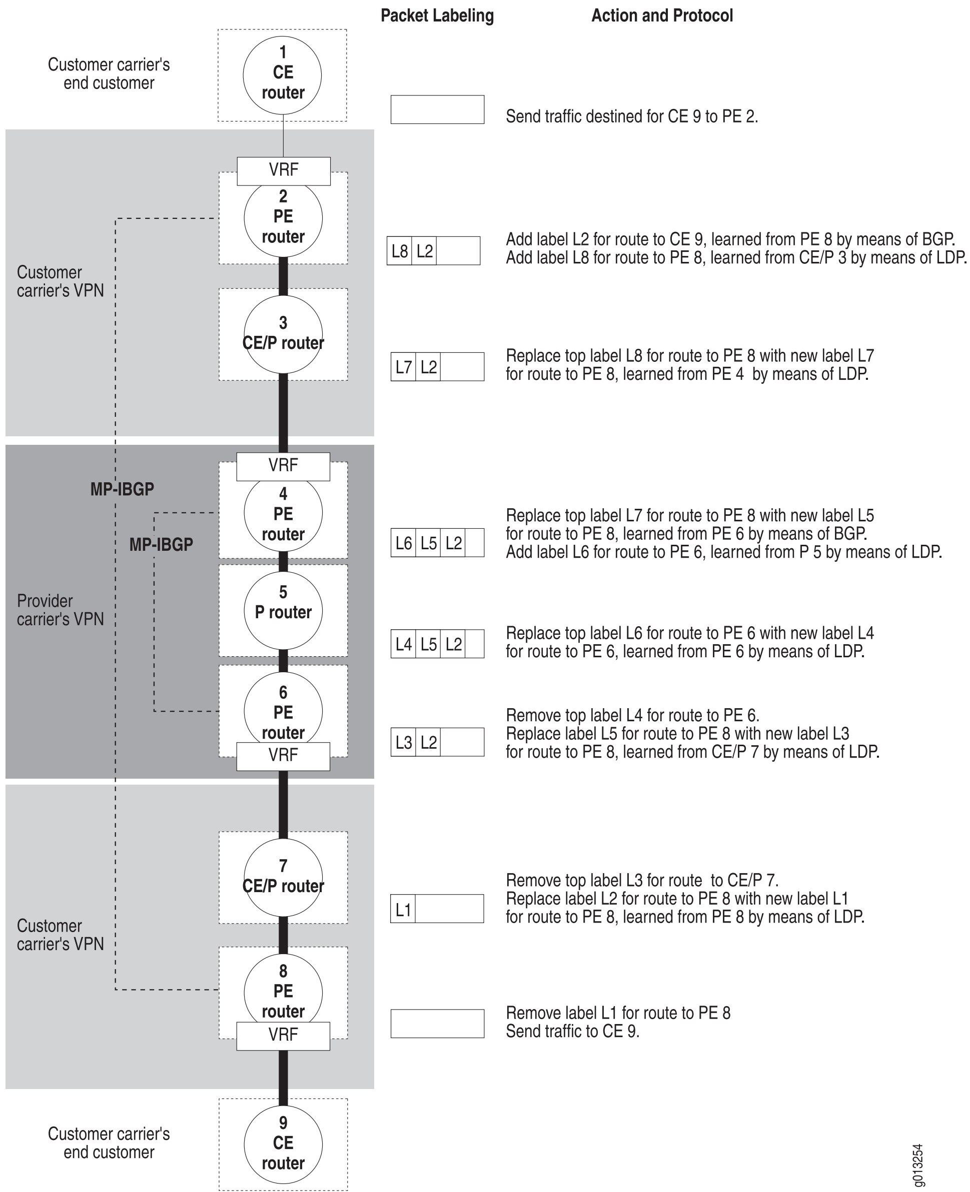
The carrier-of-carriers VPN can be used to create two-tiered hierarchical VPNs. In a hierarchical VPN, the provider carrier's VPN is the backbone, or tier-1 VPN, and the customer carrier provides the tier-2 VPN services to its customers.

In a hierarchical VPN environment, each carrier maintains the internal routes of its customers in VRF tables on its PE routers. Routes are learned and maintained as follows:

- In the provider carrier's VPN, PE routers use MP-IBGP to exchange labeled VPN routes that correspond to the internal routes of the customer carrier's VPN sites.
- In the customer carrier's VPN, PE routers use MP-IBGP sessions to exchange labeled VPN routes that correspond to the end customer's VPN routes.

Figure 110 on page 490 shows a sample carrier-of-carriers environment in which the customer carrier provides VPN services to its customers.

Figure 110: Carrier-of-Carriers VPN Service



NOTE: In a carrier-of-carriers environment, a provider carrier creates a backbone VPN that is used by a customer carrier. You must enable carrier-of-carriers support on the VRF of the provider carrier's PE device that connects to the PE device of the customer carrier.

NOTE: You can run BGP instead of LDP as the label distribution protocol on the PE-CE link between the Tier 1 and the Tier 2 carriers in a carrier-of-carriers topology. This capability is available for carriers providing Internet access or VPN service to end users.

Related Documentation

- Carrier-of-Carriers IPv4 VPNs Overview on page 485
- Configuring Customer Carrier as a VPN Service Provider on page 491

Configuring Customer Carrier as a VPN Service Provider

You must complete the following configuration process when the customer carrier provides VPN services for its customers.

On the provider carrier's PE router:

1. "Basic MPLS Configuration Tasks" on page 280.
2. Configure BGP.
3. Configure an IGP.
4. Configure LDP.
5. Configure VRF.
6. Enable carrier-of-carriers support on the VRF; use the **mpls topology-driven-lsp** command in the context of the VRF virtual router to enable MPLS support.
7. Enable LDP on the interface in the VRF that connects to the customer carrier's PE router.
8. Use the **show ip bgp vpnv4 vrf** *vrfname* **summary** command to verify that carrier-of-carriers support is enabled.

On all of the customer carrier's routers, configure:

1. MPLS
2. An IGP
3. LDP

On the customer carrier's PE router that connects to the end customer's CE router, additionally configure:

1. BGP
2. VRF

Related Documentation

- Customer Carrier as a VPN Service Provider on page 489

- *mpls topology-driven-lsp*

IPv6 VPNs

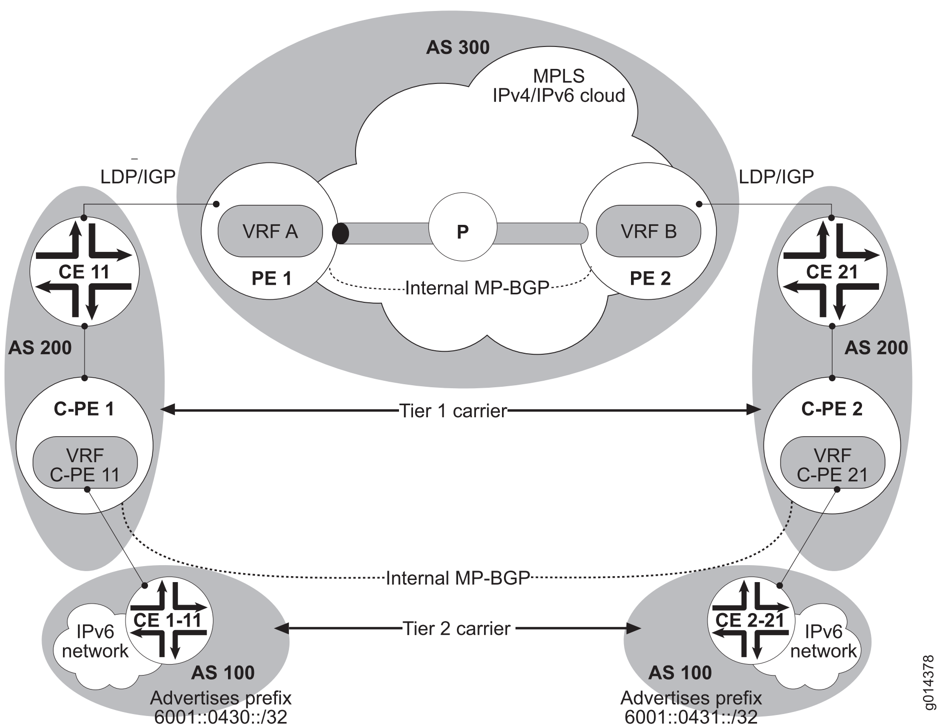
Carrier-of-Carriers IPv6 VPNs Overview

Figure 111 on page 492 illustrates a carrier-of-carrier scenario with IPv6 VPNs. MPLS labels are exchanged on the PE–CE link for customer-internal routes, but customer-external routes are not imported either into the VRFs on the PE router or into the core. VRFs maintain a routing table only for the customer-internal routes. Forwarding is accomplished primarily by label switching, without a routing table lookup.

Only customer-external routes (Tier 2 ISP routes as shown in Figure 111 on page 492) can be native IPv6 addresses. Because LDP over TCP over IPv6 is not currently supported, the customer-internal routes for which LDP can give out labels (Tier 1 ISP routes in Figure 111 on page 492) must be IPv4 addresses; they cannot be IPv6 addresses, whether native or IPv4-mapped.

For more information about carrier-of-carriers VPNs, see "Carrier-of-Carriers IPv4 VPNs Overview" on page 485

Figure 111: Carrier-of-Carrier IPv6 VPNs



Related Documentation

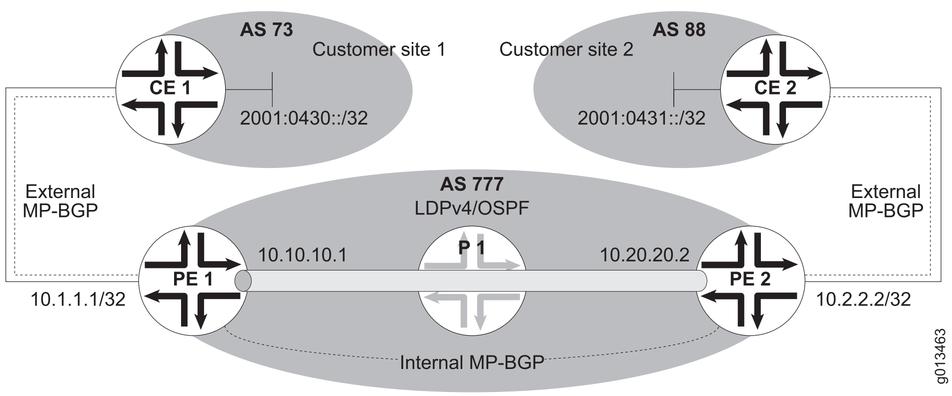
Connection of IPv6 Islands Across IPv4 Clouds with BGP Overview

If you have not upgraded your core to IPv6, you can still provide IPv6 services to customers by connecting remote IPv6 islands across IPv4 clouds by means of MP-BGP and MPLS. An IPv6 island is a network employing IPv6 addressing, such as a customer site. The IPv4 cloud consists of the PE–P–PE core.

NOTE: You must configure an IPv6 interface in the parent VR for this feature to work.

Consider Figure 112 on page 493. Each customer site is connected by means of a CE router to a PE router. The PE routers in this implementation are referred to as dual-stack BGP (DS-BGP) routers because they run both the IPv6 and IPv4 protocol stack.

Figure 112: IPv6 Tunneled over MPLS-IPv4



The PE routers learn IPv6 routes using MP-BGP over TCPv4 or TCPv6 from the CE devices. Alternatively, you can configure IPv6 static routes on the PE routers to reach the customer IPv6 networks through the CE IPv6 link. You can use any IPv6-enabled routing protocol to access the CE routers.

Use any MPLS signaling protocol to establish an MPLS base tunnel in the IPv4 core network. Each PE router runs MP-BGP over an IPv4 stack (MP-BGP/TCP/IPv4). MP-BGP advertises the customer IPv6 routes by exchanging IPv6 NLRI reachability information across the IPv4 cloud.

Each PE router announces the IPv4 address of its core-facing interface (the tunnel endpoint) to its PE peers as the BGP next hop. Because MP-BGP requires the next hop to be in the same address family as the NLRI, the IPv4 next-hop address must be embedded in an IPv6 format. The PE router advertises the IPv6 routes as labeled routes and an IPv6 next hop.

In the topology shown in Figure 112 on page 493, OSPF advertises reachability of the loopback (10.1.1.1/32 and 10.2.2.1/32) and core-facing (10.10.10.1/32 and 10.20.20.2/32) interfaces of the PE routers. LDP binds label L1 to 10.1.1.1/32 on the P router.

Router CE 1 establishes an MP-BGP session over TCPv4 to PE 1 and advertises its ability to reach the IPv6 network 2001:0430::/32. The MP-BGP update message specifies an AFI value of 2 (IPv6) and a SAFI value of 1 (unicast). As the next hop in the MP-REACH-NLRI attribute, CE 1 advertises the IPv6 address of the CE 1 interface that links to PE 1.

Both IPv4 and IPv6 addresses must be configured on the PE-CE link. The IPv6 address defaults to an IPv4-compatible address that can be overridden with policy.

PE 1 and PE 2 establish an MP-BGP session using their remote loopback IPv4 addresses as neighbor addresses. Router PE 1 installs in its IPv6 global routing table the route advertised by CE 1. MP-BGP on PE 1 then binds a second-level label, L2, and advertises the route to PE 2 with an AFI value of 2 (IPv6) and a SAFI value of 4 (labeled routes). The next hop that PE 1 advertises in the MP-REACH-NLRI attribute is the IPv4 address of its loopback interface, 10.1.1.1, encoded in IPv6 format as ::10.1.1.1.

When MP-BGP on router PE 2 receives the advertisement, it associates the base tunnel (to 10.1.1.0/24, label L1) with the next hop (::10.1.1.1) that was advertised by PE 1 to reach the customer IPv6 island, 2001:0430::/32. Router PE 2 then uses MP-BGP (AFI = 2, SAFI = 1) to advertise to CE 2 its ability to reach this network.

CE 2 sends native IPv6 packets destined for the 2001:0430::/32 network to PE 2. On receipt, PE 2 performs a lookup in its global IPv6 routing table. PE 2 prepends two labels to the IPv6 header (L1–L2–IPv6) and then forwards the packet out its core-facing interface (10.2.2.2).

The P router does a lookup on L1 and label switches the packet toward PE 1. The P router can either replace L1 with another label or pop L1 if PE 1 requested PHP.

When PE 1 receives the packet on its core-facing interface, it pops all the labels and does a lookup in the global IPv6 routing table using the destination address in the IPv6 header. PE 1 then forwards the native IPv6 packet out to CE 1 on the IPv6 link.

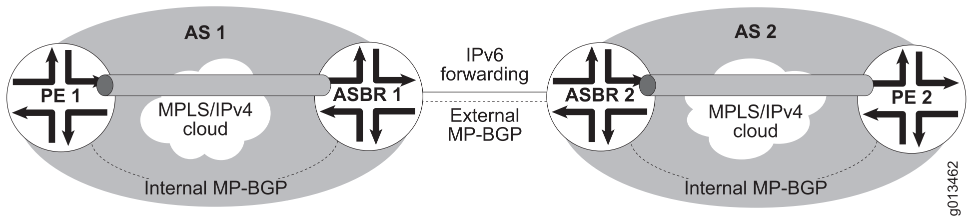
Related Documentation

• Carrier-of-Carriers IPv6 VPNs Overview on page 492

Connection of IPv6 Islands Across Multiple IPv4 Domains Overview

When the IPv6 islands are separated by multiple IPv4 domains, the autonomous system boundary routers between the IPv4 domains must be DS-BGP routers (Figure 113 on page 495).

Figure 113: IPv6 Tunneled Across IPv4 Domains



Each of these AS boundary routers establishes a peer relationship with the DS-BGP routers in its own domain, creating a separate mesh of tunnels among the DS-BGP routers of each domain. Routing between PE 1–ASBR 1 in AS 1 and between PE 2–ASBR 2 in AS 2 is accomplished by means of label-switched paths.

IPv6 unlabeled routes are exchanged through the external MP-BGP session between ASBR 1 and ASBR 2. Interdomain MPLS tunnels spanning multiple ASs are not supported.

Related Documentation

- Carrier-of-Carriers IPv6 VPNs Overview on page 492

Configuring IPv6 Tunneling over IPv4 MPLS

To configure IPv6 tunneling over MPLS:

1. On PE 1, configure both an IPv4 and an IPv6 interface toward the CE router. Use an IPv4-compatible IPv6 address.

   ```
   host1(config)#interface atm2/0.1
   host1(config)#atm pvc 1 0 1 aal5snap
   host1(config)#ip address 11.19.1.1 255.255.255.0
   host1(config)#ipv6 address ::11.19.1.1/126
   ```

2. On PE 1, configure an IPv4 interface facing the core.

   NOTE: For forwarding to work, you must configure at least one IPv6 interface on each line module where MPLS-encapsulated IPv6 traffic is expected. You can easily accomplish this by also configuring an IPv6 address on the core-facing interface.

   ```
   host1(config)#interface atm3/0.1
   host1(config)#atm pvc 30 0 30 aal5snap
   host1(config)#ip address 10.10.10.1 255.255.255.0
   host1(config)#ip address ::10.10.10.1/120
   ```

3. On PE 1, configure a loopback interface.

   ```
   host1(config)#interface loopback 1
   host1(config)#ip address 1.1.1.1 255.255.255.0
   ```

4. On PE 1, configure an IPv4 IGP and an MPLS signaling protocol in the core.
5. On PE 1, set up a base tunnel, or verify that one exists between the loopback addresses on the PE routers.
6. On PE 1, configure MP-BGP.

a. Enable BGP.

```
host1(config)#router bgp 100
```

b. Configure the MP-BGP CE and PE neighbors.

```
host1(config-router)#neighbor 11.19.1.2 remote-as 65000
host1(config-router)#neighbor 2.2.2.2 remote-as 100
```

c. Activate the neighbors in the IPv6 address-family.

```
host1(config-router)#address-family ipv6 unicast
host1(config-router-af)#neighbor 11.19.1.2 activate
host1(config-router-af)#neighbor 2.2.2.2 activate
```

d. Configure the MP-BGP PE neighbor to send labeled IPv6 prefixes.

```
host1(config-router-af)#neighbor 2.2.2.2 send-label
host1(config-router-af)#neighbor 2.2.2.2 update-source loopback 1
host1(config-router-af)#neighbor 2.2.2.2 next-hop-self
host1(config-router-af)#exit-address-family
```

7. Configure the P router with an IPv4 IGP and an MPLS signaling protocol.
8. Configure the PE 2 router as you did PE 1 in Steps 1–6.
9. Configure the CE 1 and CE 2 routers.
   a. Configure both an IPv4 and an IPv6 interface toward the PE router. Use an IPv4-compatible IPv6 address.
   b. Configure an MP-BGP session to the PE router over TCPv4, and activate the IPv6 unicast address family.

Related Documentation

- Carrier-of-Carriers IPv6 VPNs Overview on page 492
- *neighbor send-label*

OSPF and BGP/MPLS VPNs

The following sections explain OSPF and BGP/MPLS VPNs:

Understanding Usage of BGP/MPLS VPNs to Connect OSPF Domains

Before reading this section, we recommend you be thoroughly familiar with OSPF. For detailed information about that protocol, see *JunosE IP, IPv6, and IGP Configuration Guide*.

You can use BGP/MPLS VPNs to connect OSPF domains without creating OSPF adjacencies between the domains. The BGP/MPLS VPN backbone acts as either an OSPF backbone (area 0) or an OSPF area above the backbone.

In this topology, OSPF is the routing protocol between the CE router and the PE router. This OSPF link can be configured in area 0 or any other OSPF area. However, if the customer site has any connections to area 0, then at least one OSPF router configured on a CE router must have an area 0 link to a PE site. In this case, the BGP/MPLS VPN acts as if it is in an area above the OSPF backbone area. When the PE-CE link is in a nonbackbone area, the BGP/MPLS VPN acts as an OSPF backbone.

In either case, the OSPF router configured as a PE router in the BGP/MPLS VPN is always treated as an area border router (ABR) and functions as an area 0 router so that it can distribute interarea routes to the CE router. The BGP/MPLS VPN distributes both interarea and intra-area routes between PE routers as interarea, type 3 summary routes.

Distributing OSPF Routes from CE Router to PE Router

You configure OSPF in the VRF associated with the VPN and associate the interface connected to the CE router with the VRF. OSPF routes can then propagate from a CE router to a PE router when an OSPF adjacency has formed between the two routers. OSPF adds routes to the VRF's forwarding table at the PE router side with routes learned from the CE router.

Distributing Routes Between PE Routers

The OSPF routes in the VRF forwarding table are OSPF IPv4 routes, but BGP/MPLS VPNs distribute VPN-IPv4 routes by means of MP-BGP. You must configure the VRF to redistribute the OSPF routes into MP-BGP. MP-BGP converts each imported OSPF route to a VPN-IPv4 route, applies export policy to the route, and then propagates the route to a remote PE site by means of the MPLS/VPN backbone. At the destination PE router, MP-BGP places each route in the appropriate VRF forwarding table based on the import policy for each VRF and the route target associated with the route.

Related Documentation

- MBGP Overview on page 391
- Preservation of OSPF Routing Information Across the MPLS/VPN Backbone Overview on page 497
- Distribution of OSPF Routes from PE Router to CE Router Overview on page 499
- Prevention of Routing Loops Overview on page 499
- Understanding Remote Neighbors Usage to Configure OSPF Links on page 500
- Understanding OSPF Sham Links on page 501
- Configuring PE Router for OSPF on page 503

Preservation of OSPF Routing Information Across the MPLS/VPN Backbone Overview

MP-BGP attaches two new extended community attributes to the routes redistributed from OSPF:

- OSPF domain identifier extended community attribute
- OSPF route type extended community attribute

MP-BGP uses these attributes and the MED to preserve OSPF routing information across the BGP/MPLS VPN backbone.

OSPF Domain Identifier Attribute

The OSPF domain identifier attribute uniquely identifies the OSPF domain from which a route was redistributed into MP-BGP.

You must configure an OSPF domain ID for the VRFs on the PE router with the **domain-id** command. All VRFs that belong to a given OSPF domain must be configured with the same domain ID. If not configured, the domain ID defaults to zero. If you configure a value of zero, MP-BGP does not attach an OSPF domain identifier attribute.

If the OSPF domain ID for the destination PE router differs from the originating PE router, MP-BGP redistributes the route into OSPF as an OSPF type 5 external route.

OSPF Route Type Attribute

The route type attribute carries the OSPF area ID and LSA type, as indicated in Table 94 on page 498:

Table 94: Route Types and Route Origins

Type of Route	Origin of Route
1 – intra-area route	Type 1 LSA
2 – intra-area route	Type 2 LSA
3 – interarea summary route	Type 3 LSA
5 – external route (area ID = 0)	Type 5 LSA
7 – external route (area ID = 0)	Type 7 LSA

MP-BGP uses the route type conveyed by this extended community attribute to determine the best OSPF route when it installs the routes in the VRF forwarding table on the destination PE router.

Related Documentation

- Understanding Usage of BGP/MPLS VPNs to Connect OSPF Domains on page 496
- Distribution of OSPF Routes from PE Router to CE Router Overview on page 499
- Prevention of Routing Loops Overview on page 499
- Understanding Remote Neighbors Usage to Configure OSPF Links on page 500
- Understanding OSPF Sham Links on page 501

Distribution of OSPF Routes from PE Router to CE Router Overview

At the remote PE site, MP-BGP converts the OSPF routes to BGP VPN-IPv4 routes and sends them across the BGP/MPLS VPN backbone. At the destination PE router, MP-BGP must redistribute the BGP VPN-IPv4 routes back into OSPF IPv4 routes. The PE OSPF router becomes the originator of the routes, which are either type 5 external routes or type 3 internal routes. The PE router can announce the OSPF routes to the appropriate CE router through its directly connected PE-CE OSPF link.

If the route has a route type of inter or intra, it is redistributed as a type 3 summary interarea route and the destination PE router generates a type 3 LSA for it.

A route is redistributed as an external route if the route:

- Originates in an OSPF domain that is different from that of the destination PE router.
- Has a route type of 5 or 7, both of which indicate an external route.

In the first case, the PE router advertises the route as an external type 2 route. In the second case, the PE router advertises the route as an external type 2 route if the least-significant bit is set in the option byte in the route type extended community attribute; otherwise the PE router advertises the route as external type 1 route.

Related Documentation

- Understanding Usage of BGP/MPLS VPNs to Connect OSPF Domains on page 496
- Preservation of OSPF Routing Information Across the MPLS/VPN Backbone Overview on page 497
- Prevention of Routing Loops Overview on page 499
- Understanding Remote Neighbors Usage to Configure OSPF Links on page 500
- Understanding OSPF Sham Links on page 501

Prevention of Routing Loops Overview

PE routes disregard OSPF routes received from a CE router if the routes are advertised by:

- A type 3 LSA with the most-significant bit set in the LSA options field.
- A type 5 LSA that has a tag value equal to the VPN route tag associated with the OSPF VRF on that PE router.

When the destination PE router originates a type 3 LSA learned from BGP to a CE router, the PE router sets the most-significant bit in the LSA options field to identify the LSA as being generated from a PE router. Doing this prevents the LSA from being passed back to the BGP/MPLS VPN through a different PE router.

When the destination PE router originates a type 5 LSA learned from BGP to a CE router, the PE router replaces the external route tag in the LSA with the VPN route tag. You configure the VPN route tag for the OPSF VRF on the PE router with the **domain-tag** command. The value of a VPN route tag must be unique within an OSPF domain, so that the same external route is not propagated back to the BGP/MPLS VPN backbone through another PE-CE link.

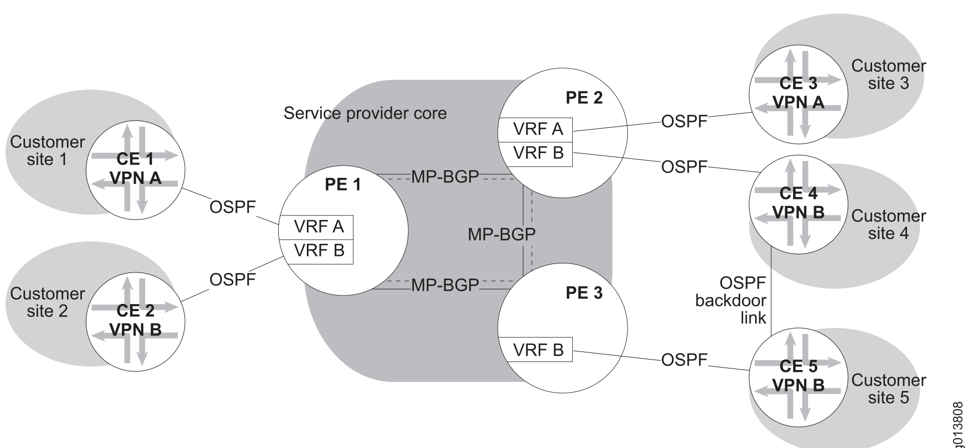
Related Documentation

- Understanding Usage of BGP/MPLS VPNs to Connect OSPF Domains on page 496
- Preservation of OSPF Routing Information Across the MPLS/VPN Backbone Overview on page 497
- Distribution of OSPF Routes from PE Router to CE Router Overview on page 499
- Understanding Remote Neighbors Usage to Configure OSPF Links on page 500
- Understanding OSPF Sham Links on page 501
- Configuring PE Router for OSPF on page 503
- *domain-tag*

Understanding Remote Neighbors Usage to Configure OSPF Links

When you employ OSPF as the PE-CE routing protocol in a BGP/MPLS VPN and also configure OSPF backdoor links between VPN sites outside the backbone, the backdoor links are always preferred over the backbone paths between the VPN links. OSPF sham links prevent this problem, and you can implement them with OSPF remote neighbors. Consider the topology shown in Figure 114 on page 500.

Figure 114: OSPF Topology with Backdoor Link



The PE routers are each running a separate logical OSPF instance for each VRF. Each of these OSPF instances has adjacencies with their directly connected CE routers and exchanges LSAs with those CE routers. The OSPF routes that are learned from a directly connected CE router are installed into the IP routing table of the VRF associated with that CE router.

The OSPF routes in the VRF's IP routing table are then redistributed into MP-BGP and advertised as VPNv4 routes to other PE routers. MP-BGP attaches extended communities to the advertised routes to carry OSPF-specific attributes such as the route type and the domain ID across the backbone.

At the remote PE router, the BGP routes are installed in the IP routing table of the VRF and then redistributed back into the logical OSPF instance for that VRF. The remote PE router uses the BGP extended communities to determine the type of LSA to send to CE routers.

As a result the intra-area OSPF routes in one VPN site appear as interarea OSPF routes at the remote VPN sites.

OSPF Backdoor Links

OSPF backdoor links typically serve as backup paths, providing a way for traffic to flow from one VPN site to the other only if the path over the backbone is broken.

However, when the OSPF backdoor link connects two sites that are in the same OSPF area, the undesired result is that the path over the OSPF backdoor link is always preferred over the path over the backbone.

In Figure 114 on page 500, the OSPF backdoor link connects customer site 4 to customer site 5 directly, without going through the backbone. OSPF uses the backdoor path for traffic flow between these two sites for the following reasons:

- At CE 4 and CE 5, the path over the OSPF backdoor link is an intra-area path, whereas the path over the backbone is an interarea path. OSPF always uses intra-area paths before interarea paths.
- At PE 2 and PE 3, the OSPF routes received from the respective directly connected CE routers have a better administrative distance than the IBGP routes received from the remote PE router. OSPF uses routes with better administrative distances.

Related Documentation

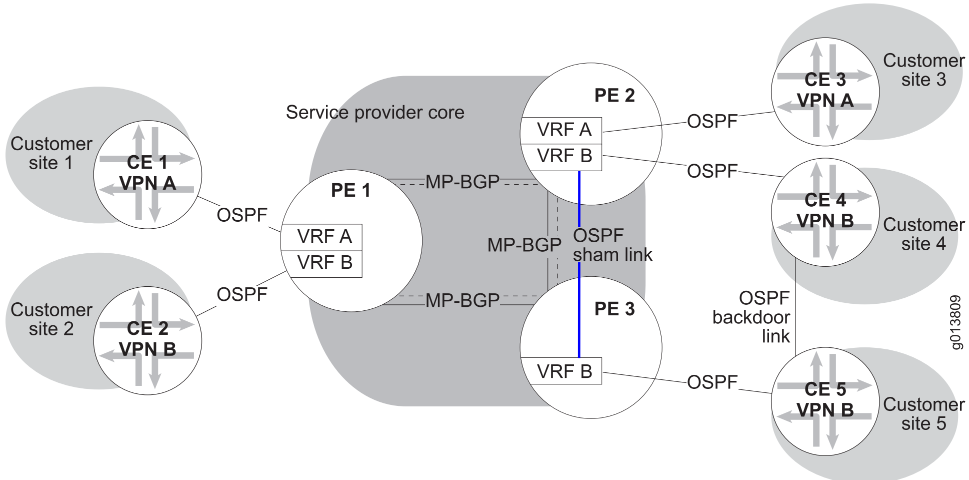
- Understanding Usage of BGP/MPLS VPNs to Connect OSPF Domains on page 496
- Preservation of OSPF Routing Information Across the MPLS/VPN Backbone Overview on page 497
- Distribution of OSPF Routes from PE Router to CE Router Overview on page 499
- Prevention of Routing Loops Overview on page 499
- Understanding OSPF Sham Links on page 501

Understanding OSPF Sham Links

Figure 115 on page 502 shows how you can use OSPF sham links to avoid the problem created by the intra-area backdoor link. The sham link is a logical intra-area link between VRF B on PE 2 and PE 3. OSPF creates an adjacency and exchanges LSAs across the sham link. As a result, OSPF sees both the path over the backdoor link and the path over the backbone as intra-area paths. OSPF then selects the best path based on the metrics of the links and selects the sham link path, ensuring that the backdoor link is not used.

NOTE: If the VPN sites are not connected by an OSPF backdoor link or if the VPN sites are in different OSPF areas, the problem does not exist and you do not need to configure an OSPF sham link.

Figure 115: OSPF Sham Link



Use the **remote-neighbor** command to configure the OSPF sham link on both VRFs joined by the link. If a BGP route and an OSPF route to the same destination are both installed in the IP routing table, OSPF uses the OSPF route because it has a better administrative distance by definition.

If you redistribute OSPF routes into BGP in each VRF, you do not want the OSPF routes that point to sham links to be redistributed into BGP. If they were redistributed, multiple BGP routes for a single OSPF route would exist: one BGP route at each endpoint of a sham link.

Use the **dont-install-routes** command to prevent OSPF routes pointing to the sham link from being installed in the IP routing table of the VRF, and thus to prevent them from being redistributed into BGP. Forwarding still works using the MP-IBGP routes received from the remote PE router.

Using this command avoids having many BGP routes to the same prefix by preventing OSPF routes learned over the sham link from being redistributed back into BGP even when you have configured redistribution of OSPF routes into BGP.

Use the **ttl** command to configure a TTL for the remote neighbor because the neighbor might be more than a single hop away. Use the **update-source** command to specify the loopback address used as the source address for the OSPF connection to the remote neighbor.

If you do not configure a sham link between each pair of PE routers for which a backdoor link exists, then you need to redistribute BGP routes back into OSPF.

For more information about OSPF remote neighbors, see *Remote Neighbors* in the *JunosE IP, IPv6, and IGP Configuration Guide*.

Related Documentation

- Understanding Usage of BGP/MPLS VPNs to Connect OSPF Domains on page 496
- Preservation of OSPF Routing Information Across the MPLS/VPN Backbone Overview on page 497

- Distribution of OSPF Routes from PE Router to CE Router Overview on page 499
- Prevention of Routing Loops Overview on page 499
- Understanding Remote Neighbors Usage to Configure OSPF Links on page 500

- *dont-install-routes*
- *remote-neighbor*
- *ttl*
- *update-source*

Configuring PE Router for OSPF

At a minimum, perform the following tasks on each PE router to configure them for OSPF. For other OSPF configuration tasks, see *OSPF Configuration Tasks* in the *JunosE IP, IPv6, and IGP Configuration Guide*.

1. Create the VRF.

   host1(config)#ip vrf ospf2
   Proceed with new VRF creation? [confirm]
   host1(config-vrf)#**rd 100:85**
   host1(config-vrf)#**exit**

2. Start OSPF on the VRF, either from the parent VR or directly from the VRF.

   From the parent VR:

   host1(config)#**router ospf 5 vrf ospf2**

   From the VRF:

   host1(config)#**virtual-router :ospf2**
   host1:default:ospf2(config)#**router ospf 5**

   The command prompts in the remaining steps reflect using the latter method for starting OSPF.

3. Configure the OSPF domain ID.

   host1:default:ospf2(config-router)#**domain-id 45**

4. Configure the VPN route tag.

   host1:default:ospf2(config-router)#**domain-tag 1200**

5. Redistribute routes learned from other PE routers back into OSPF.

   host1:default:ospf2(config-router)#**redistribute bgp**

6. Create an address family in BGP.

   host1:default(config)#**router bgp 100**
   host1:default(config-router)#**address-family ipv4 unicast vrf ospf2**

7. Redistribute OSPF routes into BGP.

   host1:default(config-router)#**redistribute ospf**

Related Documentation

- Understanding Usage of BGP/MPLS VPNs to Connect OSPF Domains on page 496
- Monitoring the VRF on page 513
- *domain-id*
- *domain-tag*

CHAPTER 7

Monitoring BGP/MPLS VPNs

This chapter describes how to monitor the BGP/MPLS VPN settings.

NOTE: The E120 router and E320 router output for **monitor** and **show** commands is identical to output from other E Series routers, except that the E120 and E320 router output also includes information about the adapter identifier in the interface specifier (*slot/adapter/port*).

Enabling the MP-BGP Events Log Display

To display information about MP-BGP logs for inbound or outbound events, or both.

- Issue the **debug ip mbgp** command:

  host1#**debug ip mbgp**

Related Documentation

- Disabling the MP-BGP Events Log Display on page 522
- *debug ip mbgp*
- *undebug ip mbgp*

Monitoring BGP Next Hops for VPN

Purpose Display information about BGP next hops.

Action To display the next hops for all the VRFs:

```
host1:pe2#show ip bgp vpnv4 all next-hops

Indirect next-hop 10.1.1.1
  Resolution in IP route table of VR
       IP indirect next-hop index 10
       Reachable (metric 3)
       Number of direct next-hops is 1
              Direct next-hop ATM4/1.20 (10.20.20.1)
  Resolution in IP tunnel-route table of VR
       MPLS indirect next-hop index 17
       Reachable (metric 3)
       Number of direct next-hops is 1
              Direct next-hop: MPLS next-hop 18
  Reference count is 1

Indirect next-hop 10.21.21.2
  Resolution in IP route table of VR
       IP indirect next-hop index 5
       Reachable (metric 0)
       Number of direct next-hops is 1
              Direct next-hop ATM4/0.21 (10.21.21.2)
  Resolution in IP tunnel-route table of VR
       MPLS indirect next-hop index 14
       Reachable (metric 0)
       Number of direct next-hops is 1
              Direct next-hop ATM4/0.21.mpls
  Reference count is 3
```

To display the next hops for specified VRFs:

```
host1:pe2#show ip bgp vpnv4 vrf pe22 next-hops
Indirect next-hop 10.61.61.2
  Resolution in IP route table of VRF pe22
       IP indirect next-hop index 3
       Reachable (metric 0)
       Number of direct next-hops is 1
              Direct next-hop ATM4/0.61 (10.61.61.2)
  Resolution in IP tunnel-route table of VRF pe22
       Not reachable
  Reference count is 2
```

Meaning Table 95 on page 506 lists the **show ip bgp next-hop** command output fields.

Table 95: show ip bgp next-hop Output Fields

Field Name	Field Description
Indirect next-hop	BGP next-hop attribute as received in the BGP update message
Resolution	Describes where the indirect next hop is resolved: the IP routing table, the IP tunnel routing table, or both, and whether this is in a VR or VRF
IP indirect next-hop index	Index number of the IP indirect next hop that this BGP indirect next hop resolves to
MPLS indirect next-hop index	Index number of the MPLS indirect next hop that this BGP indirect next hop resolves to

Table 95: show ip bgp next-hop Output Fields *(continued)*

Field Name	Field Description
Reachable	Indicates whether or not the indirect next hop is reachable.
Metric	Metric of the BGP indirect next hop
Number of direct next-hops	Number of the equal-cost legs of direct next hops that this indirect next hop resolves to
Direct next-hop	IP interface and next-hop IP address that resolve the BGP indirect next hop; the direct next hop can also be an IP indirect next hop or an MPLS indirect next hop when chains of next hops are in use
Reference count	Number of label mappings of BGP routes that use this next hop

Related Documentation

- *show ip bgp next-hops*

Monitoring VRF Interfaces

Purpose Display information about the interfaces associated with VRFs.

Action To display information about the interfaces associated with the specified VRFs:

```
host1#show ip interface vrf vpn1
null0 is up, line protocol is up
  Network Protocols: IP
  Internet address is 255.255.255.255/255.255.255.255
  Broadcast address is 255.255.255.255
  Operational MTU = 1500  Administrative MTU = 0
  Operational speed = 100000000  Administrative speed = 0
  Discontinuity Time = 0
  Router advertisement = disabled
  Administrative debounce-time = disabled
  Operational debounce-time    = disabled
  Access routing = disabled
  Multipath mode = hashed

atm4/0.77 is up, line protocol is up
  Network Protocols: IP
  Internet address is 7.8.7.7/255.255.255.0
  Broadcast address is 255.255.255.255
  Operational MTU = 9180  Administrative MTU = 0
  Operational speed = 155520000  Administrative speed = 0
  Discontinuity Time = 0
  Router advertisement = disabled
  Administrative debounce-time = disabled
  Operational debounce-time    = disabled
  Access routing = disabled
  Multipath mode = hashed

  In Received Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  In Policed Packets 0, Bytes 0
```



```
  In Error Packets 0
  In Invalid Source Address Packets 0
  Out Forwarded Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  Out Scheduler Drops Committed Packets 0, Bytes 0
  Out Scheduler Drops Conformed Packets 0, Bytes 0
  Out Scheduler Drops Exceeded  Packets 0, Bytes 0
  Out Policed Packets 0, Bytes 0
```

To display brief information about the interfaces associated with the VRFs:

```
host1#show ip interface vrf vpn1 brief
Interface               IP-Address      Status Protocol Description
null0                   255.255.255.255  up        up
atm4/0.77               7.8.7.7          up        up
```

Meaning Table 96 on page 508 lists the **show ip interface vrf** command output fields.

Table 96: show ip interface vrf Output Fields

Field Name	Field Description
Network Protocols	Type of network protocol configured
Broadcast address	IP address of the subnet
interface	Interface type and interface specifier
interface status	Status of the interface
Protocol Description	Status of the protocol
line protocol	Status of the line protocol
Link up/down trap	Status of SNMP link up/down traps on the interface
Internet address	IP address of the interface
Operational MTU	Actual MTU for the interface
Administrative MTU	Configured MTU for the interface
Operational speed	Actual speed
Administrative speed	Configured speed
Discontinuity Time	Value of sysUpTime the last time the integrity of the interface statistics was compromised
Router advertisement	Whether routes are advertised; enabled or disabled

Table 96: show ip interface vrf Output Fields *(continued)*

Field Name	Field Description
Administrative debounce-time	Configured debounce behavior, enabled or disabled. • Enabled—Indicates time in milliseconds that the router waits before generating an up or down event in response to a state change in the interface. • If the state changes back before the debounce timer expires, no state change is reported.
Operational debounce-time	Current debounce behavior, enabled or disabled. • If enabled, indicates time in milliseconds that the router waits before generating an up or down event in response to a state change in the interface. • If the state changes back before the debounce timer expires, no state change is reported.
Access routing	When enabled, an *access* route is installed to the host on the other end of the interface
Multipath mode	Algorithm used for ECMP: hashing of destination address and source address, or round-robin
In Received Packets, Bytes	Total number of packets and bytes received on an IP interface • Unicast—Number of unicast packets and bytes received on an IP interface • Multicast—Number of multicast packets and bytes received on an IP interface
In Policed Packets, Bytes	Number of packets and bytes discarded on a receive IP interface because of token bucket limiting
In Error Packets	Number of packets discarded on a receive IP interface because of IP header errors
In Invalid Source Address Packets	Number of packets discarded on a receive IP interface because of invalid IP source address (sa-validate enabled)
Out Forwarded Packets, Bytes	Number of packets and bytes forwarded out an IP interface • Unicast—Number of unicast packets and bytes forwarded out an IP interface • Multicast—Number of multicast packets and bytes forwarded out an IP interface
Out Scheduler Drops Committed Packets, Bytes	Number of committed packets and bytes dropped because of out queue threshold limit
Out Scheduler Drops Conformed Packets, Bytes	Number of conformed packets and bytes dropped because of out queue threshold limit

Table 96: show ip interface vrf Output Fields *(continued)*

Field Name	Field Description
Out Scheduler Drops Exceeded Packets, Bytes	Number of exceeded packets and bytes dropped because of out queue threshold limit
Out Policed Packets, Bytes	Number of packets and bytes discarded on a forwarding IP interface because of token bucket limiting

Related Documentation

- *show ip interface vrf*

Monitoring VRF Routing Protocols

Purpose Display information about the routing protocols associated with the VRF

Action To display information about OSPF routing protocol associated with the VRF

```
host1:pe1#show ip protocols vrf pe13
Routing Protocol is " ospf 1" with Router ID 13.13.13.1
  Distance is 110
Redistributing: bgp
  Address Summarization:
    None
  Routing for Networks:
    13.13.13.0/255.255.255.0 area 0.0.0.0
```

Meaning Table 97 on page 510 lists the **show ip protocols** command output fields.

Table 97: show ip protocols Output Fields

Field Name	Field Description
Routing Protocol	Protocol configured for routing
BGP	
Redistributing	Protocol to which BGP is redistributing routes
Default local preference	Local preference value
IGP synchronization	Status of IGP synchronization: enabled, disabled
Always compare MED	Status of multiexit discrimination: enabled, disabled
Router flap damping	Status of route dampening: enabled, disabled
Administrative Distance	External, internal, and local administrative distances
Neighbor Address	IP address of the BGP neighbor

Table 97: show ip protocols Output Fields *(continued)*

Field Name	Field Description
Neighbor Incoming/Outgoing update distribute list	Number of the access list for outgoing routes
Neighbor Incoming/Outgoing update prefix list	Number of the prefix list for incoming or outgoing routes
Neighbor Incoming/Outgoing update prefix tree	Number of the prefix tree for incoming or outgoing routes
Neighbor Incoming/Outgoing update filter list	Number of filter list for incoming routes
Routing for Networks	The network for which BGP is currently injecting routes
IS-IS	
System Id	6-byte value of the router
IS-Type	Routing type of the router: Level 1, Level 2
Distance	Administrative distance for IS-IS learned routes
Address Summarization	Aggregate addresses defined in the routing table for multiple groups of addresses at a given level or routes learned from other routing protocols
Routing for Networks	Network for which IS-IS is currently injecting routes
OSPF	
Router ID	OSPF process ID for the router
Distance	Administrative distance for OSPF learned routes
Redistributing	Protocol to which OSPF is redistributing routes
Address Summarization	Aggregate addresses defined in the routing table for multiple groups of addresses at a given level or routes learned from other routing protocols
Routing for Networks	Network for which OSPF is currently injecting routes
RIP	
Router Administrative State	RIP protocol state. Enable means it is allowed to send and receive updates. Disable means that it may be configured but it is *not* allowed to run yet.

Table 97: show ip protocols Output Fields *(continued)*

Field Name	Field Description
System Version	RIP versions allowed for sending and receiving RIP updates. The system version is currently set to RIP1, which sends RIP version 1 but will receive version 1 or 2. If the version is set to RIP2, the system will send and receive version 2 only. The default is configured for RIP1.
Update interval	Current setting of the update timer (in seconds)
Invalid after	Current setting of the invalid timer (in seconds)
hold down time	Current setting of the holddown timer (in seconds)
flushed interval	Current setting of the flush timer (in seconds)
Filter applied to outgoing route update	Access list applied to outgoing RIP route updates
Filter applied to incoming route update	Access list applied to incoming RIP route updates
Global route map	Route map that specifies all RIP interfaces on the router
Distance	Value added to RIP routes added to the IP routing table. The default is 120.
Interface	Interface type on which RIP protocol is running
Redistributing	Protocol to which RIP is redistributing routes
Routing for Networks	Network for which RIP is currently injecting routes

Related Documentation

- *show ip protocols*

Monitoring the VRF Routing Table

Purpose Display the routing table of the specified VRF

Action To display the routing table of the VRF

```
host1#show ip route vrf vpn2
Protocol/Route type codes:
I1- ISIS level 1, I2- ISIS level2,
I- route type intra, IA- route type inter, E- route type external,
i- metric type internal, e- metric type external,
O- OSPF, E1- external type 1, E2- external type2,
N1- NSSA external type1, N2- NSSA external type2

Prefix/Length   Type    Next Hop    Dist/Met     Intf
--------------- ------- ---------- --------     ------------
```



```
45.5.5.5/32     Connect  45.5.5.5    0/1          fastEthernet3/0
56.5.5.0/24     Connect  56.5.5.5    0/1          atm4/0.21
```

Meaning Table 98 on page 513 lists the **show ip route** command output fields.

Table 98: show ip route Output Fields

Field Name	Field Description
Protocol/Route type codes	Type of route
Prefix/Length	Network prefix for route in VRF routing table
Type	Protocol of route
Next Hop	IP address of the next hop to reach route
Dist/Met	Administrative distance and metric applied to route
Intf	Outgoing interface to reach route

Related Documentation

- *show ip route*

Monitoring the VRF

Purpose Display information about the VRFs in this virtual router: the route target of each VRF and the interfaces attached to each VRF.

