

Routing Protocols Configuration Guide

SmartEdge OS

Release 5.0.3

Part Number 220-0584-01



Corporate Headquarters
Redback Networks Inc.
300 Holger Way
San Jose, CA 95134-1362
USA
<http://www.redback.com>
Tel: +1 408 750 5000

© 1998–2005, Redback Networks Inc. All rights reserved.

Redback and SmartEdge are trademarks registered at the U.S. Patent & Trademark Office and in other countries. AOS, NetOp, SMS, and User Intelligent Networks are trademarks or service marks of Redback Networks Inc. All other products or services mentioned are the trademarks, service marks, registered trademarks or registered service marks of their respective owners. All rights in copyright are reserved to the copyright owner. Company and product names are trademarks or registered trademarks of their respective owners. Neither the name of any third party software developer nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission of such third party.

Rights and Restrictions

All statements, specifications, recommendations, and technical information contained are current or planned as of the date of publication of this document. They are reliable as of the time of this writing and are presented without warranty of any kind, expressed or implied. In an effort to continuously improve the product and add features, Redback Networks Inc. ("Redback") reserves the right to change any specifications contained in this document without prior notice of any kind.

Redback shall not be liable for technical or editorial errors or omissions which may occur in this document. Redback shall not be liable for any indirect, special, incidental or consequential damages resulting from the furnishing, performance, or use of this document.

Third Party Software

The following third party software may be included with this Software and is subject to the following terms and conditions:

The OpenLDAP Version 2.0.1 © 1999 The OpenLDAP Foundation; OpenSymphony Software License, Version 1.1 2001-2004 © The OpenSymphony Group; TOAD © 2004 Quest Software, Inc.; NuSOAP Web Services Toolkit for PHP © 2002 NuSphere Corporation; The PHP License, versions 2.02 and 3.0 © 1999 - 2002 The PHP Group; The OpenSSL toolkit Copyright © 1998-2003 The OpenSSL Project; Apache HTTP © 2000 The Apache Software Foundation; Java © 2003 Sun Microsystems, Inc.; ISC Dhcpd 3.0pl2 © 1995, 1996, 1997, 1998, 1999 Internet Software Consortium - DHCP; IpFilter © 2003 Darren Reed; Perl Kit © 1989-1999 Larry Wall; SNMP Monolithic Agent © 2002 SNMP Research International, Inc.; VxWorks © 1984-2000, Wind River Systems, Inc.; Point-to-Point Protocol (PPP) © 1989, Carnegie-Mellon University; Dynamic Host Configuration Protocol (DHCP) © 1997, 1998 The Internet Software Consortium; portions of the Redback SmartEdge Operating System use cryptographic software written by Eric Young (eay@cryptsoft.com); Redback adaptation and implementation of the UDP and TCP protocols developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. © 1982, 1986, 1988, 1990, 1993, 1995 The Regents of the University of California. All advertising materials mentioning features or use of this Software must display the following acknowledgment: "This product includes software developed by the University of California, Berkeley and its contributors."

This Software includes software developed by Sun Microsystems, Inc., Internet Software Consortium, Larry Wall, the Apache Software Foundation (<http://www.apache.org/>) and their contributors. Such software is provided "AS IS," without a warranty of any kind. ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. LICENSORS AND ITS CONTRIBUTORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THIS SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL LICENSOR OR ITS CONTRIBUTORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE THIS SOFTWARE, EVEN IF THE LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see <http://www.apache.org/>. Portions of this software are based upon public domain software originally written at the National Center for Supercomputing Applications, University of Illinois, Urbana-Champaign. The portions of this Software developed by Larry Wall may be distributed and are subject to the GNU General Public License as published by the Free Software Foundation.

FCC Notice

The following information is for FCC compliance of Class A devices: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio-frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case users will be required to correct the interference at their own expense.

1. MODIFICATIONS

The FCC requires the user to be notified that any changes or modifications made to this device that are not expressly approved by Redback could void the user's authority to operate the equipment.

2. CABLES

Connection to this device must be made with shielded cables with metallic RFI/EMI connector hoods to maintain compliance with FCC Rules and Regulations. (This statement only applies to copper cables, Ethernet, DS-3, E1, T1, and so forth. It does not apply to fiber cables.)

3. POWER CORD SET REQUIREMENTS

The power cord set used with the System must meet the requirements of the country, whether it is 100-120 or 220-264 VAC. For the U.S. and Canada, the cord set must be UL Listed and CSA Certified and suitable for the input current of the system.

For DC-powered systems, the installation instructions need to be followed.

VCCI Class A Statement

この装置は、情報処理装置等電波障害自主規制協議会（VCCI）の基準に基づくクラスA情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。

European Community Mark



The marking on this product signifies that it meets all relevant European Union directives.

Safety Notices

1. Laser Equipment:

CAUTION! Use of controls or adjustments of performance or procedures other than those specified herein may result in hazardous radiation exposure.

Class 1 Laser Product—Product is certified by the manufacturer to comply with DHHS Rule 21 Subchapter J.

CAUTION! Invisible laser radiation when an optical interface is open.

2. Lithium Battery Warnings:

It is recommended that, when required, Redback replace the lithium battery.

WARNING! Do not mutilate, puncture, or dispose of batteries in fire. The batteries can burst or explode, releasing hazardous chemicals. Discard used batteries according to the manufacturer's instructions and in accordance with your local regulations.

Danger of explosion if battery is incorrectly replaced. Replace only with the same or equivalent type as recommended by the manufacturer's instructions.

VARNING Explosionsfara vid felaktigt batteribyte. Använd samma batterityp eller en ekvivalent typ som rekommenderas av apparattillverkaren. Kassera använt batteri enligt fabrikantens instruktion.

ADVARSEL! Lithiumbatteri—Eksplosionsfare ved fejlagtig håndtering. Udskiftning må kun ske med batteri af samme fabrikat og type. Levér det brugte batteri tilbage til leverandøren.

VARIOTUS Paristo voi räjähtää, jos se on virheellisesti asennettu. Vaihda paristo ainoastaan valmistajan suosittelemaan tyyppiin. Hävitä käytetty paristo valmistajan ohjeiden mukaisesti.

ADVARSEL Eksplosjonsfare ved feilaktig skifte av batteri. Benytt samme batteritype eller en tilsvarende type anbefalt av apparatfabrikanten. Brukte batterier kasseres i henhold til fabrikantens instruksjoner.

WAARSCHUWING! Bij dit produkt zijn batterijen geleverd. Wanneer deze leeg zijn, moet u ze niet weggooien maar inleveren als KCA.

Contents

About This Guide	xxi
Related Publications	xxii
Intended Audience	xxiii
Organization	xxiii
Conventions	xxiv
Command Modes and Privilege Levels	xxiv
Command Syntax	xxiv
Examples	xxv
Task Tables	xxvi
Online Navigation Aids	xxvi
Ordering Documentation	xxvi

Part 1: Introduction

Chapter 1: Overview	1-1
SmartEdge Routing	1-1
Static Versus Dynamic Routing	1-2
IGPs Versus EGPs	1-2
Supported IP Routing Protocols and Routing-Related Features	1-2
Basic IP Routing	1-3
Dynamically Verified Static Routing	1-3
Virtual Router Redundancy Protocol	1-3
Routing Information Protocol	1-3
Open Shortest Path First	1-4
Bidirectional Forwarding Detection	1-4
Border Gateway Protocol	1-4
Border Gateway Protocol/Multiprotocol Label Switching Virtual Private Network	1-4
Intermediate System-to-Intermediate System Routing	1-5
IP Multicast	1-5
Routing Policy	1-5
Multiprotocol Label Switching	1-5
Layer 2 Virtual Private Network	1-6
Label Distribution Protocol	1-6
Virtual Private LAN Services	1-6
Protocol Distances	1-6
Command Mode Hierarchy	1-7

Part 2: IP Routing

Chapter 2: Basic IP Routing Configuration	2-1
Overview	2-1
Static Versus Dynamic Routing	2-2
IGPs Versus EGPs	2-2
IP Routing Protocols	2-3
Protocol Distances	2-3
Configuration Tasks	2-4
Configuring Static Routes	2-4
Configuring Additional Basic IP Routing Parameters	2-5
Configuration Examples	2-5
Command Descriptions	2-6
ip martian	2-7
ip maximum-routes	2-9
ip mstatic	2-11
ip route	2-12
ipv6 route	2-15
ip verify unicast source	2-17
router-id	2-19
service inter-context routing	2-20
tcp path-mtu-discovery	2-21
Chapter 3: DVSR Configuration	3-1
Overview	3-1
Configuration Tasks	3-2
Configuring a DVSR Profile	3-3
Configuration Examples	3-3
Basic DVSR	3-3
DVSR in Anycast Application	3-4
DVSR in Customer Multihoming Application	3-5
Command Descriptions	3-6
distance	3-7
dvsr-profile	3-8
source-address	3-9
tag	3-10
ttl	3-11
verify-set	3-12
Chapter 4: VRRP Configuration	4-1
Overview	4-1
Configuration Tasks	4-2
Configuring a VRRP Owner Router	4-2
Configuring a VRRP Backup Router	4-3
Configuration Examples	4-3
Basic VRRP	4-3
Mutual VRRP	4-4
Mutual VRRP on Different Subnets	4-5
Mutual VRRP on Multiple Subnets	4-6
MD5 Authentication	4-7
Command Descriptions	4-8
advertise-interval	4-9
authentication	4-10

preempt	4-11
priority	4-12
virtual-address	4-14
vrrp	4-15
Chapter 5: RIP Configuration	5-1
Overview	5-1
Configuration Tasks	5-2
Configuring RIP	5-2
Configure a RIP Routing Instance	5-2
Configure a RIP Interface	5-3
Configuring RIPv2	5-3
Configure a RIPv2 Routing Instance	5-4
Configure a RIPv2 Interface	5-4
Configuration Examples	5-5
Command Descriptions	5-6
authentication	5-7
default-information originate	5-9
default-metric	5-11
distance	5-12
distribute-list	5-13
flash-update-threshold	5-14
interface	5-15
interface-cost	5-17
listen	5-18
maximum-paths	5-19
offset-list	5-20
output-delay	5-21
redistribute	5-22
router rip	5-24
router ripng	5-25
split-horizon	5-26
summary-address	5-28
supply	5-30
timers basic	5-31
Chapter 6: OSPF Configuration	6-1
Overview	6-1
Areas	6-3
Normal and Backbone	6-3
Stub	6-3
Not-So-Stubby-Area	6-3
Router Functions	6-4
Route Selection Process	6-4
Packet Types	6-4
Link-State Advertisements	6-5
Sham Links	6-6
Virtual Links	6-6
OSPFv3	6-7
Configuration Tasks	6-8
Configuring OSPF	6-8
Configure an OSPF Routing Instance	6-8
Configure the Redistribution of Routes into OSPF	6-10
Configure an OSPF Area	6-10

Configure an OSPF Interface	6-11
Configure an OSPF Sham Link	6-12
Configure an OSPF Virtual Link	6-13
Configuring OSPFv3	6-13
Configure an OSPFv3 Routing Instance	6-14
Configure the Redistribution of Routes into OSPFv3	6-15
Configure an OSPFv3 Area	6-15
Configure an OSPFv3 Interface	6-15
Configure an OSPF Virtual Link	6-17
Configuration Examples	6-17
Basic OSPF	6-18
Redistribution	6-20
MD5 Authentication	6-21
Simple Key Chain	6-22
Command Descriptions	6-23
area	6-24
area-type	6-26
authentication	6-28
auto-cost	6-30
block-flooding	6-31
capabilities	6-32
cost	6-34
default-metric	6-35
default-route	6-36
demand-circuit	6-38
distance	6-40
fast-hello	6-41
fast-lsa-origination	6-43
flood-reduction	6-44
graceful-restart	6-45
hello-interval	6-46
interface	6-48
log-neighbor-up-down	6-50
maximum redistribute	6-51
maximum redistribute-quantum	6-52
mpls shortcuts	6-53
mpls traffic-engineering	6-54
neighbor	6-55
network-type	6-57
nssa-range	6-59
originate-default	6-61
passive	6-63
range	6-64
redistribute	6-65
retransmit-interval	6-68
router-dead-interval	6-69
router-id	6-71
router ospf	6-72
router ospf3	6-73
router-priority	6-74
sham-link	6-75
spf-timers	6-77
stub-router	6-78
summary-address	6-80

transmit-delay	6-82
virtual-link	6-83
Chapter 7: BFD Configuration	7-1
Overview	7-1
Configuration Tasks	7-2
Configuring a BFD Neighbor	7-2
Configuring BFD on an Interface	7-3
Enabling or Disabling BFD for a Routing Interface	7-4
Configuration Examples	7-4
BFD Neighbor	7-4
BFD Interface	7-5
Disabling BFD	7-5
Command Descriptions	7-5
bfd detection	7-6
detection-multiplier	7-7
interface	7-8
minimum receive-interval	7-9
minimum transmit-interval	7-10
neighbor	7-11
router bfd	7-12
Chapter 8: BGP Configuration	8-1
Overview	8-1
iBGP and eBGP	8-3
iBGP Route Reflectors	8-4
iBGP Confederations	8-5
Route Aggregation	8-6
MBGP	8-6
Routing Policy Triggered Update	8-6
Non-Intrusive MD5 Password Change	8-7
Replace a Password	8-7
Add a New Password	8-7
Delete a Password	8-7
Configuration Tasks	8-8
Configuring BGP Routing Instances and Instance Attributes	8-8
Configure a BGP Routing Instance	8-8
Configure IPv4 Address Family Attributes for a BGP Routing Instance	8-9
Configure IPv6 Address Family Attributes for a BGP Routing Instance	8-10
Configure Graceful Restart for a BGP Routing Instance	8-10
Configure BGP Route Reflection	8-11
Configure a BGP Confederation	8-11
Configuring BGP Neighbors and Neighbor Attributes	8-12
Configure a BGP Neighbor	8-12
Configure IPv4 Address Family Attributes for a BGP Neighbor	8-14
Configure IPv6 Address Family Attributes for a BGP Neighbor	8-15
Configure Graceful Restart for a BGP Neighbor	8-16
Configuring BGP Peer Groups and Peer Group Attributes	8-16
Configure a BGP Peer Group	8-16
Configure IPv4 Address Family Attributes for a BGP Peer Group	8-17
Configure IPv6 Address Family Attributes for a BGP Peer Group	8-18
Apply Peer Group Attributes	8-19

Configuration Examples	8-19
Basic BGP	8-19
iMBGP Peer	8-20
iMBGP Peer Group	8-21
eMBGP Peer	8-22
eMBGP Peer Group	8-23
Command Descriptions	8-25
accept filter prefix-list	8-26
address-family ipv4	8-28
address-family ipv6 unicast	8-30
advertisement-interval	8-32
aggregate-address	8-34
asloop-in	8-36
as-override	8-38
as-path-list	8-40
bestpath med always-compare	8-42
client-to-client reflection	8-43
cluster-id	8-44
confederation identifier	8-45
confederation peers	8-46
dampening	8-47
default-originate	8-49
description	8-51
distance	8-52
ebgp-multihop	8-53
enforce ttl	8-54
fast-reset	8-56
flap-statistics	8-57
local-as	8-58
local-preference	8-60
log-neighbor-changes	8-61
maximum prefix	8-62
maximum restart-time	8-64
maximum retain-time	8-65
maximum update-delay	8-67
multi-paths	8-68
neighbor	8-70
network	8-72
next-hop-self	8-74
password	8-76
peer-group	8-77
prefix-list	8-80
redistribute	8-82
remote-as	8-84
remove-private-as	8-85
retain-ibgp-routes	8-86
route-map	8-87
route-origin	8-89
router bgp	8-91
route-reflector-client	8-92
router-id	8-94
send community	8-95
send ext-community	8-96
send filter prefix-list	8-98

send label	8-100
session-dampening	8-102
shutdown	8-104
table-map	8-105
timer password	8-106
timers	8-107
update-source	8-109
Chapter 9: BGP/MPLS VPN Configuration	9-1
Overview	9-1
Virtual Private Networks	9-2
VPN Topology	9-2
Packet Labels	9-2
Multiple VPN Contexts	9-2
VPN-IPv4 Address Family	9-3
Route Distribution Among PE Routers by BGP	9-3
PE-to-CE Route Distribution	9-3
Route Target Attribute	9-4
Site of Origin Attribute	9-4
BGP/MPLS VPN over GRE	9-4
GRE over MPLS	9-4
Carrier of Carriers	9-5
Multihop eBGP Label Redistribution	9-5
Configuration Tasks	9-6
Configuring a VPN-IPv4 Address Family for BGP Sessions Between PE Routers	9-7
Creating a New VPN Context	9-7
Configuring a BGP Routing Instance in a VPN Context	9-8
Configuring Multipath Load Balancing in a BGP/MPLS VPN	9-8
Configuring the Next-Hop Reachability Check for VPN Routes	9-9
Configuring Route Targets	9-9
Configuring PE-to-CE Routing	9-10
Identifying the Specific Site from Where a Route Has Originated	9-10
Enabling Soft GRE Tunneling	9-11
Configuration Examples	9-11
Backbone Connectivity	9-11
PE-to-CE Route Distribution	9-13
VPN Using Static Routing	9-13
VPN Using RIP	9-14
VPN Using OSPF	9-14
VPN Using eBGP	9-15
Different BGP/MPLS VPN Topologies	9-16
Typical BGP/MPLS VPN	9-16
Local Import	9-19
Hub-and-Spoke	9-22
GRE over MPLS	9-26
BGP/MPLS VPN over GRE	9-28
New BGP Commands for BGP/MPLS VPN	9-30
Using the asloop-in Command	9-31
Using the as-override Command	9-31
Using the route-origin Command	9-33
CoC	9-34
Multihop eBGP Label Redistribution	9-43

Command Descriptions	9-49
address-family ipv4 vpn	9-50
context vpn-rd	9-52
export route-target	9-54
import route-target	9-56
ip soft-gre	9-58
multi-paths eibgp	9-60
next-hop-on-lsp	9-62
router bgp vpn	9-64
route-target filter	9-66
vpn	9-67
Chapter 10: IS-IS Configuration	10-1
Overview	10-1
Supported IS-IS Features	10-1
IS-IS Packets	10-2
Configuration Tasks	10-3
Configuring an IS-IS Instance	10-3
Configuring an IS-IS LSP	10-6
Configuring IS-IS SPF Calculations	10-6
Configuring an IS-IS Interface	10-7
Configuring IS-IS Hello Packets	10-8
Configuring IS-IS Interface LSPs	10-9
Configuring IS-IS Interface Metrics	10-9
Configuration Examples	10-10
Basic IS-IS	10-10
Two Routers Using IS-IS for Routing Information Exchange	10-11
IS-IS P2P-over-LAN Circuit	10-12
Three Routers Using IS-IS for Routing Information Exchange	10-13
Basic Multitopology IS-IS	10-16
Command Descriptions	10-17
address-family	10-18
attached-bit	10-20
authentication	10-22
circuit mtu	10-24
circuit type	10-25
csnp interval	10-26
csnp periodic-on-ptp	10-28
distance	10-29
dynamic-hostname	10-31
fast-convergence	10-33
hello interval	10-34
hello multiplier	10-36
hello padding	10-38
interarea-distribute	10-39
interface	10-41
is type	10-43
lsp block-flooding	10-45
lsp gen-interval	10-46
lsp interval	10-47
lsp max-lifetime	10-48
lsp receive-only-mode	10-49
lsp refresh-interval	10-50
lsp retransmit-interval	10-51

maximum paths	10-52
maximum redistribute	10-53
metric	10-54
metric-style	10-56
net	10-58
optional-checksums	10-59
passive-interface	10-60
priority	10-61
redistribute	10-63
router isis	10-65
set-overload-bit	10-66
spf holddown	10-68
spf interval	10-69
summary-address	10-70
traffic-engineering	10-72
Chapter 11: IP Multicast Configuration	11-1
Overview	11-1
Internet Group Management Protocol	11-2
IGMP Bandwidth Limitation	11-2
IGMP Membership Tracking	11-2
Membership Tracking with IGMPv2	11-3
Membership Tracking with IGMPv3	11-3
Protocol Independent Multicast	11-3
Protocol Independent Multicast-Dense Mode	11-3
Protocol Independent Multicast-Sparse Mode	11-4
Source-Specific Multicast	11-4
Multicast Source Discovery Protocol	11-5
Anycast RP	11-5
Multicast VPNs	11-6
Remote Multicast Replication	11-7
Configuration Tasks	11-7
Configuring IGMP	11-8
Configuring an IGMP Service Profile	11-9
Configuring PIM-DM	11-10
Configuring PIM-SM	11-10
Configuring MSDP	11-12
Configuring an MSDP Peer	11-12
Configuring Multicast for Subscribers	11-13
Enabling PIM Graceful Restart	11-14
Enabling SSM	11-14
Enabling Multicast VPNs	11-14
Enabling RMR	11-15
Configuration Examples	11-15
PIM-SM	11-16
MSDP for Two PIM-SM Domains	11-17
Multicast VPNs	11-21
Remote Multicast Replication	11-26
Anycast RP	11-27
Command Descriptions	11-30
default-peer	11-31
description	11-33
igmp access-group	11-34
igmp group-bandwidth	11-35

igmp join-group	11-36
igmp last-member-query-interval	11-37
igmp maximum-bandwidth	11-38
igmp mtrace-prohibit	11-40
igmp query-interval	11-41
igmp query-max-response-time	11-42
igmp robust	11-43
igmp service-profile	11-44
igmp version	11-46
instant-leave	11-47
ip igmp service-profile	11-48
ip multicast boundary	11-49
ip multicast receive	11-50
ip multicast send	11-52
max-groups	11-54
mdt default-group	11-56
mdt encapsulation	11-57
mesh-group	11-58
multicast destination	11-59
multicast output	11-61
originating-rp	11-63
originating-rp sa-filter	11-64
peer	11-65
peer-as	11-66
pim accept-rp	11-67
pim anycast-rp	11-69
pim bsr-border	11-70
pim bsr-candidate	11-71
pim dense-mode	11-72
pim dr-priority	11-73
pim graceful-restart	11-74
pim hello-interval	11-76
pim neighbor-filter	11-77
pim operation-mode	11-78
pim rp-address	11-79
pim rp-candidate	11-80
pim sparse-mode	11-81
pim spt-threshold infinity	11-82
pim ssm	11-83
pim static group	11-84
priority	11-85
router msdp	11-86
sa-filter	11-87
shutdown	11-89
static-group	11-90
sticky-groups	11-92
Chapter 12: Routing Policy Configuration	12-1
Overview	12-1
Configuration Tasks	12-2
Configuring AS Path Lists	12-2
Create an AS Path List	12-2
Configure an AS Path List Permit or Deny Condition	12-2

Configuring BGP Community Lists	12-3
Create a BGP Community List	12-3
Configure a BGP Community List Permit or Deny Condition	12-4
Configuring BGP Extended Community Lists	12-4
Create a BGP Extended Community List	12-5
Configure a BGP Extended Community List Permit or Deny Condition	12-5
Configuring IP Prefix Lists	12-6
Create an IP Prefix List	12-6
Configure an IP Prefix List Permit or Deny Condition	12-6
Configuring IPv6 Prefix Lists	12-7
Create an IPv6 Prefix List	12-7
Configure an IPv6 Prefix List Permit or Deny Condition	12-7
Configuring Route Maps	12-8
Create a Route Map	12-8
Configure a Match Condition	12-8
Configure a Set Condition	12-9
Configuring BGP Attribute-Based Accounting	12-10
Configuring BGP Destination-Based QoS	12-11
Configuration Examples	12-12
Simple IP Prefix List	12-12
Complex IP Prefix List	12-12
Simple AS Path List	12-13
Complex AS Path List	12-13
Simple Community List	12-14
Complex Community List	12-14
Simple Route Map	12-14
Complex Route Map	12-15
BGP Attribute-Based Accounting	12-15
BGP Destination-Based QoS	12-16
Command Descriptions	12-18
as-path-list	12-19
community-list	12-21
description	12-22
ext-community-list	12-23
ip prefix-list	12-25
ipv6 prefix-list	12-26
mark dscp destination	12-27
match as-path-list	12-29
match community-list	12-30
match ext-community-list	12-32
match ip address prefix-list	12-34
match ip next-hop prefix-list	12-35
match ipv6 address prefix-list	12-36
match ipv6 next-hop prefix-list	12-37
match metric	12-38
match route-type	12-39
match tag	12-41
{permit deny}	12-42
resequence as-path-list	12-46
resequence community-list	12-47
resequence ext-community-list	12-48
resequence ip prefix-list	12-49
resequence ipv6 prefix-list	12-50
resequence route-map	12-51

route-map	12-52
set as-path	12-54
set community	12-56
set community-list	12-58
set dampening	12-59
set dscp	12-61
set ext-community	12-62
set ip next-hop	12-64
set ipv6 next-hop	12-65
set label	12-66
set level	12-67
set local-preference	12-69
set metric	12-70
set metric-type	12-71
set origin	12-72
set tag	12-73
set traffic-index	12-74
set weight	12-75
traffic-index accounting	12-76

Part 3: MPLS Routing

Chapter 13: MPLS Configuration	13-1
Overview	13-1
MPLS Architecture	13-1
MPLS QoS	13-2
MPLS TTL	13-3
Next-Hop Fast Reroute	13-3
NFRR for Link Protection	13-4
NFRR for Node Protection	13-4
Configuration Tasks	13-5
Configuring MPLS	13-5
Create an MPLS Routing Instance	13-5
Configure the MPLS TTL	13-5
Configuring MPLS Static	13-6
Create an MPLS Static Routing Instance	13-6
Configure an MPLS Static interface	13-6
Configure an MPLS Static LSP	13-7
Configuring RSVP	13-7
Create an RSVP Routing Instance	13-7
Configure an RSVP LSP	13-8
Configure a Bypass RSVP LSP	13-9
Configure an Explicit Route	13-10
Configure an RSVP Interface	13-11
Configure the RSVP Reservation State Lifetime	13-11
Configure RSVP Graceful Restart	13-12
Configuration Examples	13-12
MPLS Static LSP Tunnel	13-12
RSVP LSP Tunnel	13-13
Command Descriptions	13-14
authentication	13-15
bandwidth	13-16
decrement ttl	13-17

description	13-18
egress	13-19
explicit-null	13-20
explicit-route	13-21
fast-reroute	13-22
graceful-restart	13-23
hello interval	13-24
hello keep-multiplier	13-26
igp-shortcut	13-27
ingress	13-29
interface	13-30
keep-multiplier	13-32
label-action	13-33
local-protection	13-35
log-lsp-up-down	13-36
lsp	13-37
next-hop	13-39
out-label	13-40
propagate ttl ip-to-mpls	13-41
propagate ttl mpls-to-ip	13-42
record-route	13-43
refresh-interval	13-44
router mpls	13-45
router mpls-static	13-46
router rsvp	13-47
rro-prefix-type	13-48
setup-priority	13-49
shutdown	13-50
source-path	13-51
Chapter 14: L2VPN Configuration	14-1
Overview	14-1
L2VPN Implementation	14-1
Supported Encapsulation Types	14-2
Frame Relay Martini Encapsulation	14-2
Ethernet VLAN	14-3
Ethernet	14-3
ATM AAL5	14-3
Supported Encapsulation Interconnectivity	14-3
QoS Policies for L2VPN Circuits	14-4
L2VPN over GRE	14-4
Configuration Tasks	14-4
Enabling an L2 Circuit for L2VPN Operation	14-5
Configuring an LDP L2VPN Cross-Connection	14-5
Configuring a Static L2VPN Cross-Connection	14-6
Enabling Soft GRE Tunneling	14-6
Configuration Examples	14-6
Static L2VPN	14-7
LDP L2VPN	14-7
LDP L2VPN with Frame Relay Martini Encapsulation	14-8
LDP L2VPN with Ethernet VLAN Encapsulation	14-10
LDP L2VPN with Ethernet Encapsulation	14-11
LDP L2VPN with ATM DS-3 Encapsulation	14-12
LDP L2VPN with ATM OC Encapsulation	14-13

CE Router with RFC 1483 Bridged Encapsulation for ATM AAL5	14-14
L2VPN for Extreme Networks Equipment Interoperability	14-14
QoS Rate Limiting Policy on Ingress L2VPN Circuits	14-17
QoS Metering Policies on Egress L2VPN Circuits	14-18
EXP-Bit for L2VPN VCs	14-18
dot1q Bit Propagation on L2VPN Cross-Connections	14-20
ATM RFC 1483 Bridged to dot1q Interconnection	14-21
ATM RFC 1483 Bridged to Ethernet Interconnection	14-22
L2VPN over GRE	14-23
Command Descriptions	14-24
ip soft-gre	14-25
l2vpn	14-27
l2vpn-cct-bindings ldp	14-28
l2vpn-cct-bindings static	14-29
l2vpn ctx-name	14-30
xc vc-id	14-32
xc vpn-label	14-35
Chapter 15: LDP Configuration	15-1
Overview	15-1
LDP Implementation	15-1
LDP Neighbor Discovery	15-2
LDP Hello Messages	15-2
Configuration Tasks	15-2
Configuring an LDP Routing Instance	15-3
Configuring the Hello Adjacency Holdtime (Optional)	15-5
Configuring the Hello Message Interval	15-6
Configuration Examples	15-6
Basic LDP	15-6
Targeted LDP	15-8
Command Descriptions	15-9
create-lsp-circuit	15-10
explicit-null	15-11
graceful-restart	15-13
hello holdtime	15-14
hello interval	15-16
interface	15-18
label-binding	15-20
neighbor password	15-22
neighbor targeted	15-23
router-id	15-25
router ldp	15-27
targeted-hello holdtime	15-29
targeted-hello interval	15-31
track-igp-metric	15-33
transport address	15-34
Chapter 16: VPLS Configuration	16-1
Overview	16-1
Configuration Tasks	16-2
Configuring a Bridge Profile	16-3
Configuring a VPLS Profile	16-4
Configuring a VPLS-Enabled Bridge	16-5

Configuration Examples	16-6
Bridge Profile	16-7
VPLS Profile	16-7
VPLS-Enabled Bridge	16-7
Command Descriptions	16-8
counters	16-9
description	16-10
disable	16-11
local-mode	16-13
neighbor	16-15
pe-type	16-17
profile	16-19
pw-encap	16-21
pw-id	16-22
pw-name	16-24
standby-for	16-26
vpls	16-28
vpls profile	16-29
 Index	 1
 Commands	 1
 Modes	 1

About This Guide

This guide describes the tasks and commands used to configure the following SmartEdge® OS routing protocol features:

- Basic IP routing
- Dynamically verified static routing (DVSR)
- Virtual Router Redundancy Protocol (VRRP)
- Routing Information Protocol (RIP) and RIP next generation (RIPng)
- Open Shortest Path First (OSPF) and OSPF Version 3 (OSPFv3) protocols
- Bidirectional Forwarding Detection (BFD)
- Border Gateway Protocol (BGP)
- Border Gateway Protocol/Multiprotocol Label Switching Virtual Private Network (BGP/MPLS VPN)
- Intermediate System-to-Intermediate System (IS-IS) routing
- IP multicast routing, including Internet Group Management Protocol (IGMP), Multicast Source Discovery Protocol (MSDP), and Protocol Independent Multicast (PIM)
- Routing policies
- MPLS
- Layer 2 Virtual Private Network (L2VPN)
- Label Distribution Protocol (LDP)
- Virtual Private LAN Services (VPLS)

This preface contains the following sections:

- Related Publications
- Intended Audience
- Organization
- Conventions
- Ordering Documentation

Related Publications

In parallel with this guide, use the *Routing Protocols Operations Guide* for the SmartEdge OS, which describes the tasks and the commands used to monitor, administer, and troubleshoot routing protocol features.

Use this guide and the *Routing Protocols Operations Guide* for the SmartEdge OS in conjunction with the following publications:

- *Basic System Configuration Guide* for the SmartEdge OS

Describes the tasks and commands used to configure the following SmartEdge OS features: how to use the SmartEdge command-line interface (CLI), configuration file management, access to the system; basic system parameters; contexts, interfaces, and subscribers; system-wide management features, including bulk statistics, logging facilities, and the Simple Network Management Protocol (SNMP) and Remote Monitoring (RMON) functions.

- *Ports, Circuits, and Tunnels Configuration Guide* for the SmartEdge OS

Describes the tasks and commands to use the CLI and manage SmartEdge OS releases and configuration files; describes the tasks and commands used to configure the following SmartEdge OS features: traffic cards, their ports, channels, and subchannels, and Automatic Protection Switching (APS); circuits, including clientless IP service selection (CLIPS) circuits and link aggregation; bridging and cross-connections between circuits; Generic Routing Encapsulation (GRE) tunnels (including IP Version 6 [IPv6] over GRE tunnels), Layer 2 Tunneling Protocol (L2TP) tunnels, and overlay tunnels (IPv6 over IP Version 4 [IPv4]); static and dynamic bindings between ports, channels, subchannels, and circuits to interfaces, either directly or indirectly.

- *IP Services and Security Configuration Guide* for the SmartEdge OS

Describes the tasks and commands used to configure the following SmartEdge OS features: Address Resolution Protocol (ARP), Neighbor Discovery (ND) protocol for IPv6 routers, Dynamic Host Configuration Protocol (DHCP), Network Time Protocol (NTP), Domain Name System (DNS), HTTP redirect, access control lists (ACLs), forward policies, Network Address Translation (NAT) policies, service policies, quality of service (QoS) policies, authentication, authorization, and accounting (AAA), Remote Authentication Dial-In User Service (RADIUS), Terminal Access Controller Access Control System Plus (TACACS+), key chains, and lawful intercept (LI).

- *Basic System Operations Guide* for the SmartEdge OS

Describes the tasks and commands used to monitor, administer, and troubleshoot the SmartEdge OS features described in the *Basic System Configuration Guide*; commands include all **clear**, **debug**, **monitor**, **process**, and **show** commands that monitor and test system-wide functions and features, such as software processes.

- *Ports, Circuits, and Tunnels Operations Guide* for the SmartEdge OS

Describes the tasks and commands used to monitor, administer, and troubleshoot the SmartEdge OS features described in the *Ports, Circuits, and Tunnels Configuration Guide*; commands include all **clear**, **debug**, **monitor**, and **show** commands, along with other operations-based commands, such as device management and on-demand diagnostics.

- *IP Services and Security Operations Guide* for the SmartEdge OS

Describes the tasks and commands used to monitor, administer, and troubleshoot the SmartEdge OS features described in the *IP Services and Security Configuration Guide*; commands include all **clear**, **debug**, and **show** commands, along with other operations-based commands.

- *SmartEdge 800 Router Hardware Guide*

Describes the SmartEdge 800 hardware and provides site preparation information and installation, monitoring, and maintenance procedures for the chassis and cards.

- *SmartEdge 400 Router Hardware Guide*

Describes the SmartEdge 400 hardware and provides site preparation information and installation, monitoring, and maintenance procedures for the chassis and cards.

Intended Audience

This guide is intended for system and network administrators experienced in access and internetwork administration.

Organization

This guide is organized as follows:

- Part 1, “Introduction”

Describes network routing with the SmartEdge OS, supported routing protocols and routing related-features, the routing-related command hierarchy, and the routing-related access command modes and system prompts.

- Part 2, “IP Routing”

Describes the SmartEdge OS tasks and commands used to configure basic IP routing, including static IP routing; DVSR; VRRP; RIP and RIPng; OSPF and OSPFv3; BFD; BGP; BGP/MPLS VPNs; IS-IS; IP multicast routing, including IGMP, MSDP, and PIM; and routing policies.

- Part 3, “MPLS Routing”

Describes the SmartEdge OS tasks and commands used to configure MPLS, L2VPNs, LDP, and VPLS.

Note There are three indexes in this guide: an index of tasks and features, an index of commands, and an index of CLI modes with the commands found within each mode.

Conventions

This guide uses special conventions for the following elements:

- Command Modes and Privilege Levels
- Command Syntax
- Examples
- Task Tables
- Online Navigation Aids

Command Modes and Privilege Levels

Commands are entered in exec mode or in one of many configuration modes. By default, the majority of commands in exec mode have a privilege level of 3, while commands in any configuration mode have a privilege level of 10. Exceptions are noted in parentheses () in the “Command Mode” section in any command description; for example, “exec (15)”.

For a list of command modes and a figure displaying the command mode hierarchy, see the “Command Mode Hierarchy” section in Chapter 1, “Overview.”

For detailed information about command modes and privilege levels, see the “User Interface” section (in the “Overview” chapter) in the *Basic System Configuration Guide* for the SmartEdge OS.

Command Syntax

Table 1 lists the descriptions of the elements used in a command syntax statement.

Table 1 Command Syntax Terminology

Syntax Element	Description	Example Fragment
Argument	An item for which you must supply a value.	<i>slot</i>
Construct	A combination of: • A keyword and its argument. • Two or more keywords that cannot be specified independently. • Two or more arguments that cannot be specified independently.	• min-wait <i>seconds</i> • line fdl ansi • <i>src src-wildcard</i>
Keyword	An optional or required item that must be entered exactly as shown.	all

Table 2 describes separator characters used in command syntax statements.

Table 2 Separator Characters in Command Syntax

Character	Use	Example Fragment
@	Separates the prefix name from the suffix name.	<i>sub-name@ctx-name</i>
/	Separates slot from port, IP address from prefix length, and separates fields in URLs.	<i>slot[/port]</i> <i>{ip-addr /prefix-length}</i> <i>ldevice[/directory]/filename.ext</i>

Table 2 Separator Characters in Command Syntax (*continued*)

Character	Use	Example Fragment
:	Separates a port from a channel and a channel from a subchannel.	<i>port[:chan-num]</i> <i>ds3-chan-num[:ds1-chan-num]</i>
-	Separates starting value from ending value.	<i>start-end</i>
	Separates output modifiers from keywords and arguments in show commands. ¹	show configuration include port

1. For more information about the use of the pipe (|) character, see the “Using the CLI” chapter in the *Basic System Configuration Guide* for the SmartEdge OS.

The following guidelines apply to separator characters in Table 2:

- The separator character between the prefix and suffix names in a structured username is configurable; the @ character is the default and is used in command syntax throughout this guide.
- Separator characters act as one-character keywords; therefore, they are always shown in bold.

Table 3 lists the characters and formats used in command syntax statements.

Table 3 Text Formats and Characters in Command Syntax

Convention	Example
Commands and keywords are indicated in bold .	no ip unnumbered
Arguments for which you must supply the value are indicated in <i>italics</i> .	banner login <i>delimited-text</i>
Square brackets ([]) indicate optional arguments, keywords, and constructs within scripts or commands.	show clock [universal] enable [<i>level</i>]
Alternative arguments, keywords, and constructs within commands are separated by the pipe character ().	public-key { DSA RSA } [after-key <i>existing-key</i> position <i>key-position</i>] { <i>new-key</i> ftp url }
Alternative, but required arguments, keywords, and constructs are shown within grouped braces ({ }), and are separated by the pipe character ().	debug ssh { all ssh-general sshd-detail sshd-general } ip address <i>ip-addr</i> { <i>netmask</i> <i>lprefix-length</i> } [secondary]
Optional and required arguments, keywords, and constructs can be nested with grouped braces and square brackets, where the syntax requires such format.	enable authentication { none <i>method</i> [<i>method</i> [<i>method</i>]]}

Examples

Examples use the following conventions:

- System prompts are of the form [context]hostname(mode)#, [context]hostname#, or [context]hostname>.

In this case, *context* indicates the current context, *hostname* represents the configured name of the SmartEdge system, and *mode* indicates the string for the current configuration mode, if applicable.

Whether the prompt includes the # or the > symbol depends on the privilege level. For further information on privilege levels, see the “User Interface” section (in the “Overview” chapter) in the *Basic System Configuration Guide* for the SmartEdge OS.

For example, the prompt in the `local` context on the Redback system in `context` configuration mode is:

```
[local]Redback(config-ctx)#
```

- Information displayed by the system is in `Courier` font.
- Information that you enter is in **`Courier bold`** font.

Task Tables

Tasks to configure features are described in task tables under the “Configuration Tasks” section in each chapter. The command syntax displays only the root command, which is hyperlinked to the location where the complete command syntax is described in the “Command Descriptions” section of the chapter. Table 4 displays an example of a configuration task table.

Table 4 Task Table Example

Task	Root Command	Notes
Enable static MPLS routing within a context and enter MPLS static router configuration mode.	router mpls-static	Enter this command in context configuration mode.
Create a static LSP and enter MPLS static LSP configuration mode.	lsp	Enter this command in MPLS static router configuration mode.

Online Navigation Aids

To aid in accessing information in the online format for this guide, the following types of cross-references are hyperlinks:

- Cross-references to chapters, sections, tables, and figures in the text
- Lists of section headings within a chapter or appendix
- Commands listed in the “Related Commands” section at the end of each command description
- Entries in the table of contents
- Entries in indexes

Note Hyperlinks in PDF files appear the same as regular text; however, your cursor changes from an open hand icon to a pointing finger icon when you move your cursor over a hyperlink.

Ordering Documentation

Redback[®] documentation is available on CD-ROM, which ships with Redback products. The appropriate CD-ROMS are included with your products as follows:

- SMS[™] product
- SmartEdge router product
- NetOp[™] product (includes NetOp Element Manager System [EMS] and NetOp Policy Manager [PM])

To order additional copies of the appropriate CD-ROM or printed, bound books, perform the following steps:

1. Log on to the Redback Networks Support web site at <http://www.redback.com> and enter a username and password.

If you do not have a logon username and password, contact your Redback Networks support representative, or send an e-mail to supportlogin@redback.com with a copy of the **show hardware** command output, your contact name, company name, address, and telephone number.

2. On the Redback Networks Support web site, select one of the Redback Networks product line tabs at the bottom of the web page, click **Documentation** on the navigation bar, and then click **To Order Books** on the navigation bar.

To electronically provide feedback on our documentation, perform the following steps:

1. On the Documentation web page, click **Feedback** on the navigation bar.
2. Complete and submit the documentation feedback form.

We appreciate your comments.

Introduction

This part describes SmartEdge[®] OS network routing, supported routing protocols and routing related-features, the routing-related command hierarchy, and the routing-related access command modes and system prompts; it consists of Chapter 1, “Overview.”

Overview

This chapter describes the routing protocols and related services available in the SmartEdge® OS software in the following sections:

- SmartEdge Routing
- Command Mode Hierarchy

SmartEdge Routing

Network routing moves information across an internetwork from a source to a destination, typically passing through one or more intermediate nodes along the way. The primary difference between routing and bridging is that the two access different levels of information to determine how to transport packets from source to destination—routing occurs at layer 3 (the network layer), while bridging occurs at layer 2 (the link layer) of the Open Systems Interconnection (OSI) reference model.

In addition to transporting packets through an internetwork, routing involves determining optimal paths to a destination. Routing algorithms use metrics, or standards of measurement, to establish these optimal paths, initializing and maintaining routing tables that contain all route information.

The SmartEdge OS routing table stores routes to directly attached devices, static IP routes, and routes learned dynamically from the Routing Information Protocol (RIP), the Open Shortest Path First (OSPF) protocol, the Border Gateway Protocol (BGP), and the Intermediate System-to-Intermediate System (IS-IS) routing protocol. In the routing table, next-hop associations specify that a destination can be reached by sending packets to a next-hop router located on an optimal path to the destination. Routing algorithms must converge rapidly; that is, all routers must agree on optimal routes.

When a network event causes routes either to go down or become unavailable, routers distribute routing update messages that are propagated across networks, causing a universally agreed recalculation of optimal routes. Routing algorithms that converge slowly can cause routing loops or network outages. Many algorithms can quickly select next-best paths and adapt to changes in network topology.

Methods for implementing IP routing, and the protocols used, are described in the following sections:

- Static Versus Dynamic Routing
- IGP Versus EGP
- Supported IP Routing Protocols and Routing-Related Features
- Protocol Distances

Static Versus Dynamic Routing

Static routing involves packet forwarding on the basis of static routes configured by the system administrator. Static routes work well in environments where network traffic is relatively predictable and network topology is relatively simple.

In contrast, dynamic routing algorithms adjust to changing network circumstances by analyzing incoming routing update messages. RIP, OSPF, BGP, and IS-IS all use dynamic routing algorithms. A dynamic routing algorithm can also be supplemented with static routes where appropriate. For example, a router of last resort (to which all unroutable packets are sent) can store information on such packets for troubleshooting purposes.

Some routing algorithms operate in a flat, hierarchy-free space, while others use routing hierarchies. In a flat routing system such as RIP, all routers are peers of all other routers. As networks increase in size, flat routing systems encounter scaling limitations. To address this, some routing protocols allow the administrator to partition the network into hierarchical levels, which facilitates the summary of topology information for anyone located outside the immediate level or area. An example is the OSPF protocol, which supports a two-level hierarchy where area 0 is the backbone area that interconnects all other areas.

IGPs Versus EGPs

Another group of protocols that works to optimize network performance are the Interior Gateway Protocols (IGPs). These optimize the route between points within a network. Examples of commonly used IGPs are RIP, OSPF, and IS-IS.

Exterior Gateway Protocols (EGPs) support route information exchange between different networks. An example of a commonly used EGP is BGP-4. The choice of an optimal path is made based on the cost of the path measured by metrics associated with each link in the network.

IGPs and EGPs have slightly differing administrative designs. An IGP typically runs in an area under a single administrative control; this area is referred to as an autonomous system (AS) or a routing domain. In contrast, an EGP allows two different autonomous systems to exchange routing information and send data across the AS border. Policy decisions in EGPs can be shaped to decide which routing information crosses the border between the two autonomous systems.

Supported IP Routing Protocols and Routing-Related Features

Redback[®] currently supports the following IP routing protocols and routing-related features:

- Basic IP Routing
- Dynamically Verified Static Routing
- Virtual Router Redundancy Protocol
- Routing Information Protocol
- Open Shortest Path First
- Bidirectional Forwarding Detection
- Border Gateway Protocol
- Border Gateway Protocol/Multiprotocol Label Switching Virtual Private Network

- Intermediate System-to-Intermediate System Routing
- IP Multicast
- Routing Policy
- Multiprotocol Label Switching
- Layer 2 Virtual Private Network
- Label Distribution Protocol
- Virtual Private LAN Services

Basic IP Routing

Basic IP routing includes static IP routing and other basic routing features not covered by any routing protocol, including router IDs, static routes for multicast reverse path forwarding (RPF) lookup, IP Martian addresses, unicast RPF checks, maximum IP routes, and intercontext static routing among non-local contexts.

Dynamically Verified Static Routing

Dynamically verified static routing (DVSR) is a semidynamic and semistatic routing protocol used mainly for making edge routing decisions.

SmartEdge routers support DVSR as a unique edge routing feature in addition to static routing and regular IGP, such as IS-IS, OSPF, and RIP. DVSR is similar to normal static routing. The main difference is that the DVSR's next hop, or some other relevant host IP address, is dynamically verified by this protocol before the prefix can be injected into the local routing table. In many ISP networks, using static routing without proper next-hop checks results in blackholing of network traffic.

Virtual Router Redundancy Protocol

Virtual Router Redundancy Protocol (VRRP) eliminates the single point of failure that is common in the static default routed environment and provides a higher availability default path without requiring the configuration of dynamic routing or router discovery protocols on every end host.

VRRP works by dynamically assigning responsibility for a virtual router to one of the VRRP routers on a LAN. A virtual router is defined by its virtual router ID (VRID) and a set of IP addresses. There are two types of VRRP routers—owner and backup. The VRRP router controlling the IP addresses associated with a virtual router is called the owner, and it forwards packets sent to the IP addresses.

Routing Information Protocol

RIP is a distance-vector protocol that uses a hop count as its metric. Relatively old, RIP is still commonly used, especially in small homogeneous networks. Our implementation supports RIP Version 2 and provides for multiple RIP instances. Each instance maintains its own routing table and set of interfaces. Each interface can only be assigned to at most one RIP instance.

Open Shortest Path First

OSPF is an IGP that uses link-state advertisements (LSAs) to inform other routers of the state of the sender's links. In a link-state routing protocol, each router distributes information about its interfaces and neighbor relationships. The collection of the link states of individual routers forms a database that describes the AS topology. As OSPF routers accumulate link-state information, they use the Shortest Path First (SPF) algorithm to calculate the shortest path to each node, which forms the basis for developing routing information for that autonomous system.

Bidirectional Forwarding Detection

Bidirectional Forwarding Detection (BFD) is a simple Hello protocol that in many respects is similar to the detection components of some routing protocols. A pair of routers periodically transmit BFD packets over each path between the two routers, and if a system stops receiving BFD packets after a predefined time interval, some component in that particular bidirectional path to the neighboring router is assumed to have failed.

A path is only declared to be operational when two-way communication has been established between systems.

BFD provides low overhead, short-duration detection of failures in the path between adjacent forwarding engines, including the interfaces, data links, and to the extent possible, the forwarding engines themselves.

The legacy Hello mechanism run by routing protocols do not offer detections of less than one second, and for some applications, more than one second is too long and represents a great deal of lost data at gigabit rates. BFD provides the ability to detect communication failures in less than one second.

Border Gateway Protocol

Border Gateway Protocol (BGP) is an EGP based on distance-vector algorithms, and uses the Transmission Control Protocol (TCP) as its transport protocol. BGP is a protocol between exactly two BGP nodes, or BGP speakers. First, the TCP connection is established and then the two BGP speakers exchange dynamic routing information over the connection. The exchange of messages is a BGP session between BGP peers.

Border Gateway Protocol/Multiprotocol Label Switching Virtual Private Network

In its most general definition, a Virtual Private Network (VPN) is a network in which customer connectivity among multiple remote sites is deployed across a shared central infrastructure, yet still provides the same access or security as a private network.

More specifically, a Border Gateway Protocol/Multiprotocol Label Switching Virtual Private Network (BGP/MPLS VPN) is a collection of policies, and these policies control connectivity among a set of sites. A customer site is connected to the service provider network, often called a backbone, by one or more ports, where the service provider associates each port with a VPN context.

BGP/MPLS VPN allows you to implement a wide range of policies; for example, within a given VPN, you can allow every site to have a direct route to every other site (full mesh), or you can restrict certain pairs of sites from having direct routes to each other (partial mesh).

Intermediate System-to-Intermediate System Routing

IS-IS routing is an IGP that uses link-state information to make routing decisions.

IS-IS is defined in ISO 10589, *Intermediate System to Intermediate System Intra-Domain Routing Exchange Protocol for Use in Conjunction with the Protocol for Providing the Connectionlessmode Network Service (ISO 8473)*, ISO DP 10589, February 1990, and RFC 1195, *Use of OSI IS-IS for Routing in TCP/IP and Dual Environments*.

IP Multicast

IP multicast communication enables a source host to send IP packets to any number of hosts, anywhere within an IP network; it is one-to-any communication. That is, multicast communication is not limited to sending packets to a single destination host, or sending packets to every host on the network. Instead, multicast enables a source host to send IP packets to as many destination hosts as necessary, but no more than that. The advantages of multicast communication, unlike broadcast communication, which floods the network with unnecessary traffic, is that a source host can communicate with more than one destination host without sending traffic to every host on the network. This results in an economic use of bandwidth.

The main challenge for multicast communication is developing a method for determining which hosts will receive multicast traffic, and which hosts will not receive the traffic. Several different multicast protocols have been developed, each with its own unique approach to addressing the multicast challenge. The SmartEdge OS supports the following multicast protocols:

- Internet Group Management Protocol
- Protocol Independent Multicast Sparse Mode
- Multicast Source Discovery Protocol

Routing Policy

Routing policies allow network administrators to enforce various routing policy decisions onto incoming, outgoing, and redistributed routes. The tools used to configure routing policies include BGP AS path lists, BGP community lists, IP prefix lists, and route maps with match and set conditions.

Multiprotocol Label Switching

MPLS is a method for efficiently forwarding packets through a network. MPLS operates across an interface in an MPLS-enabled context.

In a conventional IP network, routers forward packets through the network, from one router to the next, with each router making an independent forwarding decision by analyzing the packet header. This conventional approach to forwarding packets has become insufficient to support current networking demands.

With MPLS, the complete analysis of the packet header is performed only once, when it enters an MPLS-enabled network. At each incoming (ingress) point of the network, packets are assigned a label by an edge LSR. Packets are forwarded along a LSP where each LSR makes forwarding decisions based on the label information. At each hop, the LSR swaps the existing label for a new label that tells the next hop how to forward the packet. At the outgoing (egress) point, an edge LSR removes the label, and forwards the packet to its destination. MPLS uses the Resource Reservation Protocol (RSVP), or the LDP, to communicate labels and their meaning among LSRs.

Layer 2 Virtual Private Network

Layer 2 Virtual Private Networks (L2VPNs) customer edge (CE) routers send L2 traffic to provider edge (PE) routers over L2 circuits configured between the PE and the CE routers. An L2 circuit can be either an Ethernet port, an 802.1Q virtual LAN (VLAN), a Frame Relay permanent virtual circuit (PVC), or an Asynchronous Transfer Mode (ATM) PVC.

An L2VPN is configured on PE routers and is used to cross-connect a local L2 circuit with a corresponding remote L2 circuit through an LSP tunnel that crosses the network backbone.

Label Distribution Protocol

LDP enables dynamic label allocation and distribution in an MPLS network. An LSR enabled with LDP can establish LSPs to other LSRs in the network. LDP creates label bindings by assigning labels to connected routers and by advertising the bindings to neighbors. LDP also assigns labels to label bindings learned from neighbors, and readvertises the binding to other neighbors. When an LSR advertises a label binding for a route, the LSR is advertising the availability of an LSP to the destination of that route. LDP can learn several LSPs from different neighbors for the same route. In this case, LDP activates only the path selected by the underlying IGP. For this reason, LDP must work together with an IGP, such as the IS-IS or OSPF protocol.

Virtual Private LAN Services

VPLS enables networks at separate geographical locations to communicate with each other across a wide area network (WAN) as if they were directly attached to each other in a LAN. The WAN becomes transparent, which is achieved by creating VPLS pseudo-wires.

A pseudo-wire is a mechanism that emulates the attributes and function of Ethernet connectivity over a WAN. Any required switching functionality or service translation is outside the scope of the pseudo-wire and of the transport network. Pseudo-wires are carried over MPLS tunnels on the network.

MPLS signaling protocols are used to automatically provision a service on a pseudo-wire end-to-end, so you can provision a pseudo-wire by pointing to its two endpoints, and MPLS automatically negotiates the path.

Protocol Distances

When determining a single optimal route among multiple routes within a single routing protocol, the SmartEdge OS selects the route that has the shortest distance. When deciding a best path among routes originating from multiple protocols, the system uses a more complex methodology. The SmartEdge routing table stores direct, static, external BGP (eBGP), OSPF, IS-IS, RIP, and internal BGP (iBGP) routes.

Table 1-1 lists the protocols and their default values for routes learned through various protocols.

Table 1-1 Protocol Distance Defaults

Protocol	Distance Value
Directly connected	0
Static IP	1
eBGP	20
OSPF	110
IS-IS	115
RIP	120
iBGP	200

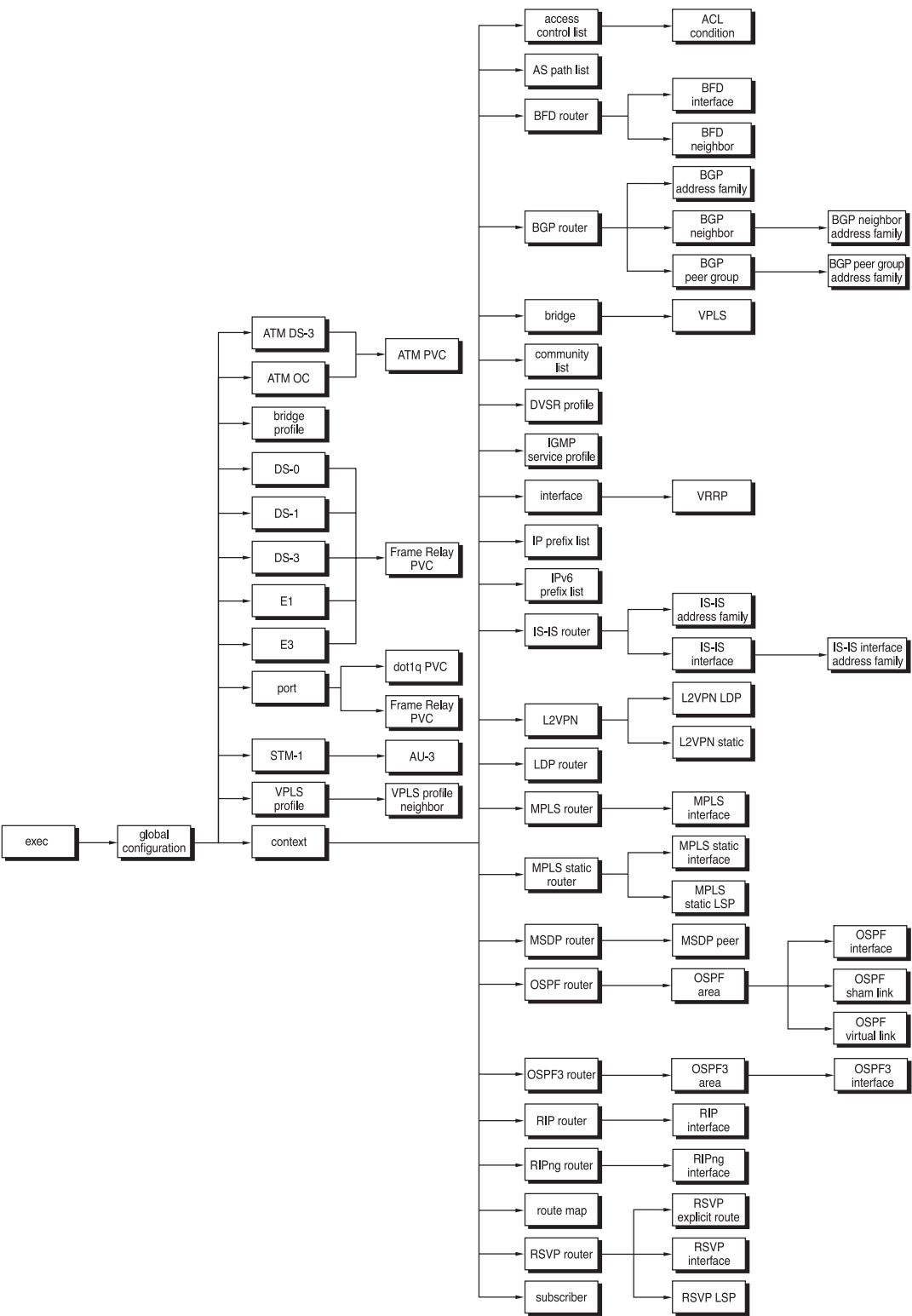
Command Mode Hierarchy

Command modes exist in a hierarchy; that is, you must access the higher-level command mode before you can access a lower-level command mode in the same chain.

Note For modes relevant to basic system features, see the “Overview” chapter in the *Basic System Configuration Guide* for the SmartEdge OS. For modes relevant to port, circuit, and tunnel features, see the “Overview” chapter in the *Ports, Circuits, and Tunnels Configuration Guide* for the SmartEdge OS. For modes relevant to IP services and security features, see the “Overview” chapter in the *IP Services and Security Configuration Guide* for the SmartEdge OS.

Figure 1-1 shows the hierarchy of the command modes used to configure routing features.

Figure 1-1 Command Mode Hierarchy



0655

Table 1-2 lists the command modes (in alphabetical order) relevant to routing features. It includes the commands that enable access to each mode and the command-line prompt for each mode.