Action To display all VRFs in this virtual router:

```
host1#show ip vrf
VRF Name        Default RD         Interfaces
 vpn1           1:1                null0
                                   atm4/0.77
 vpn2           1:3                null0
                                   fastEthernet3/0
                                   atm4/0.21
```

To display brief information about a specified VRF, specify a VRF name:

```
host1#show ip vrf vpn1
VRF Name        Default RD         Interfaces
 vpn1           1:1                null0
                                   atm4/0.77
```

To display detailed information about a specified VRF:

```
host1:pe1#show ip vrf detail
VRF  pe11; Default RD 100:11
  VRF IP Router Id: 10.11.11.1
  Default TTL: 127
  Reassemble Timeout: 30
  Interface Configured:
    null0  ATM2/0.11   tun mpls:vpnEgL17-3  ip dyn-24
  Import VPN Route Target Extended Communities:
```



```
     100:1
  Export VPN Route Target Extended Communities:
     100:1
  IPv4 Import Route-map: my-v4-import-map
  IPv6 Import Route-map: my-v6-import-map
  IPv4 Export Route-map: my-v4-export-map (cannot filter routes)
  IPv6 Export Route-map: my-v6-export-map (can filter routes)
  IPv4 Global Import Route-map: my-v4-global-import-map (max routes 5000)
  IPv6 Global Import Route-map: my-v6-global-import-map (max routes 1000)
  IPv4 Global Export Route-map: my-global-v4-export-map
  IPv6 Global Export Route-map: my-global-v6-export-map
  VRF  pe12; Default RD 100:12
  VRF IP Router Id: 10.12.12.1
  Default TTL: 127
  Reassemble Timeout: 30
  Interface Configured:
    null0  ATM2/0.12   tun mpls:vpnEgL18-4  ip dyn-25
  Import VPN Route Target Extended Communities:
     100:2
  Export VPN Route Target Extended Communities:
     100:2
  Import Route-map : importmap1
  Export Route-map : exportmap23 (can filter routes)
  Global Import Route-map : globalimportmap2
  Global Export Route-map : globalexportmap3
  VRF  pe13; Default RD 100:13
  VRF IP Router Id: 10.13.13.1
  Default TTL: 127
  Reassemble Timeout: 30
  Interface Configured:
    null0  ATM2/0.13 tun mpls:vpnEgL19-5 ip dyn-26
  Import VPN Route Target Extended Communities:
     100:3
  Export VPN Route Target Extended Communities:
     100:3
  No Import Route-map
  No Export Route-map
  No Global Import Route-map
  No Global Export Route-map
```

To display summary information about all interfaces associated with all VRFs configured in a virtual router:

```
host1:PE1#show ip vrf interfaces
Interface              IP-Address        Status Protocol   VRF
null0                  255.255.255.255/32  up         up   pe11
atm4/0.134             4.4.4.2/24          up         up   pe11
null0                  255.255.255.255/32  up         up   pe12
ip0                    6.6.6.8/24          up         up   pe12
null0                  255.255.255.255/32  up         up   pe13
loopback1              7.7.7.2/24          up         up   pe13
```

To display detailed information about the interfaces:

```
host1:PE1#show ip vrf interfaces detail
null0 is up, line protocol is up
  VRF: pe11
  Link up/down trap is disabled

  Internet address is 255.255.255.255/255.255.255.255
IP statistics:
  Rcvd:  0 local destination
```



```
         0 hdr errors, 0 addr errors
         0 unkn proto, 0 discards
  Frags: 0 reasm ok, 0 reasm req, 0 reasm fails
         0 frag ok, 0 frag creates, 0 frag fails
  Sent:  0 generated, 0 no routes, 0 discards
ICMP statistics:
  Rcvd:  0 errors, 0 dst unreach, 0 time exceed
         0 param probs, 0 src quench, 0 redirect,
         0 echo req, 0 echo rpy
         0 timestmp req, 0 timestmp rpy
         0 addr mask req, 0 addr mask rpy
  Sent:  0 errors, 0 dst unreach, 0 time excd
         0 param probs, 0 src qnch, 0 redirect
         0 timestamp req, 0 timestamp rpy
         0 addr mask req, 0 addr mask rpy

atm4/0.134 is up, line protocol is up
  VRF: pe11
  Link up/down trap is disabled

  Internet address is 4.4.4.2/255.255.255.0
IP statistics:
  Rcvd:  0 local destination
         0 hdr errors, 0 addr errors
         0 unkn proto, 0 discards
  Frags: 0 reasm ok, 0 reasm req, 0 reasm fails
         0 frag ok, 0 frag creates, 0 frag fails
  Sent:  0 generated, 0 no routes, 0 discards
ICMP statistics:
  Rcvd:  0 errors, 0 dst unreach, 0 time exceed
         0 param probs, 0 src quench, 0 redirect,
         0 echo req, 0 echo rpy
         0 timestmp req, 0 timestmp rpy
         0 addr mask req, 0 addr mask rpy
  Sent:  0 errors, 0 dst unreach, 0 time excd
         0 param probs, 0 src qnch, 0 redirect
         0 timestamp req, 0 timestamp rpy
         0 addr mask req, 0 addr mask rpy
In Received Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  In Forwarded Packets 0, Bytes 0
  In Total Dropped Packets 0, Bytes 0
    In Policed Packets 0
    In Invalid Source Address Packets 0
    In Error Packets 0
    In Discarded Packets 0
    In Fabric Dropped Packets 0

  Out Forwarded Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  Out Requested Packets 0, Bytes 0
  Out Total Dropped Packets 0, Bytes 0
    Out Scheduler Drops Committed Packets 0, Bytes 0
    Out Scheduler Drops Conformed Packets 0, Bytes 0
    Out Scheduler Drops Exceeded  Packets 0, Bytes 0
    Out Policed Packets 0
    Out Discarded Packets 0
    Out Fabric Dropped Packets 0
```

Meaning Table 99 on page 516 lists the **show ip vrf** command output fields.

Table 99: show ip vrf Output Fields

Field Name	Field Description
VRF Name	Name of each VRF
Default RD	Default route distinguisher for the VRF
Interfaces	Interfaces configured for the VRF
VRF	Name of the VRF
VRF IP Router Id	IP address that uniquely identifies the router
Default TTL	Time to live value in the IP header
Reassemble Timeout	Value to time out reassembled packets
Interface Configured	Interface configured for the VRF
Import VPN Route Target Extended Communities	List of VPNs from which the VRF accepts routing information
Export VPN Route Target Extended Communities	List of VPNs to which the VRF sends update messages
Import Route-map	Route map associated with the VRF that filters and modifies routes imported to the VRF from the global BGP VPN RIB. The map applies to both IPv4 and IPv6 routes, unless the field name is preceded by IPv4 (applies the map to only IPv4 routes) or IPv6 (applies the map to only IPv6 routes)
Export Route-map	Route map associated with the VRF that modifies and filters routes exported by the VRF to the global BGP VPN RIB. The map applies to both IPv4 and IPv6 routes, unless the field name is preceded by IPv4 (applies the map to only IPv4 routes) or IPv6 (applies the map to only IPv6 routes). The can filter routes text appears only if the filter keyword was issued for export map.
Global Import Route-map	Route map associated with the VRF that modifies routes imported to the VRF from the global BGP non-VPN RIB. The map applies to both IPv4 and IPv6 routes, unless the field name is preceded by IPv4 (applies to only IPv4 routes) or IPv6 (applies to only IPv6 routes).
Global Export Route-map	Route map associated with the VRF that modifies routes exported by the VRF to the global BGP non-VPN RIB. The map applies to both IPv4 and IPv6 routes, unless the field name is preceded by IPv4 (applies the map to only IPv4 routes) or IPv6 (applies the map to only IPv6 routes).
IP-Address	IP address of the interface
Status	Status of the interface

Table 99: show ip vrf Output Fields *(continued)*

Field Name	Field Description
Protocol	Status of the line protocol
In Forwarded Packets, Bytes	Number of packets and bytes forwarded into an output IP interface
In Total Dropped Packets, Bytes	Total number of packets and bytes discarded on a receive IP interface
In Discarded Packets	Number of packets discarded on the ingress interface because of a configuration problem rather than a problem with the packet itself
In Fabric Dropped Packets	Number of packets discarded on a receive IP interface because of internal fabric congestion
Out Requested Packets, Bytes	Number of packets and bytes requested to be forwarded out an IP interface
Out Total Dropped Packets, Bytes	Total number packets and bytes dropped by an IP interface on output
Out Discarded Packets	Number of packets discarded on the egress interface because of a configuration problem rather than a problem with the packet itself
Out Fabric Dropped Packets	Number of packets dropped because of internal fabric congestion
VRF	Name of the VRF with which the interface is associated
IP Statistics Rcvd	
local destination	Frames with this router as their destination
hdr errors	Number of packets containing header errors
addr errors	Number of packets containing addressing errors
unkn proto	Number of packets received containing unknown protocols
discards	Number of discarded packets
IP Statistics Frags	
reasm ok	Number of reassembled packets
reasm req	Number of requests for reassembly
reasm fails	Number of reassembly failures
frag ok	Number of packets fragmented successfully

Table 99: show ip vrf Output Fields *(continued)*

Field Name	Field Description
frag creates	Number of frames requiring fragmentation
frag fails	Number of packets unsuccessfully fragmented
IP Statistics Sent	
generated	Number of packets generated
no routes	Number of packets that could not be routed
discards	Number of packets that could not be routed that were discarded
ICMP Statistics Rcvd	
errors	Number of error packets received
dst unreach	Number of packets received with destination unreachable
time exceed	Number of packets received with time-to-live exceeded
param probs	Number of packets received with parameter error
src quench	Number of source quench packets received
redirect	Number of receive packet redirects
echo req	Number of echo request (PING) packets
echo rpy	Number of echo replies received
timestamp req	Number of requests for a timestamp
timestamp rpy	Number of replies to timestamp requests
addr mask req	Number of address mask requests
addr mask rpy	Number of address mask replies
ICMP Statistics Sent	
errors	Number of error packets sent
dst unreach	Number of packets sent with destination unreachable
time exceed	Number of packets sent with time-to-live exceeded
param probs	Number of packets sent with parameter error
src quench	Number of source quench packets sent

Table 99: show ip vrf Output Fields *(continued)*

Field Name	Field Description
redirect	Number of send packet redirects
timestamp req	Number of requests for a timestamp
timestamp rpy	Number of replies to timestamp requests
addr mask req	Number of address mask requests
addr mask rpy	Number of address mask replies

Related Documentation

- *show ip vrf*

Monitoring Load-Balanced Martini Circuits

Purpose Display information about load-balanced Martini circuits

Action To display brief information about load-balanced Martini circuits:

```
host1#show mpls l2transport load-balancing-group member-circuits brief
```

```
4 member ports:
      fastEthernet 2/0  down
      fastEthernet 3/0  30 member circuits
      fastEthernet 4/0  30 member circuits
      fastEthernet 5/0  30 member circuits
    90 member circuits
```

To display detailed information about load-balanced Martini circuits:

```
host1#show mpls l2transport load-balancing-group 100 member-circuits
routed to 10.9.1.3  on base LSP  tun mpls:lsp-de090103-32-3e
  load-balancing-group 100
  Martini group-id 2 vc-id 200002 mtu 1500
  State UP
  In  Label 57 on stack
    0 pkts, 0 hcPkts, 0 octets
    0 hcOctets, 0 errors, 0 discardPkts

  Out Label 59 on  tun mpls:lsp-de090103-32-3e
    0 pkts, 0 hcPkts, 0 octets
    0 hcOctets, 0 errors, 0 discardPkts
    queue 0: traffic class best-effort, bound to atm-vc ATM6/0.1
      Queue length 0 bytes
      Forwarded packets 0, bytes 0
      Dropped committed packets 0, bytes 0
      Dropped conformed packets 0, bytes 0
      Dropped exceeded packets 0, bytes 0
  Member Interfaces
    Interface fastEthernet 2/0.2  active
      Incoming Traffic Statistics
        0 pkts, 0 hcPkts, 0 octets
        0 hcOctets, 0 errors, 0 discardPkts
```

```
      Outgoing Traffic Statistics
        0 pkts, 0 hcPkts, 0 octets
        0 hcOctets, 0 errors, 0 discardPkts
    Interface fastEthernet 3/0.2
      Incoming Traffic Statistics
        0 pkts, 0 hcPkts, 0 octets
        0 hcOctets, 0 errors, 0 discardPkts
      Outgoing Traffic Statistics
        0 pkts, 0 hcPkts, 0 octets
        0 hcOctets, 0 errors, 0 discardPkts
```

NOTE: For a simpler view, the **show mpls l2transport interface** command displays only the currently active VLAN or S-VLAN subinterface. Because load-balanced circuits are configured on subinterfaces on multiple ports, only one of which is active at a given time, this command does not give a complete picture of the configuration.

Meaning Table 100 on page 520 lists the **show mpls l2transport load-balancing-group** command output fields.

Table 100: show mpls l2transport load-balancing-group Output Fields

Field Name	Field Description
routed to/base LSP	Identifies address of the router at the other end of the tunnel and the base tunnel that is selected to forward the traffic
load-balancing group	Group number
Martini group-id	Martini group ID number for the interface
state	State of the interface
vc-id	VC ID number for the interface
mtu	Maximum transmission unit for the inter
In label	Label sent to upstream neighbor for route; statistics below this field are the aggregate statistics for traffic from the core
Out label	Label received from downstream neighbor for route; statistics below this field are the aggregate statistics for traffic to the core
pkts	Number of packets sent across tunnel
hcPkts	Number of high-capacity (64-bit) packets sent across tunnel
octets	Number of octets sent across tunnel
hcOctets	Number of high-capacity (64-bit) octets sent across tunnel

Table 100: show mpls l2transport load-balancing-group Output Fields *(continued)*

Field Name	Field Description
errors	Number of packets dropped for some reason before being sent
queue 0	Number of the queue for which statistics are being displayed and whether the queue is under traffic class control
traffic class	Name of traffic class
bound to	Interface to which queue is bound
Queue length	Size of queue in length and bytes
Forwarded	Number of forwarded packets and bytes
Dropped committed	Number of committed packets and bytes dropped
Dropped conformed	Number of conformed packets and bytes dropped
Dropped exceeded	Number of exceeded packets and bytes dropped
discardPkts	Number of packets discarded due to lack of buffer space before being sent
Member Interfaces	Information about the member interfaces for the circuit
Interface	Interface specifier and status; active indicates it is being used for traffic from the core; if active is not displayed, interface is not currently being used for traffic, but the statistics may be valid
member ports	Number and type of candidate ports configured for the group, including interface specifiers and state
member circuits	Number of member circuits configured for each port and for the group

Related Documentation

- *show mpls l2transport load-balancing-group*
- *show mpls l2transport interface*

Monitoring MPLS Tunnels

Purpose Display status and configuration for all tunnels or for a specific tunnel in the current router context

Action To display the configuration for all tunnels:

```
host12#show mpls tunnels
```

```
LSP vpnIngress-21 to   3.3.3.3
State:  Up
Out label is Variable Interface
102 pkts, 0 hcPkts, 13464 octets
0 hcOctets, 0 errors, 0 discardPkts
Labels:
16 17 18 19
```

Meaning Table 101 on page 522 lists the **show mpls tunnels** command output fields.

Table 101: show mpls tunnels Output Fields

Field Name	Field Description
State	Status of tunnel, up or down
Out Label	In the default case for a BGP/MPLS VPN, the Variable Interface, which indicates that a packet exiting the interface is going through a variable interface and that one of the labels listed further in the display will be prepended to the packet
pkts	Number of packets sent across tunnel
hcpkts	Number of high-capacity (64-bit) packets sent across tunnel
octets	Number of octets sent across tunnel
hcoctets	Number of high-capacity (64-bit) octets sent across tunnel
errors	Number of packets that are dropped for some reason before being sent
discardPkts	Number of packets that are discarded due to lack of buffer space before being sent
Labels	List of labels associated with the variable interface; one will be selected to be prepended to packets before being sent across tunnel

Related Documentation

- *show mpls tunnels*

Disabling the MP-BGP Events Log Display

To disable the display of information about MP-BGP logs that was previously enabled with the **debug ip mbgp** command

- Issue the **undebug ip mbgp** command:

  host1#**undebug ip mbgp**

Related Documentation

- Enabling the MP-BGP Events Log Display on page 505
- *debug ip mbgp*

- *undebug ip mbgp*

PART 3

Layer 2 Services Over MPLS

CHAPTER 8

Layer 2 Services over MPLS Overview

This chapter contains the following sections:

Layer 2 Services over MPLS Overview

Many Internet service providers offer multiple services such as Frame Relay, Asynchronous Transfer Mode (ATM), Ethernet, High-Speed Data Link Control (HDLC), and IP to their customers, but are consolidating to a single, packet-based, optical network from several service-specific legacy layer 2 networks. Although legacy layer 2 network links are disappearing from the Internet service provider's network, the legacy layer 2 network links and services to customers must remain.

Layer 2 services over IP/MPLS enable service providers to emulate the legacy layer 2 network links and services over their IP/MPLS-based network. Layer 2 services over MPLS are especially desirable because MPLS provides features such as traffic engineering and fast reroute. These MPLS features can be used to emulate certain layer 2 service characteristics. From the perspective of the customer edge (CE) devices, all that exists is the layer 2 circuit, even though the circuit actually exists over the service provider's MPLS network.

The JunosE Software currently support the following layer 2 services over MPLS:

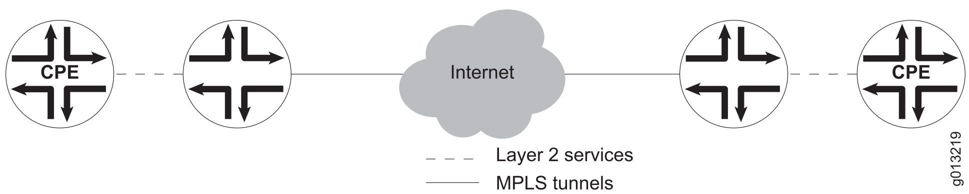
- ATM with ATM Adaptation Layer 5 (AAL5) encapsulation
- ATM with virtual channel connection (VCC) cell relay encapsulation
- Ethernet (Fast Ethernet, Gigabit Ethernet, 10-Gigabit Ethernet, bridged Ethernet, bridged Ethernet/VLAN, Ethernet/VLAN)

NOTE: For the purposes of configuring layer 2 services over MPLS, Ethernet interfaces and bridged Ethernet interfaces function identically, as do Ethernet/VLAN interfaces and bridged Ethernet/VLAN interfaces. For simplicity, the term *Ethernet* in this chapter refers to both Ethernet interfaces and bridged Ethernet interfaces, and the term *Ethernet/VLAN interfaces* refers to both Ethernet/VLAN interfaces and bridged Ethernet/VLAN interfaces.

- Frame Relay
- HDLC

Figure 116 on page 528 illustrates layer 2 services being supported over an Internet service provider's MPLS network. Customers using Frame Relay, ATM, or other legacy layer 2 connections to E Series routers are unaware that MPLS tunneling is used.

Figure 116: Layer 2 Services over a Provider's MPLS Network



Related Documentation

- Layer 2 Services over MPLS Implementation on page 530

Layer 2 Services over MPLS Platform Considerations

To configure layer 2 services over MPLS, you must first configure the underlying layer 2 service (ATM, bridged Ethernet, Fast Ethernet, Gigabit Ethernet, 10-Gigabit Ethernet, Frame Relay, or HDLC) and MPLS.

Module Requirements

For information about the modules that support the underlying layer 2 service and MPLS on ERX14xx models, ERX7xx models, and the ERX310 Broadband Services Router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support the underlying layer 2 service and MPLS.

For information about the modules that support the underlying layer 2 service and MPLS on the E120 or E320 Broadband Services Router:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support the underlying layer 2 service and MPLS.

NOTE: **In the current release, the E120 and E320 routers supports all layer 2 services over MPLS shown in the bulleted list in "Layer 2 Services over MPLS Overview" on page 527 *except* ATM with AAL5 encapsulation, ATM with VCC cell relay encapsulation, Frame Relay, and HDLC.**

Interface Specifiers

Many of the configuration task examples in this chapter use the *slot/port*[.*subinterface*] format to specify the physical interface for the underlying layer 2 service. However, the interface specifier format that you use depends on the router that you are using.

For ERX7xx models, ERX14xx models, and ERX310 routers, use the *slot/port*[.*subinterface*] format. For example, the following command specifies ATM 1483 subinterface 10 on slot 0, port 1 of an ERX7xx model, ERX14xx model, or ERX310 router.

```
host1(config)#interface atm 0/1.10
```

For the E120 router and the E320 router, use the *slot/adapter/port*[.*subinterface*] format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, adapter 0 identifies the right IOA bay (E120 router) and the upper IOA bay (E320 router); adapter 1 identifies the left IOA bay (E120 router) and the lower IOA bay (E320 router). For example, the following command specifies ATM 1483 subinterface 20 on slot 5, adapter 0, port 0 of an E320 router.

```
host1(config)#interface atm 5/0/0.20
```

Related Documentation

- *Interface Types and Specifiers*

Layer 2 Services over MPLS References

For information about layer 2 services, consult the following resources:

- Encapsulation Methods for Transport of ATM Over MPLS Networks—draft-ietf-pwe3-atm-encap-07.txt (April 2005 expiration)
- Encapsulation Methods for Transport of Ethernet Frames Over IP/MPLS Networks—draft-ietf-pwe3-ethernet-encap-05.txt (June 2004 expiration)
- Encapsulation Methods for Transport of Layer 2 Frames Over IP and MPLS Networks—draft-martini-l2circuit-encap-mpls-08.txt (March 2005 expiration)
- Encapsulation Methods for Transport of PPP/HDLC Over IP and MPLS Networks—draft-ietf-pwe3-hdlc-ppp-encap-mpls-03.txt (October 2004 expiration)

- Framework for Pseudo Wire Emulation Edge-to-Edge (PWE3)—draft-ietf-pwe3-arch-06.txt (April 2004 expiration)
- IEEE 802.3ad (Link Aggregation)
- Pseudowire Setup and Maintenance Using LDP—draft-ietf-pwe3-control-protocol-08.txt (January 2005 expiration)
- Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)—draft-ietf-pwe3-requirements-08.txt (June 2004 expiration)
- Transport of Layer 2 Frames Over MPLS—draft-martini-12circuit-trans-mpls-11.txt (October 2003 expiration)

NOTE: **IETF drafts are valid for only 6 months from the date of issuance. They must be considered as works in progress. Please refer to the IETF Website at http://www.ietf.org for the latest drafts.**

For information about configuring supported layer 2 interfaces, consult the following resources:

- *JunosE Physical Layer Configuration Guide*
- See *Configuring ATM* in *JunosE Link Layer Configuration Guide*.
- See *Configuring Frame Relay* in *JunosE Link Layer Configuration Guide*.
- See *Configuring Packet over SONET* in *JunosE Link Layer Configuration Guide*.
- See *Configuring Bridged Ethernet* in *JunosE Link Layer Configuration Guide*.

For information about configuring supported serial interfaces, which are also referred to as HDLC channels, see the following resources:

- See *Configuring Channelized T3 Interfaces* in *JunosE Physical Layer Configuration Guide*.
- See *Configuring T3 and E3 Interfaces* in *JunosE Physical Layer Configuration Guide*.
- See *Configuring Channelized OCx/STMx Interfaces* in *JunosE Physical Layer Configuration Guide*.

For information about configuring MPLS, see "Configuring MPLS" on page 279.

Related Documentation

- Layer 2 Services over MPLS Platform Considerations on page 528
- Layer 2 Services over MPLS Overview on page 527
- Layer 2 Services over MPLS Implementation on page 530

Layer 2 Services over MPLS Implementation

When layer 2 services are configured over MPLS, layer 2 traffic is encapsulated in MPLS frames and sent over MPLS tunnels. A virtual circuit (VC) label that indicates a specific layer 2 connection, such as a Frame Relay data-link connection identifier (DLCI), is pushed into the label stack between the tunnel label and the layer 2 data.

A service-specific control word may be placed between the layer 2 data and the VC label. The control word is used for frame sequencing and carrying service-specific information, such as Frame Relay forward explicit congestion notification (FECN) and backward explicit congestion notification (BECN) information. At the tunnel end, the VC label is used to find the layer 2 interface over which the traffic is sent. The control word, if present, is used to convert the encapsulated layer 2 traffic into its native format.

Because MPLS labels are unidirectional, two VC labels are required for each layer 2 connection. The VC labels are distributed by the Label Distribution Protocol (LDP) in downstream-unsolicited (DU) mode between the two routers. The layer 2 connection status signaling may be emulated by advertising and withdrawing the VC labels. For example, if the Frame Relay subinterface between customer premises equipment (CPE) and a provider edge (PE) router goes down, the corresponding VC label is withdrawn by the PE router. When the remote PE router at the other end receives the label withdrawal, it translates the label withdrawal into LMI notifications to its CPE. When the Frame Relay subinterface comes back, a VC label is advertised, and the remote PE router again translates it into LMI notifications.

Related Documentation

- Layer 2 Services over MPLS Overview on page 527
- Configuring Frame Relay Layer 2 Services on page 550
- Configuring Interoperation with Legacy Frame Relay Layer 2 Services on page 551
- Configuring Ethernet/VLAN Layer 2 Services on page 551
- Configuring S-VLAN Tunnels for Layer 2 Services on page 552

Local Cross-Connects Between Layer 2 Interfaces Using MPLS Overview

You can configure layer 2 services over MPLS to transmit data between two layer 2 interfaces that reside on the same E Series router. In this configuration, which is referred to as a local cross-connect, traffic that arrives at the router’s ingress interface is switched out the egress interface, instead of going through an MPLS core network.

A local cross-connect enables the router to function as a layer 2 switch. It operates with any supported layer 2 service. To configure local cross-connects, you must use the **mpls-relay** command.

Related Documentation

- Layer 2 Services over MPLS Overview on page 527.
- Configuring Local Cross-Connects Between Ethernet/VLAN Interfaces on page 553.

MPLS Shim Interfaces for Layer 2 Services over MPLS Overview

An MPLS shim interface is stacked on a layer 2 interface to do either of the following:

- Create a layer 2 circuit by cross-connecting the layer 2 interface to an LSP corresponding to the VC label.

- Create a local cross-connect by cross-connecting the layer 2 interface to another layer 2 interface.

You can create or remove MPLS shim interfaces with the **mpls-relay** or **route interface** commands. Shim interfaces are also created in or removed implicitly from a load-balancing group by the **member interface** command when you configure the group. Each MPLS shim interface exists in a particular virtual router.

Each MPLS shim interface points to a single MPLS next hop. When layer 2 frames arrive on the layer 2 interface below the MPLS shim interface, they are encapsulated in an MPLS packet and forwarded to that MPLS next hop. The details of the encapsulation are determined by the attributes of the shim interface.

The MPLS next hop to which the shim interface points is set by an MPLS signaling protocol, which adds a special entry with implicit null in label to the interface label-space MPLS forwarding table of the shim interface. For traffic arriving from the core, the MPLS signaling protocol adds a normal entry with a real in label to the platform label-space MPLS forwarding table whose next hop points to the MPLS shim interface.

You can configure the following attributes for each MPLS shim interface:

- The administrative state, enabled or disabled, configured with the **mpls-relay disable** command.
- The IP address of the remote PE router for the layer 2 circuit, configured with the **mpls-relay** command. The shim interface is cross-connected to an LSP corresponding to the VC label received from the specified remote PE router, or to another shim interface in the local cross-connects case. For local cross-connects the IP address is local to the PE router.
- The name of an RSVP-TE base tunnel to be used for the layer 2 circuit, configured with the **route interface** command.
- The group ID of the shim interface, configured using the **group-id** option of the **mpls-relay** and **route interface** commands. Even though you can configure the group ID, the JunosE Software does not currently use it.
- Whether the control word is used, configured with the **control-word** and **no-control-word** options of the **mpls-relay** and **route interface** commands. The layer 2 interface determines the default preference if this option is not configured. Some layer 2 interfaces require a control word; others do not support it.
- Whether sending nonzero sequence numbers in the control word is preferred, configured with the **sequencing** and **no-sequencing** options of the **mpls-relay** and **route interface** commands. The layer 2 interface determines the default preference if this option is not configured. Even when preferred, the sequence numbers might not be sent if the control word is not used due to configuration. You can only configure whether the numbers are sent. MPLS always accepts zero sequence numbers and checks the order

of nonzero sequence numbers in received MPLS packets that are forwarded to an MPLS shim interface. Out-of-order packets are discarded.

- The number of the load-balancing group of which the shim interface is a member. This attribute is set to the current load-balancing group when the shim interface is implicitly created with the **member interface** command.

You can enable statistics collection for the MPLS shim interfaces.

Related Documentation

- Understanding CE Load Balancing for Martini Layer 2 Transport on page 560
- Monitoring Layer 2 Services over MPLS on page 585

Multiple Layer 2 Services over MPLS Overview

When you configure an MPLS shim interface over an ATM, Frame Relay, or HDLC layer 2 interface, no other interface (for example, PPP or IP) can be stacked above the layer 2 interface.

By contrast, when you configure an MPLS shim interface over any Ethernet or Ethernet/VLAN interface, both the MPLS shim interface and other interfaces (such as IP, PPP, or PPPoE) can be stacked above the layer 2 interface.

When you configure both an MPLS shim interface and another interface over a layer 2 interface, traffic for a protocol explicitly configured in the interface stack is forwarded to that protocol layer for further processing. Traffic for any nonconfigured protocols is forwarded to the MPLS shim interface on the other side of the connection.

When the MPLS shim interface is the only layer stacked above the layer 2 interface, as is the case with ATM, Frame Relay, and HDLC, then all traffic is forwarded to the MPLS shim interface and across the MPLS tunnel.

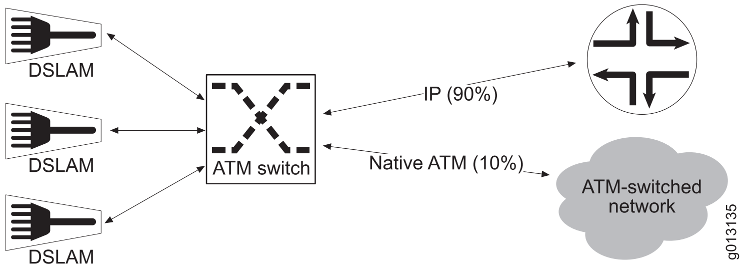
Related Documentation

- Example: Configuring Many Shim Interfaces with the Same Peer, VC Type, and VC ID on page 560
- Configuring Load-Balancing Groups on page 564
- Load-Balancing Group Configuration on page 562
- MPLS Interfaces and Labels on page 564

ATM Layer 2 Services over MPLS Overview

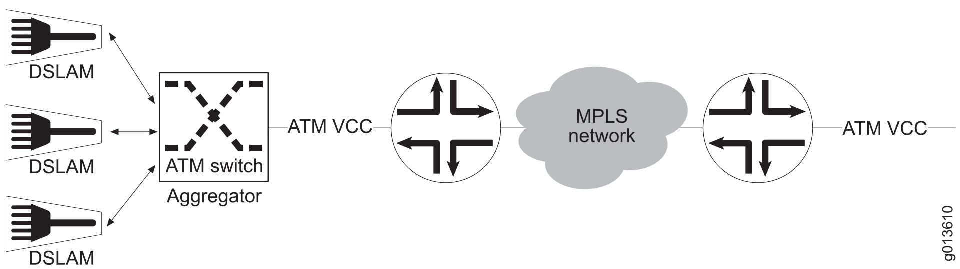
ATM layer 2 services over MPLS provide ATM switch-like functionality for E Series routers. This feature is useful for customers who run IP in the majority of their network but still have to carry a small amount of non-IP traffic, as in the example shown in Figure 117 on page 534.

Figure 117: Common ISP Network



In this scenario, it is not economical to have special ATM switches in front of E Series routers just to accommodate the small percentage of non-IP traffic. The ATM layer 2 services over MPLS feature lets you replace the ATM traffic selector switch with an E Series router, as shown in Figure 118 on page 534. The two routers pass traffic between two interfaces through an MPLS tunnel using Martini encapsulation, regardless of the contents of the packets.

Figure 118: E Series Router Replacing Remote ATM Switch



ATM layer 2 services over MPLS supports two encapsulation methods on E Series routers:

- AAL5 relay encapsulation
- VCC cell relay encapsulation

The following sections describe each of these encapsulation methods.

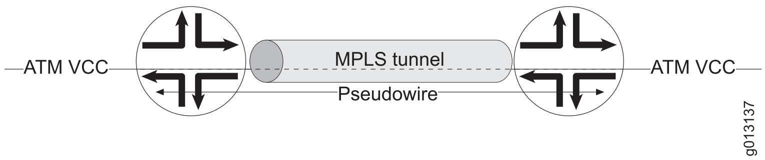
AAL5 Encapsulation

JunosE Software supports the AAL5 relay method of encapsulation that is specified in the Martini draft. This method is also referred to as AAL5 service data unit (SDU) encapsulation.

ATM Martini encapsulation emulates ATM switch behavior by creating a pseudowire between pairs of ATM virtual circuits. When the router receives AAL5 packets on one of those circuits, it reassembles them, encapsulates them using Martini encapsulation, and forwards them to an MPLS tunnel. At the end of the tunnel, the packet is de-encapsulated, segmented back, and sent to a selected ATM VC.

In Figure 119 on page 535, an MPLS tunnel connects two E Series routers, and ATM cross-connects provide a pseudowire between the ATM VCs on the two routers. All AAL5 packets on the pseudowire are encapsulated. The egress VC does not need the same ATM address as the ingress circuit.

Figure 119: AAL5 Pseudowire and MPLS Tunnel



To use AAL5 SDU encapsulation, you must use the **aal5all** encapsulation keyword when you configure ATM subinterfaces.

OAM Cells

The E Series router performs a similar operation for Operation, Administration, and Maintenance (OAM) cells, except that they do not need reassembly.

The router passes the following OAM cells transparently:

- F5 alarm indication signal (AIS) segment and end-to-end
- F5 remote defect indication (RDI) segment and end-to-end
- F5 loopback segment and end-to-end
- Resource management
- F5 continuity check segment and end-to-end

In addition, F4 OAM cell forwarding is supported.

JunosE Software does not allow for setting a segment endpoint on an ATM cross-connect interface. Segment OAM cells are forwarded to the egress interface in the same manner as end-to-end cells.

QoS Classification

Packets are subject to normal quality of service (QoS) classification according to service categories assigned to ATM virtual circuits that make up the connection.

Limitations

The JunosE implementation of the Martini draft has the following limitations:

- Only AAL5 packets and OAM cells are forwarded.
- There is no equivalent of VP switching.
- Point-to-multipoint connections are not supported.
- Automatic connection setup using user-to-network interface (UNI) signaling and private network-to-network interface (PNNI) is not supported.
- The ATM MIB cross-connected table is not supported.
- Connections between ATM circuits and non-ATM interfaces are not supported.

Control Word Support

Martini draft encapsulation includes a control word with the following fields:

- T bit for transport type
- E bit for explicit forward congestion indication (EFCI)
- C bit for cell loss priority (CLP)=1 indication
- U bit for command/response indication based on AAL5 common part convergence sublayer user-to-user indication (CPCS-UU)
- Optional sequence number

The current JunosE implementation supports the T bit and optional sequence number fields.

VCC Cell Relay Encapsulation

E Series routers support virtual channel connection (VCC) cell relay encapsulation for ATM layer 2 services over MPLS. VCC cell relay encapsulation enables a router to emulate ATM switch behavior by forwarding individual ATM cells over an MPLS pseudowire (also referred to as an MPLS tunnel) created between two ATM VCCs, or as part of a local ATM cross connect between two ATM 1483 subinterfaces on the same router. The JunosE implementation conforms to the required N-to-1 cell mode encapsulation method described in the Martini draft, with the provision that only a single ATM virtual circuit (VC) can be mapped to an MPLS pseudowire.

VCC cell relay encapsulation is useful for voice-over-ATM applications that use ATM Adaptation Layer 2 (AAL2)–encapsulated voice transmission.

AAL0 Raw Cell Mode

VCC cell relay encapsulation supports ATM Adaptation Layer 0 (AAL0) encapsulation, also referred to as raw cell mode or null encapsulation. AAL0 is often used to carry signaling ATM cells, which the router treats as raw cells.

When you create an ATM PVC as part of a VCC cell relay configuration, you must use the **aal0** encapsulation.

Cell Concatenation Parameters

VCC cell relay encapsulation supports the concatenation (aggregation) of multiple ATM cells in a single encapsulated packet that is transmitted on the MPLS pseudowire.

You can use the **atm cell-packing** and **atm mcpt-timers** commands to configure the following parameters that control how the router performs cell concatenation:

- Maximum number of ATM cells that the router can concatenate in a single packet.
- Values (in microseconds) for each of the three ATM Martini cell packing timers maintained on the router. These timers define the time threshold that the router uses to concatenate ATM cells and transmit the cells in an MPLS packet on the pseudowire.
- Numeric identifier (1, 2, or 3) that indicates which of the three ATM Martini cell packing timers you want to use to detect timeout of the cell collection threshold.

Based on this configuration, the router attempts to concatenate the specified maximum number of ATM cells into an MPLS packet within the time interval allowed by the ATM

Martini cell packing timer you selected. When the timer detects that the allotted time interval has expired, the router forwards the MPLS packet even if it contains fewer than the specified maximum number of aggregated cells per packet.

Cell Concatenation and Latency

Cell concatenation increases network latency, which is undesirable for voice-over-ATM applications. To minimize these effects, use care in choosing values for the ATM Martini cell packing timers.

We recommend that for voice-over-ATM configurations, you select timeout values between 6 microseconds and 3 x 6 microseconds. Values within this range are generally low enough to maintain a reasonable cell delay and high enough to take advantage of the cell concatenation mechanism.

If traffic shaping is enabled on the egress router, the JunosE implementation of VCC cell relay encapsulation preserves the spacing between cells.

Control Word Support

VCC cell relay encapsulation requires use of a control word. The control word contains the T, E, C, and U bits, as well as an optional sequence number field.

The JunosE implementation of VCC cell relay encapsulation supports the T bit, which is always set to indicate raw ATM cells, and the optional sequence number. The E, C, and U bits have no meaning for VCC cell relay configurations because the router forwards a complete ATM cell with as much header information as possible.

Unsupported Features

VCC cell relay encapsulation on JunosE routers does not support the following features:

- Mapping multiple ATM VCs to a single MPLS pseudowire
- Optional Martini one-to-one cell encapsulation method with cell headers removed

For information about AAL5 SDU encapsulation, see Encapsulation Methods for Transport of ATM Over MPLS Networks—draft-ietf-pwe3-atm-encap-07.txt (April 2005 expiration). For information about, VCC cell relay encapsulation, see Encapsulation Methods for Transport of ATM Over MPLS Networks—draft-ietf-pwe3-atm-encap-07.txt (April 2005 expiration).

Related Documentation

- Configuring Local ATM Cross-Connects with AAL5 Encapsulation on page 554
- Configuring an MPLS Pseudowire with VCC Cell Relay Encapsulation on page 556

HDLC Layer 2 Services over MPLS Overview

E Series routers support the creation of HDLC layer 2 circuits across an MPLS network. An HDLC layer 2 circuit can carry any standard HDLC traffic (including PPP) or Cisco HDLC traffic between two CE devices across an MPLS network. In an HDLC layer 2 circuit configuration, an E Series router functions as one of the PE routers.

You can configure an HDLC layer 2 circuit between two serial interfaces, between two packet over SONET (POS) interfaces, or between a serial interface and a POS interface. The interfaces at either end of the circuit can operate at the same speed or at different speeds. For example, you can configure an HDLC layer 2 circuit between a serial interface on a T1 circuit and a POS interface on an OC3 circuit.

Interface Stacking

When you configure an MPLS shim interface above an HDLC layer 2 interface, which is in turn stacked above a serial or POS interface, no other interfaces (for example, PPP) can be stacked above the HDLC interface. In other words, the HDLC interface can have only one upper interface.

In practice, this means that you cannot issue the **mpls-relay** command (to create the HDLC layer 2 circuit) and then issue an **encapsulation** command (such as **encapsulation ppp**) for the same HDLC interface. If you attempt to do so, the router prevents the configuration and displays an error message.

This behavior contrasts with that of bridged Ethernet and Ethernet interfaces (with and without VLANs), which allow configuration of both an MPLS shim interface and another interface (such as IP, PPP, or PPPoE) above the layer 2 interface.

Encapsulation

The JunosE implementation of HDLC layer 2 circuits supports encapsulation of either HDLC frames or PPP frames within MPLS frames. By default, the router uses VC-type HDLC signaling and HDLC encapsulation to encapsulate HDLC frames in MPLS.

However, if you want the router to encapsulate PPP frames directly in MPLS without the HDLC header, you can include the optional **relay-format ppp** keywords in the **mpls-relay** or **route interface** command to cause the router to use VC-type PPP signaling and PPP encapsulation instead of the default VC-type HDLC signaling and HDLC encapsulation. This option, which is available only for serial and POS interfaces, is useful if the traffic carried on the serial or POS interface contains actual PPP packets and not, for example, Cisco HDLC packets.

Control Word Support

The router always sends a control word for HDLC layer 2 circuits, regardless of whether or not sequencing is enabled.

Local Cross-Connects

You can configure an HDLC layer 2 circuit in a local cross-connect configuration between serial or POS interfaces within the same router. In this configuration, the pairs of HDLC interfaces are directly cross-connected to each other. The cross-connected interfaces can be different types and operate at different speeds; for example, you can cross-connect a serial interface on a T1 circuit and a POS interface on an OC3 circuit.

Related Documentation

- Configuring HDLC Layer 2 Services on page 558

CE-Side MPLS L2VPNs over LAG Overview

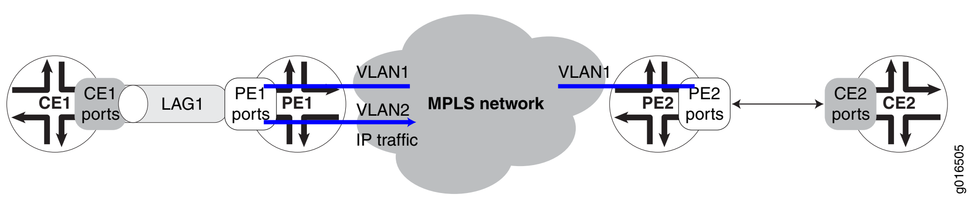
MPLS L2VPNs over link aggregation groups (LAGs) uses the functionality of both layer 2 services over MPLS and LAG. MPLS L2VPNs (Martini circuits) over LAG enable MPLS Martini circuits to use LAG in the network between the customer edge (CE) devices and the provider edge (PE) routers to distribute traffic arriving from pseudowires across multiple physical Ethernet interfaces. The criteria for distribution of packets that are transmitted from the pseudowires across the LAG bundle are determined by the hashing of VLAN and the source and destination MAC addresses. If a packet contains a VLAN stack, all the VLAN identifiers of the received packet, the source MAC address, and the destination MAC address are used in the hashing algorithm. If the packet does not contain a VLAN stack, only the source and destination MAC addresses are used in the hashing algorithm.

Using MPLS L2VPNs over the LAGs configured between CE and PE devices provides redundancy and increased bandwidth on demand. The capability to use MPLS L2VPNs over LAG is available on GE-HDE, GE-2, and ES2 4G LM.

Similarly, you can enable distribution of IP traffic from pseudowires across multiple physical interfaces on the PE routers facing the CE-side devices. In this case, the source and destination IP addresses are used in the hashing rule to determine the distribution criteria for received packets. You must use a different VLAN for IP packets from the one used for MPLS L2VPN packets. However, you can use the same LAG bundle that you configured for MPLS L2VPN traffic for IP traffic distribution.

Figure 120 on page 539 shows an example of an MPLS L2VPN or Martini tunnel over LAG on the CE-side device. A set of Ethernet ports on a provider edge device, PE1, are configured as member ports of LAG1. An MPLS L2VPN tunnel from PE1 to another provider edge device, PE2, is configured over an interface, VLAN1, over LAG1. An MPLS shim interface is stacked on the VLAN1 interface, when the MPLS L2VPN tunnel over VLAN1 over LAG1 is configured. The Martini tunnel from PE1 to PE2 can be configured either over Ethernet or LAG. In this scenario, it is considered to be configured over LAG.

Figure 120: CE-Side MPLS L2VPN Tunnel over LAG



Layer 2 frames arrive from CE1 to PE1 on the VLAN1 interface that resides below the MPLS shim interface. These frames are encapsulated in an MPLS packet and forwarded to the MPLS next hop. The attributes of the shim interface are used to determine the encapsulation parameters. These MPLS packets are later forwarded to the remote PE2 device. PE2 processes all the MPLS labels and determines that layer 2 processing is required on the remaining layer 2 frames. After PE2 processes the Ethernet layer 2 frames, they are forwarded to CE2.

Traffic arriving at PE1 from pseudowires is distributed across all the member links of the LAG. The hashing algorithm, based on the VLAN identifier and source and destination MAC addresses, is used to determine the physical link to which the packets must be forwarded.

If you configure the Martini tunnel directly over LAG in the same topology as shown in Figure 1, packet processing at PE1 remains the same as the case described. The only difference is in processing the hash algorithm, which uses only the source and destination MAC addresses, and not the VLAN identifier, to determine the physical link for transmission of MPLS packets transmitted from the pseudowires.

If IP traffic is being transmitted from CE1, which is not designated for the MPLS L2VPN tunnel, the packets arrive at an interface that is different from the one used for MPLS traffic, VLAN2. This behavior occurs because the VLAN interface used for transmission of IP traffic must be different from that used for MPLS traffic.

The MPLS shim interface can be configured directly over LAG or stacked on a VLAN interface over LAG. For more information on the guidelines to be followed when you configure the MPLS shim interface to enable MPLS Martini circuits to use LAG on the CE-side devices, see "Multiple Layer 2 Services over MPLS Overview" on page 533.

Related Documentation

- Example: Configuring MPLS L2VPN Tunnel over LAG on page 572
- Example: Configuring MPLS L2VPN Tunnel over VLAN over LAG on page 568

Ethernet Raw Mode Encapsulation for Martini Layer 2 Transport Overview

An Ethernet pseudowire carries Ethernet/802.3 traffic over an MPLS network. JunosE Software enables Ethernet traffic to be transmitted over Martini circuits (cross-connect virtual circuits) on ES2 4G line modules (LMs), GE-2 LMs, GE/FE LMs, ES2 10G LMs, ES2 10G Uplink LMs, and ES2 10G ADV LMs. The transfer of Ethernet packets over MPLS-based pseudowire enabled service providers to provide point-to-point layer 2 Ethernet connectivity between geographically remote customer edge (CE) devices. This functionality enabled the following tasks to be performed on received packets:

- Removal of all V-LAN tags from layer 2 frames received from CE devices before transmitting that frame to the pseudowire
- Replacement of all V-LAN tags in layer 2 frames received from the pseudowire with the configured VLAN tags on the local interface before sending the frame to the CE device
- Insertion of the inner V-LAN tag while sending layer 2 frames received from the pseudowire to the CE device on a double-tagged interface

However, this mode of processing did not allow for:

- Identification of a particular VLAN tag in single-tagged or double-tagged packets as the service-delimiting tag
- Selective removal of only the service-delimiting tag when layer 2 frames were received from the CE devices to be sent to the pseudowire

- Selective insertion of only the service-delimiting tag when layer 2 frames were received from the pseudowire to be sent to the CE device

You can configure an S-VLAN subinterface to enable the provider edge (PE) device to strip the S-VLAN tag from all packets that enter the MPLS pseudowires or Martini circuits. This functionality is also referred to as operation of Martini circuits or MPLS shim interfaces in *raw mode*. An Ethernet pseudowire can operate in either tagged mode or raw mode.

- In tagged mode, each frame must contain at least one 802.1Q VLAN tag, and the tag value is processed based on a predefined rule or algorithm at the two pseudowire endpoints. This mode is not supported on E Series routers.
- In raw mode, if a frame contains an 802.1Q VLAN tag and the tag is not suitable to be processed at the two pseudowire endpoints, it passes transparently through them. The raw mode operation comprises the following tasks:
  - Identifying the outermost VLAN tag of the layer 2 frame received from the attachment circuit as a service-delimiting tag, stripping this tag, and then inserting the resulting frame into the pseudowire connection
  - Inserting the specified service-delimiting tag on the layer 2 frame received from the pseudowire connection and then transmitting the resulting frame to the attachment circuit

You can enable the raw mode configuration only for MPLS shim interfaces stacked on S-VLAN interfaces. Unified ISSU and high availability are supported when the router is configured in either of these ways:

- To strip the S-VLAN tags from all packets that enter Martini circuits
- When raw mode encapsulation is configured on the VLAN interface stacked below the MPLS shim interface

NOTE: You cannot enable the raw mode configuration for S-VLAN subinterfaces on the ES2 10G, ES2 10G ADV, and ES2 10G UPLINK LMs.

When a pseudowire operates in raw mode, service-delimiting tags, if present in the frame received from the PE device, are stripped from the frame before being sent to the next processing point in the circuit. The Ethernet frame is encapsulated according to the algorithm defined for raw mode. If the local PE device detects a failure on the Ethernet input port, or if the port is disabled, the PE device sends an appropriate pseudowire status notification message to the remote PE device. In raw mode, all Ethernet frames received on the local PE device are transmitted to the remote PE device on a single pseudowire. The raw-mode attribute of the Martini circuit is sent to the forwarding controller (FC) on the supported line modules and the Label Distribution Protocol (LDP) is notified of the correct pseudowire type to be used in the signaling messages.

Related Documentation

- Examples: Ethernet Raw Mode Encapsulation for Martini Layer 2 Transport on page 575
- *mpls-relay*

- *route interface*

S-VLAN Subinterface with an Untagged C-VLAN ID Overview

When you configure an S-VLAN subinterface with a C-VLAN ID in the range 0–4095 or with the **any** (5001) keyword for the VLAN ID, the single- tagged frames with a valid S-VLAN ID or untagged frames were ignored for the subinterface. You can also configure an S-VLAN subinterface with a C-VLAN ID to zero by using the **anyUntagged** (5003) keyword, when the C-VLAN is configured inside S-VLAN as part of a pseudowire. As a result, both customer-tagged and untagged frames can be transported inside a prescribed S-VLAN-tag over a single pseudowire.

NOTE: On ES2 10G, ES2 10G ADV, and ES2 10G UPLINK LMs, untagged frames are not transported inside a prescribed S-VLAN tag even though the C-VLAN ID is set to zero.