Table 1-2 Command Modes and Prompts

Mode Name	Commands Used to Access	Command-Line Prompt
exec	(user logon)	# or >
access control list	ip access-list and policy access-list commands from context configuration mode	(config-access-list)#
ACL condition	condition time-range command from access control list configuration mode	(config-acl-condition)#
AS path list	as-path-list command from context configuration mode	(config-as-path-list)#
ATM DS-3	port atm command from global configuration mode	(config-atm-ds3)#
ATM OC	port atm command from global configuration mode	(config-atm-oc)#
ATM PVC	atm pvc command from ATM DS-3 and ATM OC configuration modes	(config-atm-pvc)#
AU-3	au3 command from STM-1 configuration mode	(config-au3)#
BFD interface	interface command from BFD router configuration mode	(config-bfd-if)#
BFD neighbor	neighbor command from BFD router configuration mode	(config-bfd-nbr)#
BFD router	router bfd command from context configuration mode	(config-bfd)#
BGP address family	address-family command from BGP router configuration mode	(config-bgp-af)#
BGP neighbor	neighbor command from BGP router configuration mode	(config-bgp-neighbor)#
BGP neighbor address family	address-family command from BGP neighbor configuration mode	(config-bgp-af)#
BGP peer group	peer-group command from BGP router configuration mode	(config-bgp-peer-group)#
BGP peer group address family	address-family command from BGP peer group configuration mode	(config-bgp-peer-af)#
BGP router	router bgp command from context configuration mode	(config-bgp)#
bridge	bridge command from context configuration mode	(config-bridge)#
bridge profile	bridge profile command from global configuration mode	(config-bridge-profile)#
community list	community-list command from context configuration mode	(config-community-list)#
context	context command from global configuration mode	(config-ctx)#
dot1q PVC	dot1q pvc command from port configuration mode	(config-dot1q-pvc)#
DS-0	port ds0s command from global configuration mode	(config-ds0s)#
DS-1	port ds1 command from global configuration mode	(config-ds1)#
DS-3	port ds3 and port channelized-d3 commands from global configuration modes	(config-ds3)#
DVSR profile	dvsvr-profile command from context configuration mode	(config-dvsvr)#
E1	port e1 command from global configuration mode	(config-e1)#
E3	port e3 command from global configuration mode	(config-e3)#

Table 1-2 Command Modes and Prompts (*continued*)

Mode Name	Commands Used to Access	Command-Line Prompt
Frame Relay PVC	frame-relay pvc command from DS-0, DS-1, DS-3, E1, E3, and port configuration modes	(config-fr-pvc)#
global	configure command from exec mode	(config)#
IGMP service profile	igmp service-profile command from context configuration mode	(config-igmp-service-profile)#
interface	interface command from context configuration mode	(config-if)#
IP prefix list	ip prefix-list command from context configuration mode	(config-prefix-list)#
IPv6 prefix list	ipv6 prefix-list command from context configuration mode	(config-ipv6-prefix-list)#
IS-IS address family	address-family command from IS-IS router configuration mode	(config-isis-af)#
IS-IS interface	interface command from IS-IS router configuration mode	(config-isis-if)#
IS-IS interface address family	address-family command from IS-IS interface configuration mode	(config-isis-if-af)#
IS-IS router	router isis command from context configuration mode	(config-isis)#
L2VPN	l2vpn command from context configuration mode	(config-l2vpn)#
L2VPN LDP	l2vpn ldp command from L2VPN configuration mode	(config-l2vpn-ldp)#
L2VPN static	l2vpn static command from L2VPN configuration mode	(config-l2vpn-static)#
LDP router	router ldp command from context configuration mode	(config-ldp)#
MPLS interface	interface command from MPLS router configuration mode	(config-mpls-if)#
MPLS router	router mpls command from context configuration mode	(config-mpls)#
MPLS static interface	interface command from MPLS static router configuration mode	(config-mpls-static-if)#
MPLS static LSP	lsp command from MPLS static router configuration mode	(config-mpls-static-lsp)#
MPLS static router	router mpls-static command from context configuration mode	(config-mpls-static)#
MSDP peer	peer command from MSDP router configuration mode	(config-msdp-peer)#
MSDP router	router msdp command from context configuration mode	(config-msdp)#
OSPF area	area command from OSPF router configuration mode	(config-ospf-area)#
OSPF interface	interface command from OSPF area configuration mode	(config-ospf-if)#
OSPF router	router ospf command from context configuration mode	(config-ospf)#
OSPF sham link	sham-link command from OSPF area configuration mode ¹	(config-ospf-sham-link)#
OSPF virtual link	virtual-link command from OSPF area configuration mode ¹	(config-ospf-virt-link)#
OSPF3 area	area command from OSPF3 router configuration mode	(config-ospf3-area)#
OSPF3 interface	interface command from OSPF3 area configuration mode	(config-ospf3-if)#
OSPF3 router	router ospf3 command from context configuration mode	(config-ospf3)#
port	port ethernet , port channelized oc-12 , and port pos commands from global configuration mode	(config-port)#
RIP interface	interface command from RIP router configuration mode	(config-rip-if)#
RIP router	router rip command from context configuration mode	(config-rip)#
RIPng interface	interface command from RIPng router configuration mode	(config-ripng-if)#

Table 1-2 Command Modes and Prompts (*continued*)

Mode Name	Commands Used to Access	Command-Line Prompt
RIPng router	router ripng command from context configuration mode	(config-ripng)#
route map	route-map command from context configuration mode	(config-route-map)#
RSVP explicit route	explicit-route command from RSVP router configuration mode	(config-rsvp-explicit-route)#
RSVP interface	interface command from RSVP router configuration mode	(config-rsvp-if)#
RSVP LSP	lsp command from RSVP router configuration mode	(config-rsvp-lsp)#
RSVP router	router rsvp command from context configuration mode	(config-rsvp)#
STM-1	port channelized-stm1 command from global configuration mode	(config-stm1)#
subscriber	subscriber command from context configuration mode	(config-sub)#
VPLS	vpls command from bridge configuration mode	(config-vpls)#
VPLS profile	vpls profile command from global configuration mode	(config-vpls-profile)#
VPLS profile neighbor	neighbor command from VPLS profile configuration mode	(config-vpls-profile-neighbor)#
VRRP	vrrp command from interface configuration mode	(config-vrrp)#

1. The **sham-link** and **virtual-link** commands are available in OSPF area configuration mode for VPN-enabled contexts only.

IP Routing

This part describes the tasks and commands used to configure the SmartEdge® OS IP routing features, including static IP routing; dynamically verified static routing (DVSR); Virtual Redundancy Router Protocol (VRRP); Routing Information Protocol (RIP) and RIP next generation (RIPng); Open Shortest Path First (OSPF) and OSPF Version 3 (OSPFv3); Bidirectional Forwarding Detection (BFD); Border Gateway Protocol (BGP); BGP/Multiprotocol Label Switching Virtual Private Networks (BGP/MPLS VPNs); Intermediate System-to-Intermediate System (IS-IS); IP multicast routing, including Internet Group Management Protocol (IGMP), Multicast Source Discovery Protocol (MSDP), and Protocol Independent Multicast (PIM); and routing policies.

This part consists of the following chapters:

- Chapter 2, “Basic IP Routing Configuration”
- Chapter 3, “DVSR Configuration”
- Chapter 4, “VRRP Configuration”
- Chapter 5, “RIP Configuration”
- Chapter 6, “OSPF Configuration”
- Chapter 7, “BFD Configuration”
- Chapter 8, “BGP Configuration”
- Chapter 9, “BGP/MPLS VPN Configuration”
- Chapter 10, “IS-IS Configuration”
- Chapter 11, “IP Multicast Configuration”
- Chapter 12, “Routing Policy Configuration”

Basic IP Routing Configuration

This chapter provides an overview of IP routing and describes the tasks and commands used to configure basic IP routing features through the SmartEdge® OS.

For information about the tasks and commands used to monitor, troubleshoot, and administer basic IP routing, see the “Basic IP Routing Operations” chapter in the *Routing Protocols Operations Guide* for the SmartEdge OS.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples
- Command Descriptions

Overview

IP routing moves information across an internetwork from a source to a destination, typically passing through one or more intermediate nodes along the way. The primary difference between routing and bridging is that the two access different levels of information to determine how to transport packets from source to destination—routing occurs at layer 3 (the network layer), while bridging occurs at layer 2 (the link layer) of the Open Systems Interconnection (OSI) reference model.

In addition to transporting packets through an internetwork, routing involves determining optimal paths to a destination. Routing algorithms use metrics, or standards of measurement, to establish these optimal paths, initializing and maintaining routing tables that contain all route information.

The SmartEdge OS routing table stores routes to directly attached devices, static IP routes, and routes learned dynamically from the Routing Information Protocol (RIP), the Open Shortest Path First (OSPF) protocol, the Border Gateway Protocol (BGP), and the Intermediate System-to-Intermediate System (IS-IS) routing protocol. In the routing table, next-hop associations specify that a destination can be reached by sending packets to a next-hop router located on an optimal path to the destination. Routing algorithms must converge rapidly; that is, all routers must agree on optimal routes.

When a network event causes routes either to go down or become unavailable, routers distribute routing update messages that are propagated across networks, causing a universally agreed recalculation of optimal routes. Routing algorithms that converge slowly can cause routing loops or network outages. Many algorithms can quickly select next-best paths and adapt to changes in network topology.

Methods for implementing IP routing, and the protocols used, are described in the following sections:

- Static Versus Dynamic Routing
- IGPs Versus EGPs
- IP Routing Protocols
- Protocol Distances

Static Versus Dynamic Routing

Static routing involves packet forwarding on the basis of static routes configured by the system administrator. Static routes work well in environments where network traffic is relatively predictable and network topology is relatively simple.

In contrast, dynamic routing algorithms adjust to changing network circumstances by analyzing incoming routing update messages. RIP, OSPF, BGP, and IS-IS all use dynamic routing algorithms. A dynamic routing algorithm can also be supplemented with static routes where appropriate. For example, a router of last resort (to which all unroutable packets are sent) can store information on such packets for troubleshooting purposes.

Some routing algorithms operate in a flat, hierarchy-free space, while others use routing hierarchies. In a flat routing system such as RIP, all routers are peers of all other routers. As networks increase in size, flat routing systems encounter scaling limitations. To address this, some routing protocols allow the administrator to partition the network into hierarchical levels, which facilitates the summary of topology information for anyone located outside the immediate level or area. An example is the OSPF protocol, which supports a two-level hierarchy where area 0 is the backbone area that interconnects all other areas.

IGPs Versus EGPs

Another group of protocols that works to optimize network performance are the Interior Gateway Protocols (IGPs). These optimize the route between points within a network. Examples of commonly used IGPs are RIP, OSPF, and IS-IS.

Exterior Gateway Protocols (EGPs) support route information exchange between different networks. An example of a commonly used EGP is BGP-4. The choice of an optimal path is made based on the cost of the path measured by metrics associated with each link in the network.

IGPs and EGPs have slightly differing administrative designs. An IGP typically runs in an area under a single administrative control; this area is referred to as an autonomous system (AS) or a routing domain. In contrast, an EGP allows two different autonomous systems to exchange routing information and send data across the AS border. Policy decisions in EGPs can be shaped to decide which routing information crosses the border between the two autonomous systems.

IP Routing Protocols

Redback currently supports the following IP routing protocols:

- The Virtual Router Redundancy Protocol (VRRP) eliminates the single point of failure that is common in a static default routed environment. A VRRP router controls IP addresses associated with a virtual router. Any of the virtual router's IP addresses on a LAN can then be used as the default first hop router by end hosts, providing a dynamic failover in forwarding responsibility should the VRRP router become unavailable. The main advantage of using VRRP is having a higher availability default path without requiring configuration of dynamic routing or router discovery protocols on every end host; see Chapter 4, "VRRP Configuration."
- RIP is a distance-vector IGP that uses hop count as its metric. Each router sends all or some of the portion of its routing table, but only to its neighbors. The RIP is widely used for routing traffic in the global Internet; see Chapter 5, "RIP Configuration."
- OSPF is a link-state IGP that uses link-state advertisements (LSAs) to inform other routers of the state of the sender's links. Each router sends only the portion of the routing table that describes the state of its own links to all nodes in the internetwork. LSAs are used to build a complete picture of the network topology, enabling other routers to determine optimal routes to destinations.

In OSPF, the autonomous system can be hierarchically organized by partitioning it into areas. Each area contains a group of contiguous networks and hosts. An area border router (ABR) communicates routing information between the areas; see Chapter 6, "OSPF Configuration."

- BGP-4 is a distance-vector EGP, and uses the Transmission Control Protocol (TCP) as its transport protocol. With BGP, a TCP connection is established over which two BGP peers exchange routing information. Routers that belong to the same autonomous system run internal BGP (iBGP), while routers that belong to different autonomous systems run external BGP (eBGP); see Chapter 8, "BGP Configuration."
- IS-IS is an OSI link-state hierarchical routing protocol that floods the network with link-state information. This builds a complete and consistent picture of network topology. Hierarchical routing simplifies backbone design, and the backbone routing protocol can also change without impacting the intra-area routing protocol. See Chapter 10, "IS-IS Configuration."

Protocol Distances

When determining a single optimal route among multiple routes within a single routing protocol, the SmartEdge OS selects the route that has the shortest distance. When deciding a best path among routes originating from multiple protocols, the system uses a more complex methodology. The SmartEdge routing table stores direct, static, eBGP, OSPF, IS-IS, RIP, and iBGP routes.

Table 2-1 lists the protocols and their default values for routes learned through various protocols.

Table 2-1 Protocol Distance Defaults

Protocol	Distance Value
Directly connected	0
Static IP	1
eBGP	20
OSPF	110

Table 2-1 Protocol Distance Defaults (*continued*)

Protocol	Distance Value
IS-IS	115
RIP	120
iBGP	200

Configuration Tasks

Note In this section, the command syntax in the task tables displays only the root command; for the complete command syntax, see the full description for the command in the “Command Descriptions” section.

To configure basic IP routing, perform the tasks described in the following sections:

- Configuring Static Routes
- Configuring Additional Basic IP Routing Parameters

Configuring Static Routes

Rather than dynamically selecting the best route to a destination, you can configure one or more static routes to the destination. Once configured, a static route stays in the routing table indefinitely. When multiple static routes are configured for a single destination and the outbound interface of the current static route goes down, a backup route is activated, improving network reliability.

You can configure up to eight static routes for a single destination. Each route is assigned a default distance value and cost value. Modifying these values allows you to set a preference for one route over the next. A static route can be overridden by a dynamically learned route with a lower administrative distance.

Among multiple routes with the same destination, preferred routes are selected in the following order:

1. The route with the shortest distance value is preferred first.
2. If two or more routes have the same distance and cost values, the equal cost multipath (ECMP) is preferred.
3. When redistributing static routes, routing protocols ignore the cost value assigned to those static routes. If static routes are redistributed through dynamic routing protocols, only the active static route to a destination is advertised.

To configure a static route, perform either of the tasks described in Table 2-2. Enter all commands in context configuration mode.

Table 2-2 Configure Static IP Routing

Task	Root Command	Notes
Configure one or more IP static routes to the same destination.	ip route	
Configure one or more IPv6 static routes to the same destination.	ipv6 route	

Configuring Additional Basic IP Routing Parameters

To configure basic IP routing parameters, perform the tasks described in Table 2-3. Enter all commands in context configuration mode, unless otherwise noted.

Table 2-3 Configure Additional Basic IP Routing Parameters

Task	Root Command	Notes
Add custom IP martian addresses in the routing table to configure an upper limit for the number of routes installed in an IP routing table.	ip martian	
Configure an upper limit for the number of routes installed in an IP routing table.	ip maximum-routes	
Configure a static route for multicast RPF lookup.	ip mstatic	Enter this command in interface configuration mode.
Perform a reverse path forwarding (RPF) check to verify the source IP address on all incoming unicast packets at the specified interface.	ip verify unicast source	
Configure a global router ID for the SmartEdge router.	router-id	The global router ID must be configured for RSVP to operate correctly.
Enable intercontext static routing among non-local contexts.	service inter-context routing	Enter this command in global configuration mode. This command can only be disabled when there is no instance of non-local context static routing configured on the router.
Enable the negotiation of the maximum transmission unit (MTU) for Transmission Control Protocol (TCP) sessions.	tcp path-mtu-discovery	Enter this command in global configuration mode. Enabling MTU negotiation has no effect on existing TCP sessions. Both the SmartEdge router and the remote router must be configured for MTU negotiation to work properly.

Configuration Examples

The following example routes packets for network 10.10.0.0/16 via interface, enet1:

```
[local]Redback(config-ctx)#ip route 10.10.0.0/16 enet1
```

The following example defines a default route through interface atm5. Because no cost is defined, this route uses a cost of 0, and is therefore used as the active route. If this route goes away, the second and third routes alternate because they have the same distance and cost.

```
[local]Redback(config-ctx)#ip route 0.0.0.0/0 atm5
[local]Redback(config-ctx)#ip route 0.0.0.0/0 10.1.1.1 cost 2
[local]Redback(config-ctx)#ip route 0.0.0.0/0 172.21.200.254 cost 2
```

Command Descriptions

The following example displays the routing table for the routes configured in the previous examples:

Note Only the default route for interface atm5 displays.

```
[local]Redback>show ip route
```

```
Codes: C - connected, S - static, R - RIP, e B - EBGp, i B - IBGP
0 - OSPF, IA - OSPF inter area, N1 - OSPF NSSA external type 1
N2 - OSPF NSSA external type 2, E1 - OSPF external type 1
E2 - OSPF external type 2
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2
> - Active Route
```

Type	Network	Next Hop	Dist	Metric	UpTime	Interface
> S	0.0.0.0/0		1	0	3w0d	atm5
> S	10.10.0.0/16		1	0	3w0d	enet

The following example shows the routing table after the default route through interface atm5 is removed:

```
[local]Redback>show ip route
```

```
Codes: C - connected, S - static, R - RIP, e B - EBGp, i B - IBGP
0 - OSPF, IA - OSPF inter area, N1 - OSPF NSSA external type 1
N2 - OSPF NSSA external type 2, E1 - OSPF external type 1
E2 - OSPF external type 2
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2
> - Active Route
```

Type	Network	Next Hop	Dist	Metric	UpTime	Interface
> S	0.0.0.0/0	10.1.1.1	1	2	3w0d	
> S		172.21.200.254				
> S	0.10.0.0/16		1	0	3w0d	enet

Command Descriptions

This section describes the syntax and usage guidelines for the commands used to configure basic IP routing features. The commands are presented in alphabetical order.

ip martian

ip maximum-routes

ip mstatic

ip route

ipv6 route

ip verify unicast source

router-id

service inter-context routing

tcp path-mtu-discovery

ip martian

ip martian *ip-addr/prefix-length* [**eq** *eq-value*] [**ge** *ge-value*] [**le** *le-value*]

no ip martian *ip-addr/prefix-length* [**eq** *eq-value*] [**ge** *ge-value*] [**le** *le-value*]

Purpose

Adds custom IP martian addresses to the list of default martian IP addresses in the routing table.

Command Mode

context configuration

Syntax Description

<i>ip-addr/prefix-length</i>	IP address (in the form <i>A.B.C.D</i>) and prefix length, separated by the slash (/) character. The range of values for the <i>prefix-length</i> argument is 0 to 32.
eq <i>eq-value</i>	Optional. Equal to value. The <i>eq-value</i> argument specifies the length of the mask to be matched; the eq keyword indicates that the mask length must exactly match the specified value. The range of values for the <i>eq-value</i> argument is 1 to 32.
ge <i>ge-value</i>	Optional. Greater than or equal to value. The <i>ge-value</i> argument specifies the length of the mask to be matched; the ge keyword indicates that all masks of a length greater than or equal to the specified value will match. The range of values for the <i>ge-value</i> argument is 1 to 32.
le <i>le-value</i>	Optional. Less than or equal to value. The <i>le-value</i> argument specifies the length of the mask to be matched; the le keyword indicates that all masks of a length less than or equal to the specified value will match. The range of values for the <i>le-value</i> argument is 1 to 32.

Default

For IPv4, the martian addresses of 0.0.0.0/8 and 127.0.0.0/8 are installed in the routing table.

Usage Guidelines

Use the **ip martian** command to add custom IP martian addresses to the list of default martian IP addresses in the routing table.

IP martian addresses are host or network addresses about which all routing information is ignored. IP martian addresses are typically advertised by misconfigured routers using dynamic protocols.

Use the **no** form of this command to remove a configured IP martian address from the routing table.

Examples

The following example configures a martian address of 10.1.0.0/20 for the local context. Routes matching this prefix are ignored.

```
[local]Redback(config-ctx)#ip martian 10.1.0.0/20
```

Related Commands

ip route

ip maximum-routes

ip maximum-routes [**multicast**] [**vpn**] *route-limit* [**log-only** | **threshold** *value*]

Purpose

Configures an upper limit for the number of routes installed in an IP routing table.

Command Mode

context configuration

Syntax Description

multicast	Optional. Sets the maximum route limit for unicast routes in a multicast topology.
vpn	Optional. Sets the maximum route limit for all non-local context unicast routing tables. When the vpn keyword is used in the local context, it specifies a default maximum route setting that automatically applies to all non-local contexts; however, if the ip maximum-route command is used in a specific non-local context, then it overrides the default maximum route setting.
<i>route-limit</i>	Maximum number of routes allowed in the IP routing table. If this limit is reached, a warning is triggered and any additional routes are rejected. Range of values is 1 to 4,294,967,295.
log-only	Optional. Configures the route limit as an advisory limit. An advisory limit triggers only a warning, and additional routes are not rejected.
threshold <i>value</i>	Optional. Threshold value for the mandatory limit that triggers a warning. Range of values is 1 to 100.

Default

No maximum limit is set.

Usage Guidelines

Use the **ip maximum-routes** command to configure an upper limit for the number of routes installed in an IP routing table.

A route limit sets an upper limit for the number of prefixes installed in a routing table; for example, you can use a route limit to limit the number of routes received from the customer edge (CE) router in a Virtual Private Network (VPN) context.

There are two modes for route limits: advisory and mandatory. An advisory limit only triggers warnings, and a mandatory limit rejects any additional routes after the threshold is reached.

Use the **vpn** keyword in the local context, to specify a default maximum route setting that automatically applies to all non-local contexts. To override the default maximum route setting, use the **ip maximum-route** command in the non-local context that you want to configure.

Examples

The following example configures an upper limit of 500 routes for the IP routing table:

```
[local]Redback#ip maximum-routes 500
```

Related Commands

None

ip mstatic

ip mstatic *src-addr netmask*

no ip mstatic *src-addr netmask*

Purpose

Configures a static route for multicast reverse path forwarding (RPF) lookup.

Command Mode

context configuration

Syntax Description

<i>src-addr</i>	IP address of the multicast source.
<i>netmask</i>	Network mask for the static route in the form <i>A.B.C.D</i> .

Default

None

Usage Guidelines

Use the **ip mstatic** command to configure a static route for multicast RPF lookup.

Use the **no** form of this command to delete a static route.

Examples

The following example configures a static route for multicast RPF lookup:

```
[local]Redback(config)#context isp1  
[local]Redback(config-ctx)#ip mstatic 192.168.100.100 255.255.0.0
```

Related Commands

None

ip route

```
ip route ip-addr/prefix-length {next-hop-ip-addr | next-hop-if-name | null0 | context ctx-name}  
[dvsr dvsr-profile-name [verify-address verify-addr]] [cost cost] [description text]  
[distance distance] [permanent] [tag tag]
```

```
no ip route ip-addr/prefix-length {next-hop-ip-addr | next-hop-if-name | null0 | context ctx-name}  
[dvsr dvsr-profile-name [verify-address verify-addr]] [cost cost] [description text]  
[distance distance] [permanent] [tag tag]
```

Purpose

Configures one or more static routes when the system is not configured to dynamically select a route to the destination.

Command Mode

context configuration

Syntax Description

<i>ip-addr/prefix-length</i>	IP address (in the form <i>A.B.C.D</i>) and prefix length, separated by the slash (/) character. The range of values for the <i>prefix-length</i> argument is 0 to 32.
<i>next-hop-ip-addr</i>	IP address of the next hop that can be used to reach the network.
<i>next-hop-if-name</i>	Interface name of the next hop that can be used to reach the network.
null0	Optional. Creates a null interface to prevent routing loops.
context <i>ctx-name</i>	Another context, which can be used as a next hop to reach a network.
dvsr <i>dvsr-profile-name</i>	Optional. dynamically verified static routing (DVSr) profile name. Defines a DVSr using the specified profile name. The dvsr <i>dvsr-profile-name</i> construct cannot be used with the <i>next-hop-ip-addr</i> or <i>next-hop-if-name</i> arguments, or the null0 or permanent keywords.
verify-address <i>verify-addr</i>	Optional. Host IP address the DVSr route should verify. If the verify-address <i>verify-addr</i> construct is not configured, the <i>next-hop-ip-addr</i> or <i>next-hop-if-name</i> argument will be used for the verification.
cost <i>cost</i>	Optional. Cost of the route. The range of values is 0 to 15.
description <i>text</i>	Optional. Description for the static route.
distance <i>distance</i>	Optional. Administrative distance assigned to the route. The range of values is 1 to 255.
permanent	Optional. Indicates that the route cannot be removed, even if the interface is shut down.
tag <i>tag</i>	Optional. Route tag used as a match value for controlling redistribution through route maps. An unsigned 32-bit integer, the range of values is 1 to 4,294,967,295; the default value is 0.

Default

None

Usage Guidelines

Use the **ip route** command to configure one or more static routes when the system is not configured to dynamically select a route to the destination.

A static route can be overridden by a dynamically learned route with a lower administrative distance.

Use the **null0** keyword to prevent routing loops. A null interface is always up and can never forward or receive traffic. The null interface provides an alternative method of filtering traffic. You can avoid the overhead involved with using access control lists by directing undesired network traffic to the null interface.

Note The Open Shortest Path First (OSPF) and Intermediate System-to-Intermediate System (IS-IS) routing processes always create a route to a null interface when summarizing a group of routes.

Use the **context** *ctx-name* construct to forward traffic to another routing context (next-hop context). The **context** *ctx-name* construct can be used to configure VPN customer Internet access, or Inter-VPN routing leaks. The next-hop context must be a different routing context than the one to which the static route belongs. If the next-hop context does not exist, and the **service multiple-contexts** command is enabled on the router, the context will be created. Intercontext static routing between two non-local contexts is not allowed unless the **service inter-context routing** command is enabled on the router. The prefix using the next-hop context is considered to be valid only if the next-hop context has the routes that are being covered by this prefix. In other words, this prefix will be installed in the RIB only if the next-hop context can reach those networks.

Use the **dvsr** *dvsr-profile-name* construct to configure a static route with DVSR capability. A DVSR route needs to reference an existing DVSR profile by name. Protocol redistribution can specify redistribute static dvsr to only import DVSR capable routes. The verify-host address of the DVSR route is by default the next-hop IP address of the route. If the DVSR verify-host is not the same as the next-hop IP address, the user need to make sure that there is a route to reach that verify-host address, and also the nexthop of that route needs to be the same as the next-hop of the DVSR route itself.

Use the **no** form of this command to remove static routes.

Examples

The following example routes packets for network 20.0.0.0/8 to the device at IP address 121.109.3.4 if dynamic information with administrative distance less than 110 is not available:

```
[local]Redback(config-ctx)#ip route 20.0.0.0/8 121.109.3.4 distance 110
```

The following example configures a null interface for network 172.0.0.0/8:

```
[local]Redback(config-ctx)#ip route 172.0.0.0/8 null0
```

The following example routes packets for network 129.108.0.0/16 to the device at IP address 129.108.6.6:

```
[local]Redback(config-ctx)#ip route 129.108.0.0/16 129.108.6.6
```

Command Descriptions

The following example configures a static route from the `local` context using context, `vpn-abc`, as the next hop context:

```
[local]Redback(config-ctx)#ip route 12.1.1.0/24 context vpn-abc
```

Related Commands

ipv6 route
service inter-context routing

ipv6 route

ipv6 route *ipv6-addr/prefix-length* {*next-hop-ipv6-addr* | *next-hop-if-name* | **null0**} [**cost** *cost*]
[**distance** *distance*] [**permanent**] [**tag** *tag*]

no ipv6 route *ipv6-addr/prefix-length* {*next-hop-ipv6-addr* | *next-hop-if-name* | **null0**} [**cost** *cost*]
[**distance** *distance*] [**permanent**] [**tag** *tag*]

Purpose

Configures one or more static routes when the system is not configured to dynamically select a route to the destination.

Command Mode

context configuration

Syntax Description

<i>ipv6-addr/prefix-length</i>	IPv6 address (in the form <i>A:B:C:D:E:F:G:H</i>) and prefix length, separated by the slash (/) character. The range of values for the <i>prefix-length</i> argument is 0 to 128.
<i>next-hop-ipv6-addr</i>	IPv6 address of the next hop that can be used to reach the network.
<i>next-hop-if-name</i>	Interface name of the next hop that can be used to reach the network.
null0	Optional. Creates a null interface to prevent routing loops.
cost <i>cost</i>	Optional. Cost of the route. The range of values is 0 to 15.
distance <i>distance</i>	Optional. Administrative distance assigned to the route. The range of values is 1 to 255.
permanent	Optional. Indicates that the route cannot be removed, even if the interface is shut down.
tag <i>tag</i>	Optional. Route tag used as a match value for controlling redistribution through route maps. An unsigned 32-bit integer, the range of values is 1 to 4,294,967,295; the default value is 0.

Default

None

Usage Guidelines

Use the **ipv6 route** command to configure one or more static routes when the system is not configured to dynamically select a route to the destination.

A static route can be overridden by a dynamically learned route with a lower administrative distance.

Use the **null0** keyword to prevent routing loops. A null interface is always up and can never forward or receive traffic. The null interface provides an alternative method of filtering traffic. You can avoid the overhead involved with using access control lists by directing undesired network traffic to the null interface.

Note The Open Shortest Path First Version 3 (OSPFv3) and Intermediate System-to-Intermediate System (IS-IS) routing processes always create a route to a null interface when summarizing a group of routes.

Use the **no** form of this command to remove static routes.

Examples

The following example routes packets for network, 2000:8A2E:5648:CDF7:65B3:2F29:B3D5:3995/64, to the device at IPV6 address, AB34:665F:B90B:3290:EA11:2678:FFFF:3210:

```
[local]Redback(config-ctx)#ipv6 route 2000:8A2E:5648:CDF7:65B3:2F29:B3D5:3995/64  
AB34:665F:B90B:3290:EA11:2678:FFFF:3210
```

The following example configures a null interface for network, 665F:B90B:3290:EA11:CDF7:65B3:2F29:B3D5/128:

```
[local]Redback(config-ctx)#ipv6 route 665F:B90B:3290:EA11:CDF7:65B3:2F29:B3D5/128 null0
```

The following example routes packets for network, 2000:8A2E:5648:CDF7:65B3:2F29:B3D5:3995/64, to the device at IP address, AB34:665F:B90B:3290:EA11:2678:FFFF:3210, if dynamic information with administrative distance less than 110 is not available:

```
[local]Redback(config-ctx)#ipv6 route 2000:8A2E:5648:CDF7:65B3:2F29:B3D5:3995/64  
AB34:665F:B90B:3290:EA11:2678:FFFF:3210 distance 110
```

Related Commands

ip route

ip verify unicast source

```
ip verify unicast source reachable-via {any | rx} [allow-default] [allow-self-ping]
[access-group acl-name [acl-count]]
```

Purpose

Performs a reverse path forwarding (RPF) check to verify the source IP address on all incoming unicast packets at the specified interface.

Command Mode

interface configuration

Syntax Description

reachable-via any	Specifies that the source IP address can be reached through any interface.
reachable-via rx	Specifies that the source IP address can be reached through an incoming interface.
allow-default	Optional. Allows the RPF check to look up the default route for verification.
allow-self-ping	Optional. Allows an interface to ping itself.
access-group acl-name	Optional. ACLs to use for verifying source IP addresses.
acl-count	Optional. Enables the counting of ACLs.

Default

None

Usage Guidelines

Use the **ip verify unicast source** command to perform an RPF check to verify the source IP address on all incoming unicast packets at the specified interface.

If the packet passes the RPF check, the packet is forwarded as normal; however, if the router does not find a reverse path for the packet, the packet is dropped.

The unicast RPF check is a network security feature designed to address RFC 2827, *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*. That is, the Unicast RPF check feature addresses problems that are caused by the introduction of frequently changing or forged (spoofed) source IP addresses into a network by discarding IP packets that have no verifiable source IP address. Denial-of-Service (DoS) attacks use spoofed source IP addresses to give attackers the ability to circumvent efforts to locate or stop the attacks. Such attacks are eliminated by forwarding only packets that have source addresses that are valid and consistent with the IP routing table.

Note Verifying the unicast source should be applied to an inbound interface at the upstream end of a connection.

Examples

The following example performs a unicast RPF check from interface `foo` on all unicast sources reachable by any interface:

```
[local]Redback(config-ctx)#interface foo  
[local]Redback(config-if)#ip verify unicast source reachable-via any
```

Related Commands

ip route

router-id

router-id *ip-addr*

no router-id

Purpose

Configures a global router ID for the SmartEdge router.

Command Mode

context configuration

Syntax Description

ip-addr IP address of the interface to be used as the router ID.

Default

A global router ID is not preconfigured.

Usage Guidelines

Use the **router-id** command to configure a global router ID for the SmartEdge router.

The global router ID in context configuration mode provides a consistent router ID for use by all routing protocols; however, if the router ID is configured as part of an individual routing protocol, such as OSPF or BGP, it will take precedence over the global router ID in context configuration mode.

Note The global router ID must be configured for RSVP to operate correctly.

Use the **no** form of this command to remove a global router ID.

Examples

The following example configures the IP address, 193.25.105.83, as the global router ID in context configuration mode:

```
[local]Redback(config)#context local  
[local]Redback(config-ctx)#router-id 193.25.105.83
```

Related Commands

router-id—BGP router configuration mode

router-id—OSPF router configuration mode

router rsvp

service inter-context routing

service inter-context routing

no service inter-context routing

Purpose

Enables intercontext static routing among non-local contexts.

Command Mode

global configuration

Syntax Description

This command has no keywords or arguments.

Default

Disabled

Usage Guidelines

Use the **service inter-context routing** command to enable intercontext static routing among non-local contexts. When this command is not enabled, intercontext static routing can still be used between the local context and non-local contexts.

Note This command can only be disabled when there is no instance of non-local context static routing configured on the router.

For more information on creating and servicing contexts, see the “Context Configuration” chapter in the *Basic System Configuration Guide* for the SmartEdge OS.

Examples

The following example enables non-local inter - context static routing:

```
[local]Redback(config)#service inter-context routing
[local]Redback(config)#context cust-abc
[local]Redback(config-ctx)#ip route 11.1.1.0/24 context web-xyz
[local]Redback(config-ctx)#context web-xyz
[local]Redback(config-ctx)#ip route 12.2.0.0/16 context cust-abc
```

Related Commands

ip route

tcp path-mtu-discovery

tcp path-mtu-discovery

no tcp path-mtu-discovery

Purpose

Enables the negotiation of the maximum transmission unit (MTU) for Transmission Control Protocol (TCP) sessions.

Command Mode

global configuration

Syntax Description

This command has no keywords or arguments.

Default

MTU negotiation is disabled.

Usage Guidelines

Use the **tcp path-mtu-discovery** command to enable the negotiation of the MTU for TCP sessions. Enabling MTU negotiation has no effect on existing TCP sessions.

TCP has the ability to dynamically discover the largest MTU that can be used on the session pipe and that minimizes fragmentation and maximizes efficiency. As described in RFC 1191, *Path MTU Discovery*, the default size of an IP packet is 576 bytes. The IP and TCP portions of the frame occupy 40 bytes leaving 536 bytes for the data payload. This payload is referred to as the maximum segment size (MSS).

This command allows the MSS (and hence the MTU) to be negotiated. When you enter this command and start a TCP session, the SYN packet sent by the SmartEdge router contains a TCP option specifying a larger MSS. This larger MSS is the MTU of the outbound interface minus 40 bytes. If the MTU of the outbound interface is 1500 bytes, the advertised MSS is 1460.

Both the SmartEdge router and the remote router must be configured for MTU negotiation to work properly. If both routers have MTU negotiation enabled, the SYN from one router to the other contains the optional TCP value advertising the higher MSS. The returning SYN then advertises the higher MSS value. If one router has MTU negotiation enabled and the second router never advertises the larger MSS, the first router is locked into sending the default values.

Use the **no** form of this command to disable the negotiation of the MTU for TCP sessions.

Examples

The following example enables the negotiation of the MTU for TCP sessions.

```
[local]Redback(config)#tcp path-mtu-discovery
```

Related Commands

None

DVSR Configuration

This chapter provides an overview of dynamically verified static routing (DVSR), describes the tasks and commands used to configure DVSR features through the SmartEdge® OS, and provides DVSR configuration examples.

For information about the tasks and commands used to monitor, troubleshoot, and administer DVSR, see the “DVSR Operations” chapter in the *Routing Protocols Operations Guide* for the SmartEdge OS.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples
- Command Descriptions

Overview

DVSR is a semidynamic and semistatic routing protocol used mainly for making edge routing decisions.

SmartEdge routers support DVSR as a unique edge routing feature in addition to static routing and regular Interior Gateway Protocols (IGPs), such as Intermediate System-to-Intermediate System (IS-IS), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP). DVSR is similar to normal static routing. The main difference is that the DVSR's next hop, or some other relevant host IP address, is dynamically verified by this protocol before the prefix can be injected into the local routing table. In many ISP networks, using static routing without proper next-hop checks results in blackholing of network traffic.

Static routes are often used on edge routers; however, with this additional dynamic host address verification, it can be safely used in some cases where static routing is not considered to be appropriate.

The DVSR routes can be redistributed into Border Gateway Protocol (BGP) or IGPs. A number of mechanisms can be used to redistribute specific DVSR routes; for example:

- Use the **redistribute** command (in BGP, IS-IS, OSPF, or RIP router configuration mode) to redistribute all the DVSR routes into a dynamic routing protocol.
- Use the **route map** command to either match the route type of DVSR, or to match the route tag. A route tag can be defined in a DVSR profile to cover all the DVSR routes associated with the profile, or it can be explicitly specified using the **ip route** command (in context configuration mode).

There are many applications where DVSR can be applied, including the following applications:

- Anycast routing

Some ISPs use anycast routing to offer load sharing services for their Domain Name System (DNS), HTTP, File Transfer Protocol (FTP), and mail relay services. DVSR provides simple way to announce the routes of the services for the servers that are up.

- Customer access and multi-homing

With the use of DVSR, the status of remote access connections can be verified, and static routes can be removed from the router if the remote connection is not available. It can also ease the burden on customers to run BGP on their sites for the purpose of multi-homing.

- Using dynamic routing to back up static routing

Static routing is often used to back up dynamic routing. With DVSR, dynamic routing can be used to back up static routing; for example, DVSR routes can be temporarily set up to alleviate link congestion. When those DVSR routes fail, dynamic routing takes over, which avoids blackholing of traffic.

- Load sharing on multiple LAN circuits

Unlike some point-to-point circuits, LAN or virtual permanent virtual circuits (PVCs) do not always offer a mechanism to learn the next-hop status, which means that using normal static routing is not appropriate in such cases; however, DVSR can be safely used.

- Suppressing summary routes in the case of IGP area partition.

When multiple area border routers announce the same summary routes, and if there is an intra-area network partition, traffic into that area may be blackholed. With DVSR, the area border routers can detect the area partition status, and suppress the summary route announcements.

Configuration Tasks

Note In this section, the command syntax in the task tables displays only the root command; for the complete command syntax, see the full description for the command in the “Command Descriptions” section.

To configure DVSR, perform the tasks described in the “Configuring a DVSR Profile” section.

Configuring a DVSR Profile

To configure a DVSR profile, perform the tasks described in Table 3-1. Enter all commands in DVSR profile configuration mode, unless otherwise noted.

Table 3-1 Configure a DVSR Profile

Task	Root Command	Notes
Create a DVSR profile and enter DVSR profile configuration mode.	dvsr-profile	Enter this command in context configuration mode. If no DVSR parameters are set, the profile uses default values for the DVSR parameters. All DVSR routes must reference an existing DVSR profile.
Configure the distance value for a DVSR profile.	distance	You can also define the distance value when configuring a DVSR route. In that case, the defined DVSR route distance overwrites the distance specified in the DVSR profile.
Configure the packet source IP address value for the DVSR profile.	source-address	
Configure the route tag value for the DVSR profile.	tag	You can also define the route tag value when configuring a DVSR route. In that case, the specified DVSR route tag value overwrites the value in the DVSR profile.
Configure the TTL value for the DVSR profile.	ttl	
Configure verify-set values for a DVSR profile.	verify-set	

Configuration Examples

This section contains DVSR configuration examples in the following subsections:

- Basic DVSR
- DVSR in Anycast Application
- DVSR in Customer Multihoming Application

Basic DVSR

To enable DVSR, or to announce DVSR routes, you must first define a DVSR profile. DVSR routes may have different requirement, thus more than one DVSR profile can be configured. Optionally, each DVSR route can specify parameters to overwrite profile definitions.

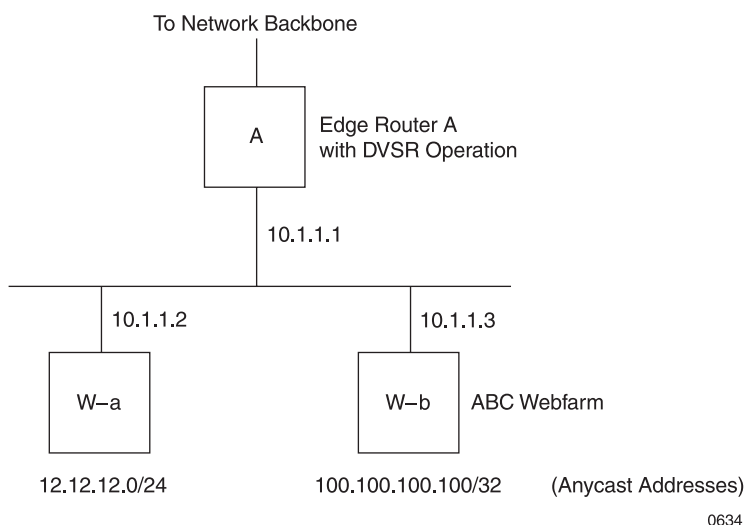
The following example shows one DVSR profile, and one DVSR route, using all default parameters. The DVSR profile `abc-web` is configured with a prefix of `10.10.0.0/16`, and with a next hop of `10.1.1.1`. The DVSR verify host is the next hop of the prefix, which is `10.1.1.1`. As long as the `10.1.1.1` host address is up, the prefix `10.10.0.0/16` is injected into the local routing table as a static route with a DVSR subtype.

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#dvsr-profile abc-web
[local]Redback(config-dvsr)#exit
[local]Redback(config-ctx)#ip route 10.10.0.0/16 10.1.1.1 dvsr abc-web
```

DVSR in Anycast Application

Figure 3-1 illustrates a network topology where a DVSR-enabled edge router, Router A, shares a LAN with two workstations in a webfarm.

Figure 3-1 Basic Anycast Network Topology



The W-a and W-b workstations serve applications with IP subnets of 12.12.12.0/24 and 100.100.100.100/32 as anycast addresses. (Somewhere else, other workstations also serve the same anycast addresses.) Edge Router A should announce those two anycast addresses only if workstations W-a and W-b are up. The anycast routes are redistributed into BGP.

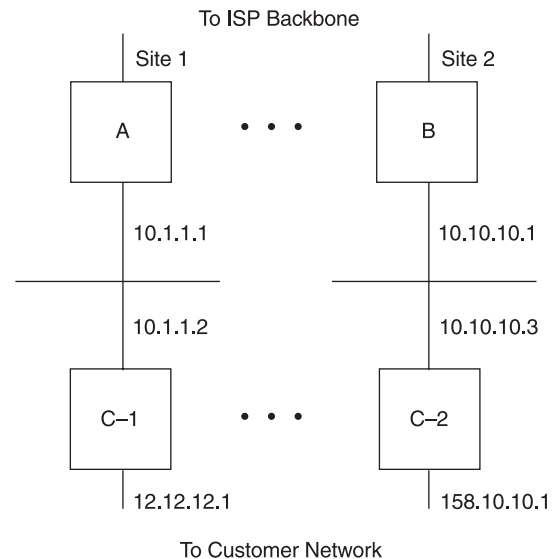
The DVSR configuration for edge router A is as follows:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#dvsr-profile abc-webfarm
[local]Redback(config-dvsr)#ttl 2
[local]Redback(config-dvsr)#verify-set 30 timeout-multiplier 4 min-success 3
[local]Redback(config-dvsr)#exit
[local]Redback(config-ctx)#ip route 12.12.12.0/24 10.1.1.2 dvsr abc-webfarm
[local]Redback(config-ctx)#ip route 100.100.100.100/32 10.1.1.3 dvsr abc-webfarm
[local]Redback(config-ctx)#router bgp 65000
[local]Redback(config-bgp)#address-family ipv4 unicast
[local]Redback(config-addrfamily)#redistribute static dvsr
```

DVSR in Customer Multihoming Application

Figure 3-2 illustrates that an ISP has a customer network multihomed into edge router A and edge router B. The customer network has IP subnets 12.12.12.0/24, 12.12.25.0/23, and 158.10.10.0/24.

Figure 3-2 Basic Customer Multihoming Network Topology



0635

Routers C-1 and C-2 do not run BGP, or any other dynamic routing protocol. DVSR is used in this case to inject customer routes into the backbone. If router C-1 or C-2 fails, or if customer internal links fail, routers A or B withdraws the DVSR routes, thus avoiding the blackholing of traffic towards the customer network.

The DVSR configuration for edge router A is as follows:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#dvsr-profile multi-home-c
[local]Redback(config-dvsr)#ttl 3
[local]Redback(config-dvsr)#tag 123
[local]Redback(config-dvsr)#exit
[local]Redback(config-ctx)#ip route 12.12.12.1/32 10.1.1.2
[local]Redback(config-ctx)#ip route 12.12.12.0/24 10.1.1.2 dvsr multi-home-c 12.12.12.1
[local]Redback(config-ctx)#ip route 12.12.25.0/23 10.1.1.2 dvsr multi-home-c 12.12.12.1
[local]Redback(config-ctx)#ip route 158.10.10.0/24 10.1.1.2 dvsr multi-home-c 12.12.12.1
[local]Redback(config-ctx)#router isis ip-backbone
[local]Redback(config-isis)#redistribute static dvsr
```

The DVSR configuration for edge router B is as follows:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#dvsr-profile multi-home-c
[local]Redback(config-dvsr)#ttl 3
[local]Redback(config-dvsr)#tag 123
[local]Redback(config-dvsr)#exit
[local]Redback(config-ctx)#ip route 158.10.10.1/32 10.10.10.3
```

```
[local]Redback(config-ctx)#ip route 12.12.12.0/24 10.10.10.3 dvsr multi-home-c
158.10.10.1
[local]Redback(config-ctx)#ip route 12.12.25.0/23 10.10.10.3 dvsr multi-home-c
158.10.10.1
[local]Redback(config-ctx)#ip route 158.10.10.0/24 10.10.10.3 dvsr multi-home-c
158.10.10.1
[local]Redback(config-ctx)#router isis ip-backbone
[local]Redback(config-isis)#redistribute static dvsr
```

Command Descriptions

This section describes the syntax and usage guidelines for the commands used to configure DVSR features. The commands are presented in alphabetical order.

distance	tag
dvsr-profile	ttl
source-address	verify-set

distance

distance *value*

Purpose

Configures the distance value for a dynamically verified static routing (DVSR) profile.

Command Mode

DVSR profile configuration

Syntax Description

value Distance value. The range of values is 1 to 255; the default value is 1.

Default

Distance value is 1, which is the same as static routes.

Usage Guidelines

Use the **distance** command to configure the distance value for a DVSR profile. The distance value is used in the route selection decision.

Note You can also define the distance value when configuring a DVSR route. In that case, the defined DVSR route distance overwrites the distance specified in the DVSR profile.

Examples

The following example defines a DVSR profile using distance of 255:

```
[local]Redback(config-ctx)#dvsr-profile abc-webfarm  
[local]Redback(config-dvsr)#distance 255
```

Related Commands

dvsr-profile
ip route
redistribute—BGP address family configuration mode
redistribute—IS-IS router configuration mode
redistribute—OSPF router configuration mode
redistribute—RIP router configuration mode
source-address
tag
ttl
verify-set

dvsr-profile

dvsr-profile *prof-name*
no dvsr-profile *prof-name*

Purpose

Creates a dynamically verified static routing (DVSR) profile and enters DVSR profile configuration mode.

Command Mode

context configuration

Syntax Description

prof-name DVSR profile name.

Default

No DVSR profile is configured.

Usage Guidelines

Use the **dvsr-profile** command to create a DVSR profile, and enter DVSR profile configuration mode. You can use the DVSR profile to set the desired values for the DVSR operation. If no DVSR parameters are set, the profile uses default values for the DVSR parameters. All DVSR routes must reference an existing DVSR profile.

Examples

The following example defines a DVSR profile, `abc-webfarm`, with a time-to-live (TTL) of 3, a verification interval of 25 seconds, a timeout multiplier of 4, and a minimum success of 2:

```
[local]Redback(config)#context foo
[local]Redback(config-ctx)#dvsr-profile abc-webfarm
[local]Redback(config-dvsr)#ttl 3
[local]Redback(config-dvsr)#verify-set 25 timeout-multiplier 4 min-success 2
```

Related Commands

distance	redistribute —RIP router configuration mode
ip route	source-address
redistribute —BGP address family configuration mode	tag
redistribute —IS-IS router configuration mode	ttl
redistribute —OSPF router configuration mode	verify-set

source-address

source-address *src-addr*

no source-address

Purpose

Configures the packet source IP address value for a dynamically verified static routing (DVSR) profile.

Command Mode

DVSR profile configuration

Syntax Description

<i>src-addr</i>	Source IP address of the verification packet. If the source IP address is not set, IP packets use the outbound interface primary IP address.

Default

Source IP address is not set.

Usage Guidelines

Use the **source-address** command to configure the packet source IP address value for a DVSR profile. Because some routers can only recognize the stable address of a router, such as the loopback address, you must configure the source IP address to ensure that the verified host has the route to reach the routers.

Use the **no** form of this command to delete the packet source IP address value from a DVSR profile.

Examples

The following example defines a DVSR profile source address of 10.1.1.1:

```
[local]Redback(config-ctx)#dvsr-profile abc-webfarm
[local]Redback(config-dvsr)#source-address 10.1.1.1
```

Related Commands

distance
dvsr-profile
ip route
redistribute—BGP address family configuration mode
redistribute—IS-IS router configuration mode
redistribute—OSPF router configuration mode
redistribute—RIP router configuration mode
tag
tth
verify-set

tag

tag *value*

no tag

Purpose

Configures the route tag value for a dynamically verified static routing (DVSR) profile.

Command Mode

DVSR profile configuration

Syntax Description

<i>value</i>	Route tag value. An unsigned 32-bit integer, the range of values is 1 to 4,294,967,295; the default value is 0.

Default

The default route tag value is 0.

Usage Guidelines

Use the **tag** command to configure the route tag value for a DVSR profile. For route redistribution, the route tag can be used for route map matches.

Note You can also define the route tag value when configuring a DVSR route. In that case, the specified DVSR route tag value overwrites the value in the DVSR profile.

Use the **no** form of this command to delete the route tag value from a DVSR profile.

Examples

The following example defines a DVSR profile using a route tag of 123; however, it is not used by the DVSR route 10.1.0.0/16, because it defines its own route tag value of 456:

```
[local]Redback(config-ctx)#dvsr-profile abc-webfarm
[local]Redback(config-dvsr)#tag 123
[local]Redback(config-dvsr)#exit
[local]Redback(config-ctx)#ip route 10.0.0.0/8 10.10.10.10 dvsr abc-webfarm
[local]Redback(config-ctx)#ip route 10.1.0.0/16 10.10.10.10 dvsr abc-webfarm tag 456
```

Related Commands

distance	redistribute —OSPF router configuration mode
dvsr-profile	redistribute —RIP router configuration mode
ip route	source-address
redistribute —BGP address family configuration mode	ttl
redistribute —IS-IS router configuration mode	verify-set

ttl

ttl *value*

no ttl

Purpose

Configures the time-to-live (TTL) value for a dynamically verified static routing (DVSR) profile.

Command Mode

DVSR profile configuration

Syntax Description

value TTL value. The range of values is 1 to 255; the default value is 5.

Default

The default TTL value is 5.

Usage Guidelines

Use the **ttl** command to configure the TTL value for a DVSR profile. The TTL value controls the maximum number of hops the verification packet can traverse; for example, if there are multiple paths to reach the verify host address, you must restrict the verification packet to the shorter paths to be considered a successful verification.

Use the **no** form of this command to delete the TTL value from a DVSR profile.

Examples

The following example defines a DVSR profile using a TTL value of 2:

```
[local]Redback(config-ctx)#dvsr-profile abc-webfarm
[local]Redback(config-dvsr)#ttl 2
```

Related Commands

distance
dvsr-profile
ip route
redistribute—BGP address family configuration mode
redistribute—IS-IS router configuration mode
redistribute—OSPF router configuration mode
redistribute—RIP router configuration mode
source-address
tag
verify-set

verify-set

verify-set *interval* [**timeout-multiplier** *count*] [**min-success** *count*]

no verify-set

Purpose

Configures the verify-set values for a dynamically verified static routing (DVSR) profile.

Command Mode

DVSR profile configuration

Syntax Description

<i>interval</i>	Interval value that defines how often DVSR route verification occurs. The interval range, in seconds, is 10 to 7,200; the default value is 20. It can only be set in 5-second increments.
timeout-multiplier <i>count</i>	Optional. Timeout multiplier. The <i>count</i> argument defines the number of verification failures that a DVSR route must have before being considered in the down state; the default value is 3.
min-success <i>count</i>	Optional. Minimum success. The <i>count</i> argument defines the number of verification successes that a DVSR route must have before being considered in the up state; the default value is 2.

Default

For a DVSR profile, the default interval value is 20 seconds, the default timeout multiplier value is 3, and the default minimum success value is 2.

Usage Guidelines

Use the **verify-set** command to configure the verify-set values for a DVSR profile. The verify set values control the frequency of the verification of DVSR routes, and change the measurement of verification. The smaller the number is, the more responsive the DVSR route becomes; however, fast response may cause network instability, especially in the case of packet loss in the network.

Use the **no** form of this command to delete the verify-set value from a DVSR profile.

Examples

The following example defines a DVSR profile using a verification interval of 25 seconds, a timeout multiplier of 4, and a minimum success of 2:

```
[local]Redback(config-ctx)#dvsr-profile abc-webfarm
[local]Redback(config-dvsr)#verify-set 25 timeout-multiplier 4 min-success 2
```

Related Commands

distance
dvsr-profile
ip route
redistribute—BGP address family configuration mode
redistribute—IS-IS router configuration mode
redistribute—OSPF router configuration mode
redistribute—RIP router configuration mode
source-address
tag
ttl

VRRP Configuration

This chapter provides an overview of the Virtual Router Redundancy Protocol (VRRP) and describes the tasks and commands used to configure VRRP features through the SmartEdge® OS.

For information about the tasks and commands used to monitor, troubleshoot, and administer VRRP, see the “VRRP Operations” chapter in the *Routing Protocols Operations Guide* for the SmartEdge OS.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples
- Command Descriptions

Overview

VRRP eliminates the single point of failure that is common in the static default routed environment and provides a higher availability default path without requiring the configuration of dynamic routing or router discovery protocols on every end host.

VRRP works by dynamically assigning responsibility for a virtual router to one of the VRRP routers on a LAN. A virtual router is defined by its virtual router ID (VRID) and a set of IP addresses. There are two types of VRRP routers—owner and backup. The VRRP router controlling the IP addresses associated with a virtual router is called the owner, and it forwards packets sent to the IP addresses.

Each VRRP router has a single well-known medium access control (MAC) address allocated to it. The MAC address is used as the source in all periodic VRRP messages sent by the owner router, enabling bridge learning in an extended LAN. Any of the virtual router’s IP addresses on a LAN can then be used as the default first-hop router by end hosts. When VRRP is configured on multiple virtual LANs (VLANs) on the same Ethernet port, unique VRIDs must be used on each VLAN to allow MAC-level filtering to be done on a port basis.

A VRRP router can associate a virtual router with its real addresses on an interface, and can also be configured with additional virtual router mapping and priorities for virtual routers it is willing to back up. The mapping between VRIDs and addresses must be coordinated among all VRRP routers on a LAN. However, there is no restriction against reusing a VRID with a different address mapping on different LANs. The scope of each virtual router is restricted to a single LAN.

To minimize network traffic, only the owner for each virtual router sends periodic VRRP advertisement messages. A backup router will not attempt to preempt the owner unless it has higher priority. This eliminates service disruption unless a more preferred path is available. The one exception is that a VRRP router always becomes owner of any virtual router associated with addresses it owns. If the owner becomes unavailable, the highest priority backup router transitions to owner status after a short delay, thus providing a controlled transition of the virtual router responsibility with minimal service interruption.

The typical operational scenarios are defined as two redundant routers, multiple routers with distinct path preferences among each router, or a combination of both. When more than two redundant paths have equal preference, duplicate packets may be forwarded for a brief period during owner election. However, typical operational scenarios cover most deployments. Loss of the owner router is infrequent, and the expected duration in owner election convergence is minimal (less than one second). These VRRP optimizations represent significant simplifications in the protocol design, while incurring an insignificant probability of brief network degradation.

The SmartEdge OS supports a standard authentication method plus a proprietary Message Digest 5 (MD5) method, providing simple deployment in insecure environments, added protection against misconfiguration, and strong sender authentication in security-conscious environments.

For more details on VRRP, see RFC 2338, *Virtual Router Redundancy Protocol*.

Configuration Tasks

Note In this section, the command syntax in the task tables displays only the root command; for the complete command syntax, see the full description for the command in the “Command Descriptions” section.

To configure VRRP, perform the tasks described in the following sections:

- Configuring a VRRP Owner Router
- Configuring a VRRP Backup Router

Configuring a VRRP Owner Router

To configure a VRRP owner router, perform the tasks described in Table 4-1. Enter all commands in VRRP configuration mode, unless otherwise noted.

Table 4-1 Configure a VRRP Owner Router

Task	Root Command	Notes
Enter VRRP configuration mode and configure the VRRP ID.	vrrp	Enter this command in interface configuration mode. Use the following command syntax: vrrp router-id owner
Configure the interval at which VRRP advertisements are sent out from the specified interface.	advertise-interval	
Configure authentication of VRRP exchanges.	authentication	
Configure the virtual IP address for the VRRP interface.	virtual-address	

Configuring a VRRP Backup Router

To configure a VRRP backup router, perform the tasks described in Table 4-2. Enter all commands in VRRP configuration mode, unless otherwise noted.