Observe the following guidelines when you configure an S-VLAN subinterface with any untagged VLAN, for which the C-VLAN ID 0 is reserved.

- Only Ethertype 0x8100 is supported on an S-VLAN subinterface that is configured with the **anyUntagged** C-VLAN ID. If you attempt to configure Ethertype 0x9100 and 0x8a88 on this S-VLAN subinterface, an error message is displayed.
- The default Ethertype of an S-VLAN subinterface that is configured with the **anyUntagged** C-VLAN id is set to 0x8100. For other VLAN subinterfaces, it is set to 0x9100.
- You cannot configure an S-VLAN subinterface with an S-VLAN ID value and the C-VLAN ID as **anyUntagged** if its Ethertype is 0x8100 and if any VLAN subinterface on the same major VLAN is configured with a VLAN ID value. Similarly, you cannot configure a VLAN subinterface with a VLAN ID value if any S-VLAN subinterface on the same major VLAN is configured with the same S-VLAN ID value, C-VLAN ID as **anyUntagged**, and Ethertype as 0x8100. If you attempt these procedures, an error message is displayed.
- If you attempt to configure the Ethertype as 0x8100 on an S-VLAN subinterface, which is configured with a valid S-VLAN ID value and C-VLAN ID as **anyUntagged**, and if any VLAN subinterface is previously configured with a VLAN ID value on the same major VLAN, an error message is displayed.

Related Documentation

- Examples: Configuring S-VLAN Subinterface with an Untagged C-VLAN ID on page 579
- *svlan id*

Multiple ATM Virtual Circuits over a Single Pseudowire Overview

In JunosE releases earlier than Release 10.2.x, ATM Martini circuit functionality was supported on ERX14xx models, ERX7xx models, and the ERX310 router that enabled the ATM cells that pertained to a particular ATM virtual circuit (VC) to be transported over a single pseudowire. This behavior was achieved by emulating connectivity between two ATM ports for a single virtual circuit. However, if you wanted to emulate the connectivity between two ATM ports instead of between ATM VCs using a single VC over a single pseudowire, all the necessary VCs had to be configured separately. Also, in such cases, the corresponding pseudowires for each of the VCs had to be configured individually. This method was not efficient because of the amount of manual configuration and MPLS signaling protocol (LDP) state that had to be maintained.

Now, you can emulate physical connectivity between two ATM ports that are not directly connected. This emulation is made possible by transporting ATM cells belonging to a subset of matching ATM VCs on both the ATM ports over a single pseudowire. Multiple VCs over a single pseudowire is useful in scenarios when ATM switches are connected using a high-speed packet switched network, instead of expensive physical cables.

Support for configuration of multiple ATM VCs over a single pseudowire is based on RFC 4816, Pseudowire Emulation Edge-to-Edge (PWE3) Asynchronous Transfer Mode (ATM) Transparent Cell Transport Service. Although this RFC requires all the ATM cells, corresponding to all possible ATM virtual circuits, received on an ATM port to be transported on the associated single pseudowire, the current implementation does not enable all possible ATM virtual circuits on an ATM port to be transported. This condition occurs because of hardware limitations on the ATM line modules supported on ERX14xx models, ERX7xx models, and the ERX310 router. As a result, it is necessary to explicitly open virtual circuits on the segmentation and reassembly (SAR) scheduler to enable ATM cells corresponding to those virtual circuits to be received and transported over a pseudowire. Because of the scaling limitations on the number of virtual circuits that can be opened on the SAR scheduler, all possible ATM virtual circuits for a single ATM port cannot be opened on the SAR device. Therefore, to enable multiple VCs over a single pseudowire to be configured on the ERX routers, you must specify the subset of ATM virtual circuits on a port that must be carried on the single pseudowire.

To configure the subset of ATM virtual circuits, you must configure a VPI/VCI range using the new **mpls-relay atm vpi-range** *vpiStart vpiEnd* **vci-range** *vciStart vciEnd* command in global configuration mode. You can configure this VPI/VCI range only for ATM ports for which you have associated a pseudowire using the MPLS Martini circuit configuration. Before the support for multiple ATM VCs over a single pseudowire was available, MPLS Martini circuit configuration was allowed only on ATM subinterfaces and not on ATM ports. To enable a subset of ATM virtual circuits to be transported over a single pseudowire, you must add the MPLS Martini circuit configuration on an ATM port (associating the ATM port with the single pseudowire) and then specify the subset of ATM virtual circuits whose cells need to be transported on the single pseudowire using the VPI/VCI range configuration.

You can specify a maximum of four non-overlapping VPI/VCI ranges for each ATM port. The cumulative number of ATM virtual circuits in the specified VPI/VCI ranges must not exceed the scaling limitation of the SAR scheduler. The SAR scheduler limitation is not for each port, but for the entire line module. The VPI/VCI range configurations specified on the ATM ports on both ends of the pseudowire must match. Even if the VPI/VCI range configurations do not match on both ends of the pseudowire, Label Discovery Protocol (LDP) brings up the pseudowire. However, on the remote provider edge (PE) router, ATM cells received from the pseudowire that are not within the configured ranges are discarded.

For more information on the guidelines to be followed when you configure VCI/VPI ranges for transportation of a subset of ATM VCs on a single pseudowire, see "Guidelines for Configuring VPI/VCI Ranges of ATM Virtual Circuits" on page 545.

You can also specify concatenation of multiple ATM cells to be sent in a single MPLS-labeled packet for efficient usage of the backbone bandwidth. If you do not specify cell concatenation, each individual ATM cell is MPLS-labeled and transmitted on the pseudowire. You can use the **mpls-relay atm cell-packing mcpt-timers** command to configure the following parameters that control how the router performs cell concatenation:

- Maximum number of ATM cells that the router can concatenate in a single packet.
- Values (in microseconds) for each of the three ATM Martini cell packing timers maintained on the router. These timers define the time threshold that the router uses to concatenate ATM cells and transmit the cells in an MPLS packet on the pseudowire.

Based on this configuration, the router attempts to concatenate the specified maximum number of ATM cells into an MPLS packet within the time interval allowed by the ATM Martini cell packing timer you selected. When the router detects that the allotted time interval has expired, the router forwards the MPLS packet even if it contains fewer than the specified maximum number of aggregated cells per packet. The cell concatenation functionality is controlled by the timer values and the maximum number of cells to be concatenated. The LDP signaling protocol option to negotiate cell concatenation (maximum number of concatenated ATM cells) is not used.

For more information on the guidelines to be followed when you configure cell concatenation and cell packing timer identifiers for transportation of ATM VCs on a single pseudowire, see "Guidelines for Configuring Cell Concatenation and Cell Packing Timer for an ATM Port" on page 546.

When you add the MPLS Martini circuit configuration on an ATM port, you cannot add the interface label space RSVP configuration on the same ATM port. Therefore, you can configure an ATM port with either the interface label space RSVP configuration or the MPLS Martini circuit at the same time. You cannot configure both the interface label space RSVP configuration or the MPLS Martini circuit on the same ATM port at the same time.

The ATMx port is not changed to the Loss of Signal (LOS) state, which denotes the number of times for which the incoming optical signal is all zeros for at least 100 microseconds, when a failure is detected on the pseudowire. Possible causes might include a cable disconnection, excessive attenuation of the signal, or faulty equipment. The changeover to the LOS state for the ATM port is not performed because only a subset of the ATM virtual circuits are configured to be transported on the pseudowire. The ATM virtual circuits on the same ATM port that are not associated with the single pseudowire and are not present in the specified subset continue to function in the desired manner, without being affected by the failure detected on the pseudowire. If the ATM port was moved to an LOS state, all ATM virtual circuits on the same port that are configured for functionality other than the multiple VCs over single pseudowire functionality are also disrupted.

The F5 Operations Administration and Maintenance (OAM) cells and Integrated Local Management Interface (ILMI) cells are carried on the pseudowire because F5 OAM cells arrive with the same VPI/VCI values as the data cells. In such cases, you can configure the ILMI VPI/VCI as part of the range to enable the ILMI cells be carried on the pseudowire. However, F4 OAM cells are not carried over the pseudowire because the router does not enable opening of the VP-level OAM circuits to be transported transparently on the pseudowire. Because only a subset of the ATM virtual circuits on an ATM port are carried on the pseudowire, LDP uses the ATM n-to-one VCC cell transport (0x0009) pseudowire (PW) type instead of the ATM transparent cell transport (0x0003) PW type in the signaling messages.

Guidelines for Configuring VPI/VCI Ranges of ATM Virtual Circuits

Observe the following guidelines when you specify a single VPI/VCI range of ATM VCs whose cells need to be transported on the single pseudowire:

- Use to specify a single VPI/VCI range of ATM VCs whose cells need to be transported on the single pseudowire.
- You can configure the **mpls-relay atm vpi-range vci-range** command on the ATM port only after you associated a pseudowire with the port by using the **mpls-relay** or **route interface tunnel** command.
- You can use this command only on an ATM port (ATM AAL5 over ATM major interface). When you attempt to run this command on other interface types, such as ATM subinterfaces or Ethernet interfaces, this setting is not saved and an error message is displayed.
- When you run the **mpls-relay** or **route interface tunnel** command, no default VPI/VCI range is configured. You must specifically configure the VPI/VCI ranges.
- The VPI/VCI values that are not part of the specified range can be used for other existing applications. For example, you can configure an ATM subinterface on the same ATM port with a VPI/VCI value that is not included by the specified range specification and use that interface as an IP interface or subscriber interface.
- If you configured a VPI/VCI range on the shim interface and try to configure an ATM subinterface with the VPI/VCI value that is encompassed by the VPI/VCI range specified earlier on the shim interface, an error message is displayed stating the VPI/VICI range that you are trying to configure is already reserved and the configuration fails.
- If some of the VPI/VCI values specified in the range are used for other applications, the specified range is marked inactive and this setting is saved on the router, without being used to transport traffic. You must remove the conflicting range, and remove or reconfigure this range to activate it for transportation of ATM cells.
- If the specified VPI/VCI range includes the F4 OAM VCI values, the range is saved. The F4 OAM VCs are not opened, but the rest of the VCs are opened if they are not configured for other applications. If some of the VCs other than the F4 OAM VCs are opened for other applications, the range is marked inactive. You must specify ranges that do not overlap with the F4 OAM VCI values.

- You can specify up to four non-overlapping VPI/VCI ranges on the ATM port that has been associated with the single pseudowire using the **mpls-relay** or **router interface tunnel** command.
- When a VPI/VCI range that you enter overlaps with already specified ranges, the newly specified range becomes effective.
- When a VPI/VCI range that you enter is a subset of an already defined range, the specified range is not saved and an appropriate message is displayed on the CLI interface.
- When a VPI/VCI range that you enter encompasses one or more of the previously defined ranges, the configuration attempt fails and an appropriate message is displayed on the CLI interface.

Guidelines for Configuring Cell Concatenation and Cell Packing Timer for an ATM Port

Observe the following guidelines when you configure the maximum number of ATM cells that the router can concatenate in a single packet and the identifier of the ATM Martini cell packing timer that you want to use to detect timeout of the cell collection threshold:

- You can configure the **mpls-relay atm cell-packing mcpt-timer** command on the ATM port only after you associated a pseudowire with the port by using the **mpls-relay** or **route interface tunnel** command.
- You can use this command only on an ATM port (ATM AAL5 over ATM major interface). When you attempt to run this command on other interface types, such as ATM subinterfaces or Ethernet interfaces, this setting is not saved and an error message is displayed.

Performance Impact and Scalability Considerations

Because the support for multiple ATM VCs over a single pseudowire requires one pseudowire per ATM port and the number of ATM ports in a fully populated ERX chassis is in the order of a few tens of ports, the number of pseudowires required is also of the same range. As a result, no performance impact is caused by LDP signaling and state management. The amount of memory needed and initial CPU activity on the line module for a specified range are proportional to the number of VCs in the range.

You can scale the number of virtual circuits configured on an ATM line module up to 16,000. The VPI/VCI range specification on the ATM ports for this feature is controlled by this limit. Depending on other VPI/VCI configuration on the ATM line module, the range specification must not be greater than this scaled limit subtracted from the other VPI/VCI configuration.

A VPI/VCI range with the maximum number of VCs does not cause the line module to become unstable. Support for unified ISSU and high availability with a VPI/VCI range configured with the maximum number of VCs is provided.

Related Documentation

- Example: Multiple ATM Virtual Circuits over a Single Pseudowire on page 581
- *mpls-relay atm vpi-range vci-range*

- *mpls-relay atm cell-packing mcpt-timer*

CHAPTER 9

Configuring Layer 2 Services over MPLS

This chapter describes how to configure layer 2 services over MPLS on the router, and contains the following sections:

Before You Configure Layer 2 Services over MPLS

Before you configure layer 2 services over Multiprotocol Label Switching (MPLS), we recommend you be thoroughly familiar with MPLS and the type of layer 2 interfaces that you want to configure.

Before you configure layer 2 services over MPLS, you must configure the layer 2 interfaces and MPLS.

NOTE: To provide uninterrupted service during an SRP switchover in a scaled configuration, such as one with 32,000 Martini circuits, set the LDP graceful restart reconnect time to the maximum 300 seconds and set the LDP graceful restart recovery timer to the maximum 600 seconds. This requirement is true for all SRP switchovers, including those in the context of a unified in-service software upgrade.

This chapter describes how to configure different types of layer 2 services over MPLS. Each procedure uses either the **mpls-relay** command or the **route-interface** command to configure MPLS tunneling.

Related Documentation

- Layer 2 Services over MPLS Overview on page 527
- Layer 2 Services over MPLS References on page 529

Configuring Frame Relay Layer 2 Services

To configure Frame Relay layer 2 services over MPLS with the RFC-4619 Frame Relay pseudowire type:

1. Configure the Frame Relay interface.

```
host1(config)#interface serial 4/1:1/1
host1(config-if)#encapsulation frame-relay ietf
host1(config-if)#frame-relay intf-type dte
host1(config-if)#frame-relay lmi-type ansi
host1(config-if)#interface serial 4/1:1/1.1
host1(config-subif)#frame-relay interface-dlci 17 ietf
```

2. Specify MPLS tunneling by using the appropriate command.

```
host1(config-if)#mpls-relay 10.10.100.2 45
```

or

```
host1(config-if)#route interface tunnel mpls:tunnel6 45
```

3. Configure Frame Relay and MPLS on the remote PE router.

Related Documentation

- Layer 2 Services over MPLS Overview on page 527
- Example: Configuring Frame Relay over MPLS on page 565

- *encapsulation frame-relay ietf*
- *frame-relay interface-dlci ietf*
- *frame-relay intf-type*
- *frame-relay lmi-type*
- *interface serial*
- *mpls-relay*

- *route interface*

Configuring Interoperation with Legacy Frame Relay Layer 2 Services

To configure the router to interoperate with a router that uses the legacy Frame Relay pseudowire type for layer 2 services over MPLS:

1. Configure the Frame Relay interface.

   host1(config)#**interface serial 4/1:1/1**
   host1(config-if)#**encapsulation frame-relay ietf**
   host1(config-if)#**frame-relay intf-type dte**
   host1(config-if)#**frame-relay lmi-type ansi**
   host1(config-if)#**interface serial 4/1:1/1.1**
   host1(config-subif)#**frame-relay interface-dlci 17 ietf**

2. Specify MPLS tunneling by using the appropriate command.

   host1(config-if)#**mpls-relay 10.10.100.2 45 relay-format frame-relay**

   or

   host1(config-if)#**route interface tunnel mpls:tunnel6 45** relay-format frame-relay

3. Configure Frame Relay and MPLS on the remote PE router.

Related Documentation

- Example: Configuring Frame Relay over MPLS on page 565
- *encapsulation frame-relay ietf*
- *frame-relay interface-dlci ietf*
- *frame-relay intf-type*
- *frame-relay lmi-type*
- *interface serial*
- *mpls-relay*
- *route interface*

Configuring Ethernet/VLAN Layer 2 Services

To configure Ethernet/VLAN layer 2 services over MPLS:

1. Configure the Ethernet/VLAN interface.

   host1(config)#**interface fastEthernet 4/0**
   host1(config-if)#**encapsulation vlan**
   host1(config-if)#**interface fastEthernet 4/0.3**
   host1(config-if)#**vlan id 201**

2. Specify MPLS tunneling by using the appropriate command.

   host1(config-if)#**mpls-relay 10.10.100.2 45**

   or

host1(config-if)#route interface tunnel mpls:tunnel6 45

3. Configure Ethernet/VLAN and MPLS on the remote PE router.

Related Documentation

- *encapsulation vlan*
- *interface fastEthernet*
- *mpls-relay*
- *route interface*
- *vlan id*

Configuring S-VLAN Tunnels for Layer 2 Services

When you configure Ethernet or bridged Ethernet layer 2 services over MPLS, you can use the **svlan id** command with the **any** keyword to create a stacked VLAN (S-VLAN) tunnel that uses a single interface to tunnel traffic from multiple VLANs across an MPLS network. The S-VLAN tunnel enables multiple VLANs, each configured with a different VLAN ID tag and a common S-VLAN ID, to share a common VC label while traversing an MPLS network.

You can use the **svlan id** command with the **any** keyword only with the **mpls-relay** command or the **route interface** command to configure layer 2 services over MPLS.

To configure S-VLAN tunnels for Ethernet/VLAN layer 2 services over MPLS:

1. Configure the Ethernet/VLAN interface.

   host1(config)#**interface fastEthernet 8/1**
   host1(config-if)#**encapsulation vlan**
   host1(config-if)#**interface fastEthernet 8/1.1**

2. Create the S-VLAN tunnel and assign the S-VLAN Ethertype. For example, the following commands tunnel traffic from VLANs configured with an S-VLAN ID of 33 and any VLAN ID to the same destination across the MPLS network.

   host1(config-if)#**svlan id 33 any**
   host1(config-if)#**svlan ethertype 8100**

3. Specify MPLS tunneling by using the appropriate command. For example:

   host1(config-if)#**route interface tunnel mpls:tunnel3 45**

   or

   host1(config-if)#**mpls-relay 10.10.100.2 45**

4. Repeat these steps, using unique values to configure the S-VLAN tunnel and MPLS on the remote PE router.

For more information about S-VLANs, including complete configuration instructions, see the *JunosE Link Layer Configuration Guide*.

Related Documentation

- *encapsulation vlan*

- *interface fastEthernet*
- *mpls-relay*
- *route interface*
- *svlan ethertype*
- *svlan id*

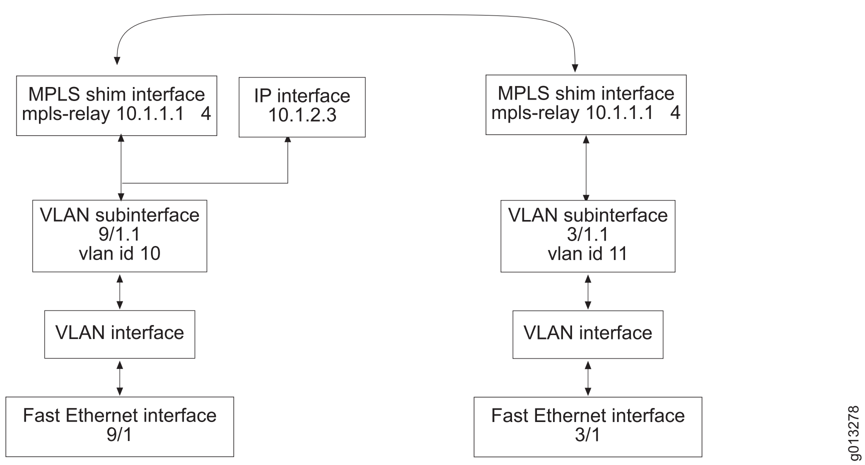
Configuring Local Cross-Connects Between Ethernet/VLAN Interfaces

This section provides an example for configuring a local cross-connect that uses MPLS between two Ethernet/VLAN interfaces.

NOTE: You must use the **mpls-relay** command instead of the **route interface** command to configure a local cross-connect, regardless of the MPLS tunneling method used in the core network.

Figure 121 on page 553 shows the interface stack that the router builds for this configuration.

Figure 121: Local Cross-Connect Between Ethernet/VLAN Interfaces



To configure the application shown in Figure 121 on page 553:

1. Configure a local IP address.

   You can use any reachable local IP address. This example uses a loopback interface to provide the local IP address.

   ```
   host1(config)#interface loopback 0
   host1(config-if)#ip address 10.1.1.1 255.255.255.255
   host1(config-if)#exit
   ```

2. Configure the Ethernet/VLAN interface on one side of the local cross-connect.

   ```
   host1(config)#interface fastEthernet 9/1
   host1(config-if)#encapsulation vlan
   ```

```
host1(config-if)#exit
host1(config)#interface fastEthernet 9/1.1
host1(config-if)#vlan id 10
```

3. (Optional) If you are configuring a multiservice local cross-connect, assign an IP address and mask to the Ethernet/VLAN interface.

   ```
   host1(config-if)#ip address 10.1.2.3 255.255.255.0
   ```

4. Configure MPLS tunneling on this side of the connection by issuing the **mpls-relay** command.

   When you issue the **mpls-relay** command, you must use a reachable local IP address and the same VC ID value (4) on both sides of the connection.

   ```
   host1(config-if)#mpls-relay 10.1.1.1 4
   host1(config-if)#exit
   ```

5. Configure the Ethernet/VLAN interface on the other side of the local cross-connect.

   ```
   host1(config)#interface fastEthernet 3/1
   host1(config-if)#encapsulation vlan
   host1(config-if)#exit
   host1(config)#interface fastEthernet 3/1.1
   host1(config-if)#vlan id 11
   ```

6. (Optional) If you are configuring a multiservice local cross-connect, assign an IP address and mask to the Ethernet/VLAN interface.

   ```
   host1(config-if)#ip address 10.1.2.4 255.255.255.0
   ```

7. Configure MPLS tunneling on this side of the connection by issuing the **mpls-relay** command.

   You must use a reachable local IP address and the same VC ID value (4) specified in Step 4.

   ```
   host1(config-if)#mpls-relay 10.1.1.1 4
   host1(config-if)#exit
   ```

Related Documentation

- *encapsulation vlan*
- *interface fastEthernet*
- *interface loopback*
- *ip address*
- *mpls-relay*
- *vlan id*

Configuring Local ATM Cross-Connects with AAL5 Encapsulation

To create a local cross-connect between two ATM 1483 subinterfaces on the same router, you create a loopback interface, configure your ATM PVCs, and then create an MPLS relay connection from the PVCs to the loopback interface. You do not need to configure any other MPLS commands.

The following commands create an ATM cross-connect between two ATM subinterfaces on the same router.

NOTE: Although this procedure uses AAL5 encapsulation to configure a local cross-connect between two ATM 1483 subinterfaces within the same router, you can also use AAL5 encapsulation when you configure an MPLS pseudowire (tunnel) connection between two ATM VCCs on different routers.

1. Create a loopback interface. All local cross-connects can share the same loopback interface.

   host1(config)#**interface loopback 0**
   host1(config-if)#**ip address 10.1.1.1 255.255.255.255**
   host1(config)#**exit**

2. Create an ATM 1483 subinterface and ATM PVC with **aal5all** encapsulation on the ingress interface.

   host1(config)#**interface atm 2/0.1**
   host1(config-subif)#**atm pvc 1 0 100 aal5all**

3. Create an MPLS relay connection to the loopback interface. Include the address of the loopback interface and a VC ID.

   host1(config-subif)#**mpls-relay 10.1.1.1 2**
   host1(config-subif)#**exit**

4. Create an ATM 1483 subinterface and ATM PVC with **aal5all** encapsulation on the egress interface.

   host1(config)#**interface atm 2/0.2**
   host1(config-subif)#**atm pvc 2 0 101 aal5all**

5. Create an MPLS relay connection to the loopback interface. The VC ID must be the same on both sides of the connection.

   host1(config-subif)#**mpls-relay 10.1.1.1 2**
   host1(config-subif)#**exit**

6. (Optional) Display your configuration.

   host1#**show mpls cross-connects atm**

```
                                                  Cate Peak
Interface  VPI VCI Interface  VPI VCI VC-ID Encap gory Rate  Status
---------- --- --- ---------- --- --- ----- ----- ---- ---- --------
ATM2/0.1     0 100 ATM2/0.2     0 101     2 AAL5  UBR     0 State UP
1 local connection(s) found
```

Related Documentation

- *atm pvc*
- *interface atm*
- *interface loopback*
- *ip address*
- *mpls-relay*

- *show mpls cross-connects atm*
- *vlan id*

Configuring an MPLS Pseudowire with VCC Cell Relay Encapsulation

The following commands create an ATM layer 2 services over MPLS pseudowire connection between two ATM 1483 subinterfaces on different routers. This procedure uses the **aal0** encapsulation keyword for each ATM PVC to indicate that the router receive raw ATM cells on these circuits and forward the cells without performing AAL5 packet reassembly. The procedure also includes optional steps for configuring nondefault values for the ATM Martini cell packing timers and cell concatenation parameters.

NOTE: Although this procedure uses AAL0 encapsulation to configure an MPLS pseudowire (tunnel) connection between two ATM VCCs on different routers, you can also use AAL0 encapsulation when you configure a local cross-connect between two ATM 1483 subinterfaces within the same router.

To create an MPLS pseudowire connection with VCC cell relay encapsulation:

1. (Optional) Configure values for the three ATM Martini cell packing timers on the ingress router to define the cell collection time threshold.

   host1(config)#**atm mcpt-timers 1500 2500 3500**

2. Configure a loopback interface.

   host1(config)#**interface loopback 0**
   host1(config-if)#**ip address 5.1.1.1 255.255.255.255**
   host1(config)#**exit**

3. Create an ATM 1483 subinterface and ATM PVC with **aal0** encapsulation on the ingress interface.

   host1(config)#**interface atm 2/0.100**
   host1(config-subif)#**atm pvc 100 0 100 aal0**

4. (Optional) Configure the following cell concatenation parameters for the ATM 1483 subinterface:

   - Maximum number of ATM cells that the router can concatenate in a single packet
   - Identifier (1, 2, or 3) of the ATM Martini cell packing timer that you want to use to detect timeout of the cell collection threshold

   host1(config-subif)#**atm cell-packing 100 mcpt-timer 2**

5. Create an MPLS relay connection to the loopback interface on the egress router. The VC ID (1 in this example) must be the same on both sides of the connection.

   host1(config-subif)#**mpls-relay 6.1.1.1 1**
   host1(config-subif)#**exit**

6. Repeat Steps 1 through 5 on the egress router, creating an MPLS relay connection to the loopback interface on the ingress router.

   The values you configure for the ATM Martini cell packing timers and cell concatenation parameters need not be the same on the ingress and egress routers, although matching values are permitted. The virtual connection ID (VC ID) value in the **mpls-relay** command, however, must be the same on the ingress and egress routers.

   ```
   host2(config)#atm mcpt-timers 1500 2500 3500
   host2(config)#interface loopback 0
   host2(config-if)#ip address 6.1.1.1 255.255.255.255
   host2(config)#exit
   host2(config)#interface atm 4/0.101
   host2(config-subif)#atm pvc 101 0 101 aal0
   host2(config-subif)#atm cell-packing 150 mcpt-timer 3
   host2(config-subif)#mpls-relay 5.1.1.1 1
   host2(config-subif)#exit
   ```

7. (Optional) Use the appropriate **show** commands to verify your configuration.

```
host1#show atm mcpt-timers
ATM Martini cell aggregation timers:
        Timer1: 1500microseconds
        Timer2: 2500microseconds
        Timer3: 3500microseconds

host1#show atm subinterface atm 2/0.100
                                 Circuit                           Interface
Interface   ATM-Prot VCD VPI VCI  Type   Encap MTU      Status       Type
---------   -------- --- --- --- ------- ----- ---- -------------- ---------
ATM 2/0.100 ATM/MPLS 100   0 100 PVC     AAL0  9180 lowerLayerDown Static

Maximum number of cells per packet: 100
Cell aggregation timeout timer:     2

SNMP trap link-status: disabled

InPackets:                0
InBytes:                  0
OutPackets:               0
OutBytes:                 0
InErrors:                 0
OutErrors:                0
InPacketDiscards:         0
InPacketsUnknownProtocol: 0
OutDiscards:              0
1 interface(s) found

host2#show atm subinterface atm 4/0.101
                                 Circuit                           Interface
Interface   ATM-Prot VCD VPI VCI  Type   Encap MTU      Status       Type
---------   -------- --- --- --- ------- ----- ---- -------------- ---------
ATM 4/0.101 ATM/MPLS 101   0 101 PVC     AAL0  9180 lowerLayerDown Static

Maximum number of cells per packet: 150
Cell aggregation timeout timer:     3

SNMP trap link-status: disabled

InPackets:                0
InBytes:                  0
OutPackets:               0
```

```
OutBytes:                 0
InErrors:                 0
OutErrors:                0
InPacketDiscards:         0
InPacketsUnknownProtocol: 0
OutDiscards:              0
1 interface(s) found
```

Related Documentation

- *atm cell-packing*
- *atm mcpt-timers*
- *atm pvc*
- *interface atm*
- *interface loopback*
- *ip address*
- *mpls-relay*
- *show atm mcpt-timers*
- *show atm subinterface*

Configuring HDLC Layer 2 Services

The following sections describe how to configure HDLC layer 2 services over MPLS and in local cross-connects:

Configuring HDLC Layer 2 Services over MPLS

The following commands configure an HDLC layer 2 circuit over MPLS between an E Series router and a remote PE device.

To configure an HDLC layer 2 circuit over MPLS:

1. Configure a serial or POS interface on the ingress router.

   host1(config)#interface serial 3/1:2/1

   or

   host1(config)#interface pos 4/0

2. Use one of the following methods to create the HDLC layer 2 circuit over MPLS:

   - Use the **mpls-relay** or **route interface** command *without* the **relay-format ppp** keywords. This command causes the router to signal VC-type HDLC on the LDP session and use HDLC encapsulation. Use this command syntax if the traffic carried on the serial or POS interface is any kind of standard HDLC (including PPP) or Cisco HDLC.

     host1(config-if)#mpls-relay 2.2.2.1 1

or

host1(config-if)#**route interface tunnel mpls:tunnel-to-pe2 1**

- Use the **mpls-relay** or **route interface** command *with* the **relay-format ppp** keywords. This command causes the router to signal VC-type PPP on the LDP session and use PPP encapsulation instead of the default VC-type HDLC signaling and HDLC encapsulation. Use this command syntax if the traffic carried on the serial or POS interface contains actual PPP packets.

  host1(config-if)#**mpls-relay 2.2.2.1 1 relay-format ppp**

  or

  host1(config-if)#**route interface tunnel mpls:tunnel-to-pe2 1** relay-format ppp

3. (Optional) Attach an MPLS policy to the HDLC layer 2 circuit by using the **mpls policy** command.

   host1(config-if)#**mpls policy input hdlc-policy**

4. Configure the serial or POS interface and MPLS on the remote PE device.

   The interfaces at either end of the HDLC layer 2 circuit can be different types and have different speeds. For example, you can configure an HDLC layer 2 circuit between a serial interface on a T1 circuit and a POS interface on an OC3 circuit.

Local Cross-Connects for HDLC Layer 2 Services Configuration Differences

You can also configure an HDLC layer 2 circuit in a local cross-connect configuration between serial or POS interfaces within the same router.

The procedure is basically the same for configuring an HDLC layer 2 interface between two PE routers and for a local cross-connect, with the following differences for local cross-connects:

- You must use the **mpls-relay** command instead of the **route interface** command to configure a local cross-connect for HDLC layer 2 services.
- You use the IP address of the local router as the value for the destination IP address (remote address) in the **mpls-relay** command.

For more information about policies and MPLS layer 2 circuits, see *Managing Policies on the E Series Router*, *Creating or Modifying Classifier Control Lists for MPLS Policy Lists*, and *Creating Policy Lists for MPLS* in the *JunosE Policy Management Configuration Guide*.

CE-Side Load Balancing for Martini Layer 2 Transport

This section contains the following subsections:

Understanding CE Load Balancing for Martini Layer 2 Transport

For layer 2 circuits over an MPLS core, each circuit normally has a single shim interface on the local router. In the case of a local cross-connects configuration, each end of the cross-connect has a single shim interface, creating a two-way cross-connect.

Alternatively, a given layer 2 circuit or each end of a local cross-connect can have many shim interfaces. In these cases, traffic destined for the CE routers is load-balanced among the multiple shim interfaces. This is known as CE-side load balancing. In the case of Ethernet/VLANs, CE-side load balancing enables an E Series router to interoperate with an 802.3ad switch.

You can configure load balancing in two different ways. You can configure many shim interfaces with the same peer, VC type, and VC ID. Alternatively, you can use the legacy method of configuring Martini circuits into load-balancing groups.

Related Documentation

Configuration of Many Shim Interfaces with the Same Peer, VC Type, and VC ID

The **mpls-relay** command enables you to specify the peer and the VC ID explicitly. The VC type is either automatically determined by the layer 2 interface type or you explicitly configure the VC type with the **relay-format** keyword in the **mpls-relay** command.

Related Documentation

Example: Configuring Many Shim Interfaces with the Same Peer, VC Type, and VC ID

In this example, the following commands create a single layer 2 circuit to the remote peer 10.9.1.3. Load balancing is established on two shim interfaces, fastEthernet 2/0.1 and fastEthernet 3/0.1. The VC type is determined by the layer 2 interface type.

```
host1(config)#interface fast 2/0.1
host1(config-subif)#vlan id 1
```


host1(config-subif)#**mpls-relay 10.9.1.3 200001**
host1(config-subif)#**exit**
host1(config)#**interface fast 3/0.1**
host1(config-subif)#**vlan id 1**
host1(config-subif)#**mpls-relay 10.9.1.3 200001**

In this example, the router advertises a single label, 53, to the remote peer, 10.9.1.3, and receives a single label, 55, from the peer, resulting in the following forwarding table:

```
host1:# show mpls forwarding brief
Platform label space
In Label  Owner                                 Action
-------- -------- --------------------------------------------------------
53       ldp      l2transport to FastEthernet3/0.1
                  l2transport to FastEthernet2/0.1
L2transport
     Interface        Owner                      Action
-------------------- -------- --------------------------------------------------
FastEthernet2/0.1    ldp      swap to 55, push 42 on ATM5/0.1, nbr 10.10.11.5
FastEthernet3/0.1    ldp      swap to 55, push 42 on ATM5/0.1, nbr 10.10.11.5
```

Traffic that arrives on either interface, 2/0.1 or 3/0.1, is forwarded to the remote peer with the same label stack (55, 42). Traffic from the remote peer with label 53 is forwarded to one of the two shim interfaces; the ECMP algorithm determines which of the two shim interfaces receives the traffic.

In the case of a local cross-connects configuration, the following commands illustrate how a three-way cross-connect is created when 10.9.1.2 is a local address:

host1(config)#**interface atm 6/0.101 point-to-point**
host1(config-subif)#**mpls-relay 10.9.1.2 600001**
host1(config-subif)#**exit**
host1(config)#**interface atm 6/2.101 point-to-point**
host1(config-subif)#**mpls-relay 10.9.1.2 600001**
host1(config-subif)#**exit**
host1(config)#**interface atm 6/2.103 point-to-point**
host1(config-subif)#**mpls-relay 10.9.1.2 600001**

This configuration results in the following forwarding table:

```
host1:two# show mpls forwarding brief
Platform label space
...
L2transport
     Interface        Owner                              Action
-------------------- -------- ----------------------------------------------------------------------
ATM6/0.101           ldp      l2transport to ATM6/2.101
                              l2transport to ATM6/2.103
ATM6/2.101           ldp      l2transport to ATM6/0.101
                              l2transport to ATM6/2.103
ATM6/2.103           ldp      l2transport to ATM6/0.101
                              l2transport to ATM6/2.101
```

Traffic that arrives on interface 6/0.101 is forwarded by means of ECMP to both interface 6/2.101 and interface 6/2.103. Traffic that arrives on interface 6/2.101 is forwarded by means of ECMP to interface 6/0.101 and interface 6/2.103. Traffic that arrives on interface 6/2.103 is forwarded by means of ECMP to interface 6/0.101 and 6/2.101.

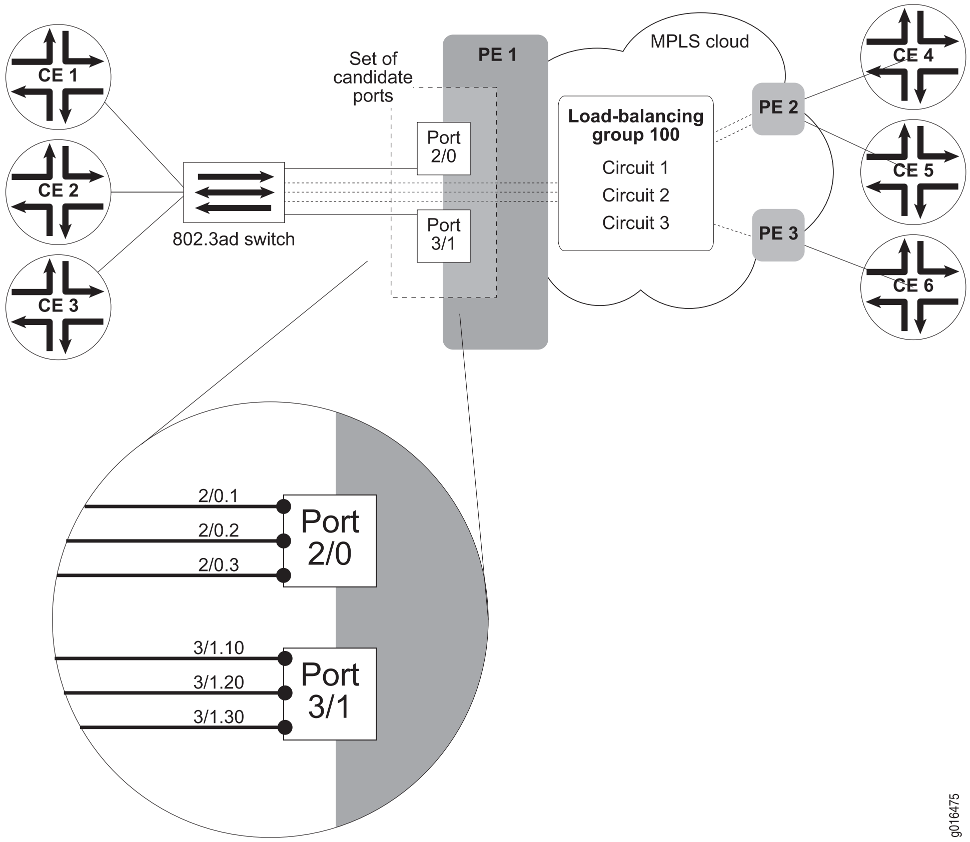
Related Documentation

- MPLS Shim Interfaces for Layer 2 Services over MPLS Overview on page 531
- Multiple Layer 2 Services over MPLS Overview on page 533
- Understanding CE Load Balancing for Martini Layer 2 Transport on page 560
- Configuration of Many Shim Interfaces with the Same Peer, VC Type, and VC ID on page 560
- Monitoring MPLS Forwarding for Layer 2 Services over MPLS on page 588
- Monitoring MPLS Layer 2 Interfaces for Layer 2 Services over MPLS on page 590
- *interface atm*
- *interface fastEthernet*
- *mpls-relay*
- *show mpls forwarding*
- *vlan id*

Load-Balancing Group Configuration

Load-balancing groups are a legacy method of configuring CE-side load balancing. It was the only method available before Release 7.1.0. Load-balancing groups enable you to configure attributes for a group that are inherited by the member shim interfaces.

Figure 122: CE-Side Load-Balancing Topology



In the topology shown in Figure 122 on page 563, the two Ethernet ports on PE 1 (2/0 and 3/1) are connected to an 802.3ad-compliant switch, and comprise the set of candidate ports. Three VLAN subinterfaces are configured on each port. Load-balancing group 100 includes three Martini circuits, one for each pair of subinterfaces on the ports. That is, three circuits were created: one for the pair 2/0.1 and 3/1.10, one for the pair 2/0.2 and 3/1.20, and one for the pair 2/0.3 and 3/1.30. Each of the three Martini circuits connects to a remote PE router. The remote PE router receives and sends only a single VC label for each circuit.

Traffic from the switch can be received on all ports and sent over the appropriate Martini circuit. For example, traffic from CE 1 to be sent over Martini circuit 1 could arrive on port 2/0 or 3/1 and still be appropriately forwarded.

You configure each circuit for VLAN or S-VLAN subinterfaces that you create across a set of candidate Ethernet ports. The router distributes traffic from the core through the candidate ports used by the load-balancing group. If a port is disabled, traffic is redistributed to a working port.

Related Documentation

- Understanding CE Load Balancing for Martini Layer 2 Transport on page 560
- MPLS Interfaces and Labels on page 564
- Configuring Load-Balancing Groups on page 564

MPLS Interfaces and Labels

When a layer 2 interface is added to a load-balancing group circuit, an MPLS shim interface is automatically created on top of that layer 2 interface. The attributes of the shim interface are inherited from the load-balancing group and cannot be configured.

All MPLS shim interfaces within a load-balancing group circuit point to the same MPLS next hop. Traffic arriving from the CE router over this set of MPLS shim interfaces is merged into a single LSP and sent to the remote PE router.

The VC in label for the layer 2 circuit points to a single ECMP MPLS next hop. The legs of this ECMP next hop are the member shim interfaces of the load-balancing group circuit. Consequently, ECMP is used to forward traffic arriving from the core across the MPLS shim interfaces to the CE router.

Related Documentation

- Multiple Layer 2 Services over MPLS Overview on page 533
- Understanding CE Load Balancing for Martini Layer 2 Transport on page 560
- Load-Balancing Group Configuration on page 562
- Configuring Load-Balancing Groups on page 564

Configuring Load-Balancing Groups

You configure Martini circuits with load-balancing groups in a separate mode, in which the member layer 2 subinterfaces are entered one by one. For example, the following commands configure two Martini circuits to different PE routers, in the same load-balancing group 100, sharing the candidate Ethernet ports 2/0 and 3/0:

host1(config)#**mpls l2transport load-balancing-group 100 mpls-relay 2.2.2.2 202**
host1(config-mpls-l2-group)#**member interface fast 2/0.500**

You can also configure member interfaces as described in the following sections:

- Adding a Member Interface to a Group Circuit on page 564
- Removing Member Subinterfaces from a Circuit on page 564

Adding a Member Interface to a Group Circuit

You specify the lower interface as a member interface, as in the following example.

host1(config)#**mpls l2transport load-balancing-group 100 mpls-relay 2.2.2.2 202**
host1(config-mpls-l2-group)#**member interface fast 2/0.500**

Removing Member Subinterfaces from a Circuit

To remove a member subinterface from a circuit, either issue the **no member interface** command (from the L2 Transport Load-Balancing-Group Configuration mode) or issue the **no mpls-relay** command at the VLAN or S-VLAN subinterface level. Each of the following examples removes member Fast Ethernet subinterface 13/0.2 from the load-balanced Martini circuit:

host1(config)#**mpls l2transport load-balancing-group 100 mpls-relay 2.2.2.2 202** host1(config-mpls-l2-group)#**no member interface fast 13/0.2**

or

host1()#**interface fast 13/0.2**
host1(config-subif)#**no mpls-relay**

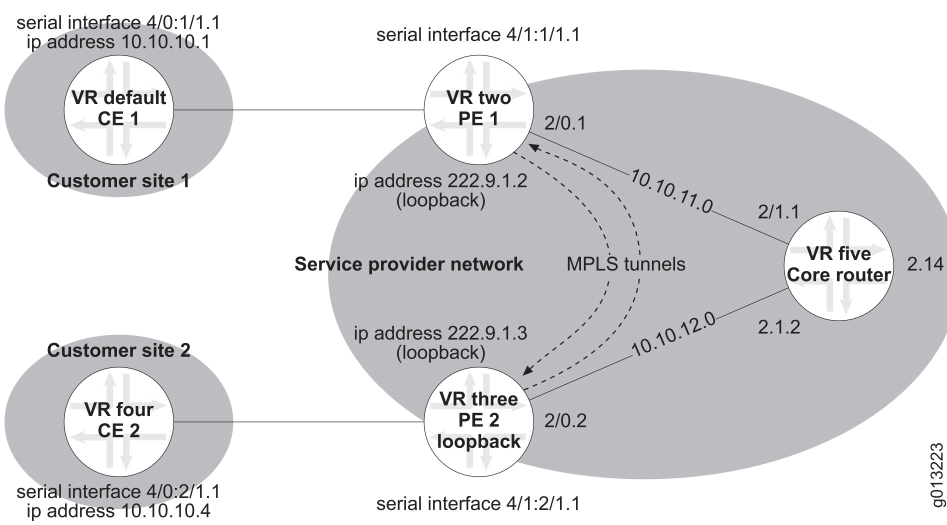
Example: Configuring Frame Relay over MPLS

The script provided in this section is one way to configure Frame Relay services over MPLS. Explanation notes are provided within the script. You must change the script for your specific configuration.

The topology example shown in Figure 123 on page 565 further explains the configuration script.

NOTE: The **route interface** command is used toward the end of the configuration script. You can substitute the **mpls-relay** command, depending on the tunneling method best for your environment.

Figure 123: Sample Frame Relay over MPLS Configuration




```
hostname "host 1"
exception protocol ftp anonymous null
!---------------------------------------------------------------------------
!Configure CT3 interfaces in slot 4 for Frame Relay.
!---------------------------------------------------------------------------
!
controller t3 4/0
 no shutdown
 clock source internal module
 cablelength 5
 t1 1 clock source internal module
 t1 1/1 timeslots 1-24 speed 64
 t1 2 clock source internal module
 t1 2/1 timeslots 1-24 speed 64
```

```
!
controller t3 4/1
 no shutdown
 clock source internal module
 cablelength 5
 t1 1 clock source internal module
 t1 1/1 timeslots 1-24 speed 64
 t1 2 clock source internal module
 t1 2/1 timeslots 1-24 speed 64
!--------------------------------------------------------------------------
!Create virtual router default.
!--------------------------------------------------------------------------
virtual-router default
interface loopback 0
  ip address 222.9.1.1 255.255.255.255
!--------------------------------------------------------------------------
!Configure Frame Relay interfaces.
!--------------------------------------------------------------------------
interface serial 4/0:1/1
 encapsulation frame-relay ietf
interface serial 4/0:1/1.1
frame-relay interface-dlci 17 ietf
!
interface serial 4/0:2/1
 encapsulation frame-relay ietf
interface serial 4/0:2/1.1
frame-relay interface-dlci 12 ietf
!
interface serial 4/1:1/1
 encapsulation frame-relay ietf
 frame-relay intf-type dce
interface serial 4/1:1/1.1
frame-relay interface-dlci 17 ietf
!
interface serial 4/1:2/1
 encapsulation frame-relay ietf
 frame-relay intf-type dce
interface serial 4/1:2/1.1
frame-relay interface-dlci 12 ietf
!--------------------------------------------------------------------------
!Create virtual router two. Configure MPLS.
!--------------------------------------------------------------------------
virtual-router two

mpls
mpls ldp tar send list 222.9.1.3

interface loopback 0
  ip address 222.9.1.2 255.255.255.255
  ip router isis

interface atm 2/0
 atm clock inter mod
interface atm 2/0.1
 atm pvc 1 1 11 aal5snap
 ip address 10.10.11.2 255.255.255.0
 ip router isis
 mpls
 mpls ldp
router isis
 net 47.0005.80FF.F800.0000.0000.0004.0000.F209.0202.00
```



```
 mpls traffic-eng router-id loopback 0
 mpls traffic-eng level-1
 metric-style wide
!---------------------------------------------------------------------------
!Create virtual router three. Configure MPLS.
!---------------------------------------------------------------------------
virtual-router three

mpls
mpls ldp tar send list 222.9.1.2

interface loopback 0
  ip address 222.9.1.3 255.255.255.255
  ip router isis

interface atm 2/0.2
 atm pvc 2 1 12 aal5snap
 ip address 10.10.12.3 255.255.255.0
 ip router isis
 mpls
 mpls ldp

router isis
 net 47.0005.80FF.F800.0000.0000.0004.0000.F209.0303.00
 mpls traffic-eng router-id loopback 0
 mpls traffic-eng level-1
 metric-style wide
!---------------------------------------------------------------------------
!Create virtual router four.
!---------------------------------------------------------------------------
virtual-router four

interface loopback 0
  ip address 222.9.1.4 255.255.255.255

!---------------------------------------------------------------------------
!Create virtual router five. Configure MPLS.
!---------------------------------------------------------------------------

virtual-router five

mpls

interface loopback 0
  ip address 222.9.1.5 255.255.255.255
  ip router isis

interface atm 2/1.1
 atm pvc 1 1 11 aal5snap
 ip address 10.10.11.5 255.255.255.0
 ip router isis
 mpls
 mpls ldp

interface atm 2/1.2
 atm pvc 2 1 12 aal5snap
 ip address 10.10.12.5 255.255.255.0
 ip router isis
 mpls
 mpls ldp
```

```
router isis
 net 47.0005.80FF.F800.0000.0000.0004.0000.F209.0505.00
 mpls traffic-eng router-id loopback 0
 mpls traffic-eng level-1
 metric-style wide
 !---------------------------------------------------------------------------
 !Create MPLS tunnel from VR three to VR two. Route Frame Relay traffic via MPLS
 tunnel.
 !---------------------------------------------------------------------------
 vir three

 interface tunnel mpls:3
 tunnel destination 222.9.1.2

 interface serial 4/1:2/1.1
 route interface tunnel mpls:3 45

 vir two

 interface tunnel mpls:2
 tunnel destination 222.9.1.3

 int ser 4/1:1/1.1
 route interface tunnel mpls:2 45
```

Related Documentation

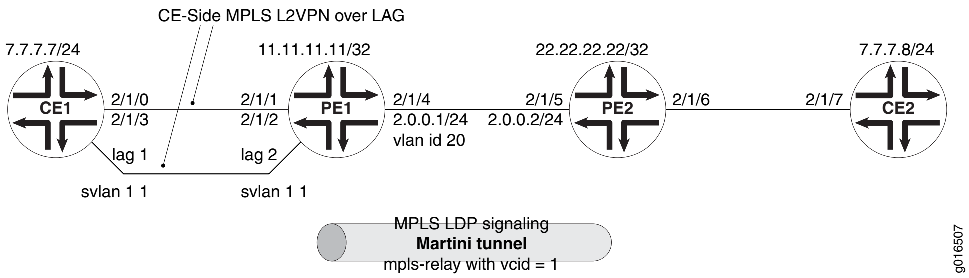
- Layer 2 Services over MPLS Overview on page 527
- Configuring Frame Relay Layer 2 Services on page 550

Example: Configuring MPLS L2VPN Tunnel over VLAN over LAG

Figure 124 on page 569 shows a sample configuration scenario of an MPLS L2VPN or Martini tunnel over VLAN over LAG. The sample topology shows a customer edge router, CE1, connected to a provider edge router, PE1 using a stacked VLAN (S-VLAN) tunnel.