Table 4-2 Configure a VRRP Backup Router

Task	Root Command	Notes
Enter VRRP configuration mode and configure the VRRP ID.	vrrp	Enter this command in interface configuration mode. Use the following command syntax: vrrp router-id backup
Configure the interval at which VRRP advertisements are sent out from the specified interface.	advertise-interval	
Configure authentication of VRRP exchanges.	authentication	
Enable a higher priority VRRP backup router to preempt a lower priority VRRP master.	preempt	
Configure VRRP owner election priority for a backup virtual router.	priority	
Configure the virtual IP address of the VRRP interface.	virtual-address	

Configuration Examples

The following sections provide examples of how to configure routers running VRRP:

- Basic VRRP
- Mutual VRRP
- Mutual VRRP on Different Subnets
- Mutual VRRP on Multiple Subnets
- MD5 Authentication

Basic VRRP

The following snapshots from two configuration files configure two routers running VRRP on a single interface, with the SE2 router backing up the SE1 router:

The SE1 router configuration is as follows:

```
[local]SE1(config)#context local
[local]SE1(config-ctx)#interface one
[local]SE1(config-if)#ip address 10.1.1.1/24
[local]SE1(config-if)#vrrp 1 owner
[local]SE1(config-vrrp)#virtual-address 10.1.1.1
[local]SE1(config-vrrp)#exit
[local]SE1(config-if)#exit
[local]SE1(config-ctx)#exit
```

```
[local]SE1(config)#port ethernet 7/2
[local]SE1(config-port)#bind interface one local
[local]SE1(config-port)#no shutdown
```

The SE2 router configuration is as follows:

```
[local]SE2(config)#context local
[local]SE2(config-ctx)#interface one
[local]SE2(config-if)#ip address 10.1.1.2/24
[local]SE2(config-if)#vrrp 1 backup
[local]SE2(config-if-vrrp)#virtual-address 10.1.1.1
[local]SE2(config-vrrp)#exit
[local]SE2(config-if)#exit
[local]SE2(config-ctx)#exit
[local]SE2(config)#port ethernet 7/2
[local]SE2(config-port)#bind interface one local
[local]SE2(config-port)#no shutdown
```

Mutual VRRP

The following snapshots from two configuration files configure two routers running VRRP on a single interface, with the two routers backing up each other:

The SE1 router configuration is as follows:

```
[local]SE1(config)#context local
[local]SE1(config-ctx)#interface one
[local]SE1(config-if)#ip address 10.1.1.1/24
[local]SE1(config-if)#vrrp 1 owner
[local]SE1(config-vrrp)#virtual-address 10.1.1.1
[local]SE1(config-vrrp)#exit
[local]SE1(config-if)#vrrp 2 backup
[local]SE1(config-vrrp)#virtual-address 10.1.1.2
[local]SE1(config-vrrp)#exit
[local]SE1(config-if)#exit
[local]SE1(config-ctx)#exit
[local]SE1(config)#port ethernet 7/2
[local]SE1(config-port)#bind interface one local
[local]SE1(config-port)#no shutdown
```

The SE2 router configuration is as follows:

```
[local]SE2(config)#context local
[local]SE2(config-ctx)#interface one
[local]SE2(config-if)#ip address 10.1.1.2/24
[local]SE2(config-if)#vrrp 1 backup
[local]SE2(config-vrrp)#virtual-address 10.1.1.1
[local]SE2(config-vrrp)#exit
[local]SE2(config-if)#vrrp 2 owner
[local]SE2(config-vrrp)#virtual-address 10.1.1.2
[local]SE2(config-vrrp)#exit
[local]SE2(config-if)#exit
[local]SE2(config-ctx)#exit
```

```
[local]SE2(config)#port ethernet 7/2
[local]SE2(config-port)#bind interface one local
[local]SE2(config-port)#no shutdown
```

Mutual VRRP on Different Subnets

The following snapshots from two configuration files configure two routers running VRRP on a single interface, with the two routers backing up each other on different subnets:

The SE1 router configuration is as follows:

```
[local]SE1(config)#context local
[local]SE1(config-ctx)#interface one
[local]SE1(config-if)#ip address 10.1.1.1/24
[local]SE1(config-if)#ip address 20.1.1.1/24 secondary
[local]SE1(config-if)#vrrp 1 owner
[local]SE1(config-vrrp)#virtual-address 10.1.1.1
[local]SE1(config-vrrp)#exit
[local]SE1(config-if)#vrrp 2 backup
[local]SE1(config-vrrp)#virtual-address 20.1.1.2
[local]SE1(config-vrrp)#exit
[local]SE1(config-if)#exit
[local]SE1(config-ctx)#exit
[local]SE1(config)#port ethernet 7/2
[local]SE1(config-port)#bind interface one local
[local]SE1(config-port)#no shutdown
```

The SE2 router configuration is as follows:

```
[local]SE2(config)#context local
[local]SE2(config-ctx)#interface one
[local]SE2(config-if)#ip address 10.1.1.2/24
[local]SE2(config-if)#ip address 20.1.1.2/24 secondary
[local]SE2(config-if)#vrrp 1 backup
[local]SE2(config-vrrp)#virtual-address 10.1.1.1
[local]SE2(config-vrrp)#exit
[local]SE2(config-if)#vrrp 2 owner
[local]SE2(config-vrrp)#virtual-address 20.1.1.2
[local]SE2(config-vrrp)#exit
[local]SE2(config-if)#exit
[local]SE2(config-ctx)#exit
[local]SE2(config)#port ethernet 7/2
[local]SE2(config-port)#bind interface one local
[local]SE2(config-port)#no shutdown
```

Mutual VRRP on Multiple Subnets

The following snapshots from three configuration files configure three routers running VRRP on a single interface, with the routers backing up each other on different subnets. For each subnet, there is an owner and two backups. Using VRRP priority, one backup is preferred over another.

The SE1 router configuration is as follows:

```
[local]SE1(config)#context local
[local]SE1(config-ctx)#interface one
[local]SE1(config-if)#ip address 10.1.1.1/24
[local]SE1(config-if)#ip address 20.1.1.1/24 secondary
[local]SE1(config-if)#ip address 30.1.1.1/24 secondary
[local]SE1(config-if)#vrrp 1 owner
[local]SE1(config-vrrp)#virtual-address 10.1.1.1
[local]SE1(config-vrrp)#exit
[local]SE1(config-if)#vrrp 2 backup
[local]SE1(config-vrrp)#virtual-address 20.1.1.2
[local]SE1(config-vrrp)#priority 100
[local]SE1(config-vrrp)#exit
[local]SE1(config-if)#vrrp 3 backup
[local]SE1(config-vrrp)#virtual-address 30.1.1.3
[local]SE1(config-vrrp)#priority 200
[local]SE1(config-vrrp)#exit
[local]SE1(config-if)#exit
[local]SE1(config-ctx)#exit
[local]SE1(config)#port ethernet 7/2
[local]SE1(config-port)#bind interface one local
[local]SE1(config-port)#no shutdown
```

The SE2 router configuration is as follows:

```
[local]SE2(config)#context local
[local]SE2(config-ctx)#interface one
[local]SE2(config-if)#ip address 10.1.1.2/24
[local]SE2(config-if)#ip address 20.1.1.2/24 secondary
[local]SE2(config-if)#ip address 30.1.1.2/24 secondary
[local]SE2(config-if)#vrrp 1 backup
[local]SE2(config-vrrp)#virtual-address 10.1.1.1
[local]SE2(config-vrrp)#priority 200
[local]SE2(config-vrrp)#exit
[local]SE2(config-if)#vrrp 2 owner
[local]SE2(config-vrrp)#virtual-address 20.1.1.2
[local]SE2(config-vrrp)#exit
[local]SE2(config-if)#vrrp 3 backup
[local]SE2(config-vrrp)#virtual-address 30.1.1.3
[local]SE2(config-vrrp)#priority 100
[local]SE2(config-vrrp)#exit
[local]SE2(config-if)#exit
[local]SE2(config-ctx)#exit
[local]SE2(config)#port ethernet 7/2
[local]SE2(config-port)#bind interface one local
[local]SE2(config-port)#no shutdown
```

The SE3 router configuration is as follows:

```
[local]SE3(config)#context local
[local]SE3(config-ctx)#interface one
[local]SE3(config-if)#ip address 10.1.1.3/24
[local]SE3(config-if)#ip address 20.1.1.3/24 secondary
[local]SE3(config-if)#ip address 30.1.1.3/24 secondary
[local]SE3(config-if)#vrrp 1 backup
[local]SE3(config-vrrp)#virtual-address 10.1.1.1
[local]SE3(config-vrrp)#priority 100
[local]SE3(config-vrrp)#exit
[local]SE3(config-if)#vrrp 2 backup
[local]SE3(config-vrrp)#virtual-address 20.1.1.2
[local]SE3(config-vrrp)#priority 200
[local]SE3(config-vrrp)#exit
[local]SE3(config-if)#vrrp 3 owner
[local]SE3(config-vrrp)#virtual-address 30.1.1.3
[local]SE3(config-vrrp)#exit
[local]SE3(config-if)#exit
[local]SE3(config-ctx)#exit
[local]SE3(config)#port ethernet 7/2
[local]SE3(config-port)#bind interface one local
[local]SE3(config-port)#no shutdown
```

MD5 Authentication

The following snapshots (from two configuration files) configure two routers running VRRP on a single interface using MD5 authentication.

The SE1 router configuration is as follows:

```
[local]SE1(config)#context local
[local]SE1(config-ctx)#interface one
[local]SE1(config-if)#ip address 10.1.1.1/24
[local]SE1(config-if)#vrrp 1 owner
[local]SE1(config-vrrp)#authentication redback-md5 rbak-md5-chain
[local]SE1(config-vrrp)#exit
[local]SE1(config-if)#exit
[local]SE1(config-ctx)#key-chain rbak-md5-chain key-id 1
[local]SE1(config-key-chain)#key-string secret
[local]SE1(config-key-chain)#exit
[local]SE1(config-ctx)#exit
[local]SE1(config)#port ethernet 7/2
[local]SE1(config-port)#bind interface one local
[local]SE1(config-port)#no shutdown
```

The SE2 router configuration is as follows:

```
[local]SE2(config)#context local
[local]SE2(config-ctx)#interface one
[local]SE2(config-if)#ip address 10.1.1.2/24
[local]SE2(config-if)#vrrp 1 backup
```

```
[local]SE2(config-vrrp)#virtual-address 10.1.1.1
[local]SE2(config-vrrp)#authentication redback-md5 rbak-md5-chain
[local]SE2(config-vrrp)#exit
[local]SE2(config-if)#exit
[local]SE2(config-ctx)#key-chain rbak-md5-chain key-id 1
[local]SE2(config-key-chain)#key-string secret
[local]SE2(config-key-chain)#exit
[local]SE2(config-ctx)#exit
[local]SE2(config)#port ethernet 7/2
[local]SE2(config-port)#bind interface one local
[local]SE2(config-port)#no shutdown
```

Command Descriptions

This section describes the syntax and usage guidelines for the commands used to configure VRRP features. The commands are presented in alphabetical order.

advertise-interval	priority
authentication	virtual-address
preempt	vrrp

advertise-interval

advertise-interval *interval*
{ **no** | **default** } **advertise-interval**

Purpose

Configures the interval at which Virtual Router Redundancy Protocol (VRRP) advertisements are sent out from the specified interface.

Command Mode

VRRP configuration

Syntax Description

<i>interval</i>	Amount of time, in seconds, between VRRP advertisements. The range of values is 1 to 255; the default value is 1.

Default

VRRP advertisements are sent out every second.

Usage Guidelines

Use the **advertise-interval** command to determine the frequency of VRRP advertisements sent from the specified interface. This command is useful for troubleshooting misconfigured routers.

Use the **no** or **default** form of this command to return the interval to its default value of 1.

Examples

The following example configures the interface, `eth0`, to send VRRP advertisements every 20 seconds:

```
[local]Redback(config)#interface eth0
[local]Redback(config-if)#vrrp 1 owner
[local]Redback(config-vrrp)#advertise-interval 20
```

Related Commands

virtual-address
vrrp

authentication

authentication { **none** | **redback-md5** *key-chain-name* | **simple** *key-chain-name* }
{ **no** | **default** } **authentication**

Purpose

Configures authentication of Virtual Router Redundancy Protocol (VRRP) exchanges.

Command Mode

VRRP configuration

Syntax Description

none	Specifies no authentication.
redback-md5 <i>key-chain-name</i>	Redback® Message Digest 5 (MD5) authentication key chain name.
simple <i>key-chain-name</i>	Simple authentication key chain name.

Default

Authentication is not enabled.

Usage Guidelines

Use the **authentication** command to enable authentication of VRRP exchanges.

Use the **no** or **default** form of this command to disable authentication of VRRP exchanges.

Examples

The following example configures a virtual router owner using our proprietary MD5 authentication:

```
[local]Redback(config-ctx)#interface one
[local]Redback(config-if)#ip address 10.1.1.1/24
[local]Redback(config-if)#vrrp 1 owner
[local]Redback(config-vrrp)#authentication redback-md5 redback-md5-chain
[local]Redback(config-vrrp)#exit
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#key-chain redback-md5-chain key-id 1 key-string secret
[local]Redback(config-key-chain)#exit
[local]Redback(config-ctx)#exit
[local]Redback(config)#port ethernet 7/2
[local]Redback(config-port)#bind interface one local
[local]Redback(config-port)#no shutdown
```

Related Commands

virtual-address
vrrp

preempt

preempt

{no | default} preempt

Purpose

Enables a higher priority Virtual Router Redundancy Protocol (VRRP) backup router to preempt a lower priority VRRP master.

Command Mode

VRRP configuration

Syntax Description

This command has no keywords or arguments.

Default

Preemption is enabled.

Usage Guidelines

Use the **preempt** command to enable a VRRP backup router that has a higher priority to preempt a lower priority VRRP master. When preemption is disabled, a higher priority VRRP backup router does not preempt a lower priority VRRP master.

Note Preemption can only be disabled for VRRP backup routers; VRRP owner routers always have preemption enabled.

Use the **no** form of this command to disable preemption.

Examples

The following example disables preemption on the VRRP backup router with virtual ID 23:

```
[local]Redback(config-if)#vrrp 23 backup
[local]Redback(config-vrrp)#no preempt
[local]Redback(config-vrrp)#
```

Related Commands

vrrp

priority

priority *priority*

no priority

Purpose

Configures the Virtual Router Redundancy Protocol (VRRP) election priority for the backup virtual router.

Command Mode

VRRP configuration

Syntax Description

priority Priority setting for the backup virtual router. The range of values is 1 to 254.

Default

The priority is set to 100.

Usage Guideline

Use the **priority** command to configure the VRRP priority for the backup virtual router.

Use the **no** form of this command to return the priority setting to its default value.

Examples

The following example configures VRRP backup routers for two separate routers, Router_A and Router_B, on the same LAN. Both VRRP backup routers have the same virtual ID, 2. The VRRP backup router on Router_A, which has a priority of 100, is preferred over the VRRP backup router on Router_B, which has a priority of 200.

The configuration for Router_A is as follows:

```
[local]Router_A(config)#context local
[local]Router_A(config-ctx)#interface foo
[local]Router_A(config-if)#ip address 1.1.1.100/24 secondary
[local]Router_A(config-if)#vrrp 2 backup
[local]Router_A(config-vrrp)#virtual-address 1.1.1.111
[local]Router_A(config-vrrp)#priority 100
```

The configuration for Router_B is as follows:

```
[local]Router_B(config)#context local
[local]Router_B(config-ctx)#interface foo
[local]Router_B(config-if)#ip address 1.1.1.200/24 secondary
[local]Router_B(config-if)#vrrp 2 backup
[local]Router_B(config-vrrp)#virtual-address 1.1.1.222
[local]Router_B(config-vrrp)#priority 200
```

Related Commands

virtual-address
vrrp

virtual-address

virtual-address *ip-addr*

no virtual-address *ip-addr*

Purpose

Configures the virtual IP address for the Virtual Router Redundancy Protocol (VRRP) interface.

Command Mode

VRRP configuration

Syntax Description

ip-addr Virtual IP address.

Default

None

Usage Guidelines

Use the **virtual-address** command to configure the virtual IP address for the VRRP interface. You can configure multiple virtual IP addresses for a single VRRP instance.

Note For a VRRP owner router, the virtual address must be match one of the interface IP addresses on which the owner VRRP is configured.



Caution Risk of conflicting IP addresses. Static Address Resolution Protocol (ARP) configuration takes precedence over a VRRP association of a virtual medium access control (MAC) address with a virtual address. To reduce the risk, avoid configuring static ARP entries for VRRP virtual addresses.

Use the **no** form of this command to remove the virtual IP address.

Examples

The following example configures a router running VRRP on interface `eth1` and assigns a virtual IP address of `10.1.1.2`:

```
[local]Redback(config-ctx)#interface eth1
[local]Redback(config-if)#ip address 10.1.1.2/24
[local]Redback(config-if)#vrrp 1 owner
[local]Redback(config-vrrp)#virtual-address 10.1.1.2
```

Related Commands

vrrp

vrrp

vrrp *router-id* {**owner** | **backup**}

no vrrp *router-id*

Purpose

Configures a virtual router as an owner or backup router, assigns a Virtual Router Redundancy Protocol (VRRP) ID and enters VRRP configuration mode.

Command Mode

interface configuration

Syntax Description

<i>router-id</i>	virtual router ID. The range of values is 1 to 255.
owner	Configures the virtual router as an owner.
backup	Configures the virtual router as a backup in the event an owner virtual router goes down.

Default

None

Usage Guidelines

Use the **vrrp** command to configure a virtual router as an owner or backup router, assign a VRRP ID, and to enter VRRP configuration mode.

For more information on VRRP, see RFC 2338, *Virtual Router Redundancy Protocol*.

Note Each virtual router corresponding to an interface that is bound to 802.1Q circuits and that uses the same Ethernet port must have a unique virtual router ID. If multiple interfaces are bound to 802.1Q circuits associated with the same Ethernet port, and there are virtual routers with duplicate router identifiers, only one of the virtual routers will be operational.

Use the **no** form of this command to remove the virtual router.

Examples

The following example configures an owner virtual router with a VRRP ID of 23:

```
[local]Redback(config-if)#vrrp 23 owner
[local]Redback(config-vrrp)#
```

Related Commands

virtual-address

RIP Configuration

This chapter provides an overview of the Routing Information Protocol (RIP) and describes the tasks and commands used to configure RIP features through the SmartEdge® OS.

For information about the tasks and commands used to monitor, troubleshoot, and administer RIP, see the “RIP Operations” chapter in the *Routing Protocols Operations Guide* for the SmartEdge OS.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples
- Command Descriptions

Overview

RIP is a distance-vector protocol that uses a hop count as its metric. Relatively old, RIP is still commonly used, especially in small homogeneous networks. Our implementation supports RIP Version 2 and provides for multiple RIP instances. Each instance maintains its own routing table and set of interfaces. Each interface can only be assigned to, at most, one RIP instance.

RIP is documented in RFC 1058, *Routing Information Protocol*, and RFC 1723, *RIP Version 2, Carrying Additional Information*.

RIP next generation (RIPng) is an enhanced version of RIP that supports IP Version 6 (IPv6)-based network routing. RIPng is documented in RFC 2080, *RIPng for IPv6*. For a description of IPv6 addressing and the types of IPv6 addresses, see RFC 3513, *Internet Protocol Version 6 (IPv6) Addressing Architecture*.

Note When IP Version 6 (IPv6) addresses are not referenced or explicitly specified, the term, IP address, can refer generally to IP Version 4 (IPv4) addresses, IPv6 addresses, or IP addressing. In instances where IPv6 addresses are referenced or explicitly specified, the term, IP address, refers only to IPv4 addresses.

Configuration Tasks

Note In this section, the command syntax in the task tables displays only the root command; for the complete command syntax, see the full description for the command in the “Command Descriptions” section.

To configure RIP or RIPng, perform the tasks described in the following sections:

- Configuring RIP
- Configuring RIPng

Configuring RIP

To configure RIP, perform the tasks described in the following sections:

- Configure a RIP Routing Instance
- Configure a RIP Interface

Configure a RIP Routing Instance

To configure a RIP routing instance, perform the tasks described in Table 5-1. Enter all commands in RIP router configuration mode, unless otherwise noted.

Table 5-1 Configure a RIP Routing Instance

Task	Root Command	Notes
Configure an instance of the RIP routing process and enter RIP router configuration mode.	router rip	Enter this command in context configuration mode.
Inject the default route (0.0.0.0) into the RIP instance.	default-information originate	
Set the default metric for the RIP instance.	default-metric	The default value is used when a route with incompatible metrics is received into the RIP instance; for example, when a route from a different routing domain is imported into RIP.
Modify the administrative distance for the RIP instance.	distance	Administrative distance specifies how desirable a route obtained from RIP is compared to the same route obtained from another protocol. The lower the value for the distance argument in comparison to other routes obtained from other protocols, the more desirable the RIP route becomes.
Apply a prefix list to RIP packets.	distribute-list	
Modify the minimum interval between consecutive RIP flash updates.	flash-update-threshold	Each flash update contains only those routes that have been changed since the most recent update.
Modify the number of multiple equal-cost RIP routes that can be used as the best paths for load balancing outgoing traffic packets.	maximum-paths	The SmartEdge router enables load balancing among these RIP paths if, in the routing table, they are the best paths among paths provided by all running routing protocols.

Table 5-1 Configure a RIP Routing Instance (*continued*)

Task	Root Command	Notes
Configure a RIP offset list.	offset-list	A RIP offset list adds to the cost metric of inbound or outbound routes learned or advertised by RIP.
Add a delay time between packets sent in multipacket RIP updates.	output-delay	This feature is useful for situations where a high-speed router is sending updates to a low-speed router.
Redistribute routes learned through protocols other than RIP into the RIP instance.	redistribute	You must enter multiple redistribute commands to redistribute routes from several different kinds of routing protocols into the RIP routing instance.
Modify RIP timers for the specified RIP instance.	timers basic	

Configure a RIP Interface

To configure a RIP interface, perform the tasks described in Table 5-2. Enter all commands in RIP interface configuration mode, unless otherwise noted.

Table 5-2 Configure a RIP Interface

Task	Root Command	Notes
Enable an interface to both send and receive RIP packets, and to access RIP interface configuration mode.	interface	Enter this command in RIP router configuration mode.
Enable authentication and specify the authentication scheme for the RIP interface.	authentication	
Configure the RIP interface to originate the default route (0.0.0.0).	default-information originate	
Modify the cost value of an interface.	interface-cost	The cost value is used by RIP as a metric for route selection. The lower the cost, the more likely an interface is to be used to forward data traffic.
Enable an interface to receive and process RIP packets.	listen	
Enable RIP split-horizon processing on an interface.	split-horizon	Simple split-horizon processing is enabled by default.
Summarize routes in RIP update packets on the specified interface.	summary-address	
Enable an interface to send RIP packets.	supply	
Modify RIP timers for the specified interface.	timers basic	

Configuring RIPng

To configure RIPng, perform the tasks described in the following sections:

- Configure a RIPng Routing Instance
- Configure a RIPng Interface

Configure a RIPng Routing Instance

To configure a RIPng routing instance, perform the tasks described in Table 5-3. Enter all commands in RIPng router configuration mode, unless otherwise noted.

Table 5-3 Configure a RIPng Routing Instance

Task	Root Command	Notes
Create an instance of the RIPng routing process and enter RIPng router configuration mode.	router ripng	Enter this command in context configuration mode.
Inject the default route (::/0) into the RIPng instance.	default-information originate	
Set the default metric for the RIPng instance.	default-metric	The default value is used when a route with incompatible metrics is received into the RIPng instance; for example, when a route from a different routing domain is imported into RIPng.
Modify the administrative distance for the RIPng instance.	distance	Administrative distance specifies how desirable a route obtained from RIPng is compared to the same route obtained from another protocol. The lower the value for the distance argument in comparison to other routes obtained from other protocols, the more desirable the RIP route becomes.
Apply a prefix list to RIPng packets.	distribute-list	
Modify the minimum interval between consecutive RIPng flash updates.	flash-update-threshold	Each flash update contains only those routes that have been changed since the most recent update.
Modify the number of multiple equal-cost RIPng routes that can be used as the best paths for load balancing outgoing traffic packets.	maximum-paths	The SmartEdge router enables load balancing among these RIPng paths if, in the routing table, they are the best paths among paths provided by all running routing protocols.
Add a delay time between packets sent in multipacket RIPng updates.	output-delay	This feature is useful for situations where a high-speed router is sending updates to a low-speed router.
Redistribute routes learned through protocols other than RIPng into the RIPng instance.	redistribute	You must enter multiple redistribute commands to redistribute routes from several different kinds of routing protocols into the RIPng routing instance.
Modify RIPng timers for the specified RIPng instance.	timers basic	

Configure a RIPng Interface

To configure a RIPng interface, perform the tasks described in Table 5-4. Enter all commands in RIPng interface configuration mode, unless otherwise noted.

Table 5-4 Configure a RIPng Interface

Task	Root Command	Notes
Enable an interface to both send and receive RIP packets, and to enter RIPng interface configuration mode.	interface	Enter this command in RIPng router configuration mode.
Configure the RIPng interface to originate the default route (::/0).	default-information originate	

Table 5-4 Configure a RIPng Interface (*continued*)

Task	Root Command	Notes
Modify the cost value of an interface.	interface-cost	The cost value is used by RIPng as a metric for route selection. The lower the cost, the more likely an interface is to be used to forward data traffic.
Enable an interface to receive and process RIPng packets.	listen	
Enable RIPng split-horizon processing on an interface.	split-horizon	Simple split-horizon processing is enabled by default.
Summarize routes in RIPng update packets on the specified interface.	summary-address	
Enable an interface to send RIPng packets.	supply	
Modify RIPng timers for the specified interface.	timers basic	

Configuration Examples

The following example configures one RIP instance, adjusts the maximum number of equal-cost paths to 4, originates a default route, and redistributes static routes into RIP with metric of 10. It then enables RIP on interface fe1.

```
[local]Redback#configure
[local]Redback(config)#context local
[local]Redback(config-ctx)#router rip edge
[local]Redback(config-rip)#maximum-paths 4
[local]Redback(config-rip)#default-information originate
[local]Redback(config-rip)#redistribute static metric 10
[local]Redback(config-rip)#interface fe1
[local]Redback(config-rip-if)#end
```

The following example configures two RIP instances in the local context. Next, it enables one RIP instance edge and a RIP instance backbone on interface fe1. An IP prefix list, prefixList1, is also applied on the outbound updates on interface fe1.

```
[local]Redback#configure
[local]Redback(config)#context local
[local]Redback(config-ctx)#router rip edge
[local]Redback(config-rip)#redistribute static metric 10
[local]Redback(config-rip)#interface fe1
[local]Redback(config-rip-if)#exit
[local]Redback(config-rip)#exit
[local]Redback(config-ctx)#router rip backbone
[local]Redback(config-rip)#distribute-list prefixList1 out fe1
[local]Redback(config-rip)#interface fe1
[local]Redback(config-rip-if)#end
```

Command Descriptions

This section describes the syntax and usage guidelines for the commands used to configure RIP features. The commands are presented in alphabetical order.

authentication	offset-list
default-information originate	output-delay
default-metric	redistribute
distance	router rip
distribute-list	router ripng
flash-update-threshold	split-horizon
interface	summary-address
interface-cost	supply
listen	timers basic
maximum-paths	

authentication

authentication {**md5** *key-chain-name* | **simple** *key-chain-name*}
 {**no** | **default**} **authentication**

Purpose

Enables authentication and specifies the authentication scheme for the Routing Information Protocol (RIP) interface.

Command Mode

RIP interface configuration

Syntax Description

md5 *key-chain-name* Message Digest 5 (MD5) authentication key chain name.
simple *key-chain-name* Simple authentication key chain name.

Default

Authentication is not enabled.

Usage Guidelines

Use the **authentication** command to enable authentication and specify the authentication scheme for the RIP interface.

Key chains allow you to control authentication keys used by various routing protocols in the system. All routers connected to the same IP subnet must use the same authentication scheme and key ID. If multiple key IDs have been configured, the one with the most current send time is used. For information on the **key-chain key-id** command, see the “Key Chain Configuration” chapter in the *IP Services and Security Configuration Guide* for the SmartEdge OS.

Use the **no** or **default** form of this command to disable authentication.

Examples

The following example configures MD5 authentication for the RIP interface, fe0, and simple authentication for the RIP interface, su12:

```
[local]Redback(config-ctx)#router rip rip001
[local]Redback(config-rip)#interface fe0
[local]Redback(config-rip-if)#authentication md5 auth01
[local]Redback(config-rip-if)#exit
[local]Redback(config-rip)#interface su12
[local]Redback(config-rip-if)#authentication simple auth02
[local]Redback(config-rip-if)#exit
[local]Redback(config-rip)#exit
[local]Redback(config-ctx)#key-chain auth01 keyid 1
[local]Redback(config-key-chain)#key-string secret
```

```
[local]Redback(config-key-chain)#exit  
[local]Redback(config-ctx)#key-chain auth02 keyid 1  
[local]Redback(config-key-chain)#key-string password
```

Related Commands

interface—RIP router configuration mode

interface-cost

listen

router rip

split-horizon

summary-address

supply

default-information originate

default-information originate [**route-map** *map-name*]

{**no** | **default**} **default-information originate** [**route-map** *map-name*]

Purpose

In RIP interface configuration mode, configures the specified Routing Information Protocol (RIP) or RIP next generation (RIPng) interface to originate the default route.

In RIP router configuration mode, injects the default route into the RIP or RIPng instance.

Command Mode

RIP interface configuration
RIPng interface configuration
RIPng router configuration
RIP router configuration

Syntax Description

route-map *map-name* Optional. Route map name. The conditions of the route map are applied to the default route.

Default

The default route is not sent.

Usage Guidelines

Use the **default-information originate** command (in RIP or RIPng interface configuration mode) to configure the specified RIP or RIPng interface to originate the default route, which is 0.0.0.0 for IPv4 and ::/0 for IPv6.

Use the **default-information originate** command (in RIP or RIPng router configuration mode) to inject the default route into the RIP or RIPng instance.

To apply a route map to the default route, use the optional **route-map** *map-name* construct. In this case, the default route is generated only when there is a match in the specified route map.

Use the **no** or **default** form of this command (in RIP or RIPng interface configuration mode) to configure the interface to not originate the default route.

Use the **no** or **default** form of this command (in RIP or RIPng router configuration mode) to not inject the default route into the RIP or RIPng instance.

Examples

The following example injects the default route into the `rip001` RIP instance:

```
[local]Redback(config-ctx)#router rip rip001  
[local]Redback(config-rip)#default-information originate
```

Command Descriptions

The following example originates the default route from the fe1 interface for the rip002 RIP instance:

```
[local]Redback(config-ctx)#router rip rip002  
[local]Redback(config-rip)#interface fe1  
[local]Redback(config-rip-if)#default-information originate
```

Related Commands

route-map

default-metric

default-metric *metric*

{ **no** | **default** } **default-metric**

Purpose

Sets the default metric for the Routing Information Protocol (RIP) or RIP next generation (RIPng) instance.

Command Mode

RIPng router configuration

RIP router configuration

Syntax Description

<i>metric</i>	Default metric. The range of values is 0 to 16; the default value is 0.

Default

The metric value is 0.

Usage Guidelines

Use the **default-metric** command to set the default metric for the RIP or RIPng instance. The default value is used when a route with incompatible metrics is received into the RIP or RIPng instance; for example, when a route from a different routing domain is imported into RIP or RIPng.

Use the **no** or **default** form of this command to return the default metric value to 0.

Examples

The following example sets the default metric to 11 for the RIP instance, `rip001`:

```
[local]Redback(config-ctx)#router rip rip001  
[local]Redback(config-rip)#default-metric 11
```

Related Commands

redistribute

distance

distance *distance*

{**no** | **default**} **distance**

Purpose

Modifies the administrative distance for the Routing Information Protocol (RIP) or RIP next generation (RIPng) instance.

Command Mode

RIPng router configuration

RIP router configuration

Syntax Description

<i>distance</i>	Administrative distance. The range of values is 1 to 255; the default value is 120.

Default

The administrative distance is 120.

Usage Guidelines

Use the **distance** command to modify the administrative distance for the RIP or RIPng instance.

Administrative distance specifies how desirable a route obtained from RIP or RIPng is compared to the same route obtained from another protocol. The lower the value for the *distance* argument in comparison to other routes obtained from other protocols, the more desirable the RIP or RIPng route becomes.

Use the **no** or **default** form of this command to return the administrative distance to the default value of 120.

Examples

The following example sets the administrative distance for the `rip001` RIP instance to 200:

```
[local]Redback(config-ctx)#router rip rip001  
[local]Redback(config-rip)#distance 200
```

Related Commands

None

distribute-list

distribute-list prefix *pl-name* {**in** | **out**} [*if-name*]

no distribute-list prefix *pl-name* {**in** | **out**} [*if-name*]

Purpose

Applies a prefix list to Routing Information Protocol (RIP) or RIP next generation (RIPng) packets.

Command Mode

RIPng router configuration

RIP router configuration

Syntax Description

prefix <i>pl-name</i>	Name of the prefix list to be applied to RIP or RIPng packets.
in	Applies the prefix list to incoming RIP or RIPng updates.
out	Applies the prefix list to outgoing RIP or RIPng updates.
<i>if-name</i>	Optional. Name of the interface to which the prefix list is applied.

Default

Prefix lists are not applied.

Usage Guidelines

Use the **distribute-list** command to apply a prefix list to RIP or RIPng packets.

Use the **no** form of this command to remove a prefix list from RIP or RIPng packets.

Examples

The following example applies the prefix list, `list1`, to incoming updates from the `fe01` interface:

```
[local]Redback(config-ctx)#router rip rip001  
[local]Redback(config-rip)#distribute-list prefix list1 in fe01
```

Related Commands

ip prefix-list

flash-update-threshold

flash-update-threshold *seconds*

{**no** | **default**} **flash-update-threshold**

Purpose

Modifies the minimum interval between consecutive Routing Information Protocol (RIP) or RIP next generation (RIPng) flash updates.

Command Mode

RIPng router configuration

RIP router configuration

Syntax Description

seconds

Minimum number of seconds between consecutive RIP or RIPng flash updates. The range of values is 1 to 30; the default value is 5.

Default

The flash update threshold is five seconds.

Usage Guidelines

Use the **flash-update-threshold** command to modify the minimum interval between consecutive RIP or RIPng flash updates. Each flash update contains only those routes that have been changed since the most recent update.

Use the **no** or **default** form of this command to return the threshold limit to five seconds.

Examples

The following example sets a RIP flash update threshold of 10 seconds:

```
[local]Redback(config-ctx)#router rip rip001  
[local]Redback(config-rip)#flash-update-threshold 10
```

Related Commands

None

interface

interface *if-name*

no interface *if-name*

Purpose

In RIP router configuration mode, enables the specified interface to receive and send Routing Information Protocol (RIP) packets for the specified RIP instance, and enters RIP interface configuration mode.

In RIPng router configuration mode, enables the specified interface to receive and send RIP next generation (RIPng) packets for the specified RIPng instance, and enters RIPng interface configuration mode.

Command Mode

RIPng router configuration

RIP router configuration

Syntax Description

<i>if-name</i>	Name of the interface on which RIP or RIPng is to be enabled.

Default

RIP or RIPng are disabled on an interface.

Usage Guidelines

Use the **interface** command (in RIP router configuration mode) to enable the specified interface to receive and send RIP packets for the specified RIP instance, and enter RIP interface configuration mode.

Use the **interface** command (in RIPng router configuration mode) to enable the specified interface to receive and send RIPng packets for the specified RIPng instance, and enter RIPng interface configuration mode.

To enable an interface to send, but not receive RIP or RIPng packets, use the **no listen** command in RIP or RIPng interface configuration mode. To enable an interface to receive, but not send RIP or RIPng packets, use the **no supply** command in RIP or RIPng interface configuration mode.

Use the **no** form of this command to disable RIP or RIPng on the interface.

Examples

The following example enables the `fe0` interface to receive and send RIP packets for the `rip001` instance:

```
[local]Redback(config-ctx)#router rip rip001
[local]Redback(config-rip)#interface fe0
[local]Redback(config-rip-if)#
```

Related Commands

authentication
interface-cost
listen
router rip
router ripng
split-horizon
summary-address
supply

listen

listen

{**no** | **default**} **listen**

Purpose

Enables the specified interface to receive and process Routing Information Protocol (RIP) or RIP next generation (RIPng) packets.

Command Mode

RIP interface configuration
RIPng interface configuration

Syntax Description

This command has no keywords or arguments.

Default

After RIP or RIPng is enabled on an interface using the **interface** command (in RIP or RIPng router configuration mode), by default, the interface can listen to and process RIP or RIPng packets; otherwise, it cannot.

Usage Guidelines

Use the **listen** command to enable the specified interface to receive and process RIP or RIPng packets.

Note This command does not apply to loopback interfaces.

Use the **no** or **default** form of this command to disable the processing of RIP or RIPng packets by an interface.

Examples

The following example enables the fe01 interface to receive and process RIP packets:

```
[local]Redback(config-ctx)#router rip rip002  
[local]Redback(config-rip)#interface fe01  
[local]Redback(config-rip-if)#listen
```

Related Commands

authentication	router ripng
interface —RIP and RIPng router configuration mode	split-horizon
interface-cost	summary-address
router rip	supply

maximum-paths

maximum-paths *path-num*

{ **no** | **default** } **maximum-paths**

Purpose

Modifies the number of multiple equal-cost Routing Information Protocol (RIP) or RIP next generation (RIPng) routes that can be used as the best paths for load balancing outgoing traffic packets.

Command Mode

RIPng router configuration

RIP router configuration

Syntax Description

<i>path-num</i>	Maximum number of equal-cost routes used as the best paths. The range of values is 1 to 16; the default value is 8.

Default

The maximum number of equal-cost routes is 8.

Usage Guidelines

Use the **maximum-paths** command to modify the number of multiple equal-cost RIP or RIPng routes that can be used as the best paths for load balancing outgoing traffic packets. The SmartEdge router enables load balancing among these RIP or RIPng paths if, in the routing table, they are the best paths among paths provided by all running routing protocols.

Use the **no** or **default** form of this command to restore the default setting.

Examples

The following example enables load balancing between two RIP paths for outgoing traffic packets:

```
[local]Redback(config-ctx)#router rip rip001  
[local]Redback(config-rip)#maximum-paths 2
```

Related Commands

None

offset-list

offset-list *pl-name* {**in** | **out**} *offset*

no offset-list *pl-name* {**in** | **out**} *offset*

Purpose

Configure a Routing Information Protocol (RIP) offset list.

Command Mode

RIP router configuration

Syntax Description

<i>pl-name</i>	IP prefix list name.
in	Adds offset to incoming RIP updates.
out	Adds offset to outgoing RIP updates.
<i>offset</i>	Offset value. The range of values is 1 to 16.

Default

No RIP offset list is configured.

Usage Guidelines

Use the **offset-list** command to configure a RIP offset list. A RIP offset list adds to the cost metric of inbound or outbound routes learned or advertised by RIP. RIP offset lists provide a method for adding to the cost metric of routes, which moves the routing switch's route selection away from those routes.

The RIP offset list adds the offset value to the cost metric of all routes that match the specified prefix list.

Use the **no** form of this command to remove the RIP offset list.

Examples

The following example configures a RIP offset list to add 8 to the cost metric for all routes that match the IP prefix list, `foo23`:

```
[local]Redback(config-ctx)#router rip rip001  
[local]Redback(config-rip)#offset-list foo23 in 8
```

Related Commands

None

output-delay

output-delay *delay*

{**no** | **default**} **output-delay**

Purpose

Adds a delay time between packets sent in multipacket Routing Information Protocol (RIP) or RIP next generation (RIPng) updates.

Command Mode

RIPng router configuration

RIP router configuration

Syntax Description

delay

Amount of delay, in milliseconds, added between packets. The range is of values is 1 to 50.

Default

Packets are sent without a delay.

Usage Guidelines

Use the **output-delay** command to add a delay time between packets in multipacket RIP or RIPng updates.

Note This feature is useful for situations where a high-speed router is sending updates to a low-speed router.

Use the **no** or **default** form of this command to disable the delay.

Examples

The following example adds a delay time of 15 milliseconds between the sending of updates for the RIP instance, `rip001`:

```
[local]Redback(config-ctx)#router rip rip001  
[local]Redback(config-rip)#output-delay 15
```

Related Commands

None

redistribute

```
redistribute {bgp asn | connected | isis instance [level-1 | level-2 | level-1-2 ] | nat | ospf instance |  
  rip instance | static [dvsr] | subscriber [address | static]} [metric metric] [route-map map-name]  
  
no redistribute {bgp asn | connected | isis instance | nat | ospf instance | rip instance | static [dvsr] |  
  subscriber [address | static]} [metric metric] [route-map map-name]
```

Purpose

Redistributes routes learned from other routing protocols into the Routing Information Protocol (RIP) or RIP next generation (RIPng) routing instance.

Command Mode

RIPng router configuration
RIP router configuration

Syntax Description

bgp <i>asn</i>	Border Gateway Protocol (BGP) autonomous system number (ASN). Redistributes routes from the specified BGP autonomous system (AS) into the RIP routing instance. The range of values for the <i>asn</i> argument is 1 to 65,535.
connected	Redistributes directly attached networks into the RIP or RIPng routing instance.
isis <i>instance</i>	Intermediate System-to-Intermediate System (IS-IS) instance name. Redistributes routes from the specified IS-IS instance into the RIP or RIPng routing instance.
level-1	Optional. Redistributes IS-IS level 1 routes only.
level-2	Optional. Redistributes IS-IS level 2 routes only.
level-1-2	Optional. Redistributes IS-IS level 1 and level 2 routes.
nat	Redistributes network address translation (NAT) routes into the RIP or RIPng routing instance.
ospf <i>instance</i>	Open Shortest Path First (OSPF) instance ID. Redistributes routes from the specified OSPF routing instance into the RIP or RIPng routing instance. The range of values is 1 to 65,535.
rip <i>instance</i>	RIP or RIPng instance name. Redistributes routes from another RIP or RIPng routing instance into the current RIP or RIPng routing instance.
static	Redistributes static IP routes into the RIP or RIPng routing instance. Optional with the subscriber keyword. Redistributes only static subscriber routes into the RIP routing instance.
dvsr	Optional. Redistributes the dynamically verified static routing (DVSR) subtype of static routes into the RIP or RIPng routing instance.

subscriber	Redistributes routes configured within subscriber records into the RIP or RIPng routing instance.
address	Optional. Redistributes only subscriber address routes into the RIP or RIPng routing instance.
metric <i>metric</i>	Optional. Metric used for the redistributed route. The range of values is 0 to 16. If no metric is specified, the metric configured with the default-metric command is used in RIP or RIPng router configuration mode. If the default-metric command has not been configured, the default metric for redistributed routes is 0.
route-map <i>map-name</i>	Optional. Route map name. Applies the conditions of the specified route map to routes that are redistributed into the RIP or RIPng routing instance.

Default

Redistribution is not enabled.

Usage Guidelines

Use the **redistribute** command to redistribute routes learned from other routing protocols into the RIP or RIPng routing instance.

You must enter multiple **redistribute** commands to redistribute routes from several different kinds of routing protocols into the RIP or RIPng routing instance.

Use the **no** form of this command to disable the specified type of route redistribution.

Examples

The following example redistributes static routes into RIP routing instance, `rip001`:

```
[local]Redback(config-ctx)#router rip rip001
[local]Redback(config-rip)#redistribute static
```

The following example prevents routes from directly attached networks from being redistributed into RIP routing instance, `rip001`:

```
[local]Redback(config-ctx)#router rip rip001
[local]Redback(config-rip)#no redistribute connected
```

Related Commands

default-information originate
default-metric
route-map—context configuration mode

router rip

router rip *instance*

no router rip *instance*

Purpose

Creates an instance of the Routing Information Protocol (RIP) routing process and enters RIP router configuration mode.

Command Mode

context configuration

Syntax Description

instance RIP instance name.

Default

The RIP routing process is disabled.

Usage Guidelines

Use the **router rip** command to create an instance of the RIP routing process and to enter RIP router configuration mode. Each RIP instance has its own routing table. You can configure multiple RIP instances.

To configure a RIP instance on an interface, use the **rip router**, **rip listen**, or **rip supply** command in interface configuration mode.

Use the **no** form of this command to disable an instance of the RIP routing process.

Examples

The following example enables the RIP instance, `rip001`, and enters RIP router configuration mode:

```
[local]Redback(config-ctx)#router rip rip001
[local]Redback(config-rip)#
```

Related Commands

interface

listen

supply

router ripng

router ripng *instance-id*

no router ripng *instance-id*

Purpose

Creates an instance of the Routing Information Protocol next generation (RIPng) routing process and enters RIPng router configuration mode.

Command Mode

context configuration

Syntax Description

instance-id RIPng instance ID.

Default

The RIPng routing process is disabled.

Usage Guidelines

Use the **router ripng** command to create an instance of the RIPng routing process and to enter RIPng router configuration mode. Each RIPng instance has its own routing table. You can configure multiple RIPng instances.

Use the **no** form of this command to disable an instance of the RIPng routing process.

Examples

The following example enables the RIPng instance, `ripng001`, and enters RIPng router configuration mode:

```
[local]Redback(config-ctx)#router ripng ripng001
[local]Redback(config-ripng)#
```

Related Commands

interface
listen
supply

split-horizon

split-horizon [**poison** | **simple**]

{**no** | **default**} **split-horizon**

Purpose

Enables Routing Information Protocol (RIP) or RIP next generation (RIPng) split-horizon processing on the specified interface.

Command Mode

RIP interface configuration

RIPng interface configuration

Syntax Description

poison	Optional. Enables split-horizon processing with poison reverse.
simple	Optional. Enables simple split-horizon processing.

Default

Simple split-horizon processing is enabled.

Usage Guidelines

Use the **split-horizon** command to enable RIP or RIPng split-horizon processing on the specified interface.

Split-horizon processing prevents routing loops in distance-vector routing protocols. When simple split-horizon is enabled, it blocks route information from being advertised out any interface from which the information originated. The split-horizon mechanism is intended to speed up convergence after a link failure.

Split-horizon processing with poisonous reverse can break the loops more quickly by advertising routes with metric infinity (16) to the link from which they are learned.

Note This command does not apply to loopback interfaces.

Use the **no** or **default** form of this command to disable split-horizon processing on the specified interface.

Examples

The following example disables split-horizon processing on the fe01 interface:

```
[local]Redback(config-ctx)#router rip rip002
[local]Redback(config-rip)#interface fe01
[local]Redback(config-rip-if)#no split-horizon
```

Related Commands

authentication

interface—RIP and RIPng router configuration mode

interface-cost

listen

router rip

router ripng

summary-address

supply

summary-address

summary-address {*ip-addr/prefix-length* | *ipv6-addr/prefix-length*} [**metric** *metric*]

{**no** | **default**} **summary-address** {*ip-addr/prefix-length* | *ipv6-addr/prefix-length*} [**metric** *metric*]

Purpose

Summarizes information about Routing Information Protocol (RIP) or RIP next generation (RIPng) routes sent over the specified interface in RIP or RIPng update packets.

Command Mode

RIP interface configuration
RIPng interface configuration

Syntax Description

<i>ip-addr/prefix-length</i>	Specifies the IP address, in the form <i>A.B.C.D</i> , and the prefix length, separated by the slash (/) character. The range of values for the <i>prefix-length</i> argument is 0 to 32.
<i>ipv6-addr/prefix-length</i>	Specifies the IP Version 6 (IPv6) address, in the form <i>A:B:C:D:E:F:G:H</i> , and the prefix length, separated by the slash (/) character. The range of values for the <i>prefix-length</i> argument is 0 to 128.
metric <i>metric</i>	Optional. Metric used for the route. The range of values is 1 to 16. If this construct is not used, the value set through the default-metric command (in RIP or RIPng router configuration mode) is used for the route.

Default

Route address ranges are not summarized.

Usage Guidelines

Use the **summary-address** command to summarize information about RIP or RIPng routes sent over the specified interface, thereby reducing the size of the RIP or RIPng update packets.

Use the **no** or **default** form of this command to disable RIP or RIPng route summarization.

Examples

The following example summarizes routes in the 10.0.0.0 255.0.0.0 range:

```
[local]Redback(config-ctx)#router rip rip002
[local]Redback(config-rip)#interface fe01
[local]Redback(config-rip-if)#summary-address 10.0.0.0 255.0.0.0
```

Related Commands

authentication
default-metric
interface—RIP and RIPng router configuration mode
interface-cost
listen
router rip
router ripng
split-horizon
supply

supply

supply

{no | default} supply

Purpose

Enables the sending of Routing Information Protocol (RIP) or RIP next generation (RIPng) packets on the specified interface.

Command Mode

RIP interface configuration

RIPng interface configuration

Syntax Description

This command has no keywords or arguments.

Default

If the interface has been enabled through the **interface** command (in RIP or RIPng router configuration mode), it can transmit RIP or RIPng packets; otherwise, it cannot.

Usage Guidelines

Use the **supply** command to enable the sending of RIP or RIPng packets on the specified interface.

If more than one circuit is bound to the interface, RIP or RIPng updates are not sent.

Note This command does not apply to loopback interfaces.

Use the **no** or **default** form of this command to disable the sending of RIP or RIPng packets on an interface.

Examples

The following example enables the sending of RIP packets on the fe01 interface:

```
[local]Redback(config-ctx)#router rip rip002  
[local]Redback(config-rip)#interface fe01  
[local]Redback(config-rip-if)#supply
```

Related Commands

authentication

default-metric

interface—RIP and RIPng router configuration mode

interface-cost

listen

router rip

router ripng

split-horizon

summary-address

timers basic

timers basic *update-interval invalid-interval holddown-interval flush-interval*

{no | default} **timers basic**

Purpose

Modifies the Routing Information Protocol (RIP) or RIP next generation (RIPng) timers for the specified RIP or RIPng instance or interface.

Command Mode

RIP interface configuration
RIPng interface configuration
RIPng router configuration
RIP router configuration

Syntax Description

<i>update-interval</i>	Interval, in seconds, at which RIP or RIPng updates are sent. The range of values is 1 to 32,767; the default value is 30.
<i>invalid-interval</i>	Interval, in seconds, before a route is declared invalid after no updates are received. This value should be at least three times the value for the <i>update-interval</i> argument. The range of values is 1 to 32,767; the default value is 180.
<i>holddown-interval</i>	Interval, in seconds, before better routes are released. The range of values is 1 to 32,767; the default value is 180.
<i>flush-interval</i>	Interval, in seconds, before a route is flushed from the routing table. This value must be larger than the value for the <i>invalid-interval</i> argument. The range of values is 1 to 32,767; the default value is 240.

Default

RIP and RIPng updates are sent every 30 seconds, a route is declared invalid if an update is not received after 180 seconds, better routes are released after 180 seconds, and a route is flushed from the routing table after 240 seconds.

Usage Guidelines

Use the **timers basic** command in RIP or RIPng interface configuration mode to modify the RIP or RIPng timers for the specified interface.

Use the **timers basic** command in RIP or RIPng router configuration mode to modify the RIP or RIPng timers for the specified instance.

Use the **no** or **default** form of this command to restore the default RIP or RIPng timer settings.

Examples

The following example sets the RIP timers for the RIP instance `rip001`. The update interval is set to 45 seconds, the invalid interval to 200 seconds, the holddown interval to 200 seconds, and the flush interval to 260 seconds.

```
[local]Redback(config-ctx)#rip001  
[local]Redback(config-rip)#timers basic 45 200 200 260
```

The following example modifies the RIP timers for the `fe01` interface. The update interval is set to 45 seconds, the invalid interval to 200 seconds, the holddown interval to 200 seconds, and the flush interval to 260 seconds:

```
[local]Redback(config-ctx)#router rip rip002  
[local]Redback(config-rip)#interface fe01  
[local]Redback(config-rip-if)#timers basic 45 200 200 260
```

Related Commands

None

OSPF Configuration

This chapter provides an overview of the Open Shortest Path First (OSPF) and describes the tasks and commands used to configure OSPF features through the SmartEdge® OS.

For information about the tasks and commands used to monitor, troubleshoot, and administer OSPF, see the “OSPF Operations” chapter in the *Routing Protocols Operations Guide* for the SmartEdge OS.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples
- Command Descriptions

Overview

OSPF is an Interior Gateway Protocol (IGP) that uses link-state advertisements (LSAs) to inform other routers of the state of the sender’s links. In a link-state routing protocol, each router distributes information about its interfaces and neighbor relationships. The collection of the link states of individual routers forms a database that describes the autonomous system (AS) topology. As OSPF routers accumulate link-state information, they use the Shortest Path First (SPF) algorithm to calculate the shortest path to each node, which forms the basis for developing routing information for that autonomous system.

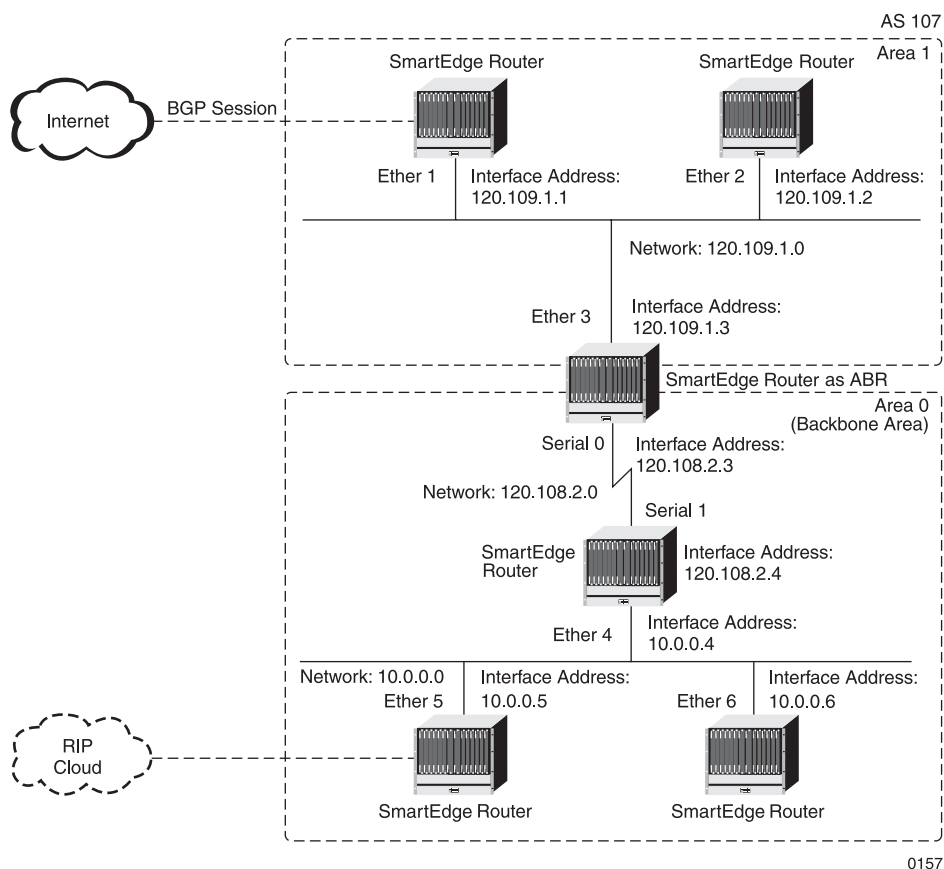
Redback® Networks supports multiple OSPF features, including those specified in the following IETF drafts and RFCs:

- RFC 2328, *OSPF Version 2*
- RFC 1587, *The OSPF NSSA Option*
- RFC 2370, *The OSPF Opaque LSA Option*
- RFC 1793, *Extending OSPF to support Demand Circuits*
- Internet Draft, *Hitless OSPF Restart*, draft-ietf-ospf-hitless-restart-04.txt
- Internet Draft, *Traffic Engineering Extensions to OSPF Version 2*, draft-katz-yeung-ospf-traffic-09.txt
- Internet Draft, *OSPF as the PE/CE Protocol in BGP/MPLS VPNs*, draft-rosen-vpn-ospf-bgp-mpls-05.txt

- Internet Draft, *OSPF Area 0 PE/CE Links in BGP/MPLS VPNs*, draft-rosen-ppvpn-ospf2547-area0-01.txt
- Internet Draft, *Point-to-point Operation over LAN in Link-State Routing Protocols*, draft-ietf-isis-igp-p2p-over-lan-01.txt

In OSPF, the autonomous system can be hierarchically organized by partitioning it into areas; see Figure 6-1.

Figure 6-1 OSPF Hierarchy



Externally derived routes, also called AS-external routes, are routes learned from other routing protocols that are redistributed into the OSPF routing process. These AS-external routes are advertised to all areas, except for stub areas and not-so-stubby-areas (NSSAs). AS-external routes can also be forwarded out to another AS through routers on its boundary.

In-depth information on how OSPF is structured, and how it operates, is described in the following sections:

- Areas
- Router Functions
- Route Selection Process
- Packet Types
- Link-State Advertisements

- Sham Links
- Virtual Links

Areas

Each area can contain a group of contiguous networks and hosts. An area border router (ABR) communicates routing information between the areas.

Because routers within the same area share the same information, they have identical topological databases. An area's topology is invisible to entities outside the area. By keeping area topologies separate, OSPF passes less routing traffic than it would if an autonomous system were not partitioned.

Area partitioning creates two different types of OSPF routing, depending on whether the source and destination are in the same or different areas. Intra-area routing occurs when the source and destination are in the same area; interarea routing occurs when they are in different areas.

The different area types are described in the following sections:

- Normal and Backbone
- Stub
- Not-So-Stubby-Area

Normal and Backbone

A normal OSPF area, including the backbone area, is distinguished by the fact that it can carry transit traffic, allowing LSAs from outside the autonomous system (type 5 AS-external-LSAs) to be flooded throughout the area. Type 5 AS-external-LSAs can be originated both by routers internal to the area or by ABRs.

Hierarchical organization of an OSPF autonomous system requires one of the areas to be configured as the backbone area. The backbone area is configured with an identity of 0 and must be contiguous, contain all area border routers, and be responsible for distributing routing information to all other nonbackbone areas.

Stub

OSPF also allows some areas to be configured as stub areas. Type 5 AS-external LSAs are not flooded into a stub area, thereby reducing the link state database size and the processor and memory usage of routers inside stub areas. While a stub area cannot propagate routes external to the autonomous system in which it resides, it can propagate a default route, intra-area routes, and interarea routes. A stub area relies on default routing to forward traffic addressed to external destinations. The backbone area cannot be configured as a stub area.

Not-So-Stubby-Area

Not-so-stubby-areas (NSSAs) are an extension of OSPF stub areas. Their intent is to preserve the properties of a stub area, while allowing limited import of external routes from other routing domains. These routes are imported as Type 7 NSSA-external LSAs, which are flooded only within the NSSA. For propagation of these routes to other areas, type 7 LSAs must be translated into type 5 external LSAs by the NSSA ABR. NSSA ABRs will also advertise a type 7 default route into the NSSA, and can be configured to summarize and to filter the translation of type 7 NSSA-external LSAs into Type 5 external LSAs.

Router Functions

Depending on its location in the OSPF hierarchy, an OSPF router can provide one or more of the following functions:

- **Internal router**—A router with all directly connected networks belonging to the same area. An internal router maintains a single topological database.
- **Backbone router**—A router that has one or more interfaces to the backbone area. The OSPF backbone is responsible for distributing routing information between areas.
- **ABR**—A router that attaches to multiple areas. ABRs maintain a separate topological database for each attached area and summarize the information for distribution to the backbone. The backbone in turn distributes the information to the other areas.
- **ASBR**—An autonomous system border router (ASBR) exchanges routing information with routers belonging to other autonomous systems, and advertises external routing information throughout its local autonomous system. The paths to each AS boundary router are known by every router in the autonomous system. ASBRs can be internal or area border routers, and may or may not participate in the backbone. ASBRs cannot be part of a stub area unless they are also ABRs; that is, connected to other non-stub areas.
- **Designated router and backup designated router**—On multi-access networks with more than one router, a designated router is responsible for generating the LSAs for the network. The designated router is elected by the Hello protocol. Designated routers allow a reduction in network traffic and in the size of the topological database. Backup designated routers provide a failsafe in case the designated router is not operational.

Route Selection Process

A routing table contains all the information necessary to forward an IP packet to a destination. When forwarding an IP data packet, the routing table entry providing the best match for the packet's IP destination is located. In the case of OSPF, the best path to a destination is determined via the SPF computation performed on the link-state database.