Two LAG bundles, LAG1 and LAG2, are created to group multiple Ethernet interfaces on CE1 and PE1 respectively. A Martini tunnel from PE1 to PE2 is configured over LAG2 with a unique subinterface number assigned to the LAG bundle. The MPLS packets that are received on PE2, which is the remote router located at the other side of the service provider core, are processed. After PE2 processes the layer 2 Ethernet frames, they are sent to CE2, which is the customer edge device at the remote site.

Figure 124: MPLS L2VPN Tunnel over VLAN over LAG Configuration Example



- Configuration on CE1 (Local CE Router) on page 569
- Configuration on PE1 (Local PE Router) on page 569
- Configuration on PE2 (Remote PE Router) on page 570
- Configuration on CE2 (Remote CE Router) on page 572

Configuration on CE1 (Local CE Router)

Use the following commands on the local CE router (CE1) to configure the MPLS L2VPN tunnel shown in Figure 124 on page 569.

```
! Configure a virtual router CE1.
host1(config)#virtual-router ce1
!
! Specify the interface for the LAG bundle lag 1 that groups all Ethernet physical
! interfaces between CE1 and PE1.
host1:ce1(config)#interface lag 1
!
! Add the Gigabit Ethernet physical interfaces to the LAG bundle named lag 1.
host1:ce1(config-if)#member-interface gigabitEthernet 2/1/0
host1:ce1(config-if)#member-interface gigabitEthernet 2/1/3
!
! Specify VLAN as the encapsulation method for the Ethernet interface.
host1:ce1(config-if)#encapsulation vlan
!
! Specify another subinterface in the LAG bundle lag 1.1.
host1:ce1(config-if)#interface lag 1.1
!
! Assign an S-VLAN ID and a VLAN ID for the subinterface, and assign an IP
! address and mask to the interface.
host1:ce1(config-subif)#svlan id 1 1
host1:ce1(config-subif)#ip address 7.7.7.7 255.255.255.0
```

Configuration on PE1 (Local PE Router)

Use the following commands on the local PE router (PE1) to configure the MPLS L2VPN tunnel shown in Figure 124 on page 569.

```
! Configure a virtual router PE1.
host1(config)#virtual-router pe1
!
! Enable MPLS on a virtual router in Global Configuration mode.
host1:pe1(config)#mpls
```

```
!
! Configure the LSR to create topology-driven LSPs. Enabling LDP automatically
! creates topology-driven LSPs.
host1:pe1(config)#mpls topology-driven-lsp
!
! On PE1, configure a loopback interface, and assign an IP address and mask to
! the interface.
host1:pe1(config)#interface loopback 0
host1:pe1(config-if)#ip address 11.11.11.11 255.255.255.255
!
! Assign the router ID using the IP address you configured for the loopback
! interface.
host1:pe1(config)#ip router-id 11.11.11.11
!
! Create an IEEE 802.3ad LAG bundle, lag2, and add the Gigabit Ethernet physical
! interfaces to lag2. Specify VLAN as the encapsulation method for the member
! interfaces.
host1:pe1(config)#interface lag 2
host1:pe1(config-if)#member-interface gigabitEthernet 2/1/1
host1:pe1(config-if)#member-interface gigabitEthernet 2/1/2
host1:pe1(config-if)#encapsulation vlan
!
! Specify a subinterface by adding a unique subinterface number to the LAG bundle
! name. Assign an S-VLAN ID and a VLAN ID for the subinterface. Also, configure
! MPLS tunneling.
host1:pe1(config-if)#interface lag 2.1
host1:pe1(config-subif)#svlan id 1 1
host1:pe1(config-subif)#mpls-relay 22.22.22.22 1
!
! Create another Gigabit Ethernet interface on PE1, specify VLAN as the
! encapsulation method, and configure another subinterface. For this subinterface,
! assign a VLAN ID, specify the encapsulation method as VLAN, configure MPLS,
! and assign an IP address and mask. Also, enable LDP and topology-driven LSP, a
! does any LDP-related command, using an implicit default profile, on this Gigabit
! Ethernet subinterface.
host1:pe1(config)#interface gigabitEthernet 2/1/4
host1:pe1(config-if)#encapsulation vlan
host1:pe1(config-if)#interface gigabitEthernet 2/1/4.1
host1:pe1(config-subif)#vlan id 20
host1:pe1(config-subif)#ip address 2.0.0.1 255.255.255.0
host1:pe1(config-subif)#mpls
host1:pe1(config-subif)#mpls ldp
!
! Configure a static route with the destination IP address, mask, and IP address of
! the next hop that can be used to reach the destination network.
host1:pe1(config)#ip route 22.22.22.22 255.255.255.255 2.0.0.2
!
! Configure LDP to advertise a non-null label for the egress routes.
host1:pe1(config)#mpls ldp egress-label non-null
```

Configuration on PE2 (Remote PE Router)

Use the following commands on the remote PE router (PE2) to configure the MPLS L2VPN tunnel shown in Figure 124 on page 569.

```
! Configure a virtual router PE2.
```



```
host1(config)#virtual-router pe1
!
! Enable MPLS on a virtual router in Global Configuration mode.
host1:pe2(config)#mpls
!
! Configure PE2 to create topology-driven LSPs. Enabling LDP automatically creates
! topology-driven LSPs.
host1:pe2(config)#mpls topology-driven-lsp
!
! On PE2, configure a loopback interface, and assign an IP address and mask to
! the interface.
host1:pe2(config)#interface loopback 0
host1:pe2(config-if)#ip address 22.22.22.22 255.255.255.255
!
! Assign the router ID using the IP address you configured for the loopback
! interface.
host1:pe1(config)#ip router-id 22.22.22.22
!
! Create a Gigabit Ethernet interface and configure MPLS tunneling with the IP
! address of the router on the remote end of the layer 2 circuit and the virtual
! circuit identifier.
host1:pe2(config)#interface gigabitEthernet 2/1/6
host1:pe2(config-if)#encapsulation vlan
!
! Create a VLAN subinterface by adding a subinterface number to the interface
! identification command. Assign an S-VLAN ID and a VLAN ID for the subinterface
! Also, configure MPLS tunneling.
host1:pe2(config-if)#interface gigabitEthernet 2/1/6.1
host1:pe2(config-subif)#svlan id 1 1
host1:pe2(config-subif)#mpls-relay 11.11.11.11 1
!
! Create a Gigabit Ethernet interface on PE2 and specify VLAN as the
! encapsulation method.
host1:pe2(config)#interface gigabitEthernet 2/1/5
host1:pe2(config-if)#encapsulation vlan
!
! Create another Gigabit Ethernet subinterface on the main interface. For this
! interface, assign a VLAN ID, specify the encapsulation method as VLAN, configure
! MPLS, and assign an IP address and mask. Also, enable LDP and topology-driven
! LSP, as does any LDP-related command, using an implicit default profile, on this
! Gigabit Ethernet subinterface.
host1:pe2(config-if)#interface gigabitEthernet 2/1/5.1
host1:pe2(config-subif)#vlan id 20
host1:pe2(config-subif)#ip address 2.0.0.2 255.255.255.0
host1:pe2(config-subif)#mpls
host1:pe2(config-subif)#mpls ldp
!
! Configure a static route with the destination IP address, mask, and IP address of
! the next hop that can be used to reach the destination network.
host1:pe2(config)#ip route 22.22.22.22 255.255.255.255 2.0.0.1
!
! Configure LDP to advertise a non-null label for the egress routes.
host1:pe2(config)#mpls ldp egress-label non-null
```


Configuration on CE2 (Remote CE Router)

Use the following commands on the remote CE router (CE2) to configure the MPLS L2VPN tunnel shown in Figure 124 on page 569.

```
! Configure a virtual router CE2
host1(config)#virtual-router ce2
!
! Specify a Gigabit Ethernet interface and assign VLAN as the encapsulation method.
host1:ce2(config)#interface gigabitEthernet 2/1/7
host1:ce2(config-if)#encapsulation vlan
!
! Create a subinterface, assign an S-VLAN ID, and configure an IP address and mask to it.
host1:ce2(config-if)#interface gigabitEthernet 2/1/7.1
host1:ce2(config-subif)#svlan id 1 1
host1:ce2(config-subif)#ip address 7.7.7.8 255.255.255.0
```

Related Documentation

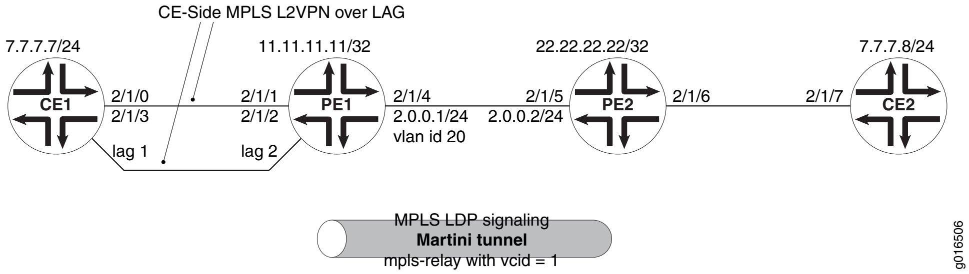
- Example: Configuring MPLS L2VPN Tunnel over LAG on page 572
- CE-Side MPLS L2VPNs over LAG Overview on page 539

Example: Configuring MPLS L2VPN Tunnel over LAG

Figure 125 on page 573 shows a sample configuration scenario of an MPLS L2VPN or Martini tunnel over LAG. The topology is the same as the one described in "Example: Configuring MPLS L2VPN Tunnel over VLAN over LAG" on page 568, with the exception of LAG bundles being used to transmit traffic from CE1 to PE1 instead of using a VLAN subinterface over a LAG bundle. Because the Martini tunnel is configured directly over LAG in this case, only the source and destination MAC addresses are used in the hashing process to determine the physical link for forwarding the received packets.

Two LAG bundles, LAG1 and LAG2, are created to group multiple Ethernet interfaces from PE1 to CE1. A Martini tunnel from PE1 to PE2 is configured over LAG2 with a unique subinterface number assigned to the LAG bundle. The MPLS packets that are received on PE2, which is the remote router located at the other side of the service provider core, are checked to determine whether they need to be processed for MPLS labels. After PE2 processes the layer 2 Ethernet frames, they are sent to CE2, which is the customer edge device at the remote site.

Figure 125: MPLS L2VPN Tunnel over LAG Configuration Example



- Configuration on CE1 (Local CE Router) on page 573
- Configuration on PE1 (Local PE Router) on page 573
- Configuration on PE2 (Remote PE Router) on page 574
- Configuration on CE2 (Remote CE Router) on page 575

Configuration on CE1 (Local CE Router)

Use the following commands on the local CE router (CE1) to configure the MPLS L2VPN tunnel over LAG shown in Figure 125 on page 573.

! Configure a virtual router CE1.
host1(config)#**virtual-router ce1**
!
! Specify the interface for the LAG bundle lag 1 that groups all Ethernet physical
! interfaces between CE1 and PE1.
host1:ce1(config)#**interface lag 1**
!
! Add the Gigabit Ethernet physical interfaces to the LAG bundle named lag 1.
! Assign an IP address and mask to it.
host1:ce1(config-if)#**member-interface gigabitEthernet 2/1/0**
host1:ce1(config-if)#**member-interface gigabitEthernet 2/1/3**
host1:ce1(config-if)#**ip address 7.7.7.7 255.255.255.0**
!

Configuration on PE1 (Local PE Router)

Use the following commands on the local PE router (PE1) to configure the MPLS L2VPN tunnel over LAG shown in Figure 125 on page 573.

! Configure a virtual router PE1.
host1(config)#**virtual-router pe1**
!
! Enable MPLS on a virtual router in Global Configuration mode.
host1:pe1(config)#**mpls**
!
! Configure the LSR to create topology-driven LSPs. Enabling LDP automatically
! creates topology-driven LSPs.
host1:pe1(config)#**mpls topology-driven-lsp**
!
! On PE1, configure a loopback interface, and assign an IP address and mask to
! the interface.
host1:pe1(config)#**interface loopback 0**
host1:pe1(config-if)#**ip address 11.11.11.11 255.255.255.255**

```
!
! Assign the router ID using the IP address you configured for the loopback
! interface.
host1:pe1(config)#ip router-id 11.11.11.11
!
! Create an IEEE 802.3ad LAG bundle, lag2, and add the Gigabit Ethernet physical
! interfaces to lag2. Configure MPLS tunneling on this side of the connection by
! issuing the mpls-relay command. When you issue the mpls-relay command, you
! must use a reachable local IP address and the same VC ID value (1) on both
! sides of the connection.
host1:pe1(config)#interface lag 2
host1:pe1(config-if)#member-interface gigabitEthernet 2/1/1
host1:pe1(config-if)#member-interface gigabitEthernet 2/1/2
host1:pe1(config-if)#mpls-relay 22.22.22.22 1
! Create another Gigabit Ethernet interface on PE1, specify VLAN as the
! encapsulation method, and configure another subinterface. For this subinterface,
! assign a VLAN ID, specify the encapsulation method as VLAN, configure MPLS,
! and assign an IP address and mask. Also, enable LDP and topology-driven LSP, as
! does any LDP-related command, using an implicit default profile, on this Gigabit
! Ethernet subinterface.
host1:pe1(config)#interface gigabitEthernet 2/1/4
host1:pe1(config-if)#encapsulation vlan
host1:pe1(config-if)#interface gigabitEthernet 2/1/4.1
host1:pe1(config-subif)#vlan id 20
host1:pe1(config-subif)#ip address 2.0.0.1 255.255.255.0
host1:pe1(config-subif)#mpls
host1:pe1(config-subif)#mpls ldp
!
! Configure a static route with the destination IP address, mask, and IP address of
! the next hop that can be used to reach the destination network.
host1:pe1(config)#ip route 22.22.22.22 255.255.255.255 2.0.0.2
!
! Configure LDP to advertise a non-null label for the egress routes.
host1:pe1(config)#mpls ldp egress-label non-null
```

Configuration on PE2 (Remote PE Router)

Use the following commands on the remote PE router (PE2) to configure the MPLS L2VPN tunnel over LAG shown in Figure 125 on page 573.

```
! Configure a virtual router PE2.
host1(config)#virtual-router pe1
!
! Enable MPLS on a virtual router in Global Configuration mode.
host1:pe2(config)#mpls
!
! Configure PE2 to create topology-driven LSPs. Enabling LDP automatically creates
! topology-driven LSPs.
host1:pe2(config)#mpls topology-driven-lsp
!
! On PE2, configure a loopback interface, and assign an IP address and mask to
! the interface.
host1:pe2(config)#interface loopback 0
host1:pe2(config-if)#ip address 22.22.22.22 255.255.255.255
!
! Assign the router ID using the IP address you configured for the loopback
```



```
! interface.
host1:pe1(config)#ip router-id 22.22.22.22
!
! Create a Gigabit Ethernet interface and configure MPLS tunneling with the IP
! address of the router on the remote end of the layer 2 circuit and the virtual
! circuit identifier. Configure MPLS tunneling on this side of the connection by
! issuing the mpls-relay command.
host1:pe2(config)#interface gigabitEthernet 2/1/6
host1:pe2(config-if)#mpls-relay 11.11.11.11 1
!
! Create a Gigabit Ethernet interface on PE2 and specify VLAN as the
! encapsulation method.
host1:pe2(config)#interface gigabitEthernet 2/1/5
host1:pe2(config-if)#encapsulation vlan
!
! Create another Gigabit Ethernet subinterface on the main interface. For this
! interface, assign a VLAN ID, specify the encapsulation method as VLAN, configure
! MPLS, and assign an IP address and mask. Also, enable LDP and topology-driven
! LSP, as does any LDP-related command, using an implicit default profile, on this
! Gigabit Ethernet subinterface.
host1:pe2(config-if)#interface gigabitEthernet 2/1/5.1
host1:pe2(config-subif)#vlan id 20
host1:pe2(config-subif)#ip address 2.0.0.2 255.255.255.0
host1:pe2(config-subif)#mpls
host1:pe2(config-subif)#mpls ldp
!
! Configure a static route with the destination IP address, mask, and IP address of
! the next hop that can be used to reach the destination network.
host1:pe2(config)#ip route 22.22.22.22 255.255.255.255 2.0.0.1
!
! Configure LDP to advertise a non-null label for the egress routes.
host1:pe2(config)#mpls ldp egress-label non-null
```

Configuration on CE2 (Remote CE Router)

Use the following commands on the remote CE router (CE2) to configure the MPLS L2VPN tunnel over LAG shown in Figure 125 on page 573.

```
! Configure a virtual router CE2
host1(config)#virtual-router ce2
!
! Specify a Gigabit Ethernet interface and assign an IP address and mask to it
host1:ce2(config)#interface gigabitEthernet 2/1/7
host1:ce2(config-subif)#ip address 7.7.7.8 255.255.255.0
```

Related Documentation

- Example: Configuring MPLS L2VPN Tunnel over VLAN over LAG on page 568
- CE-Side MPLS L2VPNs over LAG Overview on page 539

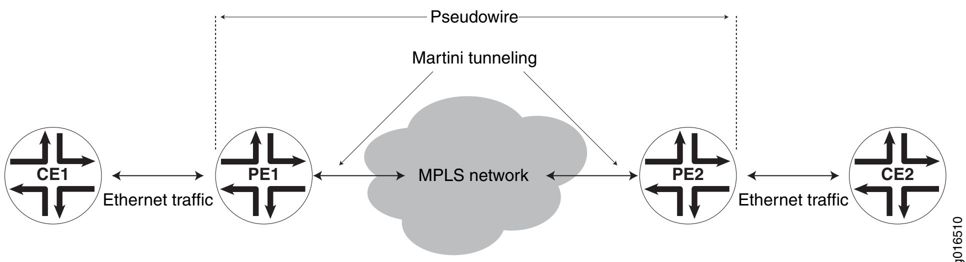
Examples: Ethernet Raw Mode Encapsulation for Martini Layer 2 Transport

When a Martini circuit operates in Ethernet raw mode, you can configure the provider edge (PE) devices that receive packets from the customer edge (CE)-facing devices to remove the S-VLAN tags from all packets entering the circuit. A device or network is said to be S-VLAN-aware, if the packets passed through it contain the S-VLAN tag configured on the subinterface. Similarly, if the S-VLAN tag is not configured on the subinterface, the device is said to be S-VLAN-unaware of the packet reaching it because the S-VLAN tag is not passed to it.

A particular S-VLAN tag that is identified as the service-delimiting tag is removed from the incoming Ethernet frame from the CE device before it is sent on the pseudowires. In this release, only raw mode operation is supported on the S-VLAN subinterfaces of PE routers. Therefore, scenarios in which the S-VLAN tag is made available to the MPLS network and the S-VLAN tag is not advertised to the CE device are not supported. Depending on how Ethernet raw mode is configured on PE devices and the configuration of the S-VLAN subinterface on CE-facing devices, a device in the Martini circuit can be either S-VLAN-aware or S-VLAN-unaware.

Figure 126 on page 576 shows a Martini circuit deployment in which the CE-side devices on either side of the network send and receive Ethernet frames. The packets reaching the CE-side devices can be S-VLAN-aware or not. The MPLS network might also be S-VLAN-aware or not, which means that S-VLAN tags might or might not be sent over the MPLS cloud.

Figure 126: MPLS L2VPN Tunnel over LAG Configuration Example



The cases in which the MPLS network is S-VLAN-aware, but the CE-side device is not S-VLAN-aware, are supported because the Ethernet pseudowire operates in tagged mode. When the pseudowire is configured for raw mode, only two cases are supported: whether the CE-side device is S-VLAN-aware or not aware.

Table 102 on page 576 describes the different scenarios in which the Martini circuit shown in Figure 126 on page 576 can be deployed depending on whether the various network segments are S-VLAN-aware or not.

Table 102: Martini Circuit Scenarios Without Ethernet Raw Mode

Case number	Sending CE device (CE1)	MPLS network between local and remote routers, PE1 and PE2	Receiving CE Device (CE2)	Whether scenario is supported, when raw mode is not configured on the S-VLAN interface
1	S-VLAN-Aware	S-VLAN-Aware	S-VLAN-Aware	Supported
2	S-VLAN-Aware	S-VLAN-Unaware	S-VLAN-Aware	Unsupported
3	S-VLAN-Aware	S-VLAN-Aware	S-VLAN-Unaware	Unsupported

Table 102: Martini Circuit Scenarios Without Ethernet Raw Mode *(continued)*

Case number	Sending CE device (CE1)	MPLS network between local and remote routers, PE1 and PE2	Receiving CE Device (CE2)	Whether scenario is supported, when raw mode is not configured on the S-VLAN interface
4	S-VLAN-Aware	S-VLAN-Unaware	S-VLAN-Unaware	Unsupported
5	S-VLAN-Unaware	S-VLAN-Aware	S-VLAN-Unaware	Unsupported
6	S-VLAN-Unaware	S-VLAN-Unaware	S-VLAN-Unaware	Supported

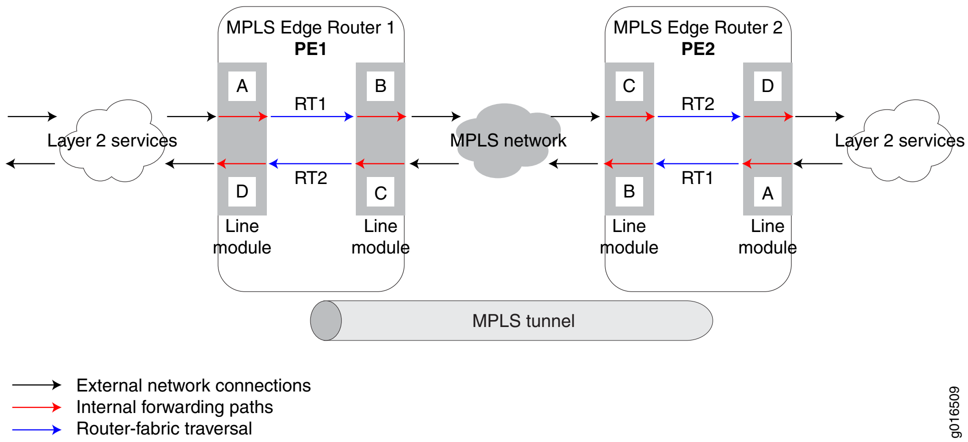
Table 103 on page 577 describes the different scenarios in which the Martini circuit configuration is supported, when Ethernet raw mode encapsulation is configured on the S-VLAN interfaces.

Table 103: Martini Circuit Scenarios with Ethernet Raw Mode

Case number	Sending CE device (CE1)	MPLS network between local and remote routers, PE1 and PE2	Receiving CE Device (CE2)	Whether scenario is supported, when raw mode is not configured on the S-VLAN interface
1	S-VLAN-Aware	S-VLAN-Aware	S-VLAN-Aware	Supported
2	S-VLAN-Aware	S-VLAN-Unaware	S-VLAN-Aware	Supported
3	S-VLAN-Aware	S-VLAN-Aware	S-VLAN-Unaware	Unsupported
4	S-VLAN-Aware	S-VLAN-Unaware	S-VLAN-Unaware	Supported
5	S-VLAN-Unaware	S-VLAN-Aware	S-VLAN-Unaware	Unsupported
6	S-VLAN-Unaware	S-VLAN-Unaware	S-VLAN-Unaware	Supported

Figure 127 on page 578 shows the transmission of Ethernet packets over a Martini circuit with ES2 4G, GE-2, GE/FE, ES2 10G, ES2 10G Uplink, and ES2 10G ADV LMs. The different processing points inside the PE-facing routers are denoted as A, B, C, and D.

Figure 127: Ethernet Packet Distribution over Martini Circuits



Consider a scenario in which Ethernet raw mode is not enabled on the S-VLAN subinterface of the PE-facing devices. When a packet reaches the S-VLAN subinterface on an ingress line module, point A, inside PE1, all packets, regardless of whether they are tagged or not, are forwarded to the subinterface on the egress line module, B, inside PE1 without any change. This behavior applies to both ES2 4G LMs, ES2 10G LMs, ES2 10G Uplink LMs, and ES2 10G ADV LMs. At point B, the MPLS encapsulation header is added to the packet and the egress line module forwards it to the MPLS network. This functionality is the same for both ES2 4G, ES2 10G, ES2 10G Uplink, and ES2 10G ADV LMs. When the packet reaches the subinterface on the ingress line module (ES2 4G LM and ES2 10G LM), point C, inside PE2, the added MPLS header is removed and the packet is sent to the subinterface on the egress line module, point D, for further processing. At point D, for ES2 4G LMs, the packet is sent to the remote CE-facing device, depending on the configuration of the S-VLAN subinterface. If the packet arrives with a single or no tag, the router adds the S-VLAN tag and sends it to the CE-facing device. On ES2 10G LMs, ES2 10G Uplink LMs, and ES2 10G ADV LMs, at point D, the packet is forwarded to the CE-facing device without any modification.

NOTE: **You cannot enable Ethernet raw mode for S-VLAN subinterfaces on the ES2 10G, ES2 10G ADV, and ES2 10G UPLINK LMs.**

In the same scenario, when Ethernet raw mode is enabled on the S-VLAN subinterface of the PE devices, the processing of Ethernet packets is performed in a slightly different way. At the S-VLAN subinterface on an ingress ES2 4G LM, point A, inside PE1, the S-VLAN tag is removed from the received packet before being forwarded to the subinterface on the egress line module, B, inside PE1. On ES2 10G LMs, ES2 10G Uplink LMs, and ES2 10G ADV LMs, at point A, the packet is forwarded to the subinterface on the egress line module at point B without any modification. At point B, the MPLS encapsulation header is added to the packet and the egress line module forwards it to the MPLS network. This functionality is the same for both ES2 4G LMs, ES2 10G LMs, ES2 10G Uplink LMs, and ES2 10G ADV LMs. When the packet reaches the subinterface on the ingress ES2 4G LMs, ES2 10G LMs, ES2 10G Uplink LMs, and ES2 10G ADV LMs, point C, inside PE2, the added MPLS header is removed and the packet is sent to the subinterface on the egress line module, point D, for further processing. This behavior is the same, regardless of whether raw mode encapsulation is enabled or not. At point D, for ES2 4G LMs, the S-VLAN tag is inserted into the packet and sent to the CE-facing device at the remote site. On ES2 10G LMs, ES2 10G Uplink LMs, and ES2 10G ADV LMs, at point D, the packet is forwarded to the CE-facing device without any modification.

Related Documentation

- Ethernet Raw Mode Encapsulation for Martini Layer 2 Transport Overview on page 540
- *mpls-relay*
- *route interface*

Examples: Configuring S-VLAN Subinterface with an Untagged C-VLAN ID

Consider the following example in which an S-VLAN interface is configured with an S-VLAN ID of 4 and the C-VLAN tag as any. Only packets that are double tagged, which contain both the S-VLAN and C-VLAN tags, are matched for this S-VLAN subinterface. Packets that are only S-VLAN tagged and do not contain a C-VLAN tag are not matched for the S-VLAN subinterface.

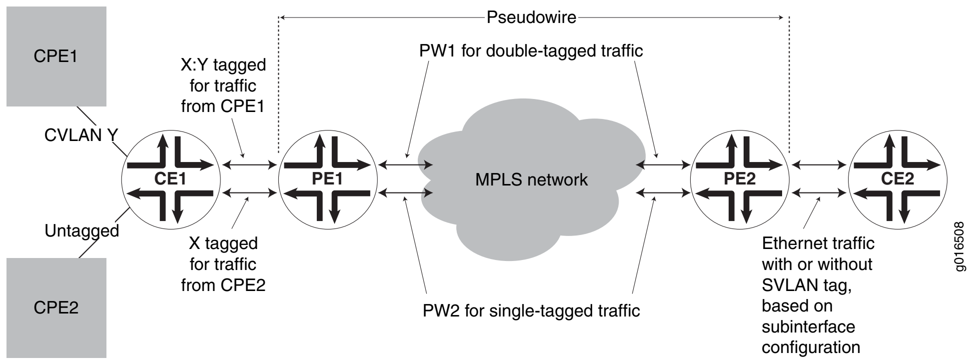
```
host1(config)#interface fastEthernet 2/0
host1(config-if)#encapsulation vlan
host1(config-if)#interface fastEthernet 2/0.1
host1(config-if)#svlan id 4 any
host1(config-if)#svlan ethertype 8100
```

The following are the limitations in this setup when the user at the CE-side tags all the packets from the CE-side, including C-VLAN tagged or untagged packets, with an S-VLAN ID to identify a particular user:

- If the S-VLAN Ethertype is 0x8100, then you must create two sub-interfaces, one for the double-tagged packets, and the other for single-tagged packets to process both these types of packets.
- If the S-VLAN Ethertype is a value other than 0x8100, S-VLAN-tagged packets without a C-VLAN ID cannot even be processed.

To enable Ethernet raw mode encapsulation and to enable the S-VLAN subinterface to remove or add the S-VLAN tag at the PE-facing routers in the first scenario, two pseudowires need to be created. Figure 128 on page 580 illustrates such a scenario in which a sending CE device and a receiving CE device are connected using PE routers through a MPLS network.

Figure 128: Martini Circuit with Two Pseudowires Between PE-Facing Routers



Two customer premise equipment (CPE)-facing interfaces send a packet each to the local CE device, CE1. The Ethernet packet is tagged with both an S-VLAN ID and a C-VLAN ID from CPE1, while the packet from CPE2 contains only the S-VLAN ID. These packets are forwarded to the local PE-router, PE1. In this case, two pseudowires are required for the two conditions. Configuring the C-VLAN ID as 0 by using the **anyUntagged** keyword in the **svlan id** *svlanIDValue* command enables both double-tagged and single-tagged traffic that matches the S-VLAN ID to pass through the S-VLAN subinterface. Using such a configuration, you need to create only one pseudowire and only one S-VLAN subinterface on PE routers.

NOTE: **On ES2 10G, ES2 10G ADV, and ES2 10G UPLINK LMs, customer-VLAN-untagged traffic matching the configured S-VLAN ID are not forwarded through the subinterface even though the C-VLAN ID is set to zero.**

If an S-VLAN subinterface is configured with Ethertype as 0x8100 and C-VLAN ID as **any**, a VLAN subinterface for the same ID shall cannot be configured, and vice-versa. For example, you cannot configure the following two commands simultaneously:

```
host1(config-if)#svlan id X any
host1(config-if)#svlan ethertype 8100
```

and

```
host1(config-if)#vlan id X
```

If you configure such a setting, the set of commands that is configured first takes precedence. Consider the following two cases with this configuration:

- Case 1: Assume that svlan id X any 8100 is configured on the subinterface. In this case, all packets with the outer tag as X and Ethertype as 8100 are matched to the sub-interface.
- Case 2: Assume that vlan id X is configured on the subinterface. In this case, only single-tagged packets with VLAN ID X and Ethertype of 8100 are matched to this subinterface. For VLAN ID of 0, the untagged traffic and the priority tagged traffic are matched the subinterface. This behavior is how it works in existing implementation.

If the S-VLAN subinterface is configured with any value other than an Ethertype 8100, both S-VLAN and VLAN interfaces for the same ID can be configured together as listed earlier. In such cases, the packets are expected to arrive with an S-VLAN ID other than 0x8100 and the C-VLAN Ethertype as 0x8100, which helps to uniquely identify the subinterface for all the following traffic patterns:

- Untagged
- C-VLAN tag only
- S-VLAN tag only
- Both S-VLAN and C-VLAN tagged

The following set of commands generates in appropriate error messages when you attempt to configure them on a VLAN major interface:

```
host1(config-if)#interface fastEthernet 1/1.1
host1(config-if)#svlan id X anyUntagged
host1(config-if)#svlan Ethertype 0x8100
host1(config-if)#interface fastEthernet 1/1.2
host1(config-if)#vlan id X
```

where **X** represents a valid V-LAN ID.

The following examples are valid configurations on a VLAN major interface:

```
host1(config-if)#interface fastEthernet 1/1.1
host1(config-if)#svlan id X anyUntagged
host1(config-if)#svlan Ethertype 0x8100
host1(config-if)#interface fastEthernet 1/1.2
host1(config-if)#vlan id Y
host1(config-if)#interface fastEthernet 1/1.3
host1(config-if)#svlan id X X
```

where the V-LAN ID value, **X**, is not the same as **Y**.

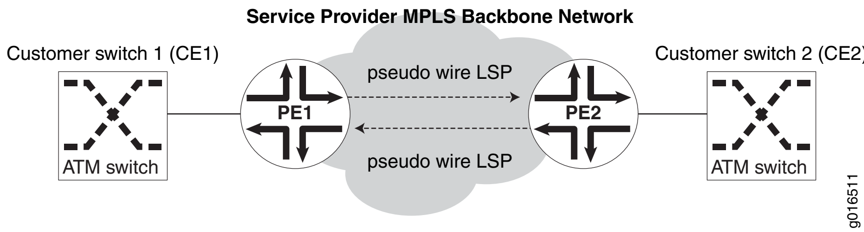
Related Documentation

- S-VLAN Subinterface with an Untagged C-VLAN ID Overview on page 542
- *svlan id*

Example: Multiple ATM Virtual Circuits over a Single Pseudowire

Figure 129 on page 582 shows a Martini circuit deployment in which the CE-side devices on either side of the network send and receive ATM cells. An MPLS tunnel that connects two E Series provider edge routers, PE1 and PE2, and ATM cross-connects provide a pseudowire between the ATM VCs on the two routers. PE2 is the remote router located at the other side of the service provider core. A customer edge ATM switch, CE1, and the provider edge router, PE1, on one side of the core are connected by an ATM port. Similarly, PE2 and CE2 are connected by an ATM port. The necessary MPLS Martini circuit configuration and VPI/VCI range configuration are added to the ATM ports on PE1 and PE2. The LDP signaling protocol performed the signaling operation and set up the pseudowire (a pair of unidirectional LSPs, from PE1 to PE2 and from PE2 to PE1) for transporting the ATM cells between CE1 and CE2.

Figure 129: Martini Circuit Deployment for Transmission of Multiple ATM VCs over a Single Pseudowire



CE1 transmits ATM cells on the ATM port connected to PE1. The transmitted cells contain VPI/VCI values that are within the range specified as part of the MPLS Martini configuration on the ATM port of PE1. If cell concatenation is configured on that ATM port of PE1, PE1 accumulates the received ATM cells. If cell concatenation is not specified, cell concatenation count is reached, or the concatenation timer expired, PE1 encapsulates the ATM cells with a control word, adds the pseudowire label and transport label, and forwards the resulting MPLS labeled packet to be delivered to PE2.

PE2 receives the MPLS-labeled packet and strips the labels on the packet. The bottommost label indicates to PE2 that the packet contains ATM cells to be forwarded on the ATM port connected to CE2. PE2 extracts the ATM cells from the packet, checks whether the VPI/VCI value on the ATM cells fall within the configured VPI/VCI range on the ATM port, and forwards those ATM cells whose VPI/VCI values fall within the configured range to CE2. PE2 discards the ATM cells that do not fall within the configured VPI/VCI range. For data traffic traversing from CE2 to CE1, the same workflow is followed with the roles reserved—PE2 as the transmitter of MPLS labeled packet and PE2 as the receiver of MPLS labeled packet.

NOTE: The support for multiple VCs over a single pseudowire uses the ATM n-to-one VCC cell transport (0x0009) pseudowire (PW) type in LDP signaling messages. This PW type is also used in the signaling messages by the ATM Martini circuit feature, which enables cell relay encapsulation on an ATM subinterface, that transports ATM cells associated with an ATM subinterface on a single pseudowire.

Because the same PW type is used for both the cell relay encapsulation for a single VC and multiple VCs, you can successfully configure an ATM subinterface Martini circuit (cell relay for a single VC) on one PE router and an ATM port Martini circuit (cell relay for multiple VCs) on another PE router, and bring up the pseudowires for this configuration. In such a scenario, traffic black holes might occur and rewriting of the ATM cell header takes place. In this topology, on the PE router configured with an ATM subinterface Martini circuit, the VPI/VCI values of all the ATM cells received on the pseudowire are rewritten to match with the configuration on the ATM subinterface. On the PE router configured with an ATM port Martini circuit, all the ATM cells received on the pseudowire are examined to determine whether they fall within the configured VPI/VCI range. If the ATM cells do not fall within the configured VPI/VCI range, they are discarded.

Related Documentation

- Multiple ATM Virtual Circuits over a Single Pseudowire Overview on page 542
- *mpls-relay atm vpi-range vci-range*
- *mpls-relay atm cell-packing mcpt-timer*

CHAPTER 10

Monitoring Layer 2 Services over MPLS

This chapter describes the commands you can use to monitor and troubleshoot layer 2 services over MPLS on E Series routers.

NOTE: The E120 and E320 Broadband Services Routers output for monitor and show commands is identical to output from other E Series routers, except that the E120 and E320 router output also includes information about the adapter identifier in the interface specifier (slot/adapter/port).

This chapter contains the following sections:

Setting Baselines for Layer 2 Services over MPLS Statistics

You can set a baseline for the statistics for all MPLS major interface statistics on the specified interface with the **baseline mpls interface** command. The router implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline when you retrieve baseline-relative statistics.

Use the **delta** keyword with the **show mpls** commands to display baselined statistics. The following statistics are maintained for each MPLS shim interface:

• receive packets and octets	• transmit packets and octets
• receive discarded packets	• transmit discarded packets
• receive error packets	• transmit error packets

To set a statistics baseline for layer 2 services over MPLS:

- Issue the **baseline mpls interface** command:

  host1#**baseline mpls interface**

Related Documentation

- *baseline mpls interface*

Monitoring ATM Martini Cell Packing Timers for Layer 2 Services over MPLS

Purpose Display the current systemwide values configured on the router for the three ATM Martini cell packing timers.

The ATM Martini cell packing timers define the time threshold that the router uses to collect and concatenate ATM cells in a single VCC cell relay–encapsulated packet.

Action To display ATM Martini cell packing timers:

```
host1(config)#show atm mcpt-timers
ATM Martini cell aggregation timers:
        Timer1: 1500microseconds
        Timer2: 2500microseconds
        Timer3: 3500microseconds
```

Meaning Table 104 on page 586 lists the **show atm mcpt-timers** command output fields.

Table 104: show atm mcpt-timers Output Fields

Field Name	Field Description
Timer1	Value in microseconds for the first ATM Martini cell packing timer
Timer2	Value in microseconds for the second ATM Martini cell packing timer
Timer3	Value in microseconds for the third ATM Martini cell packing timer

Related Documentation

- *show atm mcpt-timers*

Monitoring ATM Subinterfaces for Layer 2 Services over MPLS

Purpose Display the current state of all specified ATM subinterfaces.

Action To display the current state of all ATM subinterfaces:

```
host1#show atm subinterface
Interface   ATM-Prot VCD VPI VCI Type Encap MTU  Status Address
----------- -------- --- --- --- ---- ----- ---- ------ -------
ATM 2/0.100 ATM/MPLS 100   0 100 PVC  AAL0  9180 up     --
ATM 2/0.101 ATM/MPLS 101   0 101 PVC  AAL5  9180 up     --
ATM 2/0.200 RFC-1483 200   0 200 PVC  SNAP  9180 up     --
ATM 2/0.201 RFC-1483 201   0 201 PVC  SNAP  9180 up     --
4 interface(s) found
```


To display the current state of a specific ATM subinterface:

```
host1#show atm subinterface atm 2/0.100
                                 Circuit                                Interface
Interface    ATM-Prot VCD VPI VCI  Type   Encap MTU      Status           Type
---------    -------- --- --- --- ------- ----- ---- -------------- ---------
ATM 2/0.100 ATM/MPLS 100   0 100 PVC     AAL0  9180 lowerLayerDown Static

Maximum number of cells per packet: 100
Cell aggregation timeout timer:     2

SNMP trap link-status: disabled

InPackets:                 0
InBytes:                   0
OutPackets:                0
OutBytes:                  0
InErrors:                  0
OutErrors:                 0
InPacketDiscards:          0
InPacketsUnknownProtocol: 0
OutDiscards:               0
1 interface(s) found
```

Meaning Table 105 on page 587 lists the **show atm subinterface** command output fields; for a description of the other fields in this display, see *Monitoring ATM* in the *JunosE Link Layer Configuration Guide*.

Table 105: show atm subinterface Output Fields

Field Name	Field Description
Encap	Encapsulation type: • AAL0—VCC cell relay–encapsulated circuits that receive raw ATM cells • AAL5—ATM cross-connect interfaces
Maximum number of cells per packet	Maximum number of ATM cells that the router can concatenate in a single packet; if this value is not configured, “Martini cell aggregation: disabled” appears instead of this field. Displayed for an individual ATM over MPLS interface with AAL0 encapsulation
Cell aggregation timeout timer	Identifier (1, 2, or 3) of the ATM Martini cell packing timer that detects timeout of the cell collection threshold; if this value is not configured, “Martini cell aggregation: disabled” appears instead of this field. Displayed for an individual ATM over MPLS interface with AAL0 encapsulation

NOTE: For ATM over MPLS interfaces, the ATM-Prot field displays ATM/MPLS.

Related Documentation

- *show atm subinterface*

Monitoring ATM Cross-Connects for Layer 2 Services over MPLS

Purpose Display all ATM cross-connects (passthrough connections between local subinterfaces).

Action To display ATM cross-connects:

```
host1#show mpls cross-connects atm
              Cate Peak
VC-ID  Encap gory Rate Interface  VPI VCI Status
------ ----- ---- ---- ---------- --- --- ------
600001 AAL5  UBR     0 ATM6/0.101   0 101 Up
                       ATM6/2.101   0 101 Up
600002 AAL5  UBR     0 ATM6/0.102   0 102 Up
                       ATM6/2.102   0 102 Up
2 local connection(s) found
```

Meaning Table 106 on page 588 lists the **show mpls cross-connects atm** command output fields.

Table 106: show mpls cross-connects atm Output Fields

Field Name	Field Description
VC-ID	VC ID number of the connection
Encap	Administered encapsulation method based on what was configured with the **atm pvc** command
Category	Configured service category
Peak Rate	Send and receive peak rate, in Kbps
Interface	Specifier and status of the first subinterface that makes up the local cross-connect
VPI	Virtual path identifier of the first subinterface
VCI	Virtual channel identifier of the first subinterface
Status	Current state of the connection

Related Documentation

- *show mpls cross-connects atm*

Monitoring MPLS Forwarding for Layer 2 Services over MPLS

Purpose Display configuration and statistics for all label-switched paths (LSPs) or for specific LSPs configured on the label-switching router (LSR). The **brief** keyword displays only the action taken for each in label.

Action To display LSP configuration and statistics from the MPLS forwarding table:

```
host1:two#show mpls forwarding
serial4/1:1/1/1/1/1.1 to   222.9.1.3
  In  label 20
    0 pkts, 0 hcPkts, 0 octets
    0 hcOctets, 0 errors, 0 discardPkts
  Out label 45 on tun mpls:1 nbr 222.9.1.3
    0 pkts, 0 hcPkts, 0 octets
    0 hcOctets, 0 errors, 0 discardPkts
```

To display summary information from the MPLS forwarding table:

```
host:two#  show mpls forwarding brief
Platform label space
In Label  Owner              Action
-------- -------- ---------------------------------------------------------
16       ldp      lookup on inner header/label
17       ldp      swap to 29 on ATM5/0.1, nbr 10.10.11.5
18       ldp      swap to 30 on ATM5/0.1, nbr 10.10.11.5
19       ldp      swap to 32 on ATM5/0.1, nbr 10.10.11.5
20       ldp      swap to 34 on ATM5/0.1, nbr 10.10.11.5
21       ldp      lookup on inner header/label
22       ldp      swap to 38 on ATM5/0.1, nbr 10.10.11.5
23       ldp      swap to 40 on ATM5/0.1, nbr 10.10.11.5
24       ldp      swap to 42 on ATM5/0.1, nbr 10.10.11.5
25       ldp      lookup on inner header/label
26       ldp      swap to 46 on ATM5/0.1, nbr 10.10.11.5
27       ldp      swap to 48 on ATM5/0.1, nbr 10.10.11.5
52       ldp      l2transport to FastEthernet2/0.2
53       ldp      l2transport to FastEthernet2/0.1
L2transport
     Interface        Owner                Action
-------------------- -------- -------------------------------------------
FastEthernet2/0.1    ldp      swap to 55, push 42 on ATM5/0.1, nbr 10.10.11.5
FastEthernet2/0.2    ldp      swap to 54, push 42 on ATM5/0.1, nbr 10.10.11.5
```

The "swap to" labels 55 and 54 under the L2transport heading in the summary example are VC labels received from the other router. The label that is pushed in this case, 42, is for the base tunnel.

Meaning Table 107 on page 589 lists the **show mpls forwarding** command output fields.

Table 107: show mpls forwarding Output Fields

Field Name	Field Description
name/id	Interface specifier
destination	Destination IP address
In label	Label sent to upstream neighbor for route
Out label	Label received from downstream neighbor for route
pkts	Number of packets sent across tunnel
hcPkts	Number of high-capacity (64-bit) packets sent across tunnel

Table 107: show mpls forwarding Output Fields *(continued)*

Field Name	Field Description
octets	Number of octets sent across tunnel
hcOctets	Number of high-capacity (64-bit) octets sent across tunnel
errors	Number of packets dropped for some reason before being sent
discardPkts	Number of packets discarded due to lack of buffer space before being sent

Related Documentation

- *show mpls forwarding*

Monitoring MPLS Layer 2 Interfaces for Layer 2 Services over MPLS

Purpose Display status and configuration information about MPLS layer 2 major, minor, and shim interfaces. Both the **show mpls interface shim** command and the **show mpls l2transport interface** command provide the same output. The **shim** keyword displays all shim interfaces. The **brief** keyword displays only limited interface information.

Action To display information about MPLS layer 2 interfaces:

NOTE: If you enable scheduler profile–based computation of service session accounting by using the **service-accounting- statistics scheduler-based** command, the forwarded packets and bytes fields, and the dropped packets and bytes fields are displayed in the rate-limit-profile section under the MPLS policy output heading. The committed, conformed, exceeded, saturated, and unconditional packets and bytes fields are not displayed in the rate-limit-profile section in the output of the command. These fields are displayed instead of the forwarded packets and bytes fields, and the dropped packets and bytes fields only if you disable scheduler profile-based computation of service session accounting.

```
host1#show mpls interface shim
MPLS shim interface FastEthernet2/0
  Remote PE address is 10.9.1.3
  Virtual circuit ID is 1
  Group ID is 0 by default
  Control word is not preferred by default
  Don't send sequence numbers by default
  Relay format is ethernet by default
  ATM VPI/VCI range is 1-2/100-109 (up on the line card)
  ATM VPI/VCI range is 5-10/200-300 (up on the line card)
  ATM cell aggregation is enabled
  ATM cell aggregation maximum cells per packet is 20
  ATM cell aggregation timeout timerId is 2
  Administrative state is enabled
```



```
  Operational state is down (shim interface does not have a next-hop)
  Operational MTU is 1500
  Received:
    0 packets
    0 bytes
    1 error
    0 discards
  Sent:
    0 packets
    0 bytes
    0 errors
    0 discards
received mtu 0

queue 0: traffic class best-effort, bound to ethernet FastEthernet2/0
    Queue length 0 bytes
    Forwarded packets 0, bytes 0
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0

MPLS policy input shimRl
    classifier-group *
      0 packets, 0 bytes
      rate-limit-profile shimRl
        committed: 0 packets, 0 bytes
        conformed: 0 packets, 0 bytes
        exceeded:  0 packets, 0 bytes
  MPLS policy output shimRl
    classifier-group *
      0 packets, 0 bytes
      rate-limit-profile shimRl
        committed: 0 packets, 0 bytes
        conformed: 0 packets, 0 bytes
        exceeded:  0 packets, 0 bytes
```

This excerpt from the command output shows the label information displayed when a circuit is up

host1#**show mpls l2transport interface**

```
...
  Out Label 49 on  tun mpls:lsp-de090100-24-37
    0 pkts, 0 hcPkts, 0 octets
    0 hcOctets, 0 errors, 0 discardPkts
    queue 0: traffic class best-effort, bound to atm-vc ATM1/0.1
      Queue length 0 bytes
      Forwarded packets 0, bytes 0
      Dropped committed packets 0, bytes 0
      Dropped conformed packets 0, bytes 0
      Dropped exceeded packets 0, bytes 0
...
```

To display summary information about MPLS shim interfaces:

host1#**show mpls interface shim brief**

```
                    Remote-PE   Virtual     Load
                       or       Circuit   Balancing    Admin    Oper
    Interface       LSP-name      ID        Group      state    state
------------------  ---------   -------   ---------   -------   -----
FastEthernet2/0.1   222.9.1.3   200001    -           enabled   up
FastEthernet2/0.2   222.9.1.3   200002    -           enabled   up
```


Meaning Table 108 on page 592 lists the **show mpls interface** and **show mpls l2transport interface** command output fields.