From the link-state database, the router uses the Dijkstra algorithm to construct a tree of shortest paths with itself as root. This shortest-path tree gives the route to each destination in the autonomous system. A separate SPF computation is performed and a different tree is constructed for each area in which the router resides. Externally derived routing information appears on the tree as leaves. OSPF intra-area and inter-area paths are preferred over external paths.

Packet Types

OSPF runs directly on top of IP (protocol 89). There are five types of packets specified in OSPF:

- **Hello**—The SmartEdge router sends Hello packets to its neighbors and receives their Hello packets. In this manner, adjacencies between neighbors are established. (Not all neighboring routers are adjacent.)
- **Database description**—Sent by adjacent routers when an adjacency is initialized, database description packets describe the contents of the respective database to synchronize the two neighboring databases.

- **Link-state request**—Requests pieces of the topological database from neighbor routers. These messages are sent after a router discovers (by examining database-description packets) that parts of its topological database are out of date.
- **Link-state update**—Responds to a link-state request packet. These messages are also used for the regular flooding of LSAs. Several LSAs can be included within a single link-state update packet.
- **Link-state acknowledgment**—Acknowledges link-state update packets.

Each packet includes a common header; see Figure 6-2.

Figure 6-2 OSPF Packet Header

Version Number	Type	Packet Length	Router ID	Area ID	Checksum	Authentication Type	Authentication	Data

0544

The OSPF packet header contains the following fields:

- **Version Number** —Identifies the OSPF version.
- **Type**—Identifies the OSPF packet type; for example, Hello, database description, link-state request, link-state update, and link-state acknowledgement.
- **Packet Length**—Specifies the packet length, including the OSPF header, in bytes.
- **Router ID** —Identifies the source of the packet.
- **Area ID** —Identifies the area to which the packet belongs. A packet is associated with a single area.
- **Checksum**—Checks the entire packet contents for any damage that may have occurred in transit.
- **Authentication Type**—Contains the authentication type. All OSPF protocol exchanges are authenticated. The authentication type is configurable on a per-area basis.
- **Authentication** —Contains authentication information.
- **Data**—Contains packet data.

Link-State Advertisements

Table 6-1 provides each LSA type and its description.

Table 6-1 LSA Type and Description

ID	Type	Description
1	Router-LSA	Originated by all routers. Describes the collected states of the router's interfaces to an area. Flooded throughout a single area only.
2	Network-LSA	Originated by the designated router. Contains the list of routers connected to the network. Flooded throughout a single area only.
3	Summary-LSA (networks)	Flooded throughout a single area only. Describes routes to networks. Each summary LSA describes a route to a destination outside the area, but still inside the autonomous system.

Table 6-1 LSA Type and Description (*continued*)

ID	Type	Description
4	Summary-LSA (routers)	Flooded throughout a single area only. Describes routes to ASBRs. Each summary LSA describes a route to a destination outside the area, but still inside the autonomous system.
5	AS-external-LSA	Originated by ASBRs and flooded throughout the autonomous system. Each AS-external LSA describes a route to a destination in another autonomous system. Default routes for the AS can also be described by AS-external LSAs.
7	NSSA-external-LSA	Flooded throughout a single area only. Type 7 LSAs are advertised only within an NSSA. When forwarded outside the NSSA to nonstub areas, Type 7 LSAs are converted into Type 5 LSAs by an ABR configured to perform translation, or by the ABR with the highest router ID. ABRs can be configured to summarize and filter Type 7 LSAs.
9	Link local scope opaque LSA	Type 9 Opaque LSAs are not flooded beyond the local (sub)network.
10	Area local scope opaque LSA	Type 10 Opaque LSAs are not flooded beyond the borders of their associated area.
11	AS scope opaque LSA	The flooding scope of Type 11 LSAs are equivalent to the flooding scope of AS-external (Type 5) LSAs. Specifically, Type 11 Opaque LSAs are: • Flooded throughout all transit areas • Not flooded into stub areas from the backbone • Not originated by routers into their connected stub areas

Sham Links

A sham link is an OSPF adjacency tunneled over a VPN backbone. Sham links allow the VPN backbone path to be preferred when there are intra-area backdoor links between customer edge (CE) routers in the VPN.

The local connected route corresponding to the source IP address for the sham link must be redistributed into BGP and advertised over the VPN infrastructure to a provider edge (PE) router containing the other end of the sham link.

The route corresponding the remote end of the sham link must be redistributed into the corresponding OSPF instance in the VPN context. VPN routing must be enabled for the OSPF instance.

The cost of the sham link can be configured or will inherit the BGP Multi-Exit Discriminator (MED) from the VPN route.

For more information on sham links, see the Internet Draft, *OSPF as the PE/CE Protocol in BGP/MPLS VPNs*, draft-rosen-vpns-ospf-bgp-mpls-04.txt.

Virtual Links

The single backbone area (0.0.0.0) cannot be disconnected, or some areas of the autonomous system will become unreachable. To establish and maintain connectivity of the backbone, virtual links can be configured through non-backbone areas. Virtual links serve to connect physically separate components of the backbone. The two endpoints of a virtual link are area border routers. The virtual link must be configured in both routers. The configuration information in each router consists of the other virtual endpoint (the other area border router), and the non-backbone area the two routers have in common, which is called the transit area. Virtual links cannot be configured through stub areas.

The virtual link is treated as if it were an unnumbered point-to-point network belonging to the backbone and joining the two area border routers. An attempt is made to establish an adjacency over the virtual link. When this adjacency is established, the virtual link is included in backbone router LSAs, and OSPF packets pertaining to the backbone area flow over the virtual adjacency.

In each endpoint router, the cost and viability of the virtual link is discovered by examining the routing table entry for the other endpoint router. An InterfaceUp event occurs for a virtual link when its corresponding routing table entry becomes reachable, and an InterfaceDown event occurs when its routing table entry becomes unreachable.

The other details concerning virtual links are as follows:

- AS-external-LSAs are NEVER flooded over virtual adjacencies.
- The cost of a virtual link is not configured.
- The IP interface address for the virtual interface and the virtual neighbor's IP address are determined by the routing table build process.
- In each endpoint's router-LSA for the backbone, the virtual link is represented as a Type 4 link whose link ID is set to the virtual neighbor's OSPF router ID and whose link data is set to the virtual interface's IP address.
- A non-backbone area can carry transit data traffic only if it serves as the transit area for one or more fully adjacent virtual links.
- The time between link state retransmissions, is configured for a virtual link.

For more information on virtual links, see RFC 2328, *OSPF Version 2*.

OSPFv3

OSPF Version 3 (OSPFv3) is the version of OSPF that supports IP Version 6 (IPv6). The fundamental mechanisms of OSPF (flooding, area support, and Shortest Path First [SPF] calculations) remain unchanged in OSPFv3; however, because of changes in protocol semantics between IPv4 and IPv6, or simply to handle the increased address size of IPv6, the following changes have been made in OSPFv3:

- Addressing semantics has been removed from OSPF packets and basic LSAs.
- New LSAs exist to carry IPv6 addresses and prefixes.
- OSPFv3 runs on a per-link basis, instead of on a per-IP-subnet basis.
- Flooding scope for LSAs has been generalized.
- Authentication has been removed from OSPFv3; it is now handled by the IPv6 authentication header and encapsulating security payload.

OSPFv3 also supports all optional OSPF capabilities, including on-demand circuits, NSSA areas, and multicast extensions.

For a description of IPv6 addressing and the types of IPv6 addresses, see RFC 3513, *Internet Protocol Version 6 (IPv6) Addressing Architecture*.

Note When IP Version 6 (IPv6) addresses are not referenced or explicitly specified, the term, IP address, can refer generally to IP Version 4 (IPv4) addresses, IPv6 addresses, or IP addressing. In instances where IPv6 addresses are referenced or explicitly specified, the term, IP address, refers only to IPv4 addresses.

Configuration Tasks

Note In this section, the command syntax in the task tables displays only the root command; for the complete command syntax, see the full description for the command in the “Command Descriptions” section.

To configure OSPF or OSPFv3, perform the tasks described in the following sections:

- Configuring OSPF
- Configuring OSPFv3

Configuring OSPF

To configure OSPF, perform the tasks described in the following sections:

- Configure an OSPF Routing Instance
- Configure the Redistribution of Routes into OSPF
- Configure an OSPF Area
- Configure an OSPF Interface
- Configure an OSPF Sham Link
- Configure an OSPF Virtual Link

Configure an OSPF Routing Instance

To configure an OSPF routing instance, perform the tasks described in Table 6-2. Enter all commands in OSPF router configuration mode, unless otherwise noted.

Table 6-2 Configure an OSPF Routing Instance

Task	Root Command	Notes
Create an OSPF routing instance and enter OSPF router configuration mode.	router ospf	Enter this command in context configuration mode.
Specify that the OSPF interface cost is computed automatically and to configure the reference bandwidth that is used in the interface cost computation.	auto-cost	The interface cost is computed by dividing the reference bandwidth by the interface speed. A cost of one is assigned if the interface speed is greater than the reference bandwidth. You can override the automatic cost setting on individual interfaces by issuing the cost command in OSPF interface configuration mode. For more information, see the “Configure an OSPF Interface” section.
Enable the advertisement of router capabilities using OSPF opaque LSAs.	capabilities	
Configure a default metric that is used for redistributed OSPF routes when no metric is specified.	default-metric	

Table 6-2 Configure an OSPF Routing Instance (*continued*)

Task	Root Command	Notes
Modify the OSPF distance value of one or more of these route types.	distance	The distance value of a route is used to select the preferred route when there are equivalent routes from multiple protocols. When a distance comparison is made the route with the lowest distance is selected. By default, OSPF external, inter-area, and intra-area routes are set to a distance value of 110.
Enable OSPF fast LSA origination for an OSPF instance.	fast-lsa-origination	Normally, OSPF originates an LSA every five seconds. Because there can be multiple changes to router or network LSAs during that five-second interval, the five-second LSA origination limit can slow network convergence. When fast LSA origination is enabled, up to four instances of the same LSA can be originated in the same five-second interval. Likewise, LSA reception is normally rate limited to one new LSA instance per second. LSA instances received in less than the one second after the previous LSA instance are dropped. When fast LSA origination is enabled, LSA reception is not restricted to one new instance per second.
Enable graceful restart for an OSPF instance.	graceful-restart	
Log neighbor transitions to and from the full neighbor adjacency state.	log-neighbor-up-down	
Enable the use of MPLS LSPs as intra-area next hops.	mpls shortcuts	
Enable the advertisement of OSPF Traffic Engineering (TE) metrics.	mpls traffic-engineering	
Originate the default route advertisement in the OSPF routing domain.	originate-default	
Configure a fixed OSPF router ID for the SmartEdge router.	router-id	The router ID is used by OSPF to identify the originating router for packets and link-state advertisements (LSAs). If the OSPF router ID is not configured, OSPF chooses the lowest loopback interface address. If there are no loopback interfaces, OSPF chooses the lowest interface address. The default OSPF router ID is selected when OSPF is started initially or restarted using the process restart ospf command in exec mode.
Configure the delay time between the receipt of a topology change and the start of the Shortest Path First (SPF) calculation, and to determine the hold time between two consecutive SPF calculations.	spf-timers	
Configure the SmartEdge router as an OSPF stub router.	stub-router	
Configure the redistribution of routes into the OSPF routing instance.		For the complete list of tasks used to configure the redistribution of routes into the OSPF routing instance, see the "Configure the Redistribution of Routes into OSPF" section.
Configure an OSPF area.		For the complete list of tasks used to configure an OSPF area, see the "Configure an OSPF Area" section.

Configure the Redistribution of Routes into OSPF

You can redistribute routes learned from other protocols into the OSPF routing instance, set a limit on the number of routes that can be redistributed into the OSPF routing instance, and set a limit on the number of routes per second that can be redistributed into the OSPF routing instance.

To configure the redistribution of routes into the OSPF routing instance, perform the tasks described in Table 6-3. Enter all commands in OSPF router configuration mode.

Table 6-3 Configure the Redistribution of Routes into OSPF

Task	Root Command	Notes
Redistribute routes learned from other protocols into the OSPF routing instance.	redistribute	
Set a maximum limit on the number of routes that can be redistributed into the specified OSPF instance.	maximum redistribute	
Set a maximum limit on the number of routes that can be redistributed per second into the OSPF routing instance.	maximum redistribute-quantum	
Summarize external routes that are redistributed into the OSPF routing instance.	summary-address	

Configure an OSPF Area

To configure an OSPF area, perform the tasks described in Table 6-4. Enter all commands in OSPF area configuration mode, unless otherwise noted.

Table 6-4 Configure an OSPF Area

Task	Root Command	Notes
Create an OSPF area and enter OSPF area configuration mode.	area	Enter this command in OSPF router configuration mode.
Define an OSPF area as a stub area or as an NSSA.	area-type	
Change the attributes of a default route originated into a stub area or an NSSA.	default-route	
Summarize NSSA routes advertised by an ABR.	nssa-range	
Summarize interarea routes advertised by an ABR.	range	
Configure an OSPF interface.	For the complete list of tasks used to configure an OSPF interface, see the "Configure an OSPF Interface" section.	
Configure an OSPF sham link.	For the complete list of tasks used to configure an OSPF sham link, see the "Configure an OSPF Sham Link" section.	
Configure an OSPF virtual link.	For the complete list of tasks used to configure an OSPF virtual link, see the "Configure an OSPF Virtual Link" section.	

Configure an OSPF Interface

To configure an OSPF interface, perform the tasks described in Table 6-5. Enter all commands in OSPF interface configuration mode, unless otherwise noted.

Table 6-5 Configure an OSPF Interface

Task	Root Command	Notes
Enable OSPF routing on an interface and enter OSPF interface configuration mode.	interface	Enter this command in OSPF area configuration mode.
Enable authentication and specify the authentication scheme for an OSPF interface.	authentication	Routes within the same area are not required to use the same authentication scheme and key ID; however, if two routers directly exchange updates, they must have the same authentication scheme and key ID.
Block the flooding of LSAs that are not self-originated.	block-flooding	Blocking flooding on an interface can result in inconsistencies between OSPF routers and their respective route tables. Exercise caution before blocking the flooding of LSAs that are not self-originated.
Configure the cost used in SPF computation for the specified OSPF-enabled interface.	cost	The lower the cost, the more likely the interface is to be used to forward data traffic.
Configure OSPF to treat a point-to-point (P2P) or a point-to-multipoint (P2MP) interface as a demand circuit.	demand-circuit	Demand circuits are network segments whose costs vary with usage; charges can be based both on connect time and on bytes or packets transmitted. OSPF routing usually requires a demand circuit's underlying data-link connection to be constantly open, resulting in unwanted usage charges. Using the demand-circuit command enables OSPF Hello packets and the refresh of OSPF routing information to be suppressed on demand circuits, allowing the underlying data-link connections to be closed when not carrying traffic. Hello suppression is not negotiated unless demand circuit support is enabled.
Enable the sending of more than one OSPF Hello packet per second on the interface.	fast-hello	Using this command results in faster OSPF convergence. The following restrictions apply to this command: - After the fast-hello command is configured, you cannot use the hello-interval or router-dead interval commands until the fast-hello command has been disabled. - After the hello-interval or router-dead interval command has been configured, you cannot use the fast-hello command until the hello-interval or router-dead interval command has been disabled.
Suppress the periodic LSA refresh in stable topologies.	flood-reduction	If demand circuit operation is implicitly or explicitly enabled, LSAs are flooded as DoNotAge LSAs on the OSPF interface, and will not be re-flooded until the network topology changes.
Configure the interval at which OSPF hello packets are sent on the interface.	hello-interval	
Configure an OSPF neighbor.	neighbor	

Table 6-5 Configure an OSPF Interface (*continued*)

Task	Root Command	Notes
Configure the OSPF network type.	network-type	You can specify any of the following network types: • Broadcast network—Broadcast networks support multiple routers and have the ability to address a single physical message to all attached routers. • Nonbroadcast multiaccess (NBMA)—A nonbroadcast network, such as frame relay, that simulates an OSPF broadcast network. • Point-to-point (P2P) network—A P2P network joins a single pair of routers. • Point-to-multipoint (P2MP) network—Acts as though the nonbroadcast network is a collection of P2P links.
Disable normal OSPF operation on an interface while still advertising the interface's IP subnet as an intra-area stub network in the OSPF routing domain.	passive	
Modify the interval at which LSAs are retransmitted in link state update packets on an interface.	retransmit-interval	
Modify the amount of time the OSPF routing process waits to receive an OSPF Hello packet from a neighbor before determining that the neighbor is not operational.	router-dead-interval	
Modify the OSPF preference value for the SmartEdge router to act as the designated router on the network.	router-priority	
Set a delay value, increasing the age of LSAs sent out through the OSPF interface.	transmit-delay	

Configure an OSPF Sham Link

To configure an OSPF sham link, perform the tasks described in Table 6-6. Enter all commands in OSPF sham link configuration mode, unless otherwise noted.

Table 6-6 Configure an OSPF Sham Link

Task	Root Command	Notes
Create an OSPF adjacency tunneled over a VPN backbone (sham link).	sham-link	Enter this command in OSPF area configuration mode.
Enable authentication and specify the authentication scheme for an OSPF sham link.	authentication	Routes within the same area are not required to use the same authentication scheme and key ID; however, if two routers directly exchange updates, they must have the same authentication scheme and key ID.
Configure the cost used in SPF computation for the an OSPF sham link.	cost	The lower the cost, the more likely the sham link is to be used to forward data traffic.
Configure the interval at which OSPF hello packets are sent out through an OSPF sham link.	hello-interval	
Modify the interval at which LSAs are retransmitted in link state update packets on an OSPF sham link.	retransmit-interval	

Table 6-6 Configure an OSPF Sham Link (*continued*)

Task	Root Command	Notes
Modify the amount of time the OSPF routing process waits to receive an OSPF Hello packet from a neighbor before determining that the neighbor is not operational.	router-dead-interval	
Set a delay value, increasing the age of LSAs sent out through an OSPF sham link.	transmit-delay	

Configure an OSPF Virtual Link

To configure an OSPF virtual link, perform the tasks described in Table 6-7. Enter all commands in OSPF virtual link configuration mode, unless otherwise noted.

Table 6-7 Configure an OSPF Virtual Link

Task	Root Command	Notes
Create a virtual link through the specified transit area.	virtual-link	Enter this command in OSPF area configuration mode.
Enable authentication and specify the authentication scheme for an OSPF virtual link.	authentication	Routes within the same area are not required to use the same authentication scheme and key ID; however, if two routers directly exchange updates, they must have the same authentication scheme and key ID.
Configure the interval at which OSPF hello packets are sent out through an OSPF virtual link.	hello-interval	
Modify the interval at which LSAs are retransmitted in link state update packets on an OSPF virtual link.	retransmit-interval	
Modify the amount of time the OSPF routing process waits to receive an OSPF Hello packet from a neighbor before determining that the neighbor is not operational.	router-dead-interval	
Set a delay value, increasing the age of LSAs sent out through an OSPF virtual link.	transmit-delay	

Configuring OSPFv3

To configure OSPFv3, perform the tasks described in the following sections:

- Configure an OSPFv3 Routing Instance
- Configure the Redistribution of Routes into OSPFv3
- Configure an OSPFv3 Area
- Configure an OSPFv3 Interface
- Configure an OSPF Virtual Link

Configure an OSPFv3 Routing Instance

To configure an OSPFv3 routing instance, perform the tasks described in Table 6-8. Enter all commands in OSPF3 router configuration mode, unless otherwise noted.

Table 6-8 Configure an OSPFv3 Routing Instance

Task	Root Command	Notes
Create an OSPFv3 routing instance and enter OSPF3 router configuration mode.	router ospf3	Enter this command in context configuration mode.
Specify that the OSPFv3 interface cost is computed automatically and to configure the reference bandwidth that is used in the interface cost computation.	auto-cost	The interface cost is computed by dividing the reference bandwidth by the interface speed. A cost of one is assigned if the interface speed is greater than the reference bandwidth. You can override the automatic cost setting on individual interfaces by issuing the cost command in OSPFv3 interface configuration mode. For more information, see the "Configure an OSPFv3 Interface" section.
Configure a default metric that is used for redistributed OSPFv3 routes when no metric is specified.	default-metric	
Modify the OSPFv3 distance value of one or more of these route types.	distance	The distance value of a route is used to select the preferred route when there are equivalent routes from multiple protocols. When a distance comparison is made the route with the lowest distance is selected. By default, OSPFv3 external, inter-area, and intra-area routes are set to a distance value of 110.
Enable graceful restart for an OSPFv3 instance.	graceful-restart	
Log neighbor transitions to and from the full neighbor adjacency state.	log-neighbor-up-down	
Originate the default route advertisement in the OSPFv3 routing domain.	originate-default	
Configure a fixed OSPFv3 router ID for the SmartEdge router.	router-id	The router ID is used by OSPFv3 to identify the originating router for packets and link-state advertisements (LSAs). If the OSPFv3 router ID is not configured, OSPFv3 chooses the lowest loopback interface address. If there are no loopback interfaces, OSPFv3 chooses the lowest interface address. The default OSPFv3 router ID is selected when OSPFv3 is started initially or restarted using the process restart ospf command in exec mode.
Configure the delay time between the receipt of a topology change and the start of the Shortest Path First (SPF) calculation, and to determine the hold time between two consecutive SPF calculations.	spf-timers	
Configure the SmartEdge router as an OSPFv3 stub router.	stub-router	
Configure the redistribution of routes into the OSPFv3 routing instance.		For the complete list of tasks used to configure the redistribution of routes into the OSPFv3 routing instance, see the "Configure the Redistribution of Routes into OSPF" section.
Configure an OSPFv3 area.		For the complete list of tasks used to configure an OSPFv3 area, see the "Configure an OSPFv3 Area" section.

Configure the Redistribution of Routes into OSPFv3

You can redistribute routes learned from other protocols into the OSPFv3 routing instance, set a limit on the number of routes that can be redistributed into the OSPFv3 routing instance, and set a limit on the number of routes per second that can be redistributed into the OSPFv3 routing instance.

To configure the redistribution of routes into the OSPFv3 routing instance, perform the tasks described in Table 6-9. Enter all commands in OSPF3 router configuration mode.

Table 6-9 Configure the Redistribution of Routes into OSPFv3

Task	Root Command	Notes
Redistribute routes learned from other protocols into the OSPFv3 routing instance.	redistribute	
Set a maximum limit on the number of routes that can be redistributed into the specified OSPFv3 instance.	maximum redistribute	
Set a maximum limit on the number of routes that can be redistributed per second into the OSPFv3 routing instance.	maximum redistribute-quantum	
Summarize external routes that are redistributed into the OSPFv3 routing instance.	summary-address	

Configure an OSPFv3 Area

To configure an OSPFv3 area, perform the tasks described in Table 6-10. Enter all commands in OSPF3 area configuration mode, unless otherwise noted.

Table 6-10 Configure an OSPFv3 Area

Task	Root Command	Notes
Create an OSPFv3 area and enter OSPF3 area configuration mode.	area	Enter this command in OSPF3 router configuration mode.
Define an OSPFv3 area as a stub area or as an NSSA.	area-type	
Change the attributes of a default route originated into a stub area or an NSSA.	default-route	
Summarize NSSA routes advertised by an ABR.	nssa-range	
Summarize interarea routes advertised by an ABR.	range	
Configure an OSPFv3 interface.	For the complete list of tasks used to configure an OSPF interface, see the "Configure an OSPFv3 Interface" section.	

Configure an OSPFv3 Interface

To configure an OSPFv3 interface, perform the tasks described in Table 6-11. Enter all commands in OSPF3 interface configuration mode, unless otherwise noted.

Table 6-11 Configure an OSPFv3 Interface

Task	Root Command	Notes
Enable OSPFv3 routing on an interface and enter OSPF3 interface configuration mode.	interface	Enter this command in OSPF3 area configuration mode.
Block the flooding of LSAs that are not self-originated.	block-flooding	Blocking flooding on an interface can result in inconsistencies between OSPFv3 routers and their respective route tables. Exercise caution before blocking the flooding of LSAs that are not self-originated.
Configure the cost used in SPF computation for the specified OSPFv3-enabled interface.	cost	The lower the cost, the more likely the interface is to be used to forward data traffic.
Configure OSPFv3 to treat a P2P or a P2MP interface as a demand circuit.	demand-circuit	Demand circuits are network segments whose costs vary with usage; charges can be based both on connect time and on bytes or packets transmitted. OSPFv3 routing usually requires a demand circuit's underlying data-link connection to be constantly open, resulting in unwanted usage charges. Using the demand-circuit command enables OSPFv3 Hello packets and the refresh of OSPFv3 routing information to be suppressed on demand circuits, allowing the underlying data-link connections to be closed when not carrying traffic. Hello suppression is not negotiated unless demand circuit support is enabled.
Enable the sending of more than one OSPFv3 Hello packet per second on the interface.	fast-hello	Using this command results in faster OSPFv3 convergence. The following restrictions apply to this command: • After the fast-hello command is configured, you cannot use the hello-interval or router-dead interval commands until the fast-hello command has been disabled. • After the hello-interval or router-dead interval command has been configured, you cannot use the fast-hello command until the hello-interval or router-dead interval command has been disabled.
Suppress the periodic LSA refresh in stable topologies.	flood-reduction	If demand circuit operation is implicitly or explicitly enabled, LSAs are flooded as DoNotAge LSAs on the OSPFv3 interface, and will not be re-flooded until the network topology changes.
Configure the interval at which OSPFv3 hello packets are sent on the interface.	hello-interval	
Configure an OSPFv3 neighbor.	neighbor	
Configure the OSPFv3 network type.	network-type	You can specify any of the following network types: • Broadcast network—Broadcast networks support multiple routers and have the ability to address a single physical message to all attached routers. • Nonbroadcast multiaccess (NBMA)—A nonbroadcast network, such as frame relay, that simulates an OSPFv3 broadcast network. • Point-to-point (P2P) network—A P2P network joins a single pair of routers. • Point-to-multipoint (P2MP) network—Acts as though the nonbroadcast network is a collection of P2P links.

Table 6-11 Configure an OSPFv3 Interface (*continued*)

Task	Root Command	Notes
Disable normal OSPFv3 operation on an interface while still advertising the interface's IP subnet as an intra-area stub network in the OSPFv3 routing domain.	passive	
Modify the interval at which LSAs are retransmitted in link-state update packets on an interface.	retransmit-interval	
Modify the amount of time the OSPFv3 routing process waits to receive an OSPFv3 Hello packet from a neighbor before determining that the neighbor is not operational.	router-dead-interval	
Modify the OSPFv3 preference value for the SmartEdge router to act as the designated router on the network.	router-priority	
Set a delay value, increasing the age of LSAs sent out through the OSPFv3 interface.	transmit-delay	

Configure an OSPF Virtual Link

To configure an OSPFv3 virtual link, perform the tasks described in Table 6-12. Enter all commands in OSPF3 virtual link configuration mode, unless otherwise noted.

Table 6-12 Configure an OSPFv3 Virtual Link

Task	Root Command	Notes
Create an OSPFv3 virtual link through the specified transit area.	virtual-link	Enter this command in OSPF3 area configuration mode.
Configure the interval at which OSPFv3 hello packets are sent out through an OSPFv3 virtual link.	hello-interval	
Modify the interval at which LSAs are retransmitted in link state update packets on an OSPFv3 virtual link.	retransmit-interval	
Modify the amount of time the OSPFv3 routing process waits to receive an OSPFv3 Hello packet from a neighbor before determining that the neighbor is not operational.	router-dead-interval	
Set a delay value, increasing the age of LSAs sent out through an OSPFv3 virtual link.	transmit-delay	

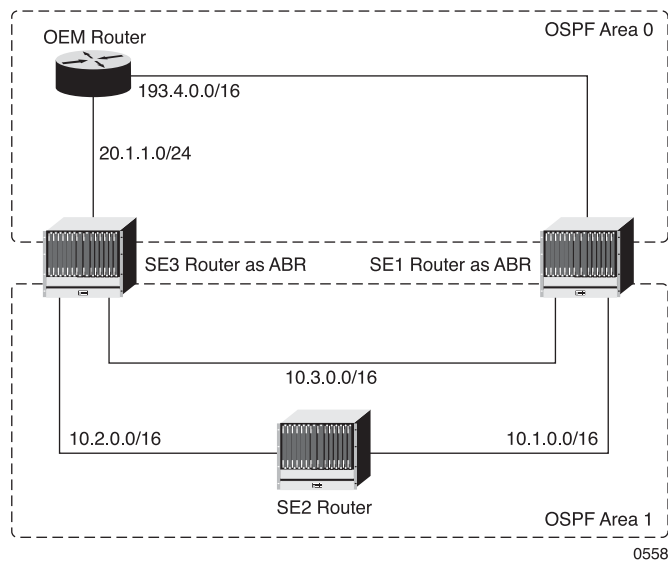
Configuration Examples

This section provides OSPF configuration examples in the following sections:

- Basic OSPF
- Redistribution
- MD5 Authentication
- Simple Key Chain

Figure 6-3 illustrates the base OSPF topology for the examples provided in this section.

Figure 6-3 OSPF Topology



Basic OSPF

This section contains the basic OSPF configuration for the three SmartEdge routers (SE1, SE2, and SE3) illustrated in Figure 6-3. Examples in proceeding sections contain only the configuration sections different from the examples here.

The basic configuration for SE1 is as follows. Because no router ID is explicitly configured, the loopback address is used as the OSPF router ID for SE1.

```
[local]SE1(config)#context local
[local]SE1(config-ctx)#ip domain-lookup
[local]SE1(config-ctx)#interface one
[local]SE1(config-if)#ip address 193.4.5.2/16
[local]SE1(config-if)#exit
[local]SE1(config-ctx)#interface two
[local]SE1(config-if)#ip address 10.1.1.1/16
[local]SE1(config-if)#exit
[local]SE1(config-ctx)#interface three
[local]SE1(config-if)#ip address 10.3.1.1/16
[local]SE1(config-if)#exit
[local]SE1(config-ctx)#interface lo1 loopback
[local]SE1(config-if)#ip address 193.10.25.7/32
[local]SE1(config-if)#exit
[local]SE1(config-ctx)#router ospf 1
[local]SE1(config-ospf)#area 0.0.0.0
[local]SE1(config-ospf-area)#interface 193.4.5.2
[local]SE1(config-ospf-if)#exit
[local]SE1(config-ospf-area)#interface 193.10.25.7
[local]SE1(config-ospf-area)#exit
```

```

[local]SE1(config-ospf)#area 0.0.0.1
[local]SE1(config-ospf-area)#interface two
[local]SE1(config-ospf-if)#exit
[local]SE1(config-ospf-area)#interface three
[local]SE1(config-ospf-if)#exit
[local]SE1(config-ospf-area)#exit
[local]SE1(config-ospf)#exit
[local]SE1(config-ctx)#exit
[local]SE1(config)#port pos 5/1
[local]SE1(config-port)#bind interface one local
[local]SE1(config-port)#no shutdown
[local]SE1(config-port)#exit
[local]SE1(config)#port pos 5/2
[local]SE1(config-port)#bind interface two local
[local]SE1(config-port)#no shutdown
[local]SE1(config-port)#exit
[local]SE1(config)#port pos 5/3
[local]SE1(config-port)#bind interface three local
[local]SE1(config-port)#no shutdown

```

The basic configuration for SE2 is as follows:

```

[local]SE2(config)#context local
[local]SE2(config-ctx)#ip domain-lookup
[local]SE2(config-ctx)#interface one
[local]SE2(config-if)#ip address 10.1.2.2/16
[local]SE2(config-if)#exit
[local]SE2(config-ctx)#interface two
[local]SE2(config-if)#ip address 10.2.1.1/16
[local]SE2(config-if)#exit
[local]SE2(config-ctx)#router ospf 1
[local]SE2(config-ospf)#router-id 22.22.22.22
[local]SE2(config-ospf)#area 0.0.0.1
[local]SE2(config-ospf-area)#interface 10.1.2.2
[local]SE2(config-ospf-if)#exit
[local]SE2(config-ospf-area)#interface 10.2.1.1
[local]SE2(config-ospf-if)#exit
[local]SE2(config-ospf-area)#exit
[local]SE2(config-ospf)#exit
[local]SE2(config-ctx)#exit
[local]SE2(config)#port pos 3/1
[local]SE2(config-port)#bind interface one local
[local]SE2(config-port)#no shutdown
[local]SE2(config-port)#exit
[local]SE2(config)#port ethernet 4/1
[local]SE2(config-port)#bind interface two local
[local]SE2(config-port)#no shutdown

```

The basic configuration for SE3 is as follows:

```

[local]SE3(config)#context local
[local]SE3(config-ctx)#ip domain-lookup
[local]SE3(config-ctx)#interface one

```

```
[local]SE3(config-if)#ip address 10.3.2.2/16
[local]SE3(config-if)#exit
[local]SE3(config-ctx)#interface two
[local]SE3(config-if)#ip address 10.2.2.2/16
[local]SE3(config-if)#exit
[local]SE3(config-ctx)#interface three
[local]SE3(config-if)#ip address 20.1.1.1/24
[local]SE3(config-if)#exit
[local]SE3(config-ctx)#router ospf 1
[local]SE3(config-ospf)#router-id 33.33.33.33
[local]SE3(config-ospf)#area 0.0.0.0
[local]SE3(config-ospf-area)#interface 20.1.1.1
[local]SE3(config-ospf-if)#exit
[local]SE3(config-ospf-area)#exit
[local]SE3(config-ospf)#area 0.0.0.1
[local]SE3(config-ospf-area)#interface 10.2.2.2
[local]SE3(config-ospf-if)#exit
[local]SE3(config-ospf-area)#interface 10.3.2.2
[local]SE3(config-ospf-if)#exit
[local]SE3(config-ospf-area)#exit
[local]SE3(config-ospf)#exit
[local]SE3(config-ctx)#exit
[local]SE3(config)#port pos 3/1
[local]SE3(config-port)#bind interface one local
[local]SE3(config-port)#no shutdown
[local]SE3(config-port)#exit
[local]SE3(config)#port ethernet 1/1
[local]SE3(config-port)#bind interface two local
[local]SE3(config-port)#no shutdown
[local]SE3(config-port)#exit
[local]SE3(config)#port pos 3/2
[local]SE3(config-port)#bind interface three local
[local]SE3(config-port)#no shutdown
```

Redistribution

The following example illustrates how to redistribute static routes into the OSPF routing instance and how to modify the attributes of the redistributed routes. Only the routes matching the 122-nets-only IP prefix list are selected for redistribution. These routes are 122.1.1.0/24, 122.1.2.0/24, and 122.1.3.0/24. Once redistributed to OSPF, the routes are advertised with metric type 1 and metric value of 500. All modifications are accomplished by using the route map, `static-to-ospf`.

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#ip domain-lookup
[local]Redback(config-ctx)#interface one
[local]Redback(config-if)#ip address 10.1.2.2/16
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface two
[local]Redback(config-if)#ip address 10.2.1.1/16
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface three
```

```

[local]Redback(config-if)#ip address 10.5.1.1/30
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#router ospf 1
[local]Redback(config-ospf)#router-id 22.22.22.22
[local]Redback(config-ospf)#area 0.0.0.1
[local]Redback(config-ospf-area)#interface 10.1.2.2
[local]Redback(config-ospf-if)#exit
[local]Redback(config-ospf-area)#interface 10.2.1.1
[local]Redback(config-ospf-if)#exit
[local]Redback(config-ospf-area)#exit
[local]Redback(config-ospf)#redistribute static route-map static-to-ospf
[local]Redback(config-ospf)#exit
[local]Redback(config-ctx)#ip prefix-list 122-nets-only
[local]Redback(config-prefix-list)#seq 10 permit 122.0.0.0/8 le 24
[local]Redback(config-prefix-list)#seq 20 deny 0.0.0.0/0
[local]Redback(config-prefix-list)#exit
[local]Redback(config-ctx)#route-map static-to-ospf permit 10
[local]Redback(config-route-map)#match ip address prefix-list 122-nets-only
[local]Redback(config-route-map)#set metric 500
[local]Redback(config-route-map)#set metric-type type-1
[local]Redback(config-route-map)#exit
[local]Redback(config-ctx)#ip route 50.0.0.0/8 three
[local]Redback(config-ctx)#ip route 121.1.1.0/24 three
[local]Redback(config-ctx)#ip route 121.1.2.0/24 three
[local]Redback(config-ctx)#ip route 121.1.3.0/24 three
[local]Redback(config-ctx)#ip route 121.1.5.0/24 three
[local]Redback(config-ctx)#ip route 122.1.1.0/24 three
[local]Redback(config-ctx)#ip route 122.1.2.0/24 three
[local]Redback(config-ctx)#ip route 122.1.3.0/24 three
[local]Redback(config-ctx)#exit
[local]Redback(config)#port pos 3/1
[local]Redback(config-port)#bind interface one local
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#exit
[local]Redback(config)#port ethernet 4/1
[local]Redback(config-port)#bind interface two local
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#exit
[local]Redback(config)#port pos 3/2
[local]Redback(config-port)#bind interface three local
[local]Redback(config-port)#no shutdown

```

MD5 Authentication

The following example shows how to use MD5 to provide authentication between two SmartEdge routers. Authentication is only configured at the interface level. A different type of authentication can be used on each interface and no area configuration is required.

The configuration for SE1 is as follows:

```

[local]SE1(config-ctx)#router ospf 1
[local]SE1(config-ospf)#area 0.0.0.0

```

```
[local]SE1(config-ospf-area)#interface 193.4.5.2
[local]SE1(config-ospf-if)#exit
[local]SE1(config)#interface 193.10.25.7
[local]SE1(config-ospf-if)#exit
[local]SE1(config-ospf-area)#exit
[local]SE1(config-ospf)#area 0.0.0.1
[local]SE1(config-ospf-area)#interface two
[local]SE1(config-ospf-if)#authentication md5 ospf-key-chain
[local]SE1(config-ospf-if)#exit
[local]SE1(config-ospf-area)#interface three
```

The configuration for SE2 is as follows:

```
[local]SE2(config-ctx)#router ospf 1
[local]SE2(config-ospf)#router-id 22.22.22.22
[local]SE2(config-ospf)#area 0.0.0.1
[local]SE2(config-ospf-area)#interface 10.1.2.2
[local]SE2(config-ospf-if)#authentication md5 ospf-key-chain
[local]SE2(config-ospf-if)#exit
[local]SE2(config-ospf-area)#interface 10.2.1.1
```

Simple Key Chain

This example show how key chain lifetimes can be used to non-disruptively switch from one key string to another. SmartEdge OSPF will always send using the key with the most recent send-lifetime start time which is not greater than the current time. It will accept any key whose accept lifetime value includes the current time.

The configuration for both SE1 and SE2 is as follows:

```
[local]Redback(config-ctx)#key-chain ospf-key-chain key-id 1
[local]Redback(config-key-chain)#key-string secret
[local]Redback(config-key-chain)#accept-lifetime 2001:09:07:00:00:00
2002:09:07:12:00:00
[local]Redback(config-key-chain)#send-lifetime 2001:09:07:00:00:00 2002:09:07:08:00:00
[local]Redback(config-key-chain)#exit
[local]Redback(config-ctx)#key-chain ospf-key-chain key-id 2
[local]Redback(config-key-chain)#key-string psst
[local]Redback(config-key-chain)#accept-lifetime 2002:09:07:00:00:00
2003:09:07:12:00:00
[local]Redback(config-key-chain)#send-lifetime 2002:09:07:08:00:00 2003:09:07:07:00:00
```

Command Descriptions

This section describes the syntax and usage guidelines for the commands used to configure OSPF features. The commands are presented in alphabetical order.

area	mpls traffic-engineering
area-type	neighbor
authentication	network-type
auto-cost	nssa-range
block-flooding	originate-default
capabilities	passive
cost	range
default-metric	redistribute
default-route	retransmit-interval
demand-circuit	router-dead-interval
distance	router-id
fast-hello	router ospf
fast-lsa-origination	router ospf3
flood-reduction	router-priority
graceful-restart	sham-link
hello-interval	spf-timers
interface	stub-router
log-neighbor-up-down	summary-address
maximum redistribute	transmit-delay
maximum redistribute-quantum	virtual-link
mpls shortcuts	

area

area {*area-id* | *ip-addr*}

no area {*area-id* | *ip-addr*}

Purpose

In OSPF router configuration mode, configures an Open Shortest Path First (OSPF) area and enters OSPF area configuration mode.

In OSPF3 router configuration mode, configures an OSPF Version 3 (OSPFv3) area and enters OSPF3 area configuration mode.

Command Mode

OSPF router configuration

OSPF3 router configuration

Syntax Description

area-id 32-bit number. The range of values is 0 to 4,294,967,295. The 0 value is reserved for the backbone area.

ip-addr IP address. The 0.0.0.0 value is reserved for the backbone area.

Default

None

Usage Guidelines

Use the **area** command (in OSPF router configuration mode) to configure an OSPF area and enter OSPF area configuration mode.

Use the **area** command (in OSPF3 router configuration mode) to configure an OSPFv3 area and enter OSPF3 area configuration mode.

Multiple areas are supported per OSPF or OSPFv3 instance. Specify the area ID or IP address for the router to use when participating in OSPF or OSPFv3 routing. All routers in an area must use the same area ID to establish neighbor adjacencies.

To specify that the router is directly connected to the OSPF or OSPFv3 backbone, use the **area 0.0.0.0** or **area 0** construct.

Use the **no** form of this command to remove an OSPF or OSPFv3 area.

Examples

The following example configures an area using an IP address of 34.0.0.0 and enters OSPF router configuration mode:

```
[local]Redback(config-ospf)#area 34.0.0.0
[local]Redback(config-ospf-area)#
```


Related Commands

`area-type`

area-type

```
area-type {nssa [no-redistribution] [no-default] | stub [no-summary]}  
{no | default} area-type
```

Purpose

Defines an Open Shortest Path First (OSPF) or OSPF Version 3 (OSPFv3) area as a stub area or not-so-stubby-area (NSSA).

Command Mode

OSPF area configuration
OSPF3 area configuration

Syntax Description

nssa	Configures the area as an NSSA.
no-redistribution	Optional. Suppresses redistribution of non-OSPF routes by an autonomous system border router (ASBR) into an NSSA area. By default, redistributed routes are advertised using Type 7 link-state advertisements (LSAs).
no-default	Optional. Suppresses NSSA default origination. An NSSA area border router (ABR) normally advertises a type 7 or type 3 default LSA in the NSSA. This keyword suppress the default.
stub	Configures the area as a stub type.
no-summary	Optional. Suppresses the advertisement of Type 3 LSAs, or interarea routes, into a stub area. This option is only relevant when the router is configured as an area border router (ABR).

Default

The area type is normal.

Usage Guidelines

Use the **area-type** command to define an OSPF or OSPFv3 area as a stub area or as an NSSA.

A stub area relies on default routing to forward traffic addressed to external destinations. You cannot configure the backbone as a stub area.

Use the **no** or **default** form of this command to return the specified area to a normal area.

Examples

The following example configures area 4 as a stub area:

```
[local]Redback(config-ospf)#area 4  
[local]Redback(config-ospf-area)#area-type stub
```

Related Commands

`area`
`default-route`

authentication

authentication {**md5** *key-chain-name* | **none** | **simple** *key-chain-name*}
{**no** | **default**} **authentication**

Purpose

Enables authentication and specifies the authentication scheme for the specified interface, sham link, or virtual link.

Command Mode

OSPF interface configuration
OSPF sham link configuration
OSPF virtual link configuration

Syntax Description

md5 <i>key-chain-name</i>	Message Digest 5 (MD5) authentication key chain name.
none	Specifies no authentication.
simple <i>key-chain-name</i>	Simple authentication key chain name.

Default

Authentication is not enabled.

Usage Guidelines

Use the **authentication** command to enable authentication and specify the authentication scheme for the specified interface, sham link, or virtual link.

Key chains allow you to control authentication keys used by various routing protocols in the system. All routers connected to the same IP subnet must use the same authentication scheme and key ID. If multiple key IDs have been configured, the one with the most current send time is used. For information on the **key-chain key-id** command, see the “Key Chain Configuration” chapter in the *IP Services and Security Configuration Guide* for the SmartEdge OS.

Routes within the same area are not required to use the same authentication scheme and key ID. However, if two routers directly exchange updates, they must have the same authentication scheme and key ID.

Use the **no** or **default** form of this command to disable authentication.

Examples

The following example configures MD5 authentication for the interface, 193.4.5.2, and simple authentication for the interface, 10.1.1.1:

```
[local]Redback(config-ctx)#router ospf 1  
[local]Redback(config-ospf)#area 0.0.0.0  
[local]Redback(config-ospf-area)#interface 193.4.5.2
```

```
[local]Redback(config-ospf-if)#authentication md5 auth01
[local]Redback(config-ospf-if)#exit
[local]Redback(config-ospf-area)#exit
[local]Redback(config-ospf)#area 0.0.0.1
[local]Redback(config-ospf-area)#interface 10.1.1.1
[local]Redback(config-ospf-if)#authentication simple auth02
[local]Redback(config-ospf-if)#exit
[local]Redback(config-ospf-area)#exit
[local]Redback(config-ospf)#exit
[local]Redback(config-ctx)#key-chain auth01 keyid 1
[local]Redback(config-key-chain)#key-string secret
[local]Redback(config-key-chain)#exit
[local]Redback(config-ctx)#key-chain auth02 keyid 1
[local]Redback(config-key-chain)#key-string password
```

Related Commands

hello-interval
interface—OSPF area configuration mode
retransmit-interval
router-dead-interval
sham-link
transmit-delay
virtual-link

auto-cost

auto-cost [**reference-bandwidth** *bandwidth*]

no auto-cost

default auto-cost

Purpose

Specifies that the Open Shortest Path First (OSPF) or OSPF Version 3 (OSPFv3) interface cost is computed automatically, and configures the reference bandwidth that is used in the interface cost computation.

Command Mode

OSPF router configuration

OSPF3 router configuration

Syntax Description

reference-bandwidth *bandwidth* Optional. Bandwidth rate in Mbps. The range of values is 1 to 4,294,967; the default value is 100.

Default

The interface cost is computed automatically using a reference bandwidth of 100 Mbps.

Usage Guidelines

Use the **auto-cost** command to specify that the OSPF or OSPFv3 interface cost is computed automatically and to configure the reference bandwidth that is used in the interface cost computation. The interface cost is computed by dividing the reference bandwidth by the interface speed. A cost of one is assigned if the interface speed is greater than the reference bandwidth.

You can override the automatic cost setting on individual interfaces by issuing the cost command the **cost** command in OSPF or OSPF3 interface configuration mode.

Use the **no** form of this command to disable automatic cost computation.

Use the **default** form of this command to return the reference bandwidth to 100 Mbps.

Examples

The following example configures the OSPF bandwidth rate to 64 Mbps:

```
[local]Redback(config-ospf)#auto-cost reference-bandwidth 64
```

Related Commands

cost

interface

block-flooding

block-flooding

no block-flooding

Purpose

Blocks the flooding of link-state advertisements (LSAs) that are not self-originated.

Command Mode

OSPF interface configuration

OSPF3 interface configuration

Syntax Description

This command has no arguments or keywords.

Default

Flooding of LSAs that are not self-originated is not blocked.

Usage Guidelines

Use the **block-flooding** command in highly meshed topologies to block the flooding of LSAs that are not self-originated.



Caution Risk of Open Shortest Path First (OSPF) or OSPF Version 3 (OSPFv3) routing errors. Blocking flooding on an interface can result in inconsistencies between OSPF or OSPFv3 routers and their respective route tables. To reduce the risk, exercise caution before blocking the flooding of LSAs that are not self-originated.

Use the **no** form of this command to remove the LSA flooding block.

Examples

The following example blocks flooding on the OSPF interface, atm-pvc10:

```
[local]Redback(config-ospf)#area 0  
[local]Redback(config-ospf-area)#interface atm-pvc10  
[local]Redback(config-ospf-if)#block-flooding
```

Related Commands

area—OSPF or OSPF3 router configuration mode

interface—OSPF or OSPF3 area configuration mode

capabilities

capabilities {area-scope | as-scope}

no capabilities {area-scope | as-scope}

Purpose

Enables the advertisement of router capabilities using Open Shortest Path First (OSPF) opaque link-state advertisements (LSAs).

Command Mode

OSPF router configuration

Syntax Description

area-scope Advertise router capabilities using Type 10 opaque LSAs.

as-scope Advertise router capabilities using Type 11 opaque LSAs.

Default

Advertisement of router capabilities is disabled.

Usage Guidelines

Use the **capabilities** command to enable the advertisement of router capabilities using OSPF opaque LSAs.

The capabilities LSAs advertise the optional OSPF capabilities enabled on the router to all IGP neighbors. Table 6-13 shows the reserved OSPF router capability bits and the associated capabilities that can be advertised.

Table 6-13 Reserved OSPF Router Capability Bits

Bit	Capability
0–3	Reserved
4	Graceful restart capable
5	OSPF graceful restart helper
6	Stub router support
7	Traffic engineering support
8	OSPF point-to-point over LAN
9	OSPF path computation server discovery
10–31	Future assignments

Use the **no** form of this command to disable advertisement of router capabilities using OSPF opaque LSAs.

Examples

The following example enables the advertisement of router capabilities using Type 10 (area-scope) opaque LSAs:

```
[local]Redback(config-ctx)#router ospf 424  
[local]Redback(config-ospf)#capabilities area-scope
```

Related Commands

None

cost

cost *cost*

{**no** | **default**} **cost**

Purpose

Configures the cost used in Shortest Path First (SPF) computations for the specified interface, or sham link.

Command Mode

OSPF interface configuration
 OSPF sham link configuration
 OSPF3 interface configuration

Syntax Description

<i>cost</i>	Interface or sham link cost. The range of values is 1 to 65,535. By default, the value set by the auto-cost command (in OSPF or OSPF3 router configuration mode) is used. If the auto cost is not configured, the default cost is 1.

Default

If this command is not enabled, the value specified through the **auto-cost** command is used. If the auto cost is not configured, the cost value is 1.

Usage Guidelines

Use the **cost** command to configure the cost used in SPF computation for the specified interface, or sham link.

The lower the cost, the more likely the interface, or sham link, is to be used to forward data traffic. You can assign only one cost per interface.

Use the **no** or **default** form of this command to return the cost to its default value.

Examples

The following example configures cost of 3 for the ospf1 interface:

```
[ local ]Redback(config-ospf)#interface ospf1
[ local ]Redback(config-ospf-if)#cost 3
```

Related Commands

authentication	retransmit-interval
auto-cost	router-dead-interval
hello-interval	sham-link
interface —OSPF or OSPF3 area configuration mode	transmit-delay

default-metric

default-metric *metric*

no default-metric

Purpose

Configures the default metric used for redistributed Open Shortest Path First (OSPF) or OSPF Version 3 (OSPFv3) routes when no metric is specified.

Command Mode

OSPF router configuration

OSPF3 router configuration

Syntax Description

metric Metric value. The range of values is 1 to 16,777,215.

Default

No default metric is configured. If a metric value is not configured through the **redistribute** command in OSPF router configuration mode or applied via a route map, the metric in the system routing table is used.

Usage Guidelines

Use the **default-metric** command to configure the default metric used for redistributed OSPF or OSPFv3 routes when no metric is specified. You can specify a metric through the **redistribute** command (in OSPF or OSPF3 router configuration mode), or indirectly by applying a route map through the **route-map** command (in route map configuration mode).

Use the **no** form of this command to return the metric value to its default setting.

Examples

The following example configures a default metric value of 40:

```
[local]Redback(config-ospf)#default-metric 40
```

Related Commands

redistribute

route-map

default-route

default-route [**metric** *metric*] [**metric-type** *type*]

no default-route

Purpose

Changes the attributes of a default route originated into a stub area or a not-so-stubby-area (NSSA).

Command Mode

OSPF area configuration

OSPF3 area configuration

Syntax Description

metric <i>metric</i>	Optional. Metric value for the default route. The range of values is 1 to 1,677,214; the default value is 1.
metric-type <i>type</i>	Optional. External route metric type for a Type 5 default link-state advertisement (LSA). The <i>type</i> argument specifies one of the following metric types: • 1—Specifies a Type 1 metric type. • 2—Specifies a Type 2 metric type.

Default

The metric value for the default route is 1. For stub areas, a Type 3 LSA with a metric value of 1 is advertised. The metric type is ignored. For NSSAs that import summary advertisements, a Type 7 LSA with a metric value of 1 and a route metric type of 2 is advertised. For NSSAs that do not import summary advertisements, a Type 3 LSA with a metric value of 1 is advertised. The metric type is ignored.

Usage Guidelines

Use the **default-route** command to change the attributes of a default route originated into a stub area or NSSA. The LSA advertising the default route depends on the area type and whether or not summary advertisements (Type 3 and 4 LSAs) are advertised into the area.

For stub areas, a Type 3 LSA with a metric value of 1 is advertised by default. The **default-route** command can be used to modify the metric. The metric type is ignored.

For NSSAs that import summary advertisements, a Type 7 LSA with a metric value of 1 and route metric type of 2 is advertised by default. The **default-route** command can be used to modify the metric or metric type.

For NSSAs that do not import summary advertisements, a Type 3 LSA with a metric value of 1 is advertised by default. The **default-route** command can be used to modify the metric. The metric type is ignored.

If there are two routers originating a default route with the same metric value, the closest router is chosen to perform routing.

Use the **no** form of this command to restore the default attributes for the originated default route.

Examples

The following example configures a default route metric value of 3:

```
[local]Redback(config-ospf-area)#default-route metric 3
```

Related Commands

- area**
- area-type**
- neighbor**
- network-type**
- nssa-range**
- range**

demand-circuit

demand-circuit

no demand-circuit

Purpose

Configures Open Shortest Path First (OSPF) or OSPF Version 3 (OSPFv3) to treat a point-to-point (P2P) or point-to-multipoint (P2MP) interface as a demand circuit as described in RFC 1793, *Extending OSPF to Support Demand Circuits*.

Command Mode

OSPF interface configuration

OSPF3 interface configuration

Syntax Description

This command has no arguments or keywords.

Default

Demand circuit support is disabled on P2P and P2MP interfaces. Demand circuit support is implicitly enabled on virtual links and sham links.

Usage Guidelines

Use the **demand-circuit** command to configure OSPF or OSPFv3 to treat a P2P or P2MP interface as a demand circuit, as described in RFC 1793, *Extending OSPF to Support Demand Circuits*.

Demand circuits are network segments whose costs vary with usage; charges can be based both on connect time and on bytes or packets transmitted. OSPF or OSPFv3 routing usually requires a demand circuit's underlying data-link connection to be constantly open, resulting in unwanted usage charges. Using the **demand-circuit** command enables OSPF or OSPFv3 Hello packets and the refresh of OSPF or OSPFv3 routing information to be suppressed on demand circuits, allowing the underlying data-link connections to be closed when not carrying traffic.

Note Hello suppression is not negotiated unless demand circuit support is enabled.

Use the **no** form of this command to remove the demand circuit designation.

Examples

The following example configures the OSPF interface POS1/2 in area 0 to be a demand circuit:

```
[local]Redback(config-ospf)#area 0
[local]Redback(config-ospf-area)#interface POS1/2
[local]Redback(config-ospf-if)#demand-circuit
```

Related Commands

`area`
`interface`
`router ospf`

distance

distance [**external** *distance*] [**inter-area** *distance*] [**intra-area** *distance*]
 {**no** | **default**} **distance** [**external** *distance*] [**inter-area** *distance*] [**intra-area** *distance*]

Purpose

Modifies the Open Shortest Path First (OSPF) or OSPF Version 3 (OSPFv3) distance value of one or more route types.

Command Mode

OSPF router configuration
 OSPF3 router configuration

Syntax Description

external <i>distance</i>	Optional. OSPF or OSPFv3 distance for external routes. The range of values is 10 to 255; the default value is 110.
inter-area <i>distance</i>	Optional. OSPF or OSPFv3 distance for interarea routes. The range of values is 10 to 255; the default value is 110.
intra-area <i>distance</i>	Optional. OSPF or OSPFv3 distance for intraarea routes. The range of values is 10 to 255; the default value is 110.

Default

Each distance is set to 110.

Usage Guidelines

Use the **distance** command to modify the OSPF or OSPFv3 distance value of one or more route types. OSPF and OSPFv3 use distances to compare and prioritize routes. The lower the distance, the more preferred the route. When you enter this command without any optional keywords, the distance for all route types are set to 110.

Use the **no** or **default** form of this command to return the values to their default settings.

Examples

The following example sets the OSPF distance for external routes to 120:

```
[local]Redback(config-ospf)#distance external 120
```

Related Commands

None

fast-hello

fast-hello count-per-second *count*

no fast-hello

default fast-hello

Purpose

Enables the sending of more than one Open Shortest Path First (OSPF) or OSPF Version 3 (OSPFv3) Hello packet per second on the interface.

Command Mode

OSPF interface configuration

OSPF3 interface configuration

Syntax Description

count-per-second *count* Number of OSPF or OSPFv3 Hello packets to be sent on the specified interface each second. The range of values is 2 to 5.

Default

Four OSPF Hello packets are sent each second.

Usage Guidelines

Use the **fast-hello** command to enable the sending of more than one OSPF or OSPFv3 Hello packet per second on the interface.

Note Using the **fast-hello** command results in faster OSPF convergence.

The following restrictions apply to the **fast-hello** command:

- After the **fast-hello** command is configured, you cannot use the **hello-interval** or **router-dead interval** command until the **fast-hello** command has been disabled.
- After the **hello-interval** or **router-dead interval** command has been configured, you cannot use the **fast-hello** command until the **hello-interval** or **router-dead interval** command has been disabled.

Use the **no** form of this command to disable the sending of more than one OSPF or OSPFv3 Hello packet per second on the interface.

Use the **default** form of this command to send four OSPF or OSPFv3 Hello packets each second.

Examples

The following example configures Hello packets to be sent 2 times per second, indicating that the interval between Hello packets to 500 ms:

```
[local]Redback(config-ospf-if)#fast-hello 2
```

Related Commands

area
fast-lsa-origination
hello-interval
interface
router-dead-interval
router ospf

fast-lsa-origination

fast-lsa-origination

{no | default} **fast-lsa-origination**

Purpose

Enables fast link-state advertisement (LSA) origination for an Open Shortest Path First (OSPF) instance.

Command Mode

OSPF router configuration

Syntax Description

This command has no arguments or keywords.

Default

Fast LSA origination is disabled.

Usage Guidelines

Use the **fast-lsa-origination** command to enable fast LSP origination for an OSPF instance.

Normally, OSPF originates an LSA every five seconds. Because there can be multiple changes to router or network LSAs during that five-second interval, the five-second LSA origination limit can slow network convergence. When fast LSA origination is enabled, up to four instances of the same LSA can be originated in the same five-second interval.

Likewise, LSA reception is normally rate limited to one new LSA instance per second. LSA instances received in less than the one second after the previous LSA instance are dropped. When fast LSA origination is enabled, LSA reception is not restricted to one new instance per second.

Use the **no** or **default** form of this command to disable fast LSA origination.

Examples

The following example enables fast LSA origination:

```
[local]Redback(config-ctx)#router ospf 1  
[local]Redback(config-ospf)#fast-lsa origination
```

Related Commands

fast-hello

flood-reduction

flood-reduction

no flood-reduction

Purpose

Suppresses periodic link-state advertisement (LSA) refresh in stable topologies.

Command Mode

OSPF interface configuration

OSPF3 interface configuration

Syntax Description

This command has no arguments or keywords.

Default

Flood reduction is disabled on the interface.

Usage Guidelines

Use the **flood-reduction** command to suppress periodic LSA refresh in stable topologies