Table 108: show mpls interface and show mpls l2transport interface Output Fields

Field Name	Field Description
MPLS shim interface	Interface specifier
Remote PE address	Address of the remote PE router for the layer 2 circuit
Virtual circuit ID	VC ID number for the interface
Group ID	Group ID number for the interface
Control word	Configuration of the control word
Sequence number	Statement regarding configuration of sequence number
Relay format	Configuration of relay format
ATM VPI/VCI range	Starting and ending (inclusive) virtual path identifiers and virtual circuit identifiers of the reserved VC range and whether the range specification is activated on the line module for transportation of ATM cells that belong to multiple VCs over a single pseudowire.
ATM cell aggregation	Concatenation of multiple ATM cells to be sent in a single MPLS-labeled packet for an ATM port, enabled or disabled.
ATM cell aggregation maximum cells per packet	Maximum number of ATM cells that the router can concatenate in a single packet; if this value is not configured, "ATM cell aggregation: disabled" appears instead of this field. Displayed for an ATM port (ATM AAL5 over ATM major interface).
ATM cell aggregation timeout timerID	Identifier (1, 2, or 3) of the ATM Martini cell packing timer that detects timeout of the cell collection threshold; if this value is not configured, "ATM cell aggregation: disabled" appears instead of this field. Displayed for an ATM port (ATM AAL5 over ATM major interface).
Administrative state	Administrative state, enabled or disabled
Operational state	Statement regarding operational state of interface
Operational MTU	Maximum transmission unit for the interface
Received, Sent	Statistics for MPLS traffic received or sent on the interface
packets	Number of packets received or sent
bytes	Number of bytes received or sent

Table 108: show mpls interface and show mpls l2transport interface Output Fields *(continued)*

Field Name	Field Description
error	Number of packets that are dropped for some reason at receipt or before being sent
discards	Number of packets that are discarded because of lack of buffer space at receipt or before being sent
received mtu	MTU specified in received packets
queue, traffic class, bound to	Queue and traffic class bound to the specified interface
Queue length	Number of bytes in the queue
Forwarded packets, bytes	Total number of packets and bytes forwarded by this interface
Dropped committed packets, bytes	Total number of committed packets and bytes dropped by this interface
Dropped conformed packets, bytes	Total number of conformed packets and bytes dropped by this interface
Dropped exceeded packets, bytes	Total number of exceeded packets and bytes dropped by this interface
MPLS policy	Type (input, output) and name of policy
classifier-group entry	Entry index
packets, bytes	Number of packets and bytes on the interface
rate-limit-profile	Name of profile
Committed	Number of packets and bytes that conform to the committed access rate
Conformed	Number of packets and bytes that exceed the committed access rate but conform to the peak access rate
Exceeded	Number of packets and bytes that exceed the peak access rate
In label	VC label sent by this router to upstream neighbor for route
Out label	VC label received by this router from downstream neighbor for route
MPLS statistics	MPLS statistics for traffic received or sent
pkts	Number of packets received or sent

Table 108: show mpls interface and show mpls l2transport interface Output Fields *(continued)*

Field Name	Field Description
hcPkts	Number of high-capacity (64-bit) packets received or sent
octets	Number of octets received or sent
hcOctets	Number of high-capacity (64-bit) octets received or sent
errors	Number of packets that are dropped for some reason at receipt or before being sent
discardPkts	Number of packets that are discarded because of lack of buffer space at receipt or before being sent
Interface	Interface specifier
Remote-PE or LSP-name	IP address of the remote PE router or name of the tunnel
Virtual Circuit ID	VC ID number for the interface
Load Balancing Group	Load-balancing group associated with the layer 2 Martini transport circuit
Admin state	Administrative state of the interface, enabled or disabled
Oper state	Operational state of the interface, up or down

Related Documentation

- *show mpls interface*
- *show mpls l2transport interface*

PART 4

Virtual Private LAN Service

CHAPTER 11

VPLS Overview

This chapter describes the virtual private LAN service (VPLS), and contains the following sections:

VPLS Protocol Overview

JunosE Software enables you to configure one or more instances of VPLS, referred to as VPLS instances, on the router. VPLS employs an Ethernet-based layer 2 VPN to connect multiple individual LANs across a service provider's MPLS core network. The geographically dispersed multiple LANs function as a single virtual LAN. VPLS provides a point-to-multipoint capability for traffic forwarding. In contrast, L2VPNs that enable a virtual private wire service (VPWS) provide only a point-to-point traffic forwarding capability.

VPLS preserves the broadcast and multicast capabilities of the physical LANs. Consequently, any broadcast or multicast traffic from a given customer end station is sent to all sites that participate in the VPLS instance.

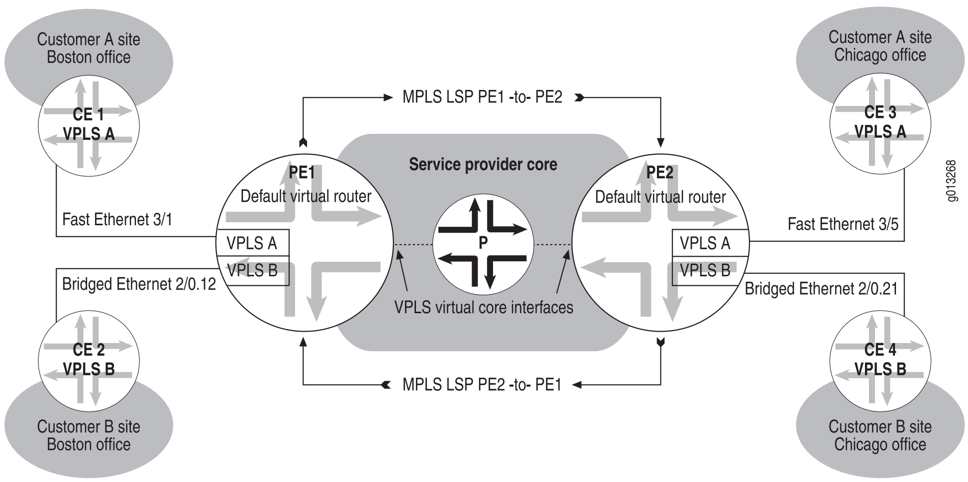
You can use either BGP or LDP to provide signaling for VPLS, as follows:

- BGP signaling—VPLS with BGP signaling, which is referred to as BGP-based VPLS, uses BGP as the protocol that signals reachability for the VPLS domain in which the VPLS instance participates. You must configure BGP on each provider edge (PE) router in your topology to provide signaling for each VPLS domain.

- LDP signaling—VPLS with LDP signaling, which is referred to as LDP-based VPLS, uses LDP as the protocol that signals reachability for the VPLS domain in which the VPLS instance participates. You must configure LDP on each PE router in your topology to provide signaling for each VPLS domain.

Figure 130 on page 598 illustrates an example of a simple VPLS topology. The basic topology of a VPLS network is the same regardless of whether BGP signaling or LDP signaling is used.

Figure 130: VPLS Sample Topology



See "Configuring BGP-MPLS Applications" on page 389 for information about configuring BGP/MPLS VPNs. See "Configuring VPWS" on page 677 for information about configuring VPWS.

Related Documentation

- BGP Signaling for VPLS Overview on page 603
- Configuring VPLS with BGP Signaling on a PE Router on page 614
- LDP Signaling for VPLS Overview on page 604
- Configuring VPLS with LDP Signaling on a PE Router on page 626

VPLS Components Overview

As illustrated in Figure 130 on page 598, a typical VPLS topology consists of the following components.

- VPLS Domains on page 598
- Customer Edge Devices on page 599
- VPLS Edge Devices on page 599

VPLS Domains

Typically, a *VPLS domain* is associated with customers who want to use Ethernet-based layer 2 VPNs to connect geographically dispersed sites in their organization across an

MPLS-based service provider core, also known as an MPLS backbone. Each VPLS domain consists of the set of PE routers running the corresponding VPLS instance that participates in that domain. In BGP-signaled VPLS, a VPLS domain is identified by the route target extended community, similarly to how a layer3 VPN domain is identified in layer 3VPNs.

Figure 130 on page 598depicts two VPLS domains: VPLS A and VPLS B. The VPLS A domain connects Customer A's Boston and Chicago offices, and consists of provider edge routers PE 1 and PE 2, each of which runs a VPLS instance named vplsA. Similarly, the VPLS B domain connects Customer B's Boston and Chicago offices, and consists of provider edge routers PE 1 and PE 2, each of which also runs a VPLS instance named vplsB.

Customer Edge Devices

Figure 130 on page 598 shows four customer edge (CE) devices: CE 1, CE 2, CE 3, and CE 4. Each CE device is located at the edge of a customer site, and participates in one or more VPLS domains. In the sample topology, CE 1 and CE 3 are members of the VPLS A domain, and CE 2 and CE 4 are members of the VPLS B domain.

A CE device can be a single host, a switch, or, most typically, a router. Each CE device is directly connected to a VPLS edge router by means of an Ethernet or bridged Ethernet network interface, but does not run VPLS. From the perspective of the CE device, the entire VPLS network appears to be a single layer 2 switch that can switch layer 2 packets, learn and filter on media access control (MAC) addresses, and flood packets that have unknown MAC destination addresses (DAs).

VPLS Edge Devices

In a VPLS configuration, E Series PE routers host VPLS edge (VE) devices, which are also referred to as VE routers or, simply, VEs. A VE device is a VPLS instance that services a particular customer site.

Figure 130 on page 598depicts two PE routers: PE 1, which is the local router, and PE 2, which is the remote router located at the other side of the service provider core. Each PE router must have a VPLS instance—the VE device—configured for each VPLS domain in which it participates. Consequently, the sample topology comprises a total of four separate VPLS instances: instances vplsA and vplsB configured on PE 1, and instances vplsA and vplsB configured with matching route target values on PE 2.

Each VPLS instance configured on the router is associated with two types of interfaces, also known as ports. The CE-facing interface is an Ethernet or bridged Ethernet network interface that directly connects the PE router to each CE device. The VPLS virtual core interface, although not an actual physical interface, is automatically generated by the router for each VPLS instance and represents all of the MPLS tunnels from the router to the remote PE devices. The router encapsulates Ethernet frames from the CE device in an MPLS packet and then forwards the encapsulated frames to the service provider core through the provider (P) router. This encapsulation is identical to Martini encapsulation for Ethernet layer 2 services over MPLS.

Each PE router in the sample topology has a total of two network interfaces and two VPLS virtual core interfaces configured, one interface of each type per VPLS instance.

Related Documentation

- VPLS Protocol Overview on page 597

VPLS and Transparent Bridging Overview

A single VPLS instance is analogous to a distributed learning bridge (also known as a bridge group) used for transparent bridging, and performs similar functions. In effect, a VPLS instance is a new or existing bridge group that has additional VPLS attributes configured.

A bridge group is a collection of bridge interfaces stacked on Ethernet layer 2 interfaces to form a broadcast domain. Similarly, a VPLS instance is a collection of network interfaces stacked on Ethernet layer 2 interfaces that transmits packets between the router, or VE device, and the CE device located at the edge of the customer's network. In addition, the VPLS virtual core interface enables a VPLS instance to forward traffic not only between bridge interfaces, like a bridge group, but also between a bridge (network) interface and the service provider core.

Like a bridge group, each VPLS instance maintains its own set of forwarding tables and filters that enables it to learn the network topology by examining the media access control (MAC) source address of every incoming packet. The VPLS instance then creates an entry in its forwarding table that includes the MAC address and associated network interface where the packet was received. For traffic on the VPLS virtual core interface, the VPLS instance captures additional information that includes an outgoing MPLS label used to reach the remote site and an incoming MPLS label used to process traffic received from the remote site.

Table 109 on page 600 through Table 112 on page 601 represent the forwarding tables on PE 1 and PE 2 for the sample VPLS topology illustrated in Figure 130 on page 598.

Table 109: VPLS Forwarding Table on PE 1 for VPLS A

Interface	MAC Address	Outgoing Label	Received Label
Fast Ethernet 3/1	1a1a.1a1a.1a1a	–	–
VPLS virtual core interface	3a3a.3a3a.3a3a	18	324

Table 110: VPLS Forwarding Table on PE 1 for VPLS B

Interface	MAC Address	Outgoing Label	Received Label
Bridged Ethernet 2/0.12	2b2b.2b2b.2b2b	–	–
VPLS virtual core interface	4b4b.4b4b.4b4b	25	526

Table 111: VPLS Forwarding Table on PE 2 for VPLS A

Interface	MAC Address	Outgoing Label	Received Label
Fast Ethernet 3/5	3a3a.3a3a.3a3a	–	–

Table 111: VPLS Forwarding Table on PE 2 for VPLS A *(continued)*

Interface	MAC Address	Outgoing Label	Received Label
VPLS virtual core interface	1a1a.1a1a.1a1a	42	107

Table 112: VPLS Forwarding Table on PE 2 for VPLS B

Interface	MAC Address	Outgoing Label	Received Label
Bridged Ethernet 2/0.21	4b4b.4b4b.4b4b	–	–
VPLS virtual core interface	2b2b.2b2b.2b2b	63	872

Related Documentation

- VPLS Protocol Overview on page 597

Subscriber Policies for VPLS Network Interfaces Overview

The router associates a VPLS network interface, as it does a bridge group interface, with a default subscriber policy that enables intelligent flooding of packets within a VPLS domain. This section describes how subscriber policies work and explains some important considerations when you use subscriber policies for VPLS instances. The requirements and procedures for subscriber policies are the same whether you employ BGP or LDP signaling for VPLS.

Network Interface Types

VPLS instances, like bridge groups, support two types of network interfaces:

- Subscriber (client)—A subscriber (client) interface is downstream from the traffic flow; that is, the traffic flow direction is from the server (trunk) to the client (subscriber). This is the default network interface type for both VPLS instances and bridge groups.
- Trunk (server)—A trunk (server) interface is upstream from the traffic flow; that is, the traffic flow direction is from the client (subscriber) to the server (trunk). To configure a trunk interface, you must specify the **subscriber-trunk** keyword as part of the **bridge-group** command. The VPLS virtual core interface always acts as a trunk interface, and cannot be configured as a subscriber interface.

Default Subscriber Policies

Each network interface is associated with a default subscriber policy for that interface type. The subscriber policy is a set of forwarding and filtering rules that defines how the specified interface handles various packet or attribute types, as follows:

- For each packet type listed in Table 113 on page 602, the subscriber policy specifies whether the network interface permits (forwards) or denies (filters or drops) packets of that type.
- For the relearn attribute, the subscriber policy specifies whether the network interface can relearn a MAC address entry on a different interface from the one initially associated with this entry in the forwarding table. Permit indicates that relearning is allowed; deny indicates that relearning is prohibited.

Table 113 on page 602 lists the default values for each packet or attribute type defined in the policies for subscriber interfaces and trunk interfaces. The default subscriber policy differs in one way from the default trunk policy: broadcast packets and packets with unknown unicast destination addresses (DAs) are denied in the subscriber policy and permitted in the trunk policy.

Table 113: Default Subscriber Policies for VPLS Network Interfaces

Packet/Attribute Type	Default Subscriber Policy	Default Trunk Policy
ARP	Permit	Permit
Broadcast	Deny	Permit
IP	Permit	Permit
MPLS	Permit	Permit
Multicast	Permit	Permit
PPPoE	Permit	Permit
Relearn	Permit	Permit
Unicast (user-to-user)	Permit	Permit
Unknown unicast DA	Deny	Permit
Unknown protocol	Permit	Permit

Modifying Subscriber Policies

For a network interface configured as a subscriber (client) interface, you can modify the default subscriber policy to change the default permit or deny value for one or more of the packet or attribute types listed in Table 113 on page 602.

You cannot, however, change the default trunk policy for a network interface configured as a trunk interface or for the VPLS virtual core interface. Trunk interfaces and the VPLS virtual core interface always use the default trunk policy, which forwards packets of all types and permits relearning.

Table 114 on page 603 lists the commands that you can use to modify subscriber policies for subscriber (client) interfaces associated with either a VPLS instance or a standard bridge group.

Table 114: Commands to Configure Subscriber Policies

arp	pppoe
bridge subscriber-policy	relearn
broadcast	subscriber-policy
ip	unicast
mpls	unknown-destination
multicast	unknown-protocol

Considerations for VPLS Network Interfaces

When you configure network interfaces for a VPLS instance, you must ensure that the subscriber policy in effect for the interface is appropriate for your network configuration.

To ensure that the network interface permits relearning and forwards (permits) packets for all of the protocol types listed in Table 113 on page 602, be sure to configure the network interface as a trunk (server) interface so that it always uses the default trunk policy. For example, the following commands associate a 10-Gigabit Ethernet interface with a VPLS instance named vplsBoston, and configure the interface as a trunk.

```
host1(config)#interface tenGigabitEthernet 4/0/1
host1(config-if)#bridge-group vplsBoston subscriber-trunk
```

If you configure a VPLS network interface as a subscriber (client) interface, use care if you modify the default subscriber policy in effect for that interface. For example, if you use the **arp** command to change the default value for ARP packets from permit (forward) to deny (filter or drop), make sure you also use the **bridge address** command to add the appropriate static (nonlearned) ARP entry to the forwarding table. If an ARP entry expires from the forwarding table and the subscriber policy is configured to deny ARP packets, the router cannot properly forward subsequent ARP packets.

For information about using these commands, see *Configuring Secure Policies* in the *JunosE Link Layer Configuration Guide*.

Related Documentation

- Configuring VPLS with BGP Signaling on a PE Router on page 614
- Configuring VPLS with LDP Signaling on a PE Router on page 626

BGP Signaling for VPLS Overview

BGP multiprotocol extensions (MP-BGP) enable BGP to support IPv4 services such as BGP/MPLS VPNs, which are sometimes known as RFC 2547bis VPNs. VPLS with BGP signaling is actually a BGP-MPLS application that has much in common with BGP/MPLS VPNs.

The procedures for configuring BGP signaling for BGP/MPLS VPNs and for VPLS are similar except, for VPLS, you must configure both of the following BGP address families:

- L2VPN—The L2VPN address family enables you to configure the PE router for VPLS (or VPWS) L2VPNs to exchange layer 2 network layer reachability information (NLRI) for all VPLS (or VPWS) instances. Optionally, you can use the **signaling** keyword with the **address-family** command for the L2VPN address family to specify BGP signaling of VPLS reachability information. Currently, you can omit the **signaling** keyword with no adverse effects.
- VPLS—The VPLS address family enables you to configure the PE router to exchange layer 2 NLRI for a specified VPLS instance.

BGP can exchange information in a VPLS topology within these address families. Specifically, BGP builds a full mesh of label-switched paths (LSPs) among all of the VPLS instances on each of the PE routers participating in a particular VPLS domain.

See "Configuring BGP Routing" on page 3 for information about configuring BGP routing. See "Configuring BGP-MPLS Applications" on page 389 for information about configuring BGP/MPLS VPNs.

Related Documentation

- Configuring BGP Signaling for VPLS on page 620

LDP Signaling for VPLS Overview

When you configure VPLS with LDP signaling, LDP supports a full mesh of pseudowires among the participating PE routers. This is analogous to BGP signaling, in which BGP builds a full mesh of label-switched paths (LSPs) among all of the VPLS instances on each of the PE routers participating in a particular VPLS domain.

Targeted Sessions

LDP establishes targeted sessions to the remote PEs configured at the edge of the service provider's MPLS core network. The number of targeted sessions supported for a local PE router is equal to the total number of other PE routers that participate in the VPLS instances configured on the local PE router. As is the case with Martini encapsulation for Ethernet layer 2 services over MPLS, a targeted session to a remote PE router can have many pseudowires that terminate at the same remote PE router.

To enable LDP to establish targeted sessions with remote PEs across the MPLS core, you must issue both the **mpls ldp vpls-id** command to configure a VPLS identifier for the VPLS instance, and the **mpls ldp vpls neighbor** command to configure a list of neighbor (peer) addresses to which LDP can send or from which LDP can receive targeted hello messages.

PWid FEC Element TLV

LDP signaling information for VPLS is carried in a label mapping message. The label mapping message contains the Generic Label type-length-value (TLV), and the pseudowire identifier (PWid) forwarding equivalence class (FEC) element. A FEC is a group of IP packets forwarded over the same path with the same path attributes applied.

The PWid FEC element (FEC Type 128 or 0x80) contains the VPLS identifier information configured for your VPLS instance with the **mpls ldp vpls-id** command. Taken together, the pseudowire type field and the PWid field in the TLV represent a unique VPLS instance. The pseudowire type field is Ethernet to identify the pseudowires that carry Ethernet traffic for multipoint connectivity between the local and remote VEs. The PWid field is a nonzero 32-bit integer that contains the VPLS identifier, which is a globally unique identifier for a VPLS domain. All VEs that participate in the same VPLS domain must use the same VPLS identifier.

Martini encapsulation for Ethernet layer 2 services over MPLS also uses the PWid FEC Element TLV. As a result, the PWid for Martini configurations must not be the same as the VPLS identifier configured for a VPLS instance. To prevent this conflict from occurring, the JunosE Software displays an error and rejects the configuration if you attempt to configure the same value for the Martini PWid and the VPLS identifier.

See "Configuring MPLS" on page 279 for information about configuring MPLS.

Related Documentation

- Configuring LDP Signaling for VPLS on page 627

BGP Multihoming for VPLS Overview

BGP multihoming enables you to connect a customer site to two or more PE routers to provide redundant connectivity while preventing the formation of layer 2 loops in the service provider's network. The redundant connectivity maintains the VPLS service and traffic forwarding to and from the multihomed site in the event of a PE router-to-CE device link failure, the failure of a PE router, or an MPLS reachability failure between the local PE router and a remote PE router. A redundant PE router can begin providing service to the customer site as soon as the failure is detected. BGP multihoming is very similar for both VPLS and VPWS, with only minor differences in behavior between the two L2VPN types.

When a CE device connects to multiple PE routers, each of these routers advertises reachability for the multihomed site—routes that have the same site ID in the layer2 NLRI. The other PE routers in the network use a BGP path selection process to select only one of the advertising routers to which they send traffic destined for the CE device. This path selection process eliminates layer 2 loops in the VPLS network.

BEST PRACTICE: To prevent the creation of layer 2 loops due to a misconfiguration or temporary loops during a topology change and subsequent convergence, we recommend that you employ the Spanning Tree Protocol (STP) on your CE devices.

You specify on each PE router connected to the CE device in the VPLS that the site is multihomed and you configure a priority. The priority serves as a site preference and is propagated by BGP in the local-preference attribute.

You configure the same site ID (sometimes referred to as a VE ID) on these connected PE routers. Each of these routers then advertises reachability for the multihomed site; the VPLS NLRI contains the site ID. The site ID shared by the connected PE routers should be different than the site IDs configured on the remote PE routers in the VPLS network; if the site ID is not different, then the pseudowire will be in a site collision state. The remote routers then use the site ID to identify where to forward traffic destined for the customer site.

Although the site ID is the same for all connected PE routers, the block offset, label range, and route distinguisher can be different for each PE router. The BGP path selection process uses the block offset and label range only to determine whether a layer 2 advertisement is relevant to the multihomed customer site. A route distinguisher is helpful to uniquely identify a particular PE router when you are troubleshooting a network.

The PE routers run the BGP path selection process on the locally originated and received layer 2 route advertisements to establish that the routes are suitable for advertisements to other peers, such as route reflectors. For this selection process, the routes advertise different prefixes, distinguished by the site ID, block offset, and route distinguisher.

The remote PE routers then run a modified selection process on these selected routes for L2VPN multihoming. Because all the prefixes advertised by multihomed local PE routers share the same site ID, the set of routes advertised for a multihomed site effectively consists of multiple routes to a single prefix, distinguished by the site ID alone. Therefore the result of the second selection process is the single best path to the multihomed site.

The PE router that originates this advertisement then becomes the designated VE device for the multihomed customer site. When the designated VE device is determined for both the local and remote customer sites for the VPLS, then a VPLS pseudowire is created between the designated VE devices.

The BGP best path selection process is run only in the core VPN address family. This first selection process does not consider the down bit for VPLS (or the status vector bit for VPWS).

The layer 2 multihoming decision process is run only in the non-core VPLS (or VPWS) layer 2 unicast address families. This second decision process treats prefixes with the same site ID but different RDs as a single prefix.

When the PE router receives a layer 2 BGP advertisement that has the down bit set, inbound policy sets the local preference attribute to zero. The selection process can then choose an existing route from an alternate PE router, if available.

When a PE router in a VPLS domain is also a BGP route reflector (RR), the path selection process to determine the VE device for the multihomed site has no effect on the path selection process performed by this PE router for the purpose of reflecting layer 2 routes.

Layer 2 prefixes that have different route distinguishers are considered to have different NLRI for route reflection. This result of the standard BGP path selection process enables the RR to reflect all routes that have different route distinguishers to all other RR clients even though only one of these routes is used to trigger the VPLS pseudowire to the multihomed site.

- Designated VE Device Selection for a Multihomed Site on page 607
- Multihoming Reaction to Failures in the Network on page 609

Designated VE Device Selection for a Multihomed Site

BGP on each PE router in the VPLS network determines the best path to the multihomed site by comparing path attributes. The PE routers receiving the advertised routes first run the standard BGP selection process. The routes from the connected multihomed PE routers all share the same site ID, but can have different route distinguishers and block offsets; the routers are advertising different prefixes. The following sequence is applied to all routes on a per-prefix basis:

1. Select a path with a reachable *next hop*.
2. Select the path with the highest *weight*.
3. If path weights are the same, select the path with the highest *local preference* value.
4. Prefer locally originated routes (network routes, redistributed routes, or aggregated routes) over received routes.
5. Select the route with the shortest *AS-path* length.
6. If all paths have the same AS-path length, select the path based on *origin*: IGP is preferred over EGP; EGP is preferred over Incomplete.
7. If the origins are the same, select the path with lowest *MED* value.
8. If the paths have the same MED values, select the path learned by means of EBGP over one learned by means of IBGP.
9. Select the path with the lowest IGP cost to the next hop.
10. Select the path with the shortest route reflection cluster list. Routes without a cluster list are treated as having a cluster list of length 0.
11. Select the path received from the peer with the lowest BGP router ID.
12. Select the path that was learned from the neighbor with the lowest peer remote address.
13. Select the path with a lower route distinguisher.

The result of this process is the best path to the multihomed customer site through each PE router connected to the site. One best path is selected for each router. The process establishes whether the route advertised by each PE router is suitable for advertising to peer routers.

Next, BGP runs the layer 2 multihoming selection process on this set of best paths to determine the one best path to the customer site. The result of this process establishes that the best path is suitable for establishing a pseudowire from the remote PE router to the PE router. That PE router is accordingly selected as the designated VE device.

The multihoming selection process is similar to the standard BGP process, but it omits the following two steps:

- The process does not prefer locally originated routes. Local origination is of no value in establishing the designated VE device. The PE routers connected to the customer site always have a local route and therefore all advertise a locally originated route. These PE routers also receive the advertisements from the other connected PE routers. If the multihoming selection process preferred local origination, each of these routers would select itself as the best path.
- The process does not take into account IGP cost in order to prevent the remote PE routers from selecting different designated VE devices in the event of a misconfiguration, such as having the same site priority on different multihomed PE routers.

When the remote PE router establishes or refreshes a pseudowire to the local PE router, it verifies whether the prefix is in the range required for the site ID based on the block offset and label range advertised by the designated VE device. If the prefix is out of range, then the pseudowire status is set to OR (out of range).

One of the following cases applies for each PE router when it completes the BGP path selection process for a layer 2 advertisement on the VPLS.

- The PE router originated one of the multihomed advertisements and selected its own advertisement as the best path.

  This PE router hosts the designated VE device. Selection as the designated VE device triggers the creation of pseudowires to and from the other PE routers in the VPLS. When the remote customer site is also multihomed, then the designated VE device triggers the creation of pseudowires to and from only the designated VE device for the remote site.
- The PE router originated one of the multihomed advertisements but did not select its own advertisement as the best path.

  This PE router is one of the redundant PE routers for the multihomed site; it does not host the designated VE device. If its status has just transitioned from being the designated VE device, then the PE router tears down all the pseudowires that it had to and from the other PE routers in the VPLS network.
- The PE router receives the multihomed advertisements and selects a best path; it does not originate any of these advertisements because it is not connected to the multihomed customer site.

  If the selected best path—and therefore the designated VE—has not changed, then nothing happens. If the best path has changed, then this PE router brings up pseudowires to and from the new designated VE device and tears down the pseudowires to and from the previous designated VE device.

  If this PE router does not select a best path after running the process, then the local PE router does not consider the remote site to exist.

When a VE device receives an advertisement for a layer 2 NLRI that matches its own site ID but the site is not multihomed, then the pseudowire between it and the transmitting PE router transitions to a site collision (SC) state and is not considered to be up.

Multihoming Reaction to Failures in the Network

The redundant connectivity provided by a multihoming configuration protects against several types of network failure.

- CE-Link failure between the CE device and the PE router—BGP on the PE router is notified when the circuit goes down. BGP then modifies the circuit status vector bit in the MP_REACH_NLRI to indicate that the circuit is down.

  If all VPLS local attachment circuits are down, then BGP modifies the down bit in the VPLS advertisement Layer2-Extended-Community to state that the site is down. When the bit is modified, BGP advertises the route to all remote PE routers to inform them that the circuit (and site) is down. The remote PE routers each run the best path selection process again and adjust the VPLS pseudowires as needed.

- Failure of MPLS reachability to the remote PE router—BGP on the PE router is notified that MPLS connectivity to the BGP next hop is gone. BGP then modifies the circuit status vector bit in the MP_REACH_NLRI to indicate that the LSP is down. When the bit is modified, BGP advertises the route to all remote PE routers to inform them that connectivity is down from the local site to the remote site.

  The down bit is set if no remote PE router is reachable by MPLS. This enables the remote PE routers to consider the other multihomed PE router as the designated VE device for the multihomed-site.

  The remote PE routers each run the best path selection process again and adjust the VPLS pseudowires as needed.

- PE router failure—When either the PE router or its BGP process fails, peer PE routers detect expiration of the holdtimer and bring down their peering sessions, and remove layer 2 advertisements from the PE router. Alternatively, the PE routers can detect unreachability to the BGP next hop that represents the failed PE router. In this case the peer routers mark the layer 2 routes advertised by PE router as unreachable. The peer PE routers each run the best path selection process again and adjust the VPLS pseudowires as needed.

  A similar response results when you adjust the multihoming priority of the PE routers connected to the multihomed site, effectively performing and administrative failover to another PE router. BGP sends a layer 2 update with the new local preference attribute to all peer PE routers. The peer PE routers each run the best path selection process again and adjust the VPLS pseudowires as needed.

To modify their pseudowires, the peer routers correct their MPLS forwarding tables and set up new entries in their pseudowire tables.

Related Documentation

- VPLS Protocol Overview on page 597
- Configuring BGP Multihoming for VPLS on page 616

VPLS Supported Features

The JunosE implementation of VPLS provides the following features:

- Single-level VPLS hierarchy within a single autonomous system (AS) using MPLS tunneling technology for the core
- Support for the following types of network interfaces between the PE router and the CE device:
  - Bridged Ethernet over ATM1483 subinterfaces
  - Fast Ethernet
  - Gigabit Ethernet
  - 10-Gigabit Ethernet
  - VLAN and S-VLAN subinterfaces over bridged Ethernet, Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet interfaces
- Autodiscovery of VPLS instance members using MP-BGP
- VPLS signaling using MP-BGP to set up and tear down the pseudowires that constitute a VPLS instance
- VPLS signaling using LDP and the PWid FEC element (FEC Type 128) to set up and tear down the pseudowires that constitute a VPLS instance
- Interworking of the VPLS instance and the VPN routing and forwarding instance (VRF) using an external cable connection
- Multihoming
- Class of service (CoS)
- Inter-AS option A, inter-AS option B, and inter-AS option C services
- Minimal filtering and policing support

As with L2VPNs, VPLS does not support BGP multipaths.

Related Documentation

- VPLS Protocol Overview on page 597
- VPLS Platform Considerations on page 610
- VPLS References on page 611

VPLS Platform Considerations

VPLS is supported on all E Series routers.

- Module Requirements on page 610
- Interface Specifiers on page 611

Module Requirements

You can configure VPLS network interfaces on all E Series module combinations that support transparent bridging. The VPLS virtual core interface can exist on all E Series module combinations that support MPLS tunnels.

For information about the modules that support VPLS network interfaces and VPLS virtual core interfaces on ERX14xx models, ERX7xx models, and ERX310 Broadband Services Router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support VPLS network interfaces and VPLS virtual core interfaces.

For information about the modules that support VPLS network interfaces and VPLS virtual core interfaces on E120 and E320 Broadband Services Routers:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support VPLS network interfaces and VPLS virtual core interfaces.

Interface Specifiers

The configuration task examples in this chapter use the *slot/port*[.*subinterface*] format to specify the physical interface on which to configure a VPLS network interface. However, the interface specifier format that you use depends on the router that you are using.

For ERX7xx models, ERX14xx models, and ERX310 routers, use the *slot/port*[.*subinterface*] format. For example, the following command specifies Fast Ethernet subinterface 6 on port 2 of the I/O module installed in slot 3 of an ERX7xx model, ERX14xx model, or ERX310 router.

host1(config)#**interface fastEthernet 3/2.6**

For E120 and E320 routers, use the *slot/adapter/port*[.*subinterface*] format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, the upper IOA bay is identified as adapter 0; the lower IOA bay is identified as adapter 1. For example, the following command specifies Gigabit Ethernet subinterface 20 on port 1 of the IOA installed in the upper adapter bay (adapter 0) of slot 4 in an E320 router.

host1(config)#**interface gigabitEthernet 4/0/1.20**

Related Documentation

- *Interface Types and Specifiers*

VPLS References

For more information about VPLS, consult the following resources:

- *JunosE Release Notes*, *Appendix A*, *System Maximums*—Refer to the Release Notes corresponding to your software release for information about the maximum values supported for VPLS configuration.
- RFC 3036—LDP Specification (January 2001)

- RFC 4447—Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP) (April 2006)
- RFC 4762—Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling (January 2007)
- Virtual Private LAN Service—draft-ietf-l2vpn-vpls-bgp-05.txt (October 2005 expiration)

NOTE: IETF drafts are valid for only 6 months from the date of issuance. They must be considered as works in progress. Please refer to the IETF Website at http://www.ietf.org for the latest drafts.

Related Documentation

- VPLS Protocol Overview on page 597
- VPLS Supported Features on page 609
- VPLS Platform Considerations on page 610

CHAPTER 12

Configuring VPLS

This chapter describes how to configure the virtual private LAN service (VPLS) on the router.

The JunosE implementation of VPLS uses features of transparent bridging, BGP, MPLS, LDP, BGP/MPLS VPNs, and layer 2 services over MPLS. We recommend that you have a thorough understanding of these protocols before you configure and use VPLS in your network.

NOTE: For more information about configuring transparent bridging, see the *JunosE Link Layer Configuration Guide*. For more information about configuring the layer 2 interfaces that support VPLS, see the *JunosE Physical Layer Configuration Guide* and the *JunosE Link Layer Configuration Guide*.

This chapter contains the following sections:

Configuring VPLS with BGP Signaling on a PE Router

To configure VPLS with BGP signaling on the PE router:

1. Configure a single instance of VPLS, known as a VPLS instance, on the PE router for each VPLS domain in which the router participates.
2. (Optional) Configure BGP multihoming for the customer site.
3. (Optional) Configure optional attributes for the VPLS instance.
4. Configure network interfaces to connect the PE router to each CE device.
5. (Optional) Configure nondefault subscriber policies for the VPLS network interface.
6. Configure a loopback interface and assign a router ID that uses the IP address of the loopback interface.
7. Configure MPLS label-switched paths (LSPs) to connect local and remote PE routers.
8. Set up BGP signaling on the autonomous system configured to signal reachability for this VPLS instance.

Related Documentation

- Configuring VPLS Instances with BGP Signaling on page 614
- Configuring Optional Attributes for VPLS Instances on page 617
- Configuring VPLS Network Interfaces on page 618
- Configuring the Loopback Interface and Router ID for VPLS on page 619
- Configuring MPLS LSPs for VPLS on page 620
- Configuring BGP Signaling for VPLS on page 620
- Example: Configuring VPLS with BGP Signaling on page 622

Configuring VPLS Instances with BGP Signaling

You must configure a VPLS instance for each VPLS domain in which the router participates. From a configuration standpoint, a VPLS instance is simply a new or existing bridge group that you configure with additional VPLS attributes.

Table 115 on page 614 lists the commands that you use to configure a basic VPLS instance, as described in this section.

Table 115: Commands to Configure Basic VPLS Instances

bridge vpls rd	**bridge vpls site-range**
bridge vpls route-target	**bridge vpls transport-virtual-routers**
bridge vpls site-name site-id	

To configure a basic VPLS instance with BGP signaling on the PE router:

1. From Global Configuration mode, create the VPLS instance by specifying the transport virtual router for this instance.

   host1(config)#**bridge customer1 vpls transport-virtual-router vr1**

   If the bridge group you specify (customer1 in this example) already exists on the router, issuing this command causes the bridge group to become a VPLS instance.

   NOTE: To configure a VPLS instance, you must issue the **bridge vpls transport-virtual-router** command before you issue any of the other **bridge vpls** commands in this procedure. If the **bridge vpls transport-virtual-router** command is not issued first, the other **bridge vpls** commands fail.

2. Specify the maximum number of customer sites that can participate in the VPLS domain represented by the VPLS instance. (By default, a VPLS domain must consist of at least one site.)

   host1(config)#**bridge customer1 vpls site-range 15**

3. Specify a name and unique identifier for the customer site that belongs to the VPLS instance.

   host1(config)#**bridge customer1 vpls site-name westford site-id 1**

   The site ID value must be greater than zero and be unique across the VPLS domain. This is not true for a multihomed customer site. See "Configuring BGP Multihoming for VPLS" on page 616 for more information.

4. Specify the unique, two-part route distinguisher (RD) for the VPLS instance.

   host1(config)#**bridge customer1 vpls rd 100:11**

   In this example, the first number in the route distinguisher (100) is the number of the AS in which the extended community resides. The second number in the route distinguisher (11) uniquely identifies the VPLS instance within the AS.

   TIP: You cannot change or remove the route distinguisher for a VPLS instance after you set it; for this reason the **no bridge vpls rd** command fails. To change the route distinguisher, you must either remove the transport virtual router configuration from the VPLS instance or delete the VPLS instance from the router. You can then reconfigure the VPLS instance with a new route distinguisher.

5. Create or add a route target to the import and export lists of VPN extended communities for this VPLS instance.

   host1(config)#**bridge customer1 vpls route-target both 100:1**

   The PE router uses the lists of VPN extended communities to determine which routes are imported by this VPLS instance.

BEST PRACTICE: **We recommend that you add the route target to both the VPLS instance's import list and export list of VPN extended communities. To do so, use the both keyword.**

Multiple VPLS instances that use the same transport virtual router cannot have the same route distinguisher. Conversely, multiple VPLS instances that use different transport virtual routers can have the same route distinguisher.

For example, the following commands configure the transport virtual router for each of three VPLS instances: vplsA, vplsB, and vplsC. The transport virtual router for both vplsA and vplsC is vr1, and the transport virtual router for vplsB is vr2.

```
host1(config)#bridge vplsA vpls transport-virtual-router vr1
host1(config)#bridge vplsB vpls transport-virtual-router vr2
host1(config)#bridge vplsC vpls transport-virtual-router vr1
```

Because vplsA and vplsC use the same transport virtual router, vr1, you cannot assign them the same route distinguisher. Consequently, the following operation fails, and the router displays an error message.

```
host1(config)#bridge vplsA vpls rd 1.1.1.1:10
host1(config)#bridge vplsC vpls rd 1.1.1.1:10
% Unable to set VPLS route distinguisher (can't re-use the route-distinguisher)
```

However, both vplsA and vplsB can use the same route distinguisher because their transport virtual routers are different. Consequently, the following commands are valid.

```
host1(config)#bridge vplsA vpls rd 1.1.1.1:10
host1(config)#bridge vplsB vpls rd 1.1.1.1:10
```

Related Documentation

- Configuring VPLS with BGP Signaling on a PE Router on page 614
- Configuring VPLS with LDP Signaling on a PE Router on page 626
- *bridge vpls rd*
- *bridge vpls route-target*
- *bridge vpls site-name site-id*
- *bridge vpls site-range*
- *bridge vpls transport-virtual-router*

Configuring BGP Multihoming for VPLS

You can configure BGP multihoming in the VPLS network to provide redundancy in the event of failures such as a PE router-to-CE device link failure, the failure of a PE router, or an MPLS reachability failure between the local PE router and a remote PE router. BGP multihoming enables you to connect a customer site to two or more PE routers. A redundant PE router can begin providing service to the customer site as soon as the failure is detected. The redundant connectivity maintains the VPLS service and traffic forwarding to and from the multihomed site while avoiding the formation of layer 2 traffic loops.

To configure BGP multihoming on a VPLS PE router:

- Configure the site as multihomed and specify a multihoming priority for the PE site for this instance.

  host1(config)#**bridge customer1 vpls site-name westford site-id 1 multi-homed priority 1**

You must configure the same site ID on all PE routers connected to the multihomed customer site. The site ID shared by the connected PE routers should be different than the site IDs configured on the remote PE routers in the VPLS network.

You can configure a different block offset, label range, and route distinguisher for each connected PE router.

Related Documentation

- Configuring VPLS with BGP Signaling on a PE Router on page 614
- Configuring VPLS with LDP Signaling on a PE Router on page 626
- *bridge vpls site-name site-id*

Configuring Optional Attributes for VPLS Instances

After you create a basic VPLS instance, you can configure one or more optional attributes to manage the MAC address entries in the VPLS instance's forwarding table, or to enable SNMP link status processing. The requirements and procedures for optional attributes are the same whether you employ BGP or LDP signaling for VPLS.

To configure these attributes, you use the same transparent bridging commands that you use to configure bridge groups that do not function as VPLS instances.

1. (Optional) Configure a VPLS instance to acquire dynamically learned MAC addresses.

   host1(config)#**bridge vplsB acquire**

2. (Optional) Enable a VPLS instance to filter (forward or discard) frames based on a specific MAC address, and to add static (nonlearned) address entries to the forwarding table.

   In this example, the VPLS instance forwards frames destined for the node with MAC address 0090.1a40.4c7c out the specified Gigabit Ethernet interface

   host1(config)#**bridge vplsA address 0090.1a40.4c7c forward gigabitEthernet 3/0.1**

   In this example, the VPLS instance drops frames sent from or destined for the node with MAC address 1011.22b2.333c

   host1(config)#**bridge vplsB address 1011.22b2.333c discard**

3. (Optional) Set the length of time that a dynamic (learned) MAC address entry can remain in the forwarding table of the specified VPLS instance before expiring.

   host1(config)#**bridge vplsB aging-time 1000**

4. (Optional) Set the maximum number of dynamic MAC address entries that the specified VPLS instance can learn.

host1(config)#**bridge vplsB learn 2500**

5. (Optional) Enable SNMP link status processing for all network interfaces associated with the specified VPLS instance.

   host1(config)#**bridge vplsB snmp-trap link-status**

For more information about using these commands, see *Configuring Optional Bridge Group Attributes* in the *JunosE Link Layer Configuration Guide*.

Related Documentation

- Configuring VPLS with BGP Signaling on a PE Router on page 614
- *bridge acquire*
- *bridge address*
- *bridge aging-time*
- *bridge learn*
- *bridge snmp-trap link-status*

Configuring VPLS Network Interfaces

You must configure one of the following types of Ethernet or bridged Ethernet network interfaces to transmit packets between the PE router and each CE device to which the PE router is connected:

- Bridged Ethernet over ATM 1483 subinterfaces
- Fast Ethernet
- Gigabit Ethernet
- 10-Gigabit Ethernet
- VLAN and S-VLAN subinterfaces over bridged Ethernet, Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet interfaces

The requirements and procedures for network interfaces are the same whether you employ BGP or LDP signaling for VPLS.

To configure a network interface for a VPLS instance:

1. From Global Configuration mode, select the interface that you want to assign to the VPLS instance.

   host1(config)#**interface gigabitEthernet 3/0**

2. From Interface Configuration mode or Subinterface Configuration mode, assign the interface to the specified VPLS instance.

   host1(config-if)#**bridge-group customer1**

   Issuing this command with no optional keywords configures the network interface as a subscriber (client) interface by default.

3. (Optional) Configure the interface as a trunk (server) interface. For more information about the differences between a subscriber (client) interface and a trunk (server) interface, see "Subscriber Policies for VPLS Network Interfaces Overview" on page 601.

   host1(config-if)#**bridge-group customer1 subscriber-trunk**

4. (Optional) Set the maximum number of MAC addresses that the network interface can learn.

   host1(config-if)#**bridge-group customer1 learn 100**

5. Enable SNMP link status processing only for the specified network interface in the VPLS instance.

   host1(config-if)#**bridge-group customer1 snmp-trap link-status**

Related Documentation

- Configuring VPLS with BGP Signaling on a PE Router on page 614
- Configuring VPLS with LDP Signaling on a PE Router on page 626

- *bridge-group*

Configuring the Loopback Interface and Router ID for VPLS

A loopback interface provides a stable address for BGP or LDP to use so that they can avoid any impact if a physical interface goes down. The loopback interface sends packets back to the router or access server for local processing. Any packets routed from the loopback interface, but not destined to the loopback interface, are dropped.

To establish a BGP session, BGP uses the IP address of the outgoing interface towards the BGP peer as the update source IP address for the TCP connection over which the BGP session runs. Typically, you configure a loopback interface as the update source interface.

LDP uses the loopback interface as the associated interface for the targeted neighbors configured with the **mpls ldp vpls neighbor** command, as described in "Configuring LDP Signaling for VPLS" on page 627.

After you configure the loopback interface, you use the **ip router-id** command to assign a router ID to uniquely identify the router within a BGP AS. The router ID is the IP address of the loopback interface.

To configure the loopback interface and router ID on the PE router:

1. Configure a loopback interface on the PE router and assign it an IP address.

   host1(config)#**interface loopback 0**
   host1(config-if)#**ip address 10.3.3.3 255.255.255.255**
   host1(config-if)#**exit**

2. Assign the router ID using the IP address you configured for the loopback interface.

   host1(config)#**ip router-id 10.3.3.3**

Related Documentation

- Configuring VPLS with BGP Signaling on a PE Router on page 614
- *interface loopback*
- *ip address*
- *ip router-id*

Configuring MPLS LSPs for VPLS

As part of a VPLS configuration, you must create MPLS label-switched paths (LSPs) to connect the local PE router and the remote PE router through the provider (P) router in the MPLS core. The requirements and procedures for subscriber policies are the same whether you employ BGP or LDP signaling for VPLS.

This section explains one way to create a basic MPLS configuration using the **mpls** and **mpls ldp** commands.

To configure MPLS LSPs on the PE router:

1. Enable MPLS on the default virtual router.

   host1(config)#**mpls**

2. Configure the core-facing interface on which you want to enable MPLS, Label Distribution Protocol (LDP), and topology-driven LSPs.

   host1(config)#**interface atm 5/0.100**
   host1(config-subif)#**atm pvc 100 1 100 aal5snap 0 0 0**
   host1(config-subif)#**ip address 192.168.5.5 255.255.255.0**

3. Create an MPLS major interface stacked on the layer 2 interface Enable MPLS on the core-facing interface.

   host1(config-subif)#**mpls**

4. Enable LDP and topology-driven LSPs on the core-facing interface, using the default values (that is, using an implicit default profile).

   host1(config-subif)#**mpls ldp**

For complete information about configuring MPLS LSPs, see "Configuring MPLS" on page 279.

Related Documentation

- Configuring VPLS with BGP Signaling on a PE Router on page 614
- Configuring VPLS with LDP Signaling on a PE Router on page 626
- *mpls*
- *mpls ldp*

Configuring BGP Signaling for VPLS

This section describes one way to configure BGP signaling for VPLS, but does not provide complete details about configuring BGP and BGP/MPLS VPNs.

Table 116 on page 621 lists the commands discussed in this section to configure BGP signaling for VPLS.

Table 116: Commands to Configure BGP Signaling for VPLS

address-family l2vpn	neighbor next-hop-self
address-family vpls	neighbor remote-as
exit-address-family	neighbor send-community
ip router-id	neighbor update-source
neighbor activate	router bgp

To configure BGP signaling for VPLS on the PE router:

1. Enable the BGP routing protocol on the PE router and specify the local AS; that is, the AS to which this BGP speaker belongs.

   host1(config)#**router bgp 100**

   The AS number identifies the PE router to other BGP routers.

2. Configure the PE-to-PE BGP session by first adding an entry to the BGP neighbor table.

   host1(config-router)#**neighbor 10.4.4.4 remote-as 100**

3. Use **neighbor** commands to specify the peers to which BGP advertises routes.

   This example configures only the update-source and next-hop-self attributes. The update-source attribute allows the BGP session to use the IP address of a specific operational interface as the update source address for TCP connections. the next-hop-self attribute forces the BGP speaker to report itself as the next hop for an advertised route that it learned from a neighbor.

   host1(config-router)#**neighbor 10.4.4.4 update-source loopback 0**
   host1(config-router)#**neighbor 10.4.4.4 next-hop-self**

4. Create the L2VPN address family to configure the router to exchange layer 2 NLRI for all VPLS instances.

   host1(config-router)#**address-family l2vpn signaling**

5. Activate the PE-to-PE session in the L2VPN address family by specifying neighbors that exchange routes from within the current address family.

   host1(config-router-af)#**neighbor 10.4.4.4 activate**

6. Use **neighbor** commands to configure additional address family parameters for the session, then exit the address family.

   This example configures only the next-hop-self attribute, forcing the BGP speaker to report itself as the next hop for an advertised route that it learned from a neighbor.

   host1(config-router-af)#**neighbor 10.4.4.4 next-hop-self**
   host1(config-router-af)#**exit-address-family**

7. Create the VPLS address family to configure the router to exchange layer 2 NLRI for each VPLS instance configured on the router.

   You must issue the **address-family vpls** command separately for each VPLS instance configured on the router.

   host1(config-router)#**address-family vpls customer1**
   host1(config-router-af)#**exit-address-family**
   host1(config-router)#**address-family vpls customer2**

After you configure MPLS LSPs and BGP signaling, the router automatically generates a VPLS virtual core interface for each VPLS instance. The VPLS virtual core interface represents all of the MPLS tunnels from the router to the remote VE device.

See "Configuring BGP Routing" on page 3 for information about configuring BGP.

See "Configuring BGP-MPLS Applications" on page 389 for information about configuring BGP/MPLS VPNs.

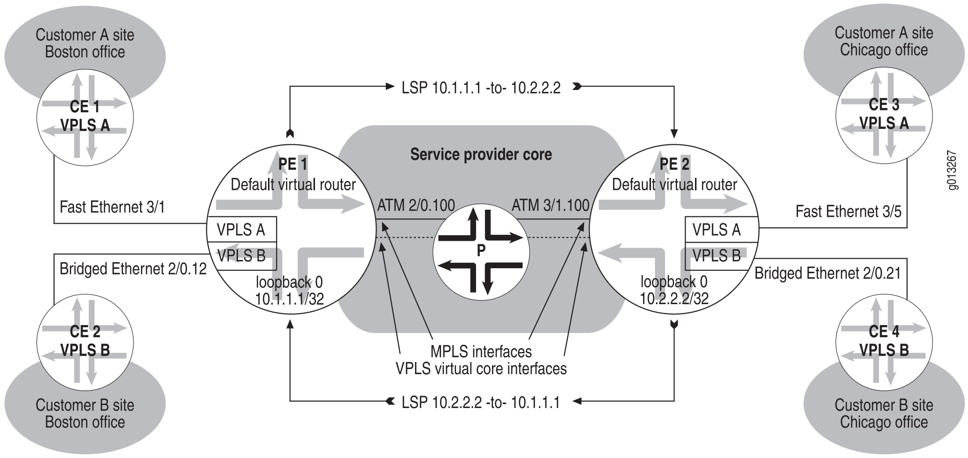
Related Documentation

- Configuring VPLS with BGP Signaling on a PE Router on page 614
- *address-family* l2vpn
- *address-family* vpls command
- *exit-address-family*
- *neighbor activate*
- *neighbor next-hop-self*
- *neighbor remote-as*
- *router bgp*

Example: Configuring VPLS with BGP Signaling

The example in this section shows how to configure the VPLS topology illustrated in Figure 131 on page 623. The example includes the commands for configuring VPLS on both the local E Series router (PE 1) and the remote E Series router (PE 2).

Figure 131: Topology for VPLS Configuration Example with BGP Signaling



- Topology Overview of VPLS with BGP Signaling on page 623
- Configuration on PE 1 (Local PE Router) on page 624
- Configuration on PE 2 (Remote PE Router) on page 625

Topology Overview of VPLS with BGP Signaling

The sample topology in Figure 131 on page 623 includes two VPLS domains, VPLS A and VPLS B. VPLS A connects CE 1, at the edge of Customer A's Boston site, with CE 3, at the edge of Customer A's Chicago site. Similarly, VPLS B connects CE 2, at the edge Customer B's Boston site, with CE 4, at the edge of Customer B's Chicago site.

The E Series routers in the topology, PE 1 and PE 2, each participate in both the VPLS A domain and the VPLS B domain. The example configures a total of four separate VPLS instances, one for each VPLS domain in which the PE router participates. The instances for the VPLS A domain are named vplsA, and the instances for the VPLS B domain are named vplsB.

For each VPLS instance, an Ethernet or bridged Ethernet network interface provides a connection to the associated CE device. Each VPLS instance maintains its own set of forwarding tables and filters to learn the network topology, in a manner that is similar to a bridge group used for transparent bridging.

Each PE router in the sample topology also has an ATM core-facing interface that connects it to the provider (P) router in the service provider core. You must configure MPLS LSPs on the core-facing interfaces to connect PE 1 and PE 2 through the P router across the service provider core. Finally, you must configure BGP on both PE 1 and PE 2 to provide signaling for both VPLS domains.

After you configure the bridging, MPLS, and BGP components of VPLS, the router automatically generates a VPLS virtual core interface for each VPLS instance. The VPLS virtual core interface represents all of the MPLS tunnels from the router to the remote VE device.

Configuration on PE 1 (Local PE Router)

Use the following commands on the local PE router (PE 1) to configure the VPLS topology shown in Figure 131 on page 623.

```
! Configure VPLS instance vplsA.
host1(config)#bridge vplsA vpls transport-virtual-router default
host1(config)#bridge vplsA vpls site-range 10
host1(config)#bridge vplsA vpls site-name boston site-id 1
host1(config)#bridge vplsA vpls rd 100:11
host1(config)#bridge vplsA vpls route-target both 100:1
!
! Configure VPLS instance vplsB.
host1(config)#bridge vplsB vpls transport-virtual-router default
host1(config)#bridge vplsB vpls site-range 20
host1(config)#bridge vplsB vpls site-name boston site-id 1
host1(config)#bridge vplsB vpls rd 100:12
host1(config)#bridge vplsB vpls route-target both 100:2
!
! Configure Fast Ethernet interface 3/0 between PE 1 and CE 1,
! and assign it to vplsA as a trunk interface.
host1(config)#interface fastEthernet 3/1
host1(config-if)#bridge-group vplsA subscriber-trunk
host1(config-if)#exit
!
! Configure bridged Ethernet interface 2/0.12 between PE 1 and CE 2,
! and assign it to vplsB as a trunk interface.
host1(config)#interface atm 2/0.12 point-to-point
host1(config-subif)#atm pvc 12 0 12 aal5snap 0 0 0
host1(config-subif)#encapsulation bridge1483 mac-address 0090.1a40.9991
host1(config-subif)#bridge-group vplsB subscriber-trunk
host1(config-if)#exit
!
! Configure a loopback interface on PE 1 and assign it an IP address.
host1(config)#interface loopback 0
host1(config-if)#ip address 10.1.1.1 255.255.255.255
host1(config-if)#exit
!
! Assign the router ID for PE 1 using the IP address of the loopback interface.
host1(config)#ip router-id 10.1.1.1
!
! Enable MPLS on the default virtual router.
host1(config)#mpls
!
! Configure ATM core-facing interface 2/0.100 between PE 1 and the P router,
! and assign it an IP address.
host1(config)#interface atm 2/0.100 point-to-point
host1(config-subif)#atm pvc 100 1 100 aal5snap 0 0 0
host1(config-subif)#ip address 192.168.1.1 255.255.255.0
!
! Enable MPLS, LDP, and topology-driven LSPs on the core-facing interface.
host1(config-subif)#mpls
host1(config-subif)#mpls ldp
host1(config-subif)#exit
!
```

```
! Configure BGP signaling.
host1(config)#router bgp 100
host1(config-router)#neighbor 10.2.2.2 remote-as 100
host1(config-router)#neighbor 10.2.2.2 update-source loopback 0
host1(config-router)#neighbor 10.2.2.2 next-hop-self
host1(config-router)#address-family l2vpn signaling
host1(config-router-af)#neighbor 10.2.2.2 activate
host1(config-router-af)#neighbor 10.2.2.2 next-hop-self
host1(config-router-af)#exit-address-family
host1(config-router)#address-family vpls vplsA
host1(config-router-af)#exit-address-family
host1(config-router)#address-family vpls vplsB
host1(config-router-af)#exit-address-family
host1(config-router)#exit
```

Configuration on PE 2 (Remote PE Router)

Use the following commands on the remote PE router (PE 2) to configure the VPLS topology shown in Figure 131 on page 623.

```
! Configure VPLS instance vplsA. The route target (100:1)
! matches the route target configured for vplsA on PE 1.
host2(config)#bridge vplsA vpls transport-virtual-router default
host2(config)#bridge vplsA vpls site-range 10
host2(config)#bridge vplsA vpls site-name chicago site-id 2
host2(config)#bridge vplsA vpls rd 100:21
host2(config)#bridge vplsA vpls route-target both 100:1
!
! Configure VPLS instance vplsB. The route target (100:2)
! matches the route target configured for vplsB on PE 1.
host2(config)#bridge vplsB vpls transport-virtual-router default
host2(config)#bridge vplsB vpls site-range 20
host2(config)#bridge vplsB vpls site-name chicago site-id 2
host2(config)#bridge vplsB vpls rd 100:22
host2 (config)#bridge vplsB vpls route-target both 100:2
! Configure Fast Ethernet interface 3/5 between PE 2 and CE 3,
! and assign it to vplsA as a trunk interface.
host2(config)#interface fastEthernet 3/5
host2(config-if)#bridge-group vplsA subscriber-trunk
host2(config-if)#exit
!
! Configure bridged Ethernet interface 2/0.21 between PE 2 and CE 4,
! and assign it to vplsB as a trunk interface.
host2(config)#interface atm 2/0.21 point-to-point
host2(config-subif)#atm pvc 21 0 21 aal5snap 0 0 0
host2(config-subif)#encapsulation bridge1483 mac-address 0090.1a40.9992
host2(config-subif)#bridge-group vplsB subscriber-trunk
host2(config-if)#exit
!
! Configure a loopback interface on PE 2 and assign it an IP address.
host2(config)#interface loopback 0
host2(config-if)#ip address 10.2.2.2 255.255.255.255
host2(config-if)#exit
!
! Assign the router ID for PE 2 using the IP address of the loopback interface.
host2(config)#ip router-id 10.2.2.2
```



```
!
! Enable MPLS on the default virtual router.
host2(config)#mpls
!
! Configure ATM core-facing interface 3/1.100 between PE 2 and the P router,
! and assign it an IP address.
host2(config)#interface atm 3/1.100 point-to-point
host2(config-subif)#atm pvc 100 1 100 aal5snap 0 0 0
host2(config-subif)#ip address 192.168.2.2 255.255.255.0
!
! Enable MPLS, LDP, and topology-driven LSPs on the on the core-facing interface.
host2(config-subif)#mpls
host2(config-subif)#mpls ldp
host2(config-subif)#exit
!
! Configure BGP signaling.
host2(config)#router bgp 100
host2(config-router)#neighbor 10.1.1.1 remote-as 100
host2(config-router)#neighbor 10.1.1.1 update-source loopback 0
host2(config-router)#neighbor 10.1.1.1 next-hop-self
host2(config-router)#address-family l2vpn signaling
host2(config-router-af)#neighbor 10.1.1.1 activate
host2(config-router-af)#neighbor 10.1.1.1 next-hop-self
host2(config-router-af)#exit-address-family
host2(config-router)#address-family vpls vplsA
host2(config-router-af)#exit-address-family
host2(config-router)#address-family vpls vplsB
host2(config-router-af)#exit-address-family
host2(config-router)#exit
```

Related Documentation

- Configuring VPLS with BGP Signaling on a PE Router on page 614

Configuring VPLS with LDP Signaling on a PE Router

To configure VPLS with LDP signaling on the PE router:

1. Configure a single instance of VPLS, known as a VPLS instance, on the PE router for each VPLS domain in which the router participates.
2. (Optional) Configure optional attributes for the VPLS instance.

   For instructions, see "Configuring Optional Attributes for VPLS Instances" on page 617.
3. Configure network interfaces to connect the PE router to each CE device.

   For instructions, see "Configuring VPLS Network Interfaces" on page 618.
4. (Optional) Configure nondefault subscriber policies for the VPLS network interface.

   For instructions, see "Subscriber Policies for VPLS Network Interfaces Overview" on page 601.
5. Configure a loopback interface to be associated with the targeted LDP neighbor, and assign a router ID that uses the IP address of the loopback interface.

For instructions, see “Configuring the Loopback Interface and Router ID for VPLS” on page 619.

6. Configure MPLS LSPs to connect local and remote PE routers.

   For instructions, see “Configuring MPLS LSPs for VPLS” on page 620.

7. Set up LDP signaling for this VPLS instance to establish targeted sessions to the remote PE neighbors configured at the edge of the MPLS core network.

8. Configure an interior gateway protocol (IGP), such as Open Shortest Path First (OSPF) or Intermediate System–to–Intermediate System (IS-IS), to enable routing within the core network.

Related Documentation

- Configuring VPLS Instances with LDP Signaling on page 627
- Configuring Optional Attributes for VPLS Instances on page 617
- Configuring VPLS Network Interfaces on page 618
- Configuring LDP Signaling for VPLS on page 627
- Example: Configuring VPLS LDP Signaling on page 629

Configuring VPLS Instances with LDP Signaling

As is the case with BGP signaling, when you use LDP signaling you must configure a VPLS instance for each VPLS domain in which the router participates. Unlike BGP signaling, however, configuring a VPLS instance for LDP signaling requires only that you specify the transport virtual router for this instance by issuing the **bridge vpls transport-virtual-router** command.

To configure a basic VPLS instance with LDP signaling on the PE router:

- From Global Configuration mode, create the VPLS instance by specifying the transport virtual router for this instance.

  host1(config)#bridge customer3 vpls transport-virtual-router vr1

  The transport virtual router specifies the name of the virtual router on which the LDP instance that signals reachability for this VPLS instance is configured. If the bridge group you specify (customer3 in this example) already exists on the router, issuing this command causes the bridge group to become a VPLS instance.

Related Documentation

- Configuring VPLS with LDP Signaling on a PE Router on page 626
- *bridge vpls transport-virtual-router*

Configuring LDP Signaling for VPLS

LDP signaling establishes targeted sessions to the remote VEs configured at the edge of the service provider’s MPLS core network. To enable LDP to establish these targeted sessions, you issue the **mpls ldp vpls-id** command to configure a VPLS identifier for the

VPLS instance, and the **mpls ldp vpls neighbor** command to configure a list of neighbor (peer) addresses to which LDP can send or from which LDP can receive targeted hello messages.

This section describes how to configure LDP signaling for a VPLS network, but does not provide complete details about configuring LDP on E Series routers.

Table 117 on page 628 lists the commands discussed in this section to configure LDP signaling for VPLS.

Table 117: Commands to Configure LDP Signaling for VPLS

mpls ldp vpls neighbor	**mpls ldp vpls vpls-id**

To configure LDP signaling for VPLS on the PE router:

1. Configure the VPLS identifier, which is a globally unique identifier for each VPLS domain.

   host1(config)#**mpls ldp vpls customer3 vpls-id 3**

2. Enable LDP signaling for a VPLS instance by configuring a list of neighbor (peer) addresses on remote VE devices in the VPLS domain to which LDP can send or from which LDP can receive targeted hello messages.

   host1(config)#**mpls ldp vpls customer3 neighbor 10.3.3.3**
   host1(config)#**mpls ldp vpls customer3 neighbor 10.3.10.100**

3. Repeat these steps for each VPLs instance on the PE router.

For more information about LDP, see “Configuring MPLS” on page 279.

Related Documentation

- Configuring VPLS with LDP Signaling on a PE Router on page 626
- *mpls ldp vpls neighbor*
- *mpls ldp vpls vpls-id*

Configuring Routing in the Core Network for VPLS

After you configure the transparent bridging, LDP, and MPLS components of the VPLS network, you must configure an IGP, such as OSPF or IS-IS, on the PE router to set up routing within the core MPLS network.

This section explains one way to configure OSPF to enable routing in the core network.

Table 118 on page 628 lists the commands discussed in this section to configure OSPF.

Table 118: Commands to Configure OSPF for a VPLS Network

network area	**router ospf**

To configure the PE router to set up OSPF routing for the core MPLS network:

1. Create the OSPF routing process.

   host1(config)#router ospf 1

2. Create the range of IP addresses associated with the routing process and the corresponding OSPF interfaces, and assign an area ID associated with each range of IP addresses.

   host1(config-router)#network 10.1.1.1 0.0.0.0 area 0.0.0.0
   host1(config-router)#network 10.10.10.0 0.0.0.255 area 0.0.0.0

   This example configures an OSPF routing process with process ID 1, and creates two OSPF interfaces in the backbone area (area 0.0.0.0): one using IP address 10.1.1.1, and one using IP address 10.10.10.0. The **network area** commands create the two OSPF areas if they do not already exist.

For complete information about configuring and using OSPF and IS-IS, see the *JunosE IP, IPv6, and IGP Configuration Guide*.

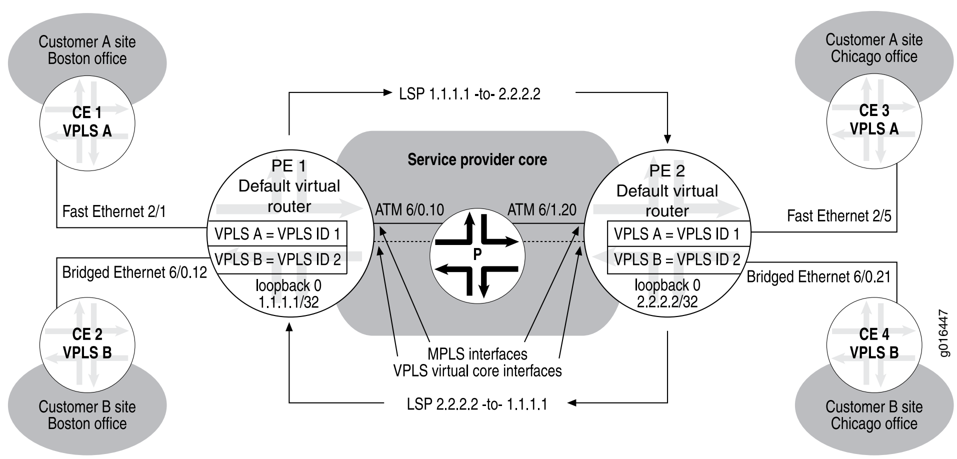
Related Documentation

- Configuring VPLS with LDP Signaling on a PE Router on page 626
- *network area*
- *router ospf*

Example: Configuring VPLS LDP Signaling

The example in this section shows how to configure the VPLS topology illustrated in Figure 132 on page 629. The example includes the commands for configuring VPLS on both the local E Series router (PE 1) and the remote E Series router (PE 2).

Figure 132: Topology for VPLS Configuration Example with LDP Signaling



- Topology Overview of VPLS with LDP Signaling on page 630
- Configuration on PE 1 (Local PE Router) on page 630
- Configuration on PE 2 (Remote PE Router) on page 631

Topology Overview of VPLS with LDP Signaling

Because the basic components of a VPLS network are the same regardless of whether BGP signaling or LDP signaling is used, the sample topology shown for LDP signaling in Figure 132 on page 629 is almost identical to the sample topology shown for BGP signaling in Figure 131 on page 623. Figure 132 on page 629 includes two VPLS domains: VPLS A, which connects CE 1 and CE 3, and VPLS B, which connects CE 2 and CE 4. The local PE router, PE 1, and the remote PE router, PE 2, each participate in both the VPLS A domain and the VPLS B domain, and have one VPLS instance associated with each domain configured on each router.

Unlike a VPLS configuration with BGP signaling, a VPLS configuration with LDP signaling requires that you configure a VPLS ID for each VPLS instance to uniquely identify each VPLS domain. In the sample topology in Figure 132 on page 629, instance vplsA is assigned VPLS ID 1, and instance vplsB is assigned VPLS ID 2 on both the local PE router and the remote PE router. You must also configure a list of remote neighbor (peer) addresses to which LDP can send or from which LDP can receive targeted hello messages. In the sample topology, the remote neighbor configured for PE 1 is PE 2 with IP address 2.2.2.2, and the remote neighbor configured for PE 2 is PE 1 with IP address 1.1.1.1.

The Ethernet and bridged Ethernet network interfaces, ATM core-facing interfaces, VPLS virtual core interfaces, and MPLS LSPs play the same role in a VPLS topology with LDP signaling as they do in a VPLS topology with BGP signaling. For more information about these components, see "Topology Overview of VPLS with BGP Signaling" on page 623.

Configuration on PE 1 (Local PE Router)

Use the following commands on the local PE router (PE 1) to configure the VPLS topology shown in Figure 132 on page 629.

```
! Configure VPLS instance vplsA.
host1(config)#bridge vplsA vpls transport-virtual-router default
!
! Configure VPLS instance vplsB.
host1(config)#bridge vplsB vpls transport-virtual-router default
!
! Configure Fast Ethernet interface 2/1 between PE 1 and CE 1,
! and assign it to vplsA as a trunk interface.
host1(config)#interface fastEthernet 2/1
host1(config-if)#bridge-group vplsA subscriber-trunk
host1(config-if)#exit
!
! Configure bridged Ethernet interface 6/0.12 between PE 1 and CE 2,
! and assign it to vplsB as a trunk interface.
host1(config)#interface atm 6/0.12 point-to-point
host1(config-subif)#atm pvc 12 0 12 aal5snap 0 0 0
host1(config-subif)#encapsulation bridge1483 mac-address 0090.1a40.9991
host1(config-subif)#bridge-group vplsB subscriber-trunk
host1(config-if)#exit
!
! Configure LDP signaling for vplsA.
host1(config)#mpls ldp vpls vplsA vpls-id 1
host1(config)#mpls ldp vpls vplsA neighbor 2.2.2.2
```



```
! Configure LDP signaling for vplsB.
host1(config)#mpls ldp vpls vplsB vpls-id 2
host1(config)#mpls ldp vpls vplsB neighbor 2.2.2.2
!
! Configure a loopback interface on PE 1 and assign it an IP address.
host1(config)#interface loopback 0
host1(config-if)#ip address 1.1.1.1 255.255.255.255
host1(config-if)#exit
!
! Assign the router ID for PE 1 using the IP address of the loopback interface.
host1(config)#ip router-id 1.1.1.1
!
! Configure ATM core-facing interface 6/0.10 between PE 1 and the P router,
! and assign it an IP address.
host1(config)#interface atm 6/0.10 point-to-point
host1(config-subif)#atm pvc 10 0 10 aal5snap 0 0 0
host1(config-subif)#ip address 10.10.10.1 255.255.255.0
!
! Enable MPLS, LDP, and topology-driven LSPs on the core-facing interface.
host1(config-subif)#mpls
host1(config-subif)#mpls ldp
host1(config-subif)#exit
!
! Configure OSPF routing in the core MPLS network.
host1(config)#router ospf 1
host1(config-router)#network 1.1.1.1 0.0.0.0 area 0.0.0.0
host1(config-router)#network 10.10.10.0 0.0.0.255 area 0.0.0.0
host1(config-router)#exit
```

Configuration on PE 2 (Remote PE Router)

Use the following commands on the remote PE router (PE 2) to configure the VPLS topology shown in Figure 132 on page 629.

```
! Configure VPLS instance vplsA.
host2(config)#bridge vplsA vpls transport-virtual-router default
!
! Configure VPLS instance vplsB.
host2(config)#bridge vplsB vpls transport-virtual-router default
!
! Configure Fast Ethernet interface 2/5 between PE 2 and CE 3,
! and assign it to vplsA as a trunk interface.
host2(config)#interface fastEthernet 2/5
host2(config-if)#bridge-group vplsA subscriber-trunk
host2(config-if)#exit
!
! Configure bridged Ethernet interface 6/0.21 between PE 2 and CE 4,
! and assign it to vplsB as a trunk interface.
host2(config)#interface atm 6/0.21 point-to-point
host2(config-subif)#atm pvc 21 0 21 aal5snap 0 0 0
host2(config-subif)#encapsulation bridge1483 mac-address 0090.1a40.9992
host2(config-subif)#bridge-group vplsB subscriber-trunk
host2(config-if)#exit
!
! Configure LDP signaling for vplsA.
host2(config)#mpls ldp vpls vplsA vpls-id 1
```

```
host2(config)#mpls ldp vpls vplsA neighbor 1.1.1.1
!
! Configure LDP signaling for vplsB.
host2(config)#mpls ldp vpls vplsB vpls-id 2
host2(config)#mpls ldp vpls vplsB neighbor 1.1.1.1
!
! Configure a loopback interface on PE 2 and assign it an IP address.
host2(config)#interface loopback 0
host2(config-if)#ip address 2.2.2.2 255.255.255.255
host2(config-if)#exit
!
! Assign the router ID for PE 2 using the IP address of the loopback interface.
host2(config)#ip router-id 2.2.2.2
!
! Configure ATM core-facing interface 6/1.20 between PE 2 and the P router,
! and assign it an IP address.
host2(config)#interface atm 6/1.20 point-to-point
host2(config-subif)#atm pvc 20 0 20 aal5snap 0 0 0
host2(config-subif)#ip address 20.20.20.2 255.255.255.0
!
! Enable MPLS, LDP, and topology-driven LSPs on the core-facing interface.
host2(config-subif)#mpls
host2(config-subif)#mpls ldp
host2(config-subif)#exit
!
! Configure OSPF routing in the core MPLS network.
host2(config)#router ospf 1
host2(config-router)#network 2.2.2.2 0.0.0.0 area 0.0.0.0
host2(config-router)#network 20.20.20.0 0.0.0.255 area 0.0.0.0
host2(config-router)#exit
```

Related Documentation

- Configuring VPLS with LDP Signaling on a PE Router on page 626

CHAPTER 13

Monitoring VPLS

This chapter describes the commands you can use to monitor and troubleshoot the virtual private LAN service (VPLS) on E Series routers.

NOTE: The E120 and E320 Broadband Services Routers output for **monitor** and **show** commands is identical to output from other E Series routers, except that the E120 and E320 router output also includes information about the adapter identifier in the interface specifier (*slot/adapter/port*).

This chapter contains the following sections:

Setting the Baseline for VPLS Statistics

You can use the following **baseline** commands to set a statistics baseline for a VPLS instance, for a network interface associated with a VPLS instance, or for the VPLS virtual core interface associated with a VPLS instance. The router implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved.

Use the **delta** keyword with the **show bridge** commands display baselined statistics.

Tasks to set a baseline for VPLS statistics are:

Setting a Baseline for a VPLS Instance

To set a statistics baseline for a VPLS instance:

- Issue the **baseline bridge** command.

  host1#**baseline bridge vplsA**

Setting a Baseline for a Network Interface associated with a VPLS Instance

To set a statistics baseline for the VPLS network interface:

- Issue the **baseline bridge interface** command.

  host1#**baseline bridge interface gigabitEthernet 4/1**

Setting a Baseline for the VPLS Virtual Core Interface associated with a VPLS Instance

To set a statistics baseline for the VPLS virtual core interface:

- Issue the **baseline bridge interface vpls** command.

  host1#**baseline bridge interface vpls vplsA**

Related Documentation

- *baseline bridge*
- *baseline bridge interface*
- *baseline bridge interface vpls*

Clearing Dynamic MAC Addresses from the VPLS Forwarding Table

You can use the following **clear** commands to remove all dynamic (learned) MAC address entries or a specific dynamic MAC address entry from the forwarding table for a VPLS instance.

Tasks to clear the VPLS forwarding table are:

Clearing All Dynamic MAC Addresses from the VPLS Forwarding Table

To clear all dynamic MAC address entries for the VPLS instance:

- Issue the **clear bridge** command.

  host1#**clear bridge vplsB**

Clearing a Specific Dynamic MAC Addresses from the VPLS Forwarding Table

To clear a specific dynamic MAC address entry for the VPLS instance:

- Issue the **clear bridge address** command.

  host1#**clear bridge vplsB address 0090.1a40.9992**

Clearing All Dynamic MAC Addresses for a Network Interface associated with a VPLS Instance

To clear all dynamic MAC address entries for a network interface associated with a VPLS instance:

- Issue the **clear bridge interface** command.

  host1#**clear bridge interface atm 3/3.2**

Clearing All Dynamic MAC Addresses for the VPLS Virtual Core Interface associated with a VPLS Instance

To clear all dynamic MAC address entries for the VPLS virtual core interface associated with a VPLS instance:

- Issue the **clear bridge interface vpls** command.

  host1#**clear bridge interface vpls vplsA**

Related Documentation

- *clear bridge*
- *clear bridge address*
- *clear bridge interface*
- *clear bridge interface vpls*

Clearing BGP Attributes for VPLS

You can use the following **clear ip bgp** commands to remove specific BGP attributes for the L2VPN address family, and in one case, for the VPLS address family associated with a specific VPLS instance.

Clearing BGP Reachability Information for the L2VPN Address Family

To clear BGP reachability information for the L2VPN address family:

- Issue the **clear ip bgp** command.

  host1#**clear ip bgp l2vpn soft in**

Use the **soft in** keywords to trigger inbound soft reconfiguration.

Clearing BGP Route Flap Dampening Information for the L2VPN Address Family

To clear route flap dampening information for the L2VPN address family:

- Issue the **clear ip bgp dampening** command.

  host1#**clear ip bgp l2vpn dampening**

Clearing BGP Route Flap Dampening Information for the VPWS Address Family

To clear route flap dampening information for the VPLS address family associated with the specified VPLS instance:

- Issue the **clear ip bgp dampening** command.

  host1#**clear ip bgp vpls vplsA dampening**

Clearing the Wait for End-of-RIB Marker for the L2VPN Address Family

To clear the wait for receiving an End-of-RIB marker from the peer for the L2VPN address family:

- Issue the **clear ip bgp wait-end-of-rib** command.

host1#**clear ip bgp l2vpn wait-end-of-rib**

Related Documentation

- *clear ip bgp*
- *clear ip bgp dampening*

- *clear ip bgp wait-end-of-rib*

Monitoring VPLS Configuration and Statistics for a Specific VPLS Instance

Purpose Display configuration and statistics information for the specified VPLS instance.

Action To display configuration information for a specified VPLS (vplsA):

`host1#`**show bridge vplsA**

```
BridgeGroup: vplsA(vpls)

      Bridge Mode:            default
      Aging Time:             300 secs
      Learning:               Enabled
      Max Learn:              Unlimited
      Link Status Snmp Traps: Disabled
      Subscriber Policy:      default Subscriber
      Port Count:             2
      Interface Count:        1
      Transport Virtual Rtr:  default
      Route Distinguisher:    1.1.1.1:10
      SiteName:               boston
      SiteId:                 1
      Multi-homed:            Yes
      Site-Priority:          45
      SiteRange:              10
      VPLS Route Targets
        Route Target: RT:100:1 (both)
        Route Target: RT:100:2 (both)
      Flood Next Hop: Index 1048577
```

Use the **all** keyword to display address table and statistics information for all network interfaces associated with the VPLS instance.

Meaning Table 119 on page 637 lists the **show bridge** command output fields.

Table 119: show bridge Output Fields

Field Name	Field Description
BridgeGroup	Name of the VPLS instance for which information is displayed
Bridge Mode	Bridging capability currently enabled; for a VPLS instance, this field always displays default
Aging Time	Length of time, in seconds, that a MAC address entry can remain in the forwarding table before expiring
Learning	Whether acquisition of dynamically learned MAC addresses is enabled or disabled
Max Learn	Maximum number of dynamic MAC addresses that the VPLS instance can learn

Table 119: show bridge Output Fields *(continued)*

Field Name	Field Description
Link Status Snmp Traps	Whether SNMP link status processing is enabled or disabled
Subscriber Policy	Name of the subscriber policy currently in effect
Port Count	Number of ports currently configured for the VPLS instance, including network interfaces and the VPLS virtual core interface
Interface Count	Number of network interfaces currently configured for the VPLS instance
Transport Virtual Rtr	Name of the transport virtual router configured for the VPLS instance
Route Distinguisher	Unique route distinguisher configured for the VPLS instance
SiteName	Site name configured for the VPLS instance
SiteId	Numerical site identifier configured for the VPLS instance
Multi-homed	Status of the site. Yes designates a multihomed site. No designates a site that is not multihomed.
Site-Priority	Priority value for the VPLS instance for the multihomed site; displayed only when the value for the Multi-homed field is Yes
SiteRange	Maximum number of sites that can participate in the VPLS domain associated with the VPLS instance
VPLS Route Targets	Extended community identifiers, also known as route targets, for each VPLS instance configured on the router
Flood Next Hop	Index of the MPLS next hop to which the router floods packets with unknown destination addresses. For more information about displaying MPLS next hops and any available next-hop statistics, see “Monitoring MPLS Next Hops” on page 361.

Related Documentation

- *show bridge*

Monitoring VPLS Configuration and Statistics for All VPLS Instances

Purpose

Display configuration and statistics information for all VPLS instances configured on the router.

Action

To display the names of all VPLS instances configured on the router:

```
host1#show bridge groups

  BridgeGroup: vplsA(vpls)
```



```
  BridgeGroup: vplsB(vpls)
```

To display configuration settings for all VPLS instances on the router:

`host1#`**show bridge groups details**

```
  BridgeGroup: vplsA(vpls)

        Bridge Mode:            default
        Aging Time:             300 secs
        Learning:               Enabled
        Max Learn:              Unlimited
        Link Status Snmp Traps: Disabled
        Subscriber Policy:      default Subscriber
        Port Count:             2
        Interface Count:        1
        Transport Virtual Rtr:  default
        Route Distinguisher:    1.1.1.1:10
        SiteName:               boston
        SiteId:                 1
        Multi-homed:            Yes
        Site-Priority:          45
        SiteRange:              10
        VPLS Route Targets
          Route Target: RT:100:1 (both)
          Route Target: RT:100:2 (both)
        Flood Next Hop: Index 1048577

  BridgeGroup: vplsB(vpls)

        Bridge Mode:            default
        Aging Time:             300 secs
        Learning:               Enabled
        Max Learn:              Unlimited
        Link Status Snmp Traps: Disabled
        Subscriber Policy:      default Subscriber
        Port Count:             2
        Interface Count:        1
        Transport Virtual Rtr:  default
        Route Distinguisher:    1.1.1.1:11
        SiteName:               boston
        SiteId:                 1
        Multi-homed:            No
        SiteRange:              20
        VPLS Route Targets
          No Route Targets configured
        Flood Next Hop: Index 1048578
```

Meaning Table 120 on page 639 lists the **show bridge group details** command output fields.

Table 120: show bridge groups details Output Fields

Field Name	Field Description
BridgeGroup	Name of the VPLS instance for which information is displayed
Bridge Mode	Bridging capability currently enabled; for a VPLS instance, this field always displays default

Table 120: show bridge groups details Output Fields *(continued)*

Field Name	Field Description
Aging Time	Length of time, in seconds, that a MAC address entry can remain in the forwarding table before expiring
Learning	Whether acquisition of dynamically learned MAC addresses is enabled or disabled
Max Learn	Maximum number of dynamic MAC addresses that the VPLS instance can learn
Link Status Snmp Traps	Whether SNMP link status processing is enabled or disabled
Subscriber Policy	Name of the subscriber policy currently in effect
Port Count	Number of ports currently configured for the VPLS instance, including network interfaces and the VPLS virtual core interface
Interface Count	Number of network interfaces currently configured for the VPLS instance
Transport Virtual Rtr	Name of the transport virtual router configured for the VPLS instance
Route Distinguisher	Unique route distinguisher configured for the VPLS instance
SiteName	Site name configured for the VPLS instance
SiteId	Numerical site identifier configured for the VPLS instance
Multi-homed	Status of the site. Yes designates a multihomed site. No designates a site that is not multihomed.
Site-Priority	Priority value for the VPLS instance for the multihomed site; displayed only when the value for the Multi-homed field is Yes
SiteRange	Maximum number of sites that can participate in the VPLS domain associated with the VPLS instance
VPLS Route Targets	Extended community identifiers, also known as route targets, for each VPLS instance configured on the router
Flood Next Hop	Index of the MPLS next hop to which the router floods packets with unknown destination addresses. For more information about displaying MPLS next hops and any available next-hop statistics, see “Monitoring MPLS Next Hops” on page 361.

Related Documentation

- *show bridge groups*

Monitoring Configuration, Statistics, and Status for VPLS Network Interfaces

Purpose Display configuration, statistics, and status information for a specified network interface or for all interfaces assigned to a VPLS instance.

Action To display information for a specified network interface:

`host1#`**show bridge interface atm 3/1.10**

```
atm3/1.10
  BridgeGroup: vplsB
  Port Number: 1
  Operational Status: Up
  Admin Status: Up
  Snmp Link Status Trap: Disabled
  Max Learn: Unlimited
  Subscriber Policy: default Trunk
  Statistics:
     In Octets:    1958
     In Frames:    14
     In Discards:  1
     In Errors:    0
     Out Octets:   1930
     Out Frames:   14
     Out Discards: 1
     Out Errors:   0
  Time since counters last reset: 00:14:32

  queue 0: traffic class best-effort, bound to bridge ATM3/1.10
    Queue length 0 bytes
    Forwarded packets 14, bytes 2238
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0
```

To display information about all interfaces in a VPLS instance, including the VPLS virtual core interface (vplsB):

`host1#`**show bridge vplsB interface**

```
FastEthernet1/1.1
  Port Number: 1
  Operational Status: Up
  Admin Status: Up
  Snmp Link Status Trap: Disabled
  Max Learn: Unlimited
  Subscriber Policy: samplepolicy
  Statistics:
     In Octets:    3770
     In Frames:    27
     In Discards:  0
     In Errors:    0
     Out Octets:   3682
     Out Frames:   27
     Out Discards: 0
     Out Errors:   0
  Time since counters last reset: 01:07:08

  queue 0: traffic class best-effort, bound to bridge FastEthernet1/1.1
    Queue length 0 bytes
```



```
          Forwarded packets 27, bytes 3898
          Dropped committed packets 0, bytes 0
          Dropped conformed packets 0, bytes 0
          Dropped exceeded packets 0, bytes 0
      vpls vplsB
        Port Number: 2
        Operational Status: Down
        Admin Status: Up
        Snmp Link Status Trap: Disabled
        Max Learn: Unlimited
        Subscriber Policy: default Trunk
        Statistics:
           In Octets:    0
           In Frames:    0
           In Discards:  0
           In Errors:    0
           Out Octets:   0
           Out Frames:   0
           Out Discards: 40
           Out Errors:   0
        Time since counters last reset: 01:04:10
```

To display a summary of all interfaces configured for the specified VPLS instance:

```
host1#show bridge vplsB interface brief
        Interface             Port      Status
-------------------------   ------   ------------
FastEthernet1/1.1            1        Up
ATM10/1.1.1                  2        Up
vpls vplsB                   3        Up
```

Meaning Table 121 on page 642 lists the **show bridge interface** command output fields.

Table 121: show bridge interface Output Fields

Field Name	Field Description
BridgeGroup	Name of the VPLS instance to which the interface belongs
Port Number	Port number on which this interface resides
Operational Status	Operational status of the physical interface: Up, Down, LowerLayerDown, NotPresent
Admin Status	State of the physical interface: Up, Down
Snmp Link Status Trap	Whether SNMP link status processing is enabled or disabled for the specified interface
Max Learn	Maximum number of dynamic MAC addresses that the interface can learn
Subscriber Policy	Name of the subscriber policy currently in effect for the interface
Statistics	Displays statistics information for the specified port

Table 121: show bridge interface Output Fields *(continued)*

Field Name	Field Description
In Octets	Number of octets received on this interface
In Frames	Number of frames received on this interface
In Discards	Number of incoming packets discarded on this interface
In Errors	Number of incoming errors received on this interface
Out Octets	Number of octets transmitted on this interface
Out Frames	Number of frames transmitted on this interface
Out Discards	Number of outgoing packets discarded on this interface
Out Errors	Number of outgoing errors on this interface
Time since counters last reset	Elapsed time since statistics counters were last reset
queue	Hardware packet queue associated with the specified traffic class and interface
Queue length	Length of the queue, in bytes
Forwarded packets, bytes	Number of packets and bytes forwarded on this queue
Dropped committed packets, bytes	Number of committed packets and bytes that were dropped
Dropped conformed packets, bytes	Number of conformed packets and bytes that were dropped
Dropped exceeded packets, bytes	Number of exceeded packets and bytes that were dropped
vpls *vplsName*	Identifies the VPLS virtual core interface for the VPLS instance

Table 122 on page 643 lists the **show bridge interface brief** command output fields.

Table 122: show bridge interface Output Fields

Field Name	Field Description
Interface	Interface type and specifier associated with the port
Port	Port number on which this interface resides

Table 122: show bridge interface Output Fields *(continued)*

Field Name	Field Description
Status	Operational status of the physical interface: Up, Down, LowerLayerDown, NotPresent

Related Documentation

- *show bridge interface*

Monitoring Configuration, Statistics, and Status for VPLS Core Interfaces

Purpose Display configuration, statistics, and status information for the VPLS virtual core interface associated with a VPLS instance.

Action To display information for the VPLS virtual core interface:

`host1#`**show bridge interface vpls vplsB**

```
vpls vplsB
  BridgeGroup: vplsB
  Port Number: 2
  Operational Status: Up
  Admin Status: Up
  Snmp Link Status Trap: Disabled
  Max Learn: Unlimited
  Subscriber Policy: default Trunk
  Statistics:
     In Octets:    0
     In Frames:    0
     In Discards:  0
     In Errors:    0
     Out Octets:   0
     Out Frames:   0
     Out Discards: 0
     Out Errors:   0
  Time since counters last reset: 00:12:53
```

Meaning Table 123 on page 644 lists the **show bridge interface vpls** command output fields.

Table 123: show bridge interface vpls Output Fields

Field Name	Field Description
BridgeGroup	Name of the VPLS instance to which the interface belongs
Port Number	Port number on which this interface resides
Operational Status	Operational status of the physical interface: Up, Down, LowerLayerDown, NotPresent
Admin Status	State of the physical interface: Up, Down
Snmp Link Status Trap	Whether SNMP link status processing is enabled or disabled for the specified interface

Table 123: show bridge interface vpls Output Fields *(continued)*

Field Name	Field Description
Max Learn	Maximum number of dynamic MAC addresses that the interface can learn
Subscriber Policy	Name of the subscriber policy currently in effect for the interface
Statistics	Displays statistics information for the specified port
In Octets	Number of octets received on this interface
In Frames	Number of frames received on this interface
In Discards	Number of incoming packets discarded on this interface
In Errors	Number of incoming errors received on this interface
Out Octets	Number of octets transmitted on this interface
Out Frames	Number of frames transmitted on this interface
Out Discards	Number of outgoing packets discarded on this interface
Out Errors	Number of outgoing errors on this interface
Time since counters last reset	Elapsed time since statistics counters were last reset
queue	Hardware packet queue associated with the specified traffic class and interface
Queue length	Length of the queue, in bytes
Forwarded packets, bytes	Number of packets and bytes forwarded on this queue
Dropped committed packets, bytes	Number of committed packets and bytes that were dropped
Dropped conformed packets, bytes	Number of conformed packets and bytes that were dropped
Dropped exceeded packets, bytes	Number of exceeded packets and bytes that were dropped
vpls *vplsName*	Identifies the VPLS virtual core interface for the VPLS instance

Related Documentation

- *show bridge interface vpls*

Monitoring Configuration, Statistics, and Status for VPLS Ports

Purpose Display configuration, statistics, and status information for ports (interfaces) associated with a VPLS instance.

Action To display information for VPLS ports:

host1#**show bridge vplsC port**

```
    FastEthernet1/1.1
      Port Number: 1
      Operational Status: Up
      Admin Status: Up
      Snmp Link Status Trap: Disabled
      Max Learn: Unlimited
      Subscriber Policy: samplepolicy
      Statistics:
         In Octets:    2018
         In Frames:    15
         In Discards:  0
         In Errors:    0
         Out Octets:   1930
         Out Frames:   14
         Out Discards: 0
         Out Errors:   0
      Time since counters last reset: 00:10:55

      queue 0: traffic class best-effort, bound to bridge FastEthernet1/1.1
        Queue length 0 bytes
        Forwarded packets 14, bytes 2042
        Dropped committed packets 0, bytes 0
        Dropped conformed packets 0, bytes 0
        Dropped exceeded packets 0, bytes 0

    vpls vplsC
      Port Number: 2
      Operational Status: Up
      Admin Status: Up
      Snmp Link Status Trap: Disabled
      Max Learn: Unlimited
      Subscriber Policy: default Trunk
      Statistics:
         In Octets:    0
         In Frames:    0
         In Discards:  0
         In Errors:    0
         Out Octets:   0
         Out Frames:   0
         Out Discards: 0
         Out Errors:   0
      Time since counters last reset: 00:07:07
```

To display a summary of all ports configured for the specified VPLS instance:

host1#**show bridge vplsTest port brief**

```
 Port            Interface              Status
------    -------------------------   ------------
1         FastEthernet1/1.1           Up
2         ATM10/1.1.1                 Up
3         vpls vplsTest               Up
```


Meaning Table 124 on page 647 lists the **show bridge port** command output fields.

Table 124: show bridge port Output Fields

Field Name	Field Description
BridgeGroup	Name of the VPLS instance to which the interface belongs
Port Number	Port number on which this interface resides
Operational Status	Operational status of the physical interface: Up, Down, LowerLayerDown, NotPresent
Admin Status	State of the physical interface: Up, Down
Snmp Link Status Trap	Whether SNMP link status processing is enabled or disabled for the specified interface
Max Learn	Maximum number of dynamic MAC addresses that the interface can learn
Subscriber Policy	Name of the subscriber policy currently in effect for the interface
Statistics	Displays statistics information for the specified port
In Octets	Number of octets received on this interface
In Frames	Number of frames received on this interface
In Discards	Number of incoming packets discarded on this interface
In Errors	Number of incoming errors received on this interface
Out Octets	Number of octets transmitted on this interface
Out Frames	Number of frames transmitted on this interface
Out Discards	Number of outgoing packets discarded on this interface
Out Errors	Number of outgoing errors on this interface
Time since counters last reset	Elapsed time since statistics counters were last reset
queue	Hardware packet queue associated with the specified traffic class and interface
Queue length	Length of the queue, in bytes
Forwarded packets, bytes	Number of packets and bytes forwarded on this queue

Table 124: show bridge port Output Fields *(continued)*

Field Name	Field Description
Dropped committed packets, bytes	Number of committed packets and bytes that were dropped
Dropped conformed packets, bytes	Number of conformed packets and bytes that were dropped
Dropped exceeded packets, bytes	Number of exceeded packets and bytes that were dropped
vpls *vplsName*	Identifies the VPLS virtual core interface for the VPLS instance

Table 125 on page 648 lists the **show bridge port brief** command output fields.

Table 125: show bridge port brief Output Fields

Field Name	Field Description
Port	Port number on which this interface resides
Interface	Interface type and specifier associated with the port
Status	Operational status of the physical interface: Up, Down, LowerLayerDown, NotPresent

Related Documentation

- *show bridge port*

Monitoring MAC Address Entries for a Specific VPLS Instance

Purpose Display information about the MAC address entries in the forwarding table for the specified VPLS instance.

Action To display information about the MAC address entries:

```
host1#show bridge vpls1 table
  Bridge: vpls1 MAC Address Table

   Address            Action                Interface                 Age
-------------  --------------------  -------------------------  -------
0009.01a0.002e  forward              ATM10/1.1.1                0
0090.1a41.3aca  forward              vpls (10)                  0
```

Meaning Table 126 on page 649 lists the **show bridge table** command output fields.

Table 126: show bridge table Output Fields

Field Name	Field Description
Bridge	Name of the VPLS instance for which the MAC address table is displayed
Address	MAC address of the entry
Action	Specifies how the VPLS instance handles this entry: forward or discard
Interface	Interface type and specifier on which the entry is forwarded; this value does not appear for entries that are discarded; vpls identifies the VPLS virtual core interface
Age	Length of time that a dynamic entry has been in the forwarding table; this value does not appear for static entries

Related Documentation

- *show bridge table*

Monitoring Subscriber Policy Rules

Purpose Display the set of forwarding and filtering rules for all subscriber policies configured on the router, or for a specified subscriber policy.

Action To display the rules for a specified subscriber policy:

```
host1#show subscriber-policy client01
  Subscriber: client01
  ARP                  : Permit
  Broadcast            : Permit
  Multicast            : Deny
  Unknown Destination : Deny
  IP                   : Permit
  Unknown Protocol     : Permit
  Unicast              : Permit
  PPPoE                : Permit
  Relearn              : Deny
  Mpls                 : Permit
```

To display the rules for all default and nondefault subscriber policies configured on the router:

```
host1#show subscriber-policy

Subscriber: default Subscriber
  ARP                  : Permit
  Broadcast            : Deny
  Multicast            : Permit
  Unknown Destination : Deny
  IP                   : Permit
  Unknown Protocol     : Permit
  Unicast              : Permit
  PPPoE                : Permit
```



```
  Relearn             : Permit
  Mpls                : Permit

Subscriber: default Trunk
  ARP                 : Permit
  Broadcast           : Permit
  Multicast           : Permit
  Unknown Destination : Permit
  IP                  : Permit
  Unknown Protocol    : Permit
  Unicast             : Permit
  PPPoE               : Permit
  Relearn             : Permit
  Mpls                : Permit

  Subscriber: client01
  ARP                 : Permit
  Broadcast           : Permit
  Multicast           : Deny
  Unknown Destination : Deny
  IP                  : Permit
  Unknown Protocol    : Permit
  Unicast             : Permit
  PPPoE               : Permit
  Relearn             : Deny
  Mpls                : Permit
```

Meaning Table 127 on page 650 lists the **show subscriber-policy** command output fields.

Table 127: show subscriber-policy Output Fields

Field Name	Field Description
Subscriber	Name of the subscriber policy
Permit	Indicates that the subscriber interface forwards packets of the specified type. For the relearn attribute, specifies that relearning a MAC address entry on a different interface from the one initially associated with this entry in the forwarding table is allowed on this interface
Deny	Indicates that the subscriber interface filters packets of the specified type. For the relearn attribute, specifies that relearning is prohibited on this interface

Related Documentation

- *show subscriber-policy*

Monitoring Layer2 NLRI for VPLS Instances

Purpose Display layer 2 NLRI for all VPLS instances in the L2VPN address family, for a particular VPLS instance in the L2VPN address family, or for a particular VPLS instance in the VPLS address family.

The **l2vpn vpls** keywords display layer 2 NLRI for a particular VPLS instance in the VPLS address family.

The **l2vpn all** keywords display layer 2 NLRI for all VPLS instances in the L2VPN address family. The output for this version of the command also includes information about any VPWS instances configured in the L2VPN address family.

To display layer 2 NLRI for the route that matches a specified prefix (site ID and block offset) in the L2VPN address family or in the VPLS address family, use the **site-id** and **block-offset** keywords.

Action To display information for all VPLS instances (and all VPWS instances) in the L2VPN address family:

```
host1#show ip bgp l2vpn all
Local BGP identifier 1.1.1.1, local AS 100
  4 routes (264 bytes)
  4 destinations (288 bytes) of which 4 have a route
  0 routes selected for route tables installation
  0 unicast/multicast routes selected for route table installation
  0 unicast/multicast tunnel-usable routes selected for route table installation

  0 tunnel-only routes selected for tunnel-route table installation
  4 path attribute entries (608 bytes)
  Local-RIB version 11. FIB version 11.
Status codes: > best, * invalid, s suppressed, d dampened, r rejected,
              a auto-summarized
Prefix                      Peer             Next-hop       MED LocPrf Weight
Origin
> 1:1                      0.0.0.0         self                          0 IGP

> 1:1                      0.0.0.0         self                          0 IGP

> 2:1                      2.2.2.2         2.2.2.2              100      0 IGP

> 2:1                      2.2.2.2         2.2.2.2              100      0 IGP
```

To display summary information for the L2VPN address family:

```
host1#show ip bgp l2vpn all summary
Display summary information for the l2vpn address-family
Local router ID 1.1.1.1, local AS 100
  Administrative state is Start
  BGP Operational state is Up
  Shutdown in overload state is disabled
  Default local preference is 100
  IGP synchronization is enabled
  Default originate is disabled
  Always compare MED is disabled
  Compare MED within confederation is disabled
  Advertise inactive routes is disabled
  Advertise best external route to internal peers is disabled
  Enforce first AS is disabled
  Missing MED as worst is disabled
  Route flap dampening is disabled
  Log neighbor changes is disabled
  Fast External Fallover is disabled
  No maximum received AS-path length
  BGP administrative distances are 20 (ext), 200 (int), and 200 (local)
  Client-to-client reflection is enabled
  Cluster ID is 1.1.1.1
  Route-target filter is enabled
```

```
  Default IPv4-unicast is enabled
  Check next-hops of vpn routes is disabled
  Redistribution of iBGP routes is disabled
  Graceful restart is globally disabled
  Global graceful-restart restart time is 120 seconds
  Global graceful-restart stale paths time is 360 seconds
  Graceful-restart path selection defer time is 360 seconds
  Graceful-restart is not ready to switch to the standby SRP
  The last restart was not graceful
  Local-RIB version 11. FIB version 11.
                                                Messages  Messages  Prefixes
Neighbor          AS State       Up/down time      Sent  Received  Received
2.2.2.2          100 Established     00:30:35        65        65         2
```

To display information for the route that matches the specified prefix (2:1) for a VPLS instance named customer1 in the VPLS address family:

`host1#`**show ip bgp l2vpn vpls customer1 site-id 2 block-offset 1**

```
BGP route information for prefix 2:1
  Received route learned from internal peer 2.2.2.2 (best route)
    Leaked route
    Route placed in IP forwarding table
    Best to advertise to external peers
    Address Family Identifier (AFI) is layer2
    Subsequent Address Family Identifier (SAFI) is unicast
    Route Distinguisher (RD) is 100:11
    Original Route Distinguisher (RD) is 100:21
    MPLS in-label is none
    MPLS in-label block size is 0
    MPLS out-label is 46
    MPLS out-label block size is 20
    Next hop IP address is 2.2.2.2 (metric 3)
    Multi-exit discriminator is not present
    Local preference is 100
    Weight is 0
    Origin is IGP
    AS path is empty
    Extended communities RT:100:1 Layer 2:19:00:0
```

To display only certain information for all VPLS instances (and all VPWS instances) in the L2VPN address family, including the status of the route:

`host1:pe1#`**show ip bgp l2vpn all fields best rd peer next-hop loc-pref extended-communities next-hop-cost**

```
Prefix  Rd     Peer    Next-hop  Next-hop-cost LocPrf Extended-communities
> 1:1  100:11  0.0.0.0   self       0       200 RT:100:1 Layer2:vpls:mtu=0
>m1:1  100:22  10.2.2.2  10.2.2.2   3       100 RT:100:1 Layer2:vpls:mtu=0
> 2:1  100:33  10.3.3.3  10.3.3.3   3       200 RT:100:1 Layer2:vpls:mtu=0
>m2:1  100:44  10.4.4.4  10.4.4.4   3       0 RT:100:1 Layer2:vpls:down,:mtu=0
> 3:1  100:55  10.5.5.5  10.5.5.5   3       100 RT:100:1 Layer2:vpls:mtu=0
```

Meaning Table 128 on page 653 lists the **show ip bgp l2vpn** command output fields.

Table 128: show ip bgp l2vpn Output Fields

Field Name	Field Description
Local BGP identifier	IP address of the local VE router
local AS	Autonomous system number
Local-RIB version	Version number of the local routing information base
FIB version	Version number of the forwarding information base
Status codes	Status codes for the route, listed before the Prefix: • >—best route • *—invalid route • s—suppressed route • d—dampened route • r—rejected route • a—auto-summarized route • m—multihomed backup route; not used to trigger pseudowire
Prefix	Route prefix in the format *siteID:blockOffset*
Peer	IP address of the peer from which the route was learned
Next-hop (or Next hop IP address)	IP address of the next router that is used when a packet is forwarded to the destination network
MED	Multiexit discriminator for the route
LocPrf	Local preference for the route
Weight	Weight of the route
Origin	Origin of the route
AS path	AS path through which this route has been advertised
Extended communities	Description of the extended communities associated with this route. Includes route target, community type, encapsulation, control word and sequencing use, L2VPN link MTU, layer 2 D-bit (Site-is-Down) setting.

Related Documentation

- *show ip bgp* l2vpn
- *show ip bgp* l2vpn vpls

Monitoring BGP Next Hops for VPLS

Purpose Display information about BGP next hops in the L2VPN address family or in the VPLS address family.

Action To display next hop information that matches the specified indirect next-hop address (2.2.2.2) in the L2VPN address family:

```
host1#show ip bgp l2vpn all next-hops 2.2.2.2
Indirect next-hop 2.2.2.2
  Resolution in IP route table of VR
       IP indirect next-hop index 2
       Reachable (metric 3)
       Number of direct next-hops is 1
              Direct next-hop ATM2/0.10 (10.10.10.2)
  Resolution in IP tunnel-route table of VR
       MPLS indirect next-hop index 19
       Reachable (metric 3)
       Number of direct next-hops is 1
              Direct next-hop 0000000c
  Reference count is 2
```

Meaning Table 129 on page 654 lists the **show ip bgp next-hops** command output fields.

Table 129: show ip bgp next-hops Output Fields

Field Name	Field Description
Indirect next-hop	BGP next-hop attribute received in the BGP update message
Resolution	Describes where the indirect next hop is resolved (the IP routing table, the IP tunnel routing table, or both) and whether this is in a VR or VRF
IP indirect next-hop index	Index number of the IP indirect next hop that corresponds to the BGP indirect next hop and its resolution
MPLS indirect next-hop index	Index number of the MPLS indirect next hop that corresponds to the BGP indirect next hop and its resolution
Reachable	Indicates whether or not the indirect next hop is reachable. For more information about the reachability rules that apply for various route types, see the command description for *show ip bgp next-hops*
metric	Metric number of the BGP indirect next hop
Number of direct next-hops	Number of the equal-cost legs of direct next hops to which this indirect next hop resolves
Direct next-hop	MPLS next-hop index that resolves the MPLS indirect next hop
Reference count	Number of label mappings of BGP routes that use this next hop

Related Documentation

- *show ip bgp next-hops*

Monitoring LDP-Related Settings for VPLS

Purpose Display MPLS configuration information for a VPLS instance that uses LDP as the signaling protocol.

Action To display information for all VPLS instances configured on the virtual router:

```
host1:ve1#show ldp vpls
  Vpls      Vpls   Remote
Instance     Id      PE       In-label    Out-label
--------    ----   -------    --------    ----------
vpls1          1   2.2.2.2          25            27
vpls2          2   2.2.2.2          26            28
```

Meaning Table 130 on page 655 lists the **show ldp vpls** command output fields.

Table 130: show ldp vpls Output Fields

Field Name	Field Description
Vpls Instance	Name of the VPLS instance for which the configuration information is displayed
Vpls Id	Globally unique identifier for the VPLS domain
Remote PE	IP address of the remote VE (also known as the PE) router
In-label	Incoming MPLS label from the remote site
Out-label	Outgoing MPLS label used to reach the remote site

Related Documentation

- *show ldp vpls*

Monitoring MPLS-Related Settings for VPLS

Purpose Display MPLS-related settings for VPLS instances.

Action To display information for a specific MPLS label being used for forwarding:

```
host1:ve1#show mpls forwarding label 17
In label: 17
  Label space: platform label space
  Owner: bgp
  Spoof check: router pe1
  Action:
    MPLS next-hop: 3, Forward to bridge-group customer1
  Statistics:
    0 in pkts
    0 in Octets
```

```
    0 in errors
    0 in discard pkts
```

To display summary information for all MPLS labels being used for forwarding:

`host1:ve1#`**show mpls forwarding brief**

```
In-label   Owner    Action
--------   -----    ----------------------------------
17         bgp      Forward to bridge-group customer1
27         bgp      Forward to bridge-group customer2
```

Meaning Table 131 on page 656 lists the **show mpls forwarding** command output fields.

Table 131: show mpls forwarding Output Fields

Field Name	Field Description
In label	Label sent to upstream neighbor for route
Out label	Label received from downstream neighbor for route
Label space	Label space in which the label is assigned
Owner	Signaling protocol that placed the label in the forwarding table: BGP, LDP, or RSVP-TE
Spoof check	Type and location of spoof checking performed on the MPLS packet, router, or interface
Action	Action taken for MPLS packets arriving with that label
in pkts	Number of packets sent with the label
in Octets	Number of octets sent with the label
in errors	Number of packets that are dropped for some reason before being sent
in discardPkts	Number of packets that are discarded due to lack of buffer space before being sent

Related Documentation

- *show mpls forwarding*

Monitoring VPLS-Specific Settings

Purpose To display information about all VPLS instances configured on the router.

Action To display detailed information about all VPLS instances configured on the router:

`host1:ce3#`**show vpls connections**

```
  BridgeGroup: vpls1(vpls)
```



```
    SiteName:               bangalore
    SiteId:                 1
    Transport Virtual Rtr:  pe1
    Multihomed priority:    200

Connections status code:
UP = Operational
SC = Local and Remote Site Identifier Collision
EM = Encapsulation Mismatch
OR = Out of Range
DN = VC Down because Remote PE Unreachable
LD = Local Site Down
RD = Remote Site Down
AS = Max BGP AS path length exceeded
OL = No Out Label
LN = Local Site not Designated
RN = Remote Site not Designated

Site   State     Remote PE     In-label  Out-label  MPLS NH Idx  Up-down Time
-----  -----  ---------------  --------  ---------  -----------  ------------
2      UP     3.3.3.3          17        36         79           00:04:09
3      UP     5.5.5.5          18        56         54           00:04:09
```

host1# **show vpls connections details**

```
  BridgeGroup: vpls1(vpls)
        Bridge Mode:            default
        Aging Time:             300 secs
        Learning:               Enabled
        Max Learn:              Unlimited
        Link Status Snmp Traps: Disabled
        Subscriber Policy:      default Subscriber
        Port Count:             2
        Interface Count:        1
        Transport Virtual Rtr:  pe1
          Route Distinguisher:    1.1.1.1:10
        SiteName:               westford
        SiteId:                 1
        SiteRange:              10
        VPLS Route Targets
          Route Target: RT:100:1 (both)
        Flood Next Hop: Index 1048577
          MPLS next-hop: 20, label 46, resolved by MPLS next-hop 19
            MPLS next-hop: 19, resolved by MPLS next-hop 17, peer 2.2.2.2
              MPLS next-hop: 17, label 82 on ATM2/0.10, nbr 10.10.10.2
        Interface            Port       Status
-------------------------   ------   ------------
FastEthernet3/1             1        Up
vpls vpls1                  2        Up
Connections status code:
UP = Operational
SC = Local and Remote Site Identifier Collision
EM = Encapsulation Mismatch
OR = Out of Range
DN = VC Down because Remote PE Unreachable
LD = Local Site Down
RD = Remote Site Down
AS = Max BGP AS path length exceeded
OL = No Out Label
Site   State     Remote PE     In-label  Out-label  MPLS NH Idx  Up-down Time
-----  -----  ---------------  --------  ---------  -----------  ------------
2      UP     2.2.2.2          17        46         20           00:02:56
```



```
 BridgeGroup: vpls2(vpls)
       Bridge Mode:            default
       Aging Time:             300 secs
       Learning:               Enabled
       Max Learn:              Unlimited
       Link Status Snmp Traps: Disabled
       Subscriber Policy:      default Subscriber
       Port Count:             2
       Interface Count:        1
       Transport Virtual Rtr:  pe1
         Route Distinguisher:    1.1.1.1:10
       SiteName:               westford
       SiteId:                 1
       SiteRange:              20
       VPLS Route Targets
         Route Target: RT:100:2 (both)
       Flood Next Hop: Index 1048578
         MPLS next-hop: 21, label 56, resolved by MPLS next-hop 19
           MPLS next-hop: 19, resolved by MPLS next-hop 17, peer 2.2.2.2
             MPLS next-hop: 17, label 82 on ATM2/0.10, nbr 10.10.10.2
       Interface                 Port       Status
-------------------------   ------   ------------
ATM2/0.12                   1        Up
vpls vpls2                  2        Up
Connections status code:
UP = Operational
SC = Local and Remote Site Identifier Collision
EM = Encapsulation Mismatch
OR = Out of Range
DN = VC Down because Remote PE Unreachable
LD = Local Site Down
RD = Remote Site Down
AS = Max BGP AS path length exceeded
OL = No Out Label
Site   State     Remote PE      In-label  Out-label  MPLS NH Idx  Up-down Time
-----  -----  ---------------  --------  ---------  -----------  ------------
2      UP     2.2.2.2          27        56         21           00:02:56
```

Meaning Table 132 on page 658 lists the **show vpls connections** command output fields.

Table 132: show vpls connections Output Fields

Field Name	Field Description
BridgeGroup	Name of the VPLS instance for which information is displayed
Bridge Mode	Bridging capability currently enabled; for a VPLS instance, this field always displays default
Aging Time	Length of time, in seconds, that a MAC address entry can remain in the forwarding table before expiring
Learning	Whether acquisition of dynamically learned MAC addresses is enabled or disabled
Max Learn	Maximum number of dynamic MAC addresses that the VPLS instance can learn

Table 132: show vpls connections Output Fields *(continued)*

Field Name	Field Description
Link Status Snmp Traps	Whether SNMP link status processing is enabled or disabled
Subscriber Policy	Name of the subscriber policy currently in effect
Port Count	Number of ports currently configured for the VPLS instance, including network interfaces and the VPLS virtual core interface
Interface Count	Number of network interfaces currently configured for the VPLS instance
Transport Virtual Rtr	Name of the transport virtual router configured for the VPLS instance
Route Distinguisher	Unique route distinguisher configured for the VPLS instance
SiteName	Site name configured for the VPLS instance
Transport Virtual Rtr	Identifier for the PE router that acts as the transport virtual router for the VPLS instance
Multihomed priority	Priority of the VPLS instance to serve as the backup PE router for the CE device in the event of a network failure in the multihomed configuration; indicates also that the site is multihomed
SiteId	Numerical site identifier configured for the VPLS instance
SiteRange	Maximum number of sites that can participate in the VPLS domain associated with the VPLS instance
VPLS Route Targets	Extended community identifiers, also known as route targets, for each VPLS instance configured on the router
Flood Next Hop	Index number of the MPLS next hop to which the router floods packets with unknown destination addresses. For more information about displaying MPLS next hops and any available next-hop statistics, see “Monitoring MPLS Next Hops” on page 361.
Interface	Type and specifier of the network interfaces and VPLS virtual core interface associated with the VPLS instance; vpls *vplsName* in this field identifies the VPLS virtual core interface
Port	Port number of the module on which the network interface or VPLS virtual core interface resides
Status	Operational status of the physical interface: Up, Down, LowerLayerDown, NotPresent
Connections status code	Possible status codes for the VPLS connection that appear in the State field
Site	Remote site identifier

Table 132: show vpls connections Output Fields *(continued)*

Field Name	Field Description
State	Status of the connection with the remote VPLS instance; possible values for this field appear in the Connections status code legend in the command display
Remote PE	IP address of the remote VPLS edge (VE) router, which is analogous to the remote provider edge (PE) router in a BGP/MPLS VPN configuration
In-label	Incoming MPLS label from the remote site
Out-label	Outgoing MPLS label used to reach the remote site
MPLS NH Idx	MPLS next-hop index number that corresponds to the outgoing MPLS label
Up-down Time	Time since the last state change for this VPLS connection

Related Documentation

- *show vpls connections*

PART 5

Virtual Private Wire Service

CHAPTER 14

VPWS Overview

This chapter describes virtual private wire service (VPWS) L2VPNs, and contains the following sections:

VPWS Overview

VPWS L2VPNs employ layer 2 services over MPLS to build a topology of point-to-point connections that connect end customer sites in a VPN. These L2VPNs provide an alternative to private networks that have been provisioned by means of dedicated leased lines or by means of layer 2 virtual circuits that employ ATM or Frame Relay. The service provisioned with these L2VPNs is known as Virtual Private Wire Service (VPWS). VPWS L2VPNs are sometimes called *Kompella* L2VPNs. You configure a VPWS *instance* on each associated edge router for each VPWS L2VPN.

Traditional VPNs over layer 2 circuits require the provisioning and maintenance of separate networks for IP and for VPN services. In contrast, VPWS enables the sharing of a provider's core network infrastructure between IP and L2VPN services, reducing the cost of providing those services.

VPWS also uses BGP as the signaling protocol, and consequently has a simpler design and requires less provisioning overhead than traditional VPNs over layer 2 circuits. BGP signaling also enables autodiscovery of L2VPN peers. VPWS is similar to BGP/MPLS VPNs and VPLS in many respects, because all three types of services employ BGP for signaling.

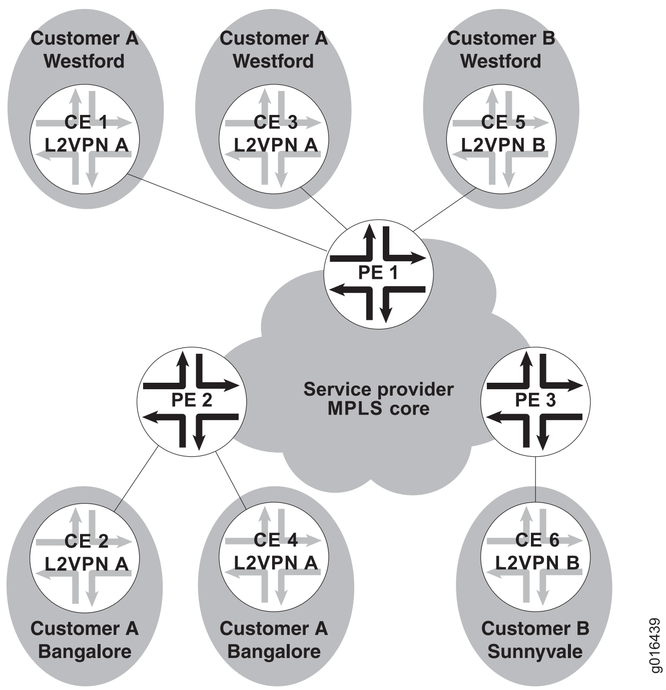
VPWS creates pseudowires that emulate layer 2 circuits. A virtual private LAN service (VPLS) network is similar to VPWS, but provides point-to-multipoint traffic forwarding in contrast to the VPWS L2VPN's point-to-point traffic forwarding.

VPWS provides the same services as layer 2 over MPLS except for CE-side load-balancing. The main differences between the VPWS and L2 over MPLS services are signaling, autodiscovery, and configuration.

A VPWS L2VPN can have either a full-mesh or a hub-and-spoke topology. The tunneling mechanism in the core network typically is MPLS. However, VPWS can also use other tunneling protocols, such as GRE. VPWS is similar to Martini layer 2 services over MPLS, and employs a similar encapsulation scheme for forwarding traffic.

Figure 133 on page 664 illustrates an example of a simple VPWS L2VPN topology.

Figure 133: VPWS Sample Topology



In this example, the service provider offers VPWS services to Customer A and Customer B. Customer A wants to create a full mesh of point-to-point links between Westford and Bangalore. Customer B needs only a single point-to-point link between Westford and Sunnyvale. The service provider uses BGP and MPLS signaling in the core, and creates a set of unidirectional pseudowires at each provider edge (PE) router to separately cross-connect each customer's layer 2 circuits.

In order to provision this service, the provider configures two VPWS L2VPNs, L2VPN A and L2VPN B. An encapsulation type is configured for each VPWS L2VPN. All interfaces in a given VPWS L2VPN must be configured with the VPWS L2VPN's encapsulation type. The layer 2 interfaces that connect the PE router and CE device pairs are configured to be members of the corresponding VPWS L2VPN, L2VPN A or L2VPN B.

Local and remote site information for the interfaces identifies the cross-connect. Local cross-connects are supported when the interfaces that are connected belong to two different sites configured in the same VPWS instance and on the same PE router.

BGP advertises reachability for the VPNs. The BGP configuration is similar to that used for other VPN services, such as layer 3 VPNs and VPLS. MPLS is configured to set up base LSPs to the remote PE routers similarly to the other VPN services.

Related Documentation

- VPWS Supported Features on page 674
- VPWS Platform Considerations on page 675
- VPWS References on page 676

BGP Signaling for L2VPNs Overview

When you configure VPWS at a given PE router for a given L2VPN customer, BGP signals reachability for all sites that belong to that L2VPN. This signaling is identical to the signaling used for BGP/MPLS VPNs and VPLS. The network layer reachability information (NLRI) for both services are encoded in a similar manner.

A new NLRI format carries the individual VPWS information listed in Table 133 on page 665. One or more of these NLRIs is carried in the MP_REACH_NLRI and MP_UNREACH_NLRI BGP attributes.

Table 133: Components of VPWS NLRI

NLRI value	Size in octets
Length	2
Route Distinguisher	8
CE-ID	2
Label-block Offset	2
Label Base	3
Variable TLVs	0–*n*

The local PE router selects a contiguous label block to cover all the remote sites for a given VPWS instance. The local PE router then advertises that label block as part of the reachability information for a given CE device in a particular VPWS instance; (the NLRI contains the CE-ID (site ID) for the CE device. This label block represents the set of demultiplexers that are used to cross-connect incoming MPLS traffic to a specific local interface in the VPWS instance.

The local PE router also processes advertisements from all remote PE routers and for each local interface in an VPWS instance. The local PE router selects a demultiplexer label from a label block received from the remote PE router associated with each remote site in the VPWS instance. Traffic coming into the local interface from the CE device is cross-connected to an MPLS next hop that corresponds to the demultiplexer. Traffic is then encapsulated in MPLS and sent across the MPLS core to the remote PE router in the L2VPN.

The same address family identifier (AFI) and subsequent address family identifier (SAFI) are used in the NLRI for VPWS and VPLS.

The VPWS NLRIs must be accompanied by a route-target extended community. PE routers that receive VPN information can filter route advertisements with the route target import lists and export lists. This route filtering enables the PE routers to control CE-to-CE connectivity or full-mesh, hub-and-spoke, and overlapping VPNs as is done in L3VPNs.

A VPWS NLRI is uniquely identified by the route distinguisher, site ID (CE-ID), and the label block offset.

In addition to the site ID and label block information, BGP also signals control flags that indicate whether a control word is included in the encapsulation and whether packets have a sequence number. If a control word mismatch occurs, the pseudowire remains in a down state with a status of control word mismatch.

A control status vector is sent along with the other NLRI information. This vector carries the operational state of the local layer 2 interfaces between the PE router and CE device for a given VPWS instance. A TLV type of 1 is used currently to interoperate with Junos OS.

For information about configuring BGP/MPLS VPNs, see "Configuring BGP-MPLS Applications" on page 389. For information about configuring VPLS, see "Configuring VPLS" on page 613.

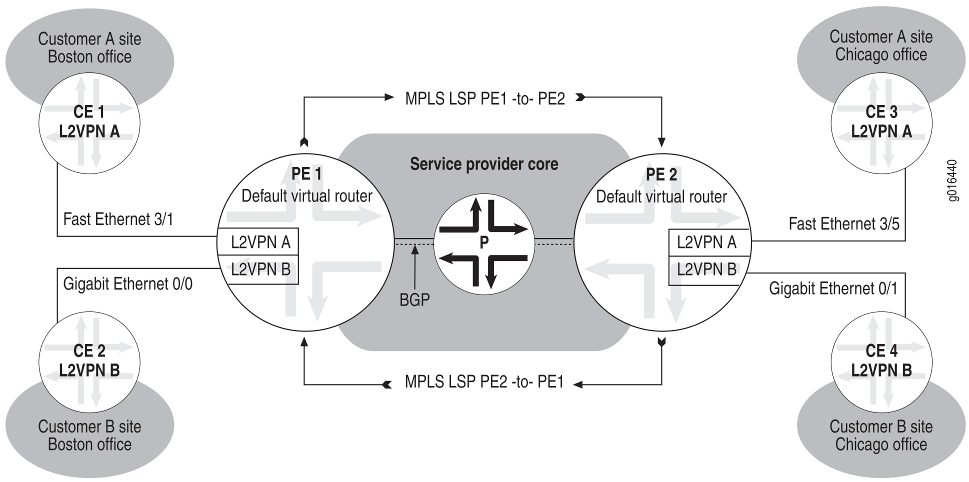
Related Documentation

- BGP/MPLS VPN Components Overview on page 394
- VPLS Protocol Overview on page 597

VPWS Components Overview

Figure 134 on page 667 shows the components of a typical VPWS L2VPN topology.

Figure 134: VPWS Components



- VPWS Instances on page 667
- Customer Edge Devices on page 667
- VPWS Provider Edge Devices on page 667

VPWS Instances

Typically, a VPWS is associated with customers who want to use L2VPNs to connect geographically dispersed sites in their organization across an MPLS-based service provider core, also known as an MPLS backbone. Each VPWS L2VPN consists of the set of provider edge routers running the corresponding VPWS instance. To provide connectivity for the L2VPN, BGP builds pseudowires between the VPWS instances on the provider edge routers participating in a particular VPWS L2VPN point-to-point connection.

Figure 134 on page 667 depicts two L2VPNs: L2VPN A and L2VPN B. L2VPN A connects Customer A's Boston and Chicago offices, and consists of provider edge routers PE 1 and PE 2, each of which runs a VPWS instance named l2vpnA. Similarly, L2VPN B connects Customer B's Boston and Chicago offices, and consists of provider edge routers PE 1 and PE 2, each of which also runs a VPWS instance named l2vpnB.

Customer Edge Devices

Figure 134 on page 667 shows four customer edge devices: CE 1, CE 2, CE 3, and CE 4. Each CE device is located at the edge of a customer site. In the sample topology, CE 1 and CE 3 are members of L2VPN A, and CE 2 and CE 4 are members of L2VPN B.

A CE device can be a single host, a switch, or, most typically, a router. Each CE device is directly connected to an VPWS provider edge router by means of a layer 2 interface.

VPWS Provider Edge Devices

In a VPWS configuration, E Series routers function as provider edge devices, which are also referred to as PE routers. These PE routers perform a similar function to PE routers in a BGP/MPLS VPN configuration.

Figure 134 on page 667 depicts two PE routers: PE 1, which is the local router, and PE 2, which is the remote router located at the other side of the service provider core. Each PE router must have an VPWS instance configured for each L2VPN in which it participates. Consequently, the sample topology comprises four separate VPWS instances: instances l2vpnA and l2vpnB configured on PE 1, and instances l2vpnA and l2vpnB configured with matching route target values on PE 2.

Each VPWS instance configured on the router is associated with two types of interfaces. The CE-facing or customer-facing interface is a layer 2 interface that directly connects the PE router to a local CE device. The router encapsulates layer 2 frames from the CE device in an MPLS packet and then forwards the encapsulated frames to the service provider core from an MPLS interface through the provider (P) router. This encapsulation is identical to Martini encapsulation for layer 2 services over MPLS.

Related Documentation

- VPWS Overview on page 663
- Configuring a VPWS Instance on page 678
- Types of Interfaces to Configure in the VPWS Instance on page 680

VPWS and BGP/MPLS VPNs Overview

BGP multiprotocol extensions (MP-BGP) enable BGP to support IPv4 services such as BGP/MPLS VPNs, which are sometimes known as RFC 2547bis VPNs. A VPWS L2VPN is actually a BGP-MPLS application that has much in common with BGP/MPLS VPNs.

The procedures for configuring BGP signaling for BGP/MPLS VPNs and for VPWS L2VPNs are similar except that for VPWS L2VPNs you must configure both of the following address families:

- L2VPN—The L2VPN address family enables you to configure the PE router for VPWS (or VPLS) L2VPNs to exchange layer 2 network layer reachability information (NLRI) for all VPWS (or VPLS) instances. Optionally, you can use the **signaling** keyword with the address-family command for the L2VPN address family to specify BGP signaling of VPWS reachability information. Currently, you can omit the signaling

  keyword with no adverse effects.
- VPWS—The VPWS address family enables you to configure the PE router to exchange layer 2 NLRI for a specified VPWS instance.

BGP can exchange information in an L2VPN topology within these address families.

For information about configuring BGP/MPLS VPNs, see “Configuring BGP-MPLS Applications” on page 389. For information about configuring BGP routing, see “Configuring BGP Routing” on page 3.

Related Documentation

- BGP/MPLS VPN Components Overview on page 394
- VPWS Overview on page 663

BGP Multihoming for VPWS Overview

BGP multihoming enables you to connect a customer site to two or more PE routers to provide redundant connectivity while preventing the formation of layer 2 loops in the service provider's network. The redundant connectivity maintains the VPWS service and traffic forwarding to and from the multihomed site in the event of a PE router-to-CE device link failure, the failure of a PE router, or an MPLS reachability failure between the local PE router and a remote PE router. A redundant PE router can begin providing service to the customer site as soon as the failure is detected. BGP multihoming is very similar for both VPLS and VPWS, with only minor differences in behavior between the two L2VPN types. Multihoming is not applicable for VPWS local cross-connects because the local and remote CE devices are connected to the same PE.

When a CE device connects to multiple PE routers, each of these routers advertises reachability for the multihomed site. The other PE routers in the network use a BGP path selection process to select only one of the advertising routers to which they send traffic destined for the CE device. This path selection process eliminates layer 2 loops in the VPWS network.

BEST PRACTICE: **To prevent the creation of layer 2 loops due to a misconfiguration or temporary loops during a topology change and subsequent convergence, we recommend that you employ the Spanning Tree Protocol (STP) on your CE devices.**

You specify on each PE router connected to the CE device in the VPWS that the site is multihomed and you configure a priority. The priority serves as a site preference and is propagated by BGP in the local-preference attribute.

You configure the same site ID (sometimes referred to as a VE ID) on these connected PE routers. Each of these routers then advertises reachability for the multihomed site; the VPWS NLRI contains the site ID. The site ID shared by the connected PE routers should be different than the site IDs configured on the remote PE routers in the VPWS network; if the site ID is not different, then the pseudowire will be in a site collision state. The remote routers then use the site ID to identify where to forward traffic destined for the customer site.

Although the site ID is the same for all connected PE routers, the block offset, label range, and route distinguisher can be different for each PE router. The BGP path selection process uses the block offset and label range only to determine whether a layer 2 advertisement is relevant to the multihomed customer site. A route distinguisher is helpful to uniquely identify a particular PE router when you are troubleshooting a network.

The PE routers run the BGP path selection process on the locally originated and received layer 2 route advertisements to establish that the routes are suitable for advertisements to other peers, such as route reflectors. For this selection process, the routes advertise different prefixes, distinguished by the site ID, block offset, and route distinguisher.

The remote PE routers then run a modified selection process on these selected routes for L2VPN multihoming. Because all the prefixes advertised by multihomed local PE routers share the same site ID, the set of routes advertised for a multihomed site effectively consists of multiple routes to a single prefix, distinguished by the site ID alone. Therefore the result of the second selection process is the single best path to the multihomed site.

The PE router that originates this advertisement then becomes the designated VE device for the multihomed customer site. When the designated VE device is determined for both the local and remote customer sites for the VPWS, then a VPWS pseudowire is created between the designated VE devices.

The BGP best path selection process is run only in the core VPN address family. This first selection process does not consider the status vector bit for VPWS (or the down bit for VPLS).

The layer 2 multihoming decision process is run only in the non-core VPWS (or VPLS) layer 2 unicast address families. This second decision process treats prefixes with the same site ID but different RDs as unique prefixes.

When the PE router receives a layer 2 BGP advertisement that has the down bit set, inbound policy sets the local preference attribute to zero. The selection process can then choose an existing route from an alternate PE router, if available.

When a PE router in a VPWS domain is also a BGP route reflector (RR), the path selection process to determine the VE device for the multihomed site has no effect on the path selection process performed by this PE router for the purpose of reflecting layer 2 routes.

Layer 2 prefixes that have different route distinguishers are considered to have different NLRI for route reflection. This result of the L2VPN multihoming decision process enables the RR to reflect all routes that have different route distinguishers to all other RR clients even though only one of these routes is used to trigger the VPWS pseudowire to the multihomed site.

Related Documentation

- VPWS Overview on page 663
- Designated VE Device Selection for a Multihomed Site on page 671
- Multihoming Reaction to Failures in the Network on page 673
- Configuring BGP Multihoming for VPWS on page 679

Designated VE Device Selection for a Multihomed Site

BGP on each PE router in the VPWS network determines the best path to the multihomed site by comparing path attributes. The PE routers receiving the advertised routes first run the standard BGP selection process. The routes from the connected multihomed PE routers all share the same site ID, but can have different route distinguishers and block offsets; the routers are advertising different prefixes. The following sequence is applied to all routes on a per-prefix basis:

1. Select a path with a reachable *next hop*.
2. Select the path with the highest *weight*.
3. If path weights are the same, select the path with the highest *local preference* value.
4. Prefer locally originated routes (network routes, redistributed routes, or aggregated routes) over received routes.
5. Select the route with the shortest *AS-path* length.
6. If all paths have the same AS-path length, select the path based on *origin*: IGP is preferred over EGP; EGP is preferred over Incomplete.
7. If the origins are the same, select the path with lowest *MED* value.
8. If the paths have the same MED values, select the path learned by means of EBGP over one learned by means of IBGP.
9. Select the path with the lowest IGP cost to the next hop.
10. Select the path with the shortest route reflection cluster list. Routes without a cluster list are treated as having a cluster list of length 0.
11. Select the path received from the peer with the lowest BGP router ID.
12. Select the path that was learned from the neighbor with the lowest peer remote address.

The result of this process is the best path to the multihomed customer site through each PE router connected to the site. One best path is selected for each router. The process establishes whether the route advertised by each PE router is suitable for advertising to peer routers.

Next, BGP runs the layer 2 multihoming selection process on this set of best paths to determine the one best path the customer site. The result of this process establishes that the best path is suitable for establishing a pseudowire from the remote PE router to the PE router. That PE router is accordingly selected as the designated VE device.

The multihoming selection process is similar to the standard BGP process, but it omits two steps:

- The process does not prefer locally originated routes. Local origination is of no value in establishing the designated VE device. The PE routers connected to the customer site always have a local route and therefore all advertise a locally-originated route. These PE router also receive the advertisements from the other connected PE routers.

If the multihoming selection process preferred local origination, each of these routers would select itself as the best path.

- The process does not consider IGP cost in order to prevent improperly designated VE device selection by the remote PE routers.

When the remote PE router establishes or refreshes a pseudowire to the local PE router, it verifies whether the prefix is in the range required for the site ID based on the block offset and label range advertised by the designated VE device. If the prefix is out of range, then the pseudowire status is set to OR (out of range).

For VPWS only, the layer 2 multihoming decision process considers the status vector bit in the layer 2 route. The status vector bit gives the combined status of the following:

- The attachment circuit between the CE device and the local PE router
- The MPLS connectivity between the local PE router and the remote PE router

When the status vector bit indicates that the connectivity between the CE device and the remote CE device is down, then the PE router runs the layer 2 multihoming decision process again, but this time excludes all advertisements that indicate this lack of connectivity between the two site IDs (CE-IDs). The status vector bit is not considered for VPLS multihoming.

This evaluation of connectivity must be supported by all PE routers in the VPWS network. Because some implementations do not support the status vector bit, The E Series routers also advertise the down bit in a VPWS network. This bit is then used by the other routers to evaluate connectivity.

One of the following cases applies for each PE router when it completes the BGP path selection process for a layer 2 advertisement on the VPWS.

- The PE router originated one of the multihomed advertisements and selected its own advertisement as the best path.

  This PE router hosts the designated VE device. Selection as the designated VE device triggers the creation of pseudowires to and from the other PE routers in the VPWS. When the remote customer site is also multihomed, then the designated VE device triggers the creation of pseudowires to and from only the designated VE device for the remote site.

- The PE router originated one of the multihomed advertisements but did not select its own advertisement as the best path.

  This PE router is one of the redundant PE routers for the multihomed site; it does not host the designated VE device. If its status has just transitioned from being the designated VE device, then the PE router tears down all the pseudowires that it had to and from the other PE routers in the VPWS network.

- The PE router receives the multihomed advertisements and selects a best path; it does not originate any of these advertisements because it is not connected to the multihomed customer site.

If the selected best path—and therefore the designated VE—has not changed, then nothing happens. If the best path has changed, then this PE router brings up pseudowires to and from the new designated VE device and tears down the pseudowires to and from the previous designated VE device.

If this PE router does not select a best path after running the process, then the local PE router does not consider the remote site to exist.

For a VPWS network, this PE router then sets the circuit status vector bit in the MP_REACH_NLRI to indicate that the remote site is unreachable. BGP advertises the route to inform remote PE routers for this new condition.

When a VE device receives an advertisement for a layer 2 NLRI that matches its own site ID but the site is not multihomed, then the pseudowire between it and the transmitting PE router transitions to a site collision (SC) state and is not considered to be up.

Related Documentation

- BGP Multihoming for VPWS Overview on page 669
- Multihoming Reaction to Failures in the Network on page 673
- Configuring BGP Multihoming for VPWS on page 679

Multihoming Reaction to Failures in the Network

The redundant connectivity provided by a multihoming configuration protects against several types of network failure.

- CE-Link failure between the CE device and the PE router—BGP on the PE router is notified when the circuit goes down. BGP then modifies the circuit status vector bit in the MP_REACH_NLRI to indicate that the circuit is down.

  If all VPWS local attachment circuits are down, then BGP modifies the down bit in the VPWS advertisement Layer2-Extended-Community to state that the site is down. When the bit is modified, BGP advertises the route to all remote PE routers to inform them that the circuit (and site) is down. The remote PE routers each run the best path selection process again and adjust the VPWS pseudowires as needed.

- Failure of MPLS reachability to the remote PE router—BGP on the PE router is notified that MPLS connectivity to the BGP next hop is gone. BGP then modifies the circuit status vector bit in the MP_REACH_NLRI to indicate that the LSP is down. When the bit is modified, BGP advertises the route to all remote PE routers to inform them that connectivity is down from the local site to the remote site.

  Unlike for VPLS, the down bit is not set when no remote PE router is reachable by MPLS.

- PE router failure—When either the PE router or its BGP process fails, peer PE routers detect expiration of the holdtimer and bring down their peering sessions, and remove layer 2 advertisements from the PE router. Alternatively, the PE routers can detect unreachability to the BGP next hop that represents the failed PE router. In this case the peer routers mark the layer 2 routes advertised by PE router as unreachable. The peer PE routers each run the best path selection process again and adjust the VPWS pseudowires as needed.

A similar response results when you adjust the multihoming priority of the PE routers connected to the multihomed site, effectively performing and administrative failover to another PE router. BGP sends a layer 2 update with the new local preference attribute to all peer PE routers. The peer PE routers each run the best path selection process again and adjust the VPWS pseudowires as needed.

To modify their pseudowires, the peer routers correct their MPLS forwarding tables and set up new entries in their pseudowire tables.

Related Documentation

- BGP Multihoming for VPWS Overview on page 669
- Designated VE Device Selection for a Multihomed Site on page 671
- Configuring BGP Multihoming for VPWS on page 679

VPWS Supported Features

The JunosE Software implementation of VPWS provides the following features:

- Support for the following types of network interfaces between the PE router and the CE device:
  - ATM with ATM Adaptation Layer 5 (AAL5) encapsulation
  - ATM with virtual channel connection (VCC) cell relay encapsulation
  - Cisco HDLC
  - Ethernet (Fast Ethernet, Gigabit Ethernet, 10-Gigabit Ethernet, Ethernet/VLAN)
  - Frame Relay
  - PPP
- Autodiscovery of L2VPN instance members using MP-BGP
- L2VPN signaling using MP-BGP to set up and tear down the pseudowires that constitute an L2VPN instance
- Multihoming
- Inter-AS option A, inter-AS option B, and inter-AS option C services

As with VPLS, VPWS does not support BGP multipaths.

Related Documentation

- VPWS Overview on page 663
- VPWS Platform Considerations on page 675
- VPWS References on page 676

VPWS Platform Considerations

VPWS is supported on all E Series routers:

- Module Requirements on page 675
- Interface Specifiers on page 675

Module Requirements

You can configure VPWS on all E Series module combinations that support MPLS tunnels.

For information about the modules that support VPWS on ERX14xx models, ERX7xx models, and ERX310 Broadband Services Router:

- See *Module Combinations* in the *ERX Module Guide* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support VPWS.

For information about the modules that support VPWS on E120 and E320 Broadband Services Routers:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support VPWS network interfaces and VPWS virtual core interfaces.

Interface Specifiers

The configuration task examples in this chapter use the *slot/port*[.*subinterface*] format to specify the physical interface on which to configure an VPWS L2VPN network interface. However, the interface specifier format that you use depends on the router that you are using.

For ERX7xx models, ERX14xx models, and ERX310 routers, use the *slot/port*[.*subinterface*] format. For example, the following command specifies Fast Ethernet subinterface 6 on port 2 of the I/O module installed in slot 3 of an ERX7xx model, ERX14xx model, or ERX310 router.

host1(config)#**interface fastEthernet 3/2.6**

For E120 and E320 routers, use the *slot/adapter/port*[.*subinterface*] format, which includes an identifier for the bay in which the I/O adapter (IOA) resides. In the software, adapter 0 identifies the right IOA bay (E120 router) and the upper IOA bay (E320 router); adapter 1 identifies the left IOA bay (E120 router) and the lower IOA bay (E320 router). For example, the following command specifies Gigabit Ethernet subinterface 20 on port 1 of the IOA installed in the upper adapter bay (adapter 0) of slot 4 in an E320 router.

host1(config)#**interface gigabitEthernet 4/0/1.20**

Related Documentation

- VPWS Overview on page 663
- VPWS Supported Features on page 674
- VPWS References on page 676
- *Interface Types and Specifiers*

VPWS References

For more information about VPWS, consult the following resources:

- Layer 2 VPNs over Tunnels—draft-kompella-l2vpn-l2vpn-01.txt (July 2006 expiration)

NOTE: IETF drafts are valid for only 6 months from the date of issuance. They must be considered as works in progress. Please refer to the IETF Website at http://www.ietf.org for the latest drafts.

Related Documentation

- VPWS Overview on page 663
- VPWS Supported Features on page 674
- VPWS Platform Considerations on page 675

CHAPTER 15

Configuring VPWS

This chapter describes how to configure virtual private wire service (VPWS) L2VPNs on the router, and contains the following sections:

Configuring VPWS on a PE Router

To configure a PE router to provide VPWS:

1. Configure an VPWS instance.

   See "Configuring BGP Multihoming for VPWS" on page 679 and (optional) "Configuring a VPWS Instance" on page 678

2. Configure the customer-facing interfaces in the VPWS instance.

   See "Configuring Customer-Facing Interfaces in the VPWS Instance" on page 680

3. (Optional) Configure local cross-connects.

   See "Configuring a Local Cross-Connect for VPWS" on page 681

4. Configure the loopback interface and router ID for BGP.

See "Configuring the Loopback Interface and Router ID for BGP for VPWS" on page 682

5. Set up BGP signaling in the autonomous system that is configured to signal reachability for this VPWS instance.

   See "Configuring BGP Signaling for VPWS" on page 683

6. Configure MPLS label-switched paths (LSPs) to connect the local and remote PE routers.

   See "Configuring MPLS LSPs for VPWS" on page 684

Related Documentation

- BGP Signaling for VPWS Overview on page 683
- Local Cross-Connects for VPWS Overview on page 681
- Types of Interfaces to Configure in the VPWS Instance on page 680
- BGP Loopback Interface and Router ID Overview on page 682
- MPLS LSPs for VPWS Overview on page 684
- Example: Configuring VPWS on Local and Remote Routers on page 685

Configuring a VPWS Instance

You must configure a VPWS instance for each L2VPN in which the router participates. From a configuration standpoint, a VPWS instance is simply a new L2VPN that you configure with additional VPWS L2VPN attributes.

Table 134 on page 678 lists the commands that you use to configure a basic VPWS instance.

Table 134: Commands to Configure Basic VPWS Instances

l2vpn control-word	**l2vpn route-target**
l2vpn encapsulation-type	**l2vpn sequencing**
l2vpn local-site-id remote-site-id	**l2vpn site-name site-id**
l2vpn rd	**l2vpn site-range**

To configure a basic VPWS instance on the PE router:

1. Create the VPWS instance by configuring the encapsulation type for all interfaces in the L2VPN.

   You must issue this command before any other **l2vpn** commands.

   ```
   host1(config)#l2vpn exampleco encapsulation-type ethernet
   ```

2. Configure the maximum number of customer sites that can participate in the L2VPN.

   ```
   host1(config)#l2vpn exampleco site-range 10
   ```

3. Configure the name and ID number for the customer sites in the L2VPN instance.

The site ID value must be greater than zero and be unique within the L2VPN for that customer site. This is not true for a multihomed customer site. See "Configuring BGP Multihoming for VPWS" on page 679 for more information.

host1(config)#l2vpn exampleco site-name westford site-id 1

4. Configure a route distinguisher for the L2VPN.

   In this example, the first number in the route distinguisher (100) is the number of the autonomous system (AS). The second number in the route distinguisher (11) uniquely identifies the L2VPN instance within that AS.

   host1(config)#l2vpn exampleco rd 100:11

5. Create or add a route target to the import and export lists of the L2VPN's route-target extended community.

   The PE router uses the lists to determine which routes are imported into the VPWS instance.

   host1(config)#l2vpn exampleco route-target both 100:1

6. Specify the local preference for use of the control word and sequencing for the layer 2 packets encapsulated in the MPLS packets that are sent to the remote PE router.

   host1(config)#l2vpn exampleco control-word
   host1(config)#l2vpn exampleco sequencing

Related Documentation

- *l2vpn control-word*
- *l2vpn encapsulation-type*
- *l2vpn rd*
- *l2vpn route-target*
- *l2vpn sequencing*
- *l2vpn site-name site-id*
- *l2vpn site-range*

Configuring BGP Multihoming for VPWS

You can configure BGP multihoming in the VPWS network to provide redundancy in the event of failures such as a PE router-to-CE device link failure, the failure of a PE router, or an MPLS reachability failure between the local PE router and a remote PE router. BGP multihoming enables you to connect a customer site to two or more PE routers. A redundant PE router can begin providing service to the customer site as soon as the failure is detected. The redundant connectivity maintains the VPWS service and traffic forwarding to and from the multihomed site while avoiding the formation of layer 2 traffic loops.

To configure BGP multihoming on a VPWS PE router:

- Configure the site as multihomed and specify a multihoming priority for the PE site for this instance.

```
host1(config)#l2vpn exampleco site-name westford site-id 1 multi-homed priority 2
```

You must configure the same site ID on all PE routers connected to the multihomed customer site. The site ID shared by the connected PE routers should be different than the site IDs configured on the remote PE routers in the VPWS network.

You can configure a different block offset, label range, and route distinguisher for each connected PE router.

Related Documentation

- Configuring VPWS on a PE Router on page 677
- *l2vpn site-name site-id*

Types of Interfaces to Configure in the VPWS Instance

You must configure one of the following types of interfaces as a member of the VPWS to transmit packets between the PE router and each CE device to which the PE router is connected:

- ATM (AAL5 VCC transport or ATM VCC cell transport)
- Cisco HDLC
- Ethernet (Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet)
- Frame Relay
- PPP
- VLAN and S-VLAN subinterfaces over Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet interfaces

Related Documentation

- VPWS Overview on page 663
- Configuring a VPWS Instance on page 678

Configuring Customer-Facing Interfaces in the VPWS Instance

To configure a customer-facing interface for an VPWS instance:

1. Access Interface Configuration mode for a layer 2 interface for the VPWS on the PE router.

   ```
   host1(config)#interface fastEthernet 4/0
   ```

2. Configure the local and remote site IDs on the interface to specify the interface as a member of the VPWS L2VPN.

   ```
   host1(config-if)#l2vpn exampleco local-site-id 1 remote-site-id 2
   host1(config-if)#exit
   ```

3. Repeat for all customer-facing interfaces in the VPWS.

   ```
   host1(config)#interface fastEthernet 4/1
   host1(config-if)#l2vpn exampleco local-site-id 1 remote-site-id 3
   ```



```
host1(config-if)#exit
```

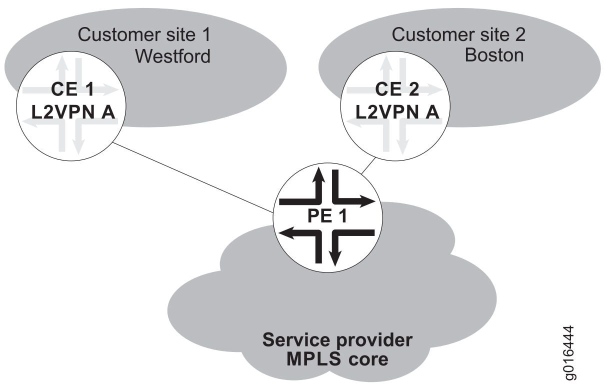
Related Documentation

- *interface fastEthernet*
- *l2vpn local-site-id remote-site-id*

Local Cross-Connects for VPWS Overview

You configure a local cross-connect between two local customer sites by first configuring the two local sites and then configuring the correct local and remote site IDs on the two local interfaces that you are cross-connecting. Figure 135 on page 681 illustrates cross-connects by showing a portion of a sample VPWS topology.

Figure 135: VPWS Cross-Connects



The following example shows the creation of a cross-connect between sites Westford and Boston. On one customer-facing interface, Westford is considered local and Boston is remote; on the other customer-facing interface, Boston is considered local and Westford is remote. From the perspective of the PE router, both sites are local.

Multihoming is not applicable for VPWS local cross-connects because the local and remote CE devices are connected to the same PE.

Related Documentation

- BGP Multihoming for VPWS Overview on page 669
- Configuring a Local Cross-Connect for VPWS on page 681

Configuring a Local Cross-Connect for VPWS

To configure a local cross-connect between two local sites:

1. Configure the two local sites.

```
host1(config)#l2vpn exampleco encapsulation-type ethernet
host1(config)#l2vpn exampleco site-name westford site-id 1
host1(config)#l2vpn exampleco site-name boston site-id 2
host1(config)#l2vpn exampleco site-range 10
host1(config)#l2vpn exampleco rd 100:11
host1(config)#l2vpn exampleco route-target both 100:1
```

2. Configure the correct local and remote site IDs on the two local interfaces that are being cross-connected.

```
host1(config)#interface fastEthernet 4/0
host1(config-if)#l2vpn exampleco local-site-id 1 remote-site-id 2
host1(config-if)#exit

host1(config)#interface fastEthernet 4/1
host1(config-if)#l2vpn exampleco local-site-id 2 remote-site-id 1
host1(config-if)#exit
```

Related Documentation

- Configuring a VPWS Instance on page 678
- Types of Interfaces to Configure in the VPWS Instance on page 680

BGP Loopback Interface and Router ID Overview

To establish a BGP session, BGP uses the IP address of the outgoing interface towards the BGP peer as the update source IP address for the TCP connection over which the BGP session runs. Typically, you configure a loopback interface as the update source interface because a loopback interface is inherently stable.

After you configure the loopback interface, use the **ip router-id** command to assign a router ID to uniquely identify the router within a BGP AS. The router ID is the IP address of the loopback interface.

Related Documentation

- Configuring the Loopback Interface and Router ID for BGP for VPWS on page 682

Configuring the Loopback Interface and Router ID for BGP for VPWS

To configure the loopback interface and router ID on the PE router:

1. Configure a loopback interface on the PE router and assign an IP address to the interface.

```
host1(config)#interface loopback 0
host1(config-if)#ip address 10.3.3.3 255.255.255.255
host1(config-if)#exit
```

2. Assign the router ID using the IP address you configured for the loopback interface.

```
host1(config)#ip router-id 10.3.3.3
```

Related Documentation

- *interface loopback*
- *ip address*
- *ip router-id*

BGP Signaling for VPWS Overview

This section describes one way to configure BGP signaling for VPWS, but does not provide complete details about configuring BGP and BGP/MPLS VPNs.

Table 135 on page 683 lists the commands used in this section to configure BGP signaling for VPWS.

Table 135: Commands to Configure BGP Signaling for VPWS

address-family l2vpn	neighbor next-hop-self
address-family vpws	neighbor remote-as
exit-address-family	neighbor update-source
neighbor activate	router bgp

Related Documentation

- BGP Signaling for L2VPNs Overview on page 665
- VPWS and BGP/MPLS VPNs Overview on page 668
- Configuring BGP Signaling for VPWS on page 683

Configuring BGP Signaling for VPWS

To configure BGP signaling for an VPWS L2VPN on the PE router:

1. Enable the BGP routing process on the PE router in the specified local AS.

   The AS number identifies the PE router to other BGP routers.

   host1(config)#**router bgp 738**

2. Configure the PE-to-PE BGP session. Use **neighbor** commands to specify the PE router peers to which BGP advertises routes and to configure additional BGP attributes.

   host1(config-router)#**neighbor 10.2.2.2 remote-as 738**
   host1(config-router)#**neighbor 10.2.2.2 update-source loopback 0**
   host1(config-router)#**neighbor 10.2.2.2 next-hop-self**

3. Create the L2VPN address family to configure the router to use BGP signaling to exchange layer 2 NLRI to peer PE routers for all VPWS instances.

   Optionally, you can use the signaling

   keyword with the

   address-family command when you configure the L2VPN address family to specify BGP signaling of VPWS reachability information. Currently, you can omit the signaling

   keyword with no adverse effects.

   host1(config-router)#**address-family l2vpn signaling**

4. Activate the neighbors with which routes of the L2VPN address family are exchanged for this PE-to-PE BGP session. Use the **bgp dampening** command and BGP **neighbor** commands to configure additional address family parameters for the session. No other commands are supported in this address family.

   host1(config-router-af)#**neighbor 10.2.2.2 activate**
   host1(config-router-af)#**neighbor 10.2.2.2 next-hop-self**

5. Exit the address family.

   host1(config-router-af)#**exit-address-family**

6. Create the VPWS address family to configure the router to exchange layer 2 NLRI for each VPWS instance configured on the router.

   You must issue the **address-family vpws** command separately for each VPWS instance configured on the router.

   host1(config-router)#**address-family vpws l2vpnA**
   host1(config-router)#**address-family vpws l2vpnB**

For information about configuring BGP, see "Configuring BGP Routing" on page 3. For information about configuring BGP/MPLS VPNs, see "Configuring BGP-MPLS Applications" on page 389.

Related Documentation

- *address-family* l2vpn
- *address-family* vpws
- *exit-address-family*
- *neighbor activate*
- *neighbor next-hop-self*
- *neighbor remote-as*
- *neighbor update-source*
- *router bgp*

MPLS LSPs for VPWS Overview

As part of a VPWS L2VPN configuration, you must create MPLS label-switched paths (LSPs) to connect the local PE router and the remote PE router.

Related Documentation

- Configuring MPLS LSPs for VPWS on page 684

Configuring MPLS LSPs for VPWS

This section explains one way to create a basic MPLS configuration using the **mpls** and **mpls ldp** commands.

To configure MPLS LSPs on the PE router:

1. Enable MPLS on the virtual router.


```
host1(config)#mpls
```

2. Configure the core-facing interface on which you want to enable MPLS, Label Distribution Protocol (LDP), and topology-driven LSPs.

```
host1(config)#interface atm 5/0.100
host1(config-subif)#atm pvc 100 1 100 aal5snap 0 0 0
host1(config-subif)#ip address 192.168.5.5 255.255.255.0
```

3. Enable MPLS on the core-facing interface.

```
host1(config-subif)#mpls
```

4. Enable LDP and topology-driven LSPs on the core-facing interface.

```
host1(config-subif)#mpls ldp
host1(config-subif)#exit
```

For information about configuring MPLS, see "Configuring MPLS" on page 279.

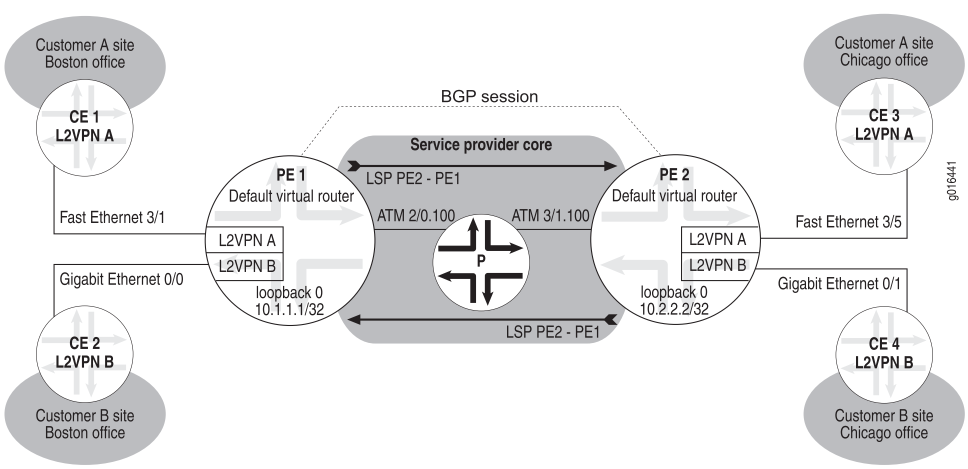
Related Documentation

- *atm pvc*
- *interface atm*
- *ip address*
- *mpls*
- *mpls ldp*

Example: Configuring VPWS on Local and Remote Routers

The example in this section shows how to configure the VPWS topology illustrated in Figure 136 on page 685. The example includes procedures for configuring VPWS on both the local E Series router (PE 1) and the remote E Series router (PE 2).

Figure 136: Topology for VPWS Configuration Example



Topology Overview

The sample topology in Figure 136 on page 685 includes two L2VPNs, L2VPN A and L2VPN B. L2VPN A connects CE 1, at the edge of Customer A's Boston site, with CE 3, at the edge of Customer A's Chicago site. Similarly, L2VPN B connects CE 2, at the edge Customer B's Boston site, with CE 4, at the edge of Customer B's Chicago site.

The E Series routers in the topology, PE 1 and PE 2, each participate in both L2VPN A and L2VPN B. The example configures a total of four separate L2VPN instances, one for each L2VPN on each PE router. The instances for L2VPN A are named l2vpnA, and the instances for L2VPN B are named l2vpnB.

For each VPWS instance, an Ethernet network interface provides a connection to the associated CE device.

Each PE router in the sample topology also has an ATM core-facing interface that connects it to the provider (P) router in the service provider core. You must configure MPLS LSPs on the core-facing interfaces to connect PE 1 and PE 2 through the P router across the service provider core. Finally, you must configure BGP on both PE 1 and PE 2 to provide signaling for both L2VPNs.

Configuration on PE 1 (Local PE Router)

Use the following commands on the local PE router (PE 1) to configure the VPWS topology shown in Figure 136 on page 685.

```
! Configure VPWS instance l2vpnA.
host1(config)#l2vpn l2vpnA encapsulation-type ethernet
host1(config)#l2vpn l2vpnA site-range 10
host1(config)#l2vpn l2vpnA site-name boston site-id 1
host1(config)#l2vpn l2vpnA rd 100:11
host1(config)#l2vpn l2vpnA route-target both 100:1
host1(config)#l2vpn l2vpnA control-word
host1(config)#l2vpn l2vpnA sequencing
!
! Configure VPWS instance l2vpnB.
host1(config)#l2vpn l2vpnB encapsulation-type atm
host1(config)#l2vpn l2vpnB site-range 20
host1(config)#l2vpn l2vpnB site-name boston site-id 2
host1(config)#l2vpn l2vpnB rd 100:12
host1(config)#l2vpn l2vpnB route-target both 100:2
host1(config)#l2vpn l2vpnB control-word
host1(config)#l2vpn l2vpnB sequencing
!
! Configure the customer-facing interface between PE 1 and CE 1
! in L2VPN instance l2vpnA.
host1(config)#interface fastEthernet 4/0
host1(config-if)#l2vpn l2vpnA local-site-id 1 remote-site-id 3
host1(config-if)#exit
!
! Configure the customer-facing interface between PE 1 and CE 2
! in L2VPN instance l2vpnB.
host1(config)#interface gigabitEthernet 1/1
host1(config-subif)#l2vpn l2vpnB local-site-id 2 remote-site-id 4
```



```
host1(config-if)#exit
!
! Configure a loopback interface on PE 1 and assign it an IP address.
host1(config)#interface loopback 0
host1(config-if)#ip address 10.1.1.1 255.255.255.255
host1(config-if)#exit
!
! Assign the router ID for PE 1 using the IP address of the loopback interface.
host1(config)#ip router-id 10.3.3.3
!
! Configure BGP signaling.
host1(config)#router bgp 738
host1(config-router)#neighbor 10.1.1.1 remote-as 738
host1(config-router)#neighbor 10.1.1.1 update-source loopback 0
host1(config-router)#neighbor 10.1.1.1 next-hop-self
host1(config-router)#address-family l2vpn signaling
host1(config-router-af)#neighbor 10.1.1.1 activate
host1(config-router-af)#neighbor 10.1.1.1 next-hop-self
host1(config-router-af)#exit-address-family
host1(config-router)#address-family vpws l2vpnA
host1(config-router-af)#exit-address-family
host1(config-router)#address-family vpws l2vpnB
host1(config-router-af)#exit-address-family
!
! Enable MPLS on the default virtual router.
host1(config)#mpls
! Configure ATM core-facing interface 2/0.100 between PE 1 and the P router,
host1(config)#interface atm 2/0.100
host1(config-subif)#atm pvc 100 1 100 aal5snap 0 0 0
! and assign it an IP address.
host1(config-subif)#ip address 192.168.5.5 255.255.255.0
!
! Enable MPLS, LDP, and topology-driven LSPs on the core-facing interface.
host1(config-subif)#mpls
host1(config-subif)#mpls ldp
host1(config-subif)#exit
```

Configuration on PE 2 (Remote PE Router)

Use the following commands on the remote PE router (PE 2) to configure the VPWS topology shown in Figure 136 on page 685.

```
! Configure VPWS instance l2vpnA. The route target (100:1)
! matches the route target configured for l2vpnA on PE 1.
host2(config)#l2vpn l2vpnA encapsulation-type ethernet
host2(config)#l2vpn l2vpnA site-range 10
host2(config)#l2vpn l2vpnA site-name chicago site-id 3
host2(config)#l2vpn l2vpnA rd 100:11
host2(config)#l2vpn l2vpnA route-target both 100:1
host2(config)#l2vpn l2vpnA control-word
host2(config)#l2vpn l2vpnA sequencing
!
! Configure VPWS instance l2vpnB. The route target (100:2)
! matches the route target configured for l2vpnB on PE 1.
host2(config)#l2vpn l2vpnB encapsulation-type ethernet
host2(config)#l2vpn l2vpnB site-range 20
```



```
host2(config)#l2vpn l2vpnB site-name chicago site-id 4
host2(config)#l2vpn l2vpnB rd 100:12
host2(config)#l2vpn l2vpnB route-target both 100:2
host2(config)#l2vpn l2vpnB control-word
host2(config)#l2vpn l2vpnB sequencing
!
! Configure the customer-facing interface between PE 2 and CE 3
! in L2VPN instance l2vpnA.
host2(config)#interface fastEthernet 3/5
host2(config-if)#l2vpn l2vpnA local-site-id 3 remote-site-id 1
host2(config-if)#exit
!
! Configure the customer-facing interface between PE 2 and CE 4
! in L2VPN instance l2vpnB.
host2(config)#interface gigabitEthernet 0/1
host2(config-subif)#l2vpn l2vpnB local-site-id 4 remote-site-id 2
host2(config-if)#exit
!
! Configure a loopback interface on PE 2 and assign it an IP address.
host2(config)#interface loopback 0
host2(config-if)#ip address 10.2.2.2 255.255.255.255
host2(config-if)#exit
!
! Assign the router ID for PE 2 using the IP address of the loopback interface.
host2(config)#ip router-id 10.2.2.2
!
! Configure BGP signaling.
host2(config)#router bgp 738
host2(config-router)#neighbor 10.2.2.2 remote-as 738
host2(config-router)#neighbor 10.2.2.2 update-source loopback 0
host2(config-router)#neighbor 10.2.2.2 next-hop-self
host2(config-router)#address-family l2vpn signaling
host2(config-router-af)#neighbor 10.2.2.2 activate
host2(config-router-af)#neighbor 10.2.2.2 next-hop-self
host2(config-router)#address-family vpws l2vpnA
host2(config-router-af)#exit-address-family
host2(config-router)#address-family vpws l2vpnB
host2(config-router-af)#exit-address-family
!
! Enable MPLS on the default virtual router.
host2(config)#mpls
!
! Configure ATM core-facing interface 3/1.100 between PE 2 and the P router,
! and assign it an IP address.
host2(config)#interface atm 3/1.100 point-to-point
host2(config-subif)#atm pvc 100 1 100 aal5snap 0 0 0
host2(config-subif)#ip address 192.168.4.4 255.255.255.0
!
! Enable MPLS, LDP, and topology-driven LSPs on the on the core-facing interface.
host2(config-subif)#mpls
host2(config-subif)#mpls ldp
host2(config-subif)#exit
!
! Enable MPLS, LDP, and topology-driven LSPs on the core-facing interface.
host1(config-subif)#mpls
host1(config-subif)#mpls ldp
```



```
host1(config-subif)#exit
```

Related Documentation

- VPWS Overview on page 663
- Configuring VPWS on a PE Router on page 677

CHAPTER 16

Monitoring VPWS

This chapter describes the commands you can use to monitor and troubleshoot Virtual Private Wire Service (VPWS) L2VPNs on E Series routers.

NOTE: The E120 and E320 Broadband Services Routers output for **monitor** and **show** commands is identical to output from other E Series routers, except that the E120 and E320 router output also includes information about the adapter identifier in the interface specifier (*slot/adapter/port*).

This chapter contains the following sections:

Clearing BGP Attributes for VPWS

You can use the following **clear ip bgp** commands to remove specific BGP attributes for the L2VPN address family, and in one case, for the VPWS address family.

Tasks to clear BGP attributes are:

Clearing BGP Reachability Information for the L2VPN Address Family

To clear BGP reachability information for a specific VPWS instance in the L2VPN address family:

- Issue the **clear ip bgp** command and specify **l2vpn vpws** *vpwsName*.

  host1#clear ip bgp l2vpn soft in

To clear BGP reachability information for all VPLS and VPWS instances in the L2VPN address family:

- Issue the **clear ip bgp** command and include the **l2vpn all** keywords.

  host1#clear ip bgp l2vpn all soft in

Use the **soft in** keywords to trigger inbound soft reconfiguration.

Clearing BGP Route Flap Dampening Information for the L2VPN Address Family

To clear route flap dampening information for a specific VPWS instance in the L2VPN address family:

- Issue the **clear ip bgp dampening** command and specify **l2vpn vpws** *vpwsName*.

  host1#clear ip bgp l2vpn dampening l2vpn vpws l2vpnBoston

To clear route flap dampening information for all VPLS and VPWS instances in the L2VPN address family:

- Issue the **clear ip bgp dampening** command and include the **l2vpn all** keywords.

  host1#clear ip bgp l2vpn all dampening

Clearing the Wait for the End-of-RIB Marker for the L2VPN Address Family

To clear the wait for receiving an End-of-RIB marker from the peer for all VPWS instances in the L2VPN address family:

- Issue the **clear ip bgp wait-end-of-rib** command and specify **l2vpn vpws** *vpwsName*.

  host1#clear ip bgp l2vpn vpws l2vpnBoston wait-end-of-rib

To clear the wait for receiving an End-of-RIB marker from the peer for all VPLS and VPWS instances in the L2VPN address family:

- Issue the **clear ip bgp wait-end-of-rib** command and include the **l2vpn all** keywords.

  host1#clear ip bgp l2vpn all wait-end-of-rib

For information about configuring BGP routing, see "Configuring BGP Routing" on page 3.

Related Documentation

- *clear ip bgp*
- *clear ip bgp dampening*
- *clear ip bgp wait-end-of-rib*

Monitoring BGP-Related Settings for VPWS L2VPNS

Purpose This section provides examples of some of the **show ip bgp** commands that you can use to monitor VPWS configurations.

You can use the **show ip bgp** commands listed in Table 136 on page 693 to display BGP-related settings for VPWS instances in the L2VPN address family. The **l2vpn all** keywords display both all VPWS instances and all VPLS instances in the L2VPN address family

Table 136: Commands for Monitoring BGP Settings for the VPWS Address Family

show ip bgp advertised-routes	**show ip bgp neighbors received-routes**
show ip bgp l2vpn all	**show ip bgp neighbors routes**
show ip bgp neighbors	**show ip bgp peer-group**
show ip bgp neighbors dampened-routes	

You can use the **show ip bgp** commands listed in Table 137 on page 693 to display BGP-related settings for L2VPNs. Specify **l2vpn vpws** *vpwsName* to display information about only the VPWS instances in the VPWS address family. Specify **l2vpn all** to display information about all VPWS instances in the VPWS address family and all VPLS instances in the VPLS address family.

Table 137: Commands for Monitoring BGP Settings for the VPWS Address Family

show ip bgp	**show ip bgp l2vpn vpws**
show ip bgp community	**show ip bgp next-hops**
show ip bgp community-list	**show ip bgp paths**
show ip bgp dampened-paths	**show ip bgp quote-regexp**
show ip bgp filter-list	**show ip bgp regexp**
show ip bgp flap-statistics	**show ip bgp summary**

For more information about using the **show ip bgp** commands that are not described in this section, see "Configuring BGP Routing" on page 3 and "Configuring BGP-MPLS Applications" on page 389.

Display layer 2 NLRI for all VPWS instances in the L2VPN address family, for a particular VPWS instance in the L2VPN address family, or for a particular VPWS instance in the VPWS address family.

The **l2vpn vpws** keywords display layer 2 NLRI for a particular VPWS instance in the VPWS address family.

The **l2vpn all** keywords display layer 2 NLRI for all VPWS instances in the L2VPN address family. The output for this version of the command also includes information about any VPLS instances configured in the L2VPN address family.

To display layer 2 NLRI for the route that matches a specified prefix (site ID and block offset) in the L2VPN address family or in the VPWS address family, use the **site-id** and **block-offset** keywords.

Action To display information for a particular L2VPN instance in the L2VPN address family:

```
host1:pe1# show ip bgp l2vpn vpws l2vpn1
Local BGP identifier 10.1.1.1, local AS 100
  2 routes (152 bytes)
  2 destinations (152 bytes) of which 2 have a route
  2 routes selected for route tables installation
  0 unicast/multicast routes selected for route table installation
  0 unicast/multicast tunnel-usable routes selected for route table installation

  0 tunnel-only routes selected for tunnel-route table installation
  4 path attribute entries (608 bytes)
  Local-RIB version 6. FIB version 6.
Status codes: > best, * invalid, s suppressed, d dampened, r rejected,
              a auto-summarized m multihomed-backup
  Prefix      Peer            Next-hop        MED LocPrf Weight Origin
> 1:1         0.0.0.0         self                           0    IGP
> 2:1         12.2.2.2        12.2.2.2             100       0    IGP
```

To display summary information for a particular VPWS instance in the VPWS address family; only the BGP operational state is useful:

```
host1:pe1# show ip bgp l2vpn vpws l2vpn1 summary
Local router ID 10.1.1.1, local AS 100
  Administrative state is Start
  BGP Operational state is Up
  Shutdown in overload state is disabled
  Default local preference is 100
  Default originate is disabled
  Always compare MED is disabled
  Compare MED within confederation is disabled
  Advertise inactive routes is disabled
  Advertise best external route to internal peers is disabled
  Enforce first AS is disabled
  Missing MED as worst is disabled
  Route flap dampening is disabled
  Log neighbor changes is disabled
  Fast External Fallover is disabled
  No maximum received AS-path length
  BGP administrative distances are 20 (ext), 200 (int), and 200 (local)
  Client-to-client reflection is enabled
  Cluster ID is not configured (local router ID used)
  Route-target filter is enabled
  Default IPv4-unicast is enabled
  Redistribution of iBGP routes is disabled
  Graceful restart is globally disabled
  Global graceful-restart restart time is 120 seconds
  Global graceful-restart stale paths time is 360 seconds
  Graceful-restart path selection defer time is 360 seconds
  Graceful-restart is not ready to switch to the standby SRP
  The last restart was not graceful
  Local-RIB version 6. FIB version 6.
  (No neighbors are configured)
```


To display information for the route that matches the specified prefix (2:1) for a VPWS instance named customer1 in the VPWS address family:

`host1#`**show ip bgp l2vpn vpws customer1 site-id 2 block-offset 1**

```
BGP route information for prefix 2:1
  Received route learned from internal peer 10.2.2.2 (best route)
    Leaked route
    Route placed in IP forwarding table
    Best to advertise to external peers
    Address Family Identifier (AFI) is layer2
    Subsequent Address Family Identifier (SAFI) is unicast
    Route Distinguisher (RD) is 100:11
    Original Route Distinguisher (RD) is 100:21
    MPLS in-label is none
    MPLS in-label block size is 0
    MPLS out-label is 46
    MPLS out-label block size is 20
    Next hop IP address is 2.2.2.2 (metric 3)
    Multi-exit discriminator is not present
    Local preference is 100
    Weight is 0
    Origin is IGP
    AS path is empty
    Extended communities RT:100:1 Layer 2:19:00:0
```

To display information for all instances that support a multihomed site with site ID 2 and block offset 1:

`host1:pe1#`**show ip bgp l2vpn all site-id 2 block-offset 1**

```
BGP route information for prefix 2:1
Received route learned from internal peer 2.2.2.2 (best route)
   Route not placed in PW table
   Best to advertise to external peers
   Suppressed by multihoming
   Address Family Identifier (AFI) is layer2
   Subsequent Address Family Identifier (SAFI) is vpn-unicast
   Route Distinguisher (RD) is 100:23
   Original Route Distinguisher (RD) is 100:23
   MPLS in-label is none
   MPLS in-label block size is 0
   MPLS out-label is 106
   MPLS out-label block size is 10
   Next hop IP address is 2.2.2.2 (metric 3)
   Multi-exit discriminator is not present
   Local preference is 100
   Weight is 0
   Origin is IGP
   AS path is empty
   Extended communities RT:100:3 Layer2:atm-vcc-cell:use-cw,use-seq,:mtu=0
   Status Vector 0x7fc0
```

Meaning Table 138 on page 695 lists the **show ip bgp l2vpn** command output fields

Table 138: show ip bgp l2vpn Output Fields

Field Name	Field Description
Local BGP identifier	IP address of the local PE router

Table 138: show ip bgp l2vpn Output Fields *(continued)*

Field Name	Field Description
local AS	Autonomous system number
Local-RIB version	Version number of the local routing information base
FIB version	Version number of the forwarding information base
Status codes	Status codes for the route, listed before the Prefix: • >—best route • *—invalid route • s—suppressed route • d—dampened route • r—rejected route • a—auto-summarized • m—multihomed backup route
Prefix	Route prefix in the format *siteID*:*blockOffset*
Peer	IP address of the peer from which the route was learned
Next-hop (or Next hop IP address)	IP address of the next router that is used when a packet is forwarded to the destination network
MED	Multiexit discriminator for the route
LocPrf	Local preference for the route
Weight	Weight of the route
Origin	Origin of the route
AS Path	AS path through which this route has been advertised
Extended communities	Description of the L2VPN extended communities that the router uses to determine which routes are imported by the specified VPWS instance. Includes route target, community type, encapsulation, control word and sequencing use, L2VPN link MTU.
BGP Operational state	Operational state, up or down
Status Vector	Hexadecimal representation of the status vector bits attached to the route

Related Documentation

- *show ip bgp*
- *show ip bgp advertised-routes*
- *show ip bgp community*

- *show ip bgp community-list*
- *show ip bgp dampened-paths*
- *show ip bgp filter-list*
- *show ip bgp flap-statistics*
- *show ip bgp neighbors*
- *show ip bgp neighbors dampened-routes*
- *show ip bgp neighbors received-routes*
- *show ip bgp neighbors routes*
- *show ip bgp next-hops*
- *show ip bgp paths*
- *show ip bgp peer-group*
- *show ip bgp quote-regexp*
- *show ip bgp regexp*
- *show ip bgp summary*

Monitoring BGP Next Hops for VPWS L2VPNS

Purpose Display information about BGP next hops in the L2VPN address family or in the VPWS address family

Action To display BGP next hop information that matches the specified indirect next-hop address (2.2.2.2) in the L2VPN address family:

```
host1#show ip bgp l2vpn all next-hops 10.2.2.2
Indirect next-hop 10.2.2.2
  Resolution in IP route table of VR
       IP indirect next-hop index 2
       Reachable (metric 3)
       Number of direct next-hops is 1
              Direct next-hop ATM2/0.10 (10.10.10.2)
  Resolution in IP tunnel-route table of VR
       MPLS indirect next-hop index 19
       Reachable (metric 3)
       Number of direct next-hops is 1
              Direct next-hop 0000000c
  Reference count is 2
```

Meaning Table 139 on page 697 lists the **show ip bgp l2vpn all next-hops** command output fields.

Table 139: show ip bgp l2vpn all next-hops Output Fields

Field Name	Field Description
Indirect next-hop	BGP next-hop attribute received in the BGP update message

Table 139: show ip bgp l2vpn all next-hops Output Fields *(continued)*

Field Name	Field Description
Resolution	Describes where the indirect next hop is resolved (the IP routing table, the IP tunnel routing table, or both) and whether this is in a VR or VRF
IP indirect next-hop index	Index number of the IP indirect next hop that corresponds to the BGP indirect next hop and its resolution
MPLS indirect next-hop index	Index number of the MPLS indirect next hop that corresponds to the BGP indirect next hop and its resolution
Reachable	Indicates whether or not the indirect next hop is reachable. For more information about the reachability rules that apply for various route types, see the command description for *show ip bgp next-hops*.
metric	Metric number of the BGP indirect next hop
Number of direct next-hops	Number of the equal-cost legs of direct next hops to which this indirect next hop resolves
Direct next-hop	MPLS next-hop index that resolves the MPLS indirect next hop
Reference count	Number of label mappings of BGP routes that use this next hop

Related Documentation

- *show ip bgp next-hops*

Monitoring VPWS Connections

Purpose Display configuration and status information for VPWS L2VPN connections configured on the router. The **details** keyword displays detailed information about the connections.

Action To display detailed information about connections for all VPWS instances configured on the router:

```
host1:pe1# show l2vpn connections details
L2VPN: l2vpn1
    Encapsulation Type Ethernet
    Use of control word is preferred
    Send sequence numbers
    Route Distinguisher 100:11
    Site Range 10
    Sites:
      Site Name westford Site Id 1
       Multi-homed:            Yes
       Site-Priority:          45    Route Targets:
      Route Target: RT:100:1 both
                                                     Admin    Oper
   Interface       Local-Site-Id   Remote-Site-Id    state    state
---------------    -------------   --------------   -------   -----
FastEthernet4/1    1               2                enabled   up
```

```
Connections status code:
UP = Operational
SC = Local and Remote Site Identifier Collision
EM = Encapsulation Mismatch
OR = Out of Range
DN = VC Down because Remote PE Unreachable
LD = Local Site Down
RD = Remote Site Down
AS = Max BGP AS path length exceeded
OL = No Out Label
CM = Control Word Mismatch
LN = Local Site not Designated
RN = Remote Site not Designated

Local   Remote          Remote                               MPLS     Up-down
Site     Site    State     PE      In-label   Out-label    NH Idx      Time
-----   ------   -----   -------   --------   ---------   --------   --------
1       2        UP      2.2.2.2   25         74          0000001d   01:49:12
L2VPN: l2vpn2
    Encapsulation Type ATM AAL5 SDU VCC transport
    Use of control word is preferred
    Send sequence numbers
    Route Distinguisher 100:12
    Site Range 20
    Sites:
      Site Name boston Site Id 3
       Multi-homed:            Yes
       Site-Priority:          100    Route Targets:
      Route Target: RT:100:2 both
                                               Admin    Oper
Interface    Local-Site-Id   Remote-Site-Id    state    state
----------   -------------   --------------   -------   -----
ATM2/0.122   1               2                enabled   up
ATM2/0.123   1               3                enabled   up
ATM2/0.124   3               1                enabled   up
ATM2/0.121   3               2                enabled   up
Connections status code:
UP = Operational
SC = Local and Remote Site Identifier Collision
EM = Encapsulation Mismatch
OR = Out of Range
DN = VC Down because Remote PE Unreachable
LD = Local Site Down
RD = Remote Site Down
AS = Max BGP AS path length exceeded
OL = No Out Label
CM = Control Word Mismatch
LN = Local Site not Designated
RN = Remote Site not Designated

Local   Remote          Remote                                        MPLS
Up-down
Site     Site    State     PE        In-label        Out-label      NH Idx
Time
-----   ------   -----   -------   -------------   -------------   --------
--------
1       2        UP      2.2.2.2   35              84              0000001d
01:50:40
1       3        UP                implicit-null   implicit-null   2d000008
02:24:45
3       1        UP                implicit-null   implicit-null   2d000007
```

```
02:24:45
3       2        UP      2.2.2.2   55              86              0000001d
01:50:40
```

To display detailed information about connections for a specific VPWS instance:

```
host1#show l2vpn connections instance l2vpn1 details
L2VPN: l2vpn1
    Encapsulation Type ATM AAL5 SDU VCC transport
    Use of control word is preferred
    Send sequence numbers
    Route Distinguisher 100:11
    Site Range 10
    Sites:
      Site Name westford Site Id 1
      Multihomed priority 200
     Route Targets:
      Route Target: RT:100:2 both

                                               Admin    Oper
Interface    Local-Site-Id   Remote-Site-Id    state    state
----------   -------------   --------------   -------   -----
ATM2/0.100   1               2                enabled   up
ATM2/0.12    1               3                enabled   up

Connections status code:
UP = Operational
SC = Local and Remote Site Identifier Collision
EM = Encapsulation Mismatch
OR = Out of Range
DN = VC Down because Remote PE Unreachable
LD = Local Site Down
RD = Remote Site Down
AS = Max BGP AS path length exceeded
OL = No Out Label
CM = Control Word Mismatch
LN = Local Site not Designated
RN = Remote Site not Designated

               Remote                                MPLS       Up-down
Site    State    PE       In-label    Out-label    NH Idx       Time
----    -----   --------  --------    ---------   --------   -----------
2       UP      2.2.2.2   17          801024      00000014   1d 08:45:34
3       UP      2.2.2.2   18          801028      00000014   1d 08:45:34
```

Meaning Table 140 on page 700 lists the **show l2vpn connections** command output fields.

Table 140: show l2vpn connections Output Fields

Field Name	Field Description
L2VPN	Name of the VPWS instance
Encapsulation Type	Encapsulation type configured for the VPWS instance
Use of control word	Local preference for control word, preferred or not preferred
sequence numbers	Local preference for sequence number, send or don't send

Table 140: show l2vpn connections Output Fields *(continued)*

Field Name	Field Description
Route Distinguisher	Route distinguisher configured for the VPWS instance
Site Range	Maximum number of customer sites allowed in the VPWS instance
Sites	Site name and site ID for each customer site in the VPWS instance
Multihomed priority	Priority of the VPWS instance to serve as the backup PE router for the CE device in the event of a network failure in the multihomed configuration; indicates also that the site is multihomed
Route Targets	Route targets configured for the VPWS instance
Interface	Layer 2 interface that is a member of the VPWS instance
Local-Site-Id	Local customer site ID configured on the layer 2 interface
Remote-Site-Id	Remote customer site ID configured on the layer 2 interface

Related Documentation

- *show l2vpn connections*

Monitoring VPWS Instances

Purpose To display configuration and status information for VPWS instances configured on the router. You can display information for all VPWS instances or information about a particular VPWS instance. The **detail** keyword displays detailed information about the specified VPWS instance or all VPWS instances.

Action To display information about all VPWS instances configured on the router:

host1#**show l2vpn all**

```
L2VPN: l2vpn1
    Encapsulation Type Ethernet
    Use of control word is preferred
    Send sequence numbers
    Route Distinguisher 100:11
    Site Range 10
    Sites:
      Site Name boston Site Id 1
    Route Targets:
      Route Target: RT:100:1 both

L2VPN: l2vpn2
    Encapsulation Type ATM AAL5 SDU VCC transport
    Use of control word is preferred
    Send sequence numbers
    Route Distinguisher 100:12
    Site Range 20
    Sites:
```



```
      Site Name westford Site Id 1
      Site Name boston Site Id 3
    Route Targets:
      Route Target: RT:100:2 both
```

To display detailed information about all VPWS instances configured on the router:

host1#**show l2vpn instance all detail**

```
L2VPN: l2vpn1
    Encapsulation Type Ethernet
    Use of control word is preferred
    Send sequence numbers
    Route Distinguisher 100:11
    Site Range 10
    Sites:
      Site Name boston Site Id 1
    Route Targets:
      Route Target: RT:100:1 both

                                                 Admin    Oper
   Interface      Local-Site-Id   Remote-Site-Id   state    state
---------------   -------------   --------------   -------  -----
FastEthernet4/1   1               2                enabled  up

L2VPN: l2vpn2
    Encapsulation Type ATM AAL5 SDU VCC transport
    Use of control word is preferred
    Send sequence numbers
    Route Distinguisher 100:12
    Site Range 20
    Sites:
      Site Name westford Site Id 1
      Site Name boston Site Id 3
    Route Targets:
      Route Target: RT:100:2 both

                                            Admin    Oper
Interface    Local-Site-Id   Remote-Site-Id   state    state
----------   -------------   --------------   -------  -----
ATM2/0.122   1               2                enabled  up
ATM2/0.123   1               3                enabled  up
ATM2/0.124   3               1                enabled  up
ATM2/0.121   3               2                enabled  up
```

To display detailed information about a particular VPWS instance:

host1#**show l2vpn instance l2vpn1 detail**

```
L2VPN: l2vpn1
    Encapsulation Type ATM AAL5 SDU VCC transport
    Use of control word is preferred
    Send sequence numbers
    Route Distinguisher 100:11
    Site Range 10
    Sites:
      Site Name westford Site Id 1
        Multi-homed:           Yes
        Site-Priority:         45

    Route Targets:
      Route Target: RT:100:2 both

                                            Admin    Oper
Interface    Local-Site-Id   Remote-Site-Id   state    state
```



```
----------    -------------    --------------    -------    -----
ATM2/0.100    1                2                 enabled    up
ATM2/0.12     1                3                 enabled    up
ATM1/1.1      1                2                 enabled    down
```

Meaning Table 141 on page 703 lists the **show l2vpn instance** command output fields.

Table 141: show l2vpn instance Output Fields

Field Name	Field Description
L2VPN	Name of VPWS instance
Encapsulation Type	Encapsulation type configured for the VPWS instance
Use of control word	Local preference for control word, preferred or not preferred
sequence numbers	Local preference for sequence number, send or don't send
Route Distinguisher	Route distinguisher configured for the VPWS instance
Site Range	Maximum number of customer sites allowed in the L2VPN instance
Sites	Site name and site ID for each customer site in the VPWS instance
Multi-homed	Status of the site. Yes designates a multihomed site. No designates a site that is not multihomed.
Site-Priority	Priority value for the VPWS instance for the multihomed site; displayed only when the value for the Multi-homed field is Yes
Route Targets	Route targets configured for the VPWS instance
Interface	Layer 2 interface that is a member of the VPWS instance
Local-Site-Id	Local customer site ID configured on the layer 2 interface
Remote-Site-Id	Remote customer site ID configured on the layer 2 interface
Admin state	Administrative state of the connection, disabled or enabled
Oper state	Operational state of the connection, up or down

Related Documentation

- *show l2vpn instance*

Monitoring L2VPN Interfaces for VPWS

Purpose Display configuration and status information for interfaces on the router that are that are configured to be members of VPWS L2VPNs in the current VR. You can display information for a specific L2VPN interface, for all L2VPN interfaces in the specified VPWS instance, or for all L2VPN interfaces in all VPWS instances. The **detail** keyword displays detailed information about the specified L2VPN interface or all L2VPN interfaces.

Action To display L2VPN interface information for a particular VPWS instance:

```
host1#show l2vpn interface instance l2vpn1
MPLS shim interface ATM2/0.100
  ATM circuit type is AAL5
  Member of L2VPN instance l2vpn1
  Local site ID is 1
  Remote site ID is 2
  Control word is preferred by default
  Do send sequence numbers by default
  Relay format is atm-aal5-sdu-vcc by default
  Administrative state is enabled
  Operational state is up
  Operational MTU is 9180
  MPLS shim interface UID is 0x2d000007
  Lower interface UID is 0x0b000005
  Condensed location is 0x00020000
  Received:
    3 packets
    204 bytes
    19 errors
    0 discards
  Sent:
    0 packets
    0 bytes
    0 errors
    0 discards
  queue 0: traffic class best-effort, bound to atm-vc ATM2/0.100
    Queue length 0 bytes
    Forwarded packets 0, bytes 0
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0
```

Meaning Table 142 on page 704 lists the **show l2vpn interface** command output fields.

Table 142: show l2vpn interface Output Fields

Field Name	Field Description
MPLS shim interface	Type and specifier for MPLS shim interface
ATM circuit type	Type of ATM circuit
Member of L2VPN instance	Name of the VPWS instance to which the interface belongs
Local site ID	Local customer site ID configured on the interface
Remote site ID	Remote customer site ID configured on the interface
Control word	Local preference for the control word, preferred or not preferred
send sequence numbers	Local preference for sequence numbers, send or don't send

Table 142: show l2vpn interface Output Fields *(continued)*

Field Name	Field Description
Relay format	Type of signaling and encapsulation used by the router for layer 2 traffic
Administrative state	Administrative state of the interface, enabled or disabled
Operational state	Operational state of the interface, up or down
Operational MTU	Maximum allowable size in bytes of the maximum transmission unit for the interface
MPLS shim interface UID	UID automatically assigned to the MPLS shim interface when it is created
Lower interface UID	UID automatically assigned to the MPLS major interface when it is created
Condensed location	Internal, platform-dependent, 32-bit representation of the interface location, used by Juniper Networks Customer support for troubleshooting.
Received	Number of packets, bytes, errors and discards received on the interface
Sent	Number of packets, bytes, errors and discards sent from the interface
queue	Number of messages queued to be sent on the interface
traffic-class	Type of traffic class configured for traffic on the interface
bound to	ATM virtual circuit to which the interface is bound
Queue length	Length of all messages queued to be sent to on this connection, in bytes
Forwarded	Number of packets and bytes that have been forwarded
Dropped committed	Number of committed packets and bytes that have been dropped
Dropped conformed packets	Number of conformed packets and bytes that have been dropped
Dropped exceeded	Number of exceeded packets and bytes that have been dropped

Related Documentation

- *show l2vpn interface*

Monitoring MPLS Forwarding Table for VPWS

Purpose Display information about MPLS labels that are being used for forwarding. The **brief** keyword displays summary information for the MPLS labels.

Action To display MPLS forwarding information for a particular label:

```
host1#show mpls forwarding label 17
In label: 17
  Label space: platform label space
  Owner: bgp
  Spoof check: router ERX-pe
  Action:
    MPLS next-hop: 28, l2transport to ATM2/0.100
  Statistics:
    0 in pkts
    0 in Octets
    0 in errors
    0 in discard pkts
```

To display brief information about MPLS forwarding for all labels:

```
host1:pe1#  show mpls forwarding brief


In Label  Owner                              Action
-------- -------- -------------------------------------------------------------
17       bgp      l2transport to ATM2/0.100
18       bgp      l2transport to ATM2/0.12
26       ldp      lookup on inner header/label
27       ldp      swap to 39 on ATM2/0.20, nbr 20.20.20.2
28       ldp      swap to 41 on ATM2/0.20, nbr 20.20.20.2
29       ldp      lookup on inner header/label
30       ldp      swap to 43 on ATM2/0.20, nbr 20.20.20.2
31       ldp      swap to 44 on ATM2/0.20, nbr 20.20.20.2
46       ldp      swap to 40 on ATM2/0.20, nbr 20.20.20.2
L2transport


     Interface     Owner                       Action
--------------- -------- ------------------------------------------
ATM2/0.12       bgp       swap to 801028, push 39 on ATM2/0.20, nbr 20.20.20.2

ATM2/0.100      bgp      swap to 801024, push 39 on ATM2/0.20, nbr 20.20.20.2
```

To display MPLS forwarding information for a particular interface:

```
host1:pe#show mpls forwarding interface atm2/0.100
In label: n/a, ATM2/0.100
  Owner: bgp
  Spoof check: router erx-pe
  Action:
    MPLS next-hop: 27, label 801024, resolved by MPLS next-hop 8
      MPLS next-hop: 8, resolved by MPLS next-hop 9, peer 10.3.2.2
        MPLS next-hop: 9, label 39 on ATM2/0.20, nbr 10.20.20.2
  Statistics: Disabled
```

Meaning Table 143 on page 707 lists the **show mpls forwarding** command output fields.

Table 143: show mpls forwarding Output Fields

Field Name	Field Description
In label	Label sent to upstream neighbor for route
Out label	Label received from downstream neighbor for route
Label space	Label space in which the label is assigned
Owner	Signaling protocol that placed the label in the forwarding table: BGP, LDP, or RSVP-TE
Spoof check	Type and location of spoof checking performed on the MPLS packet, router, or interface
Action	Action taken for MPLS packets arriving with that label
in pkts	Number of packets sent with the label
in Octets	Number of octets sent with the label
in errors	Number of packets that are dropped for some reason before being sent
in discardPkts	Number of packets that are discarded due to lack of buffer space before being sent
Interface	Layer 2 interface that is a member of an L2VPN

Related Documentation

- *show mpls forwarding*

PART 6

Index

Index

Symbols

M

R

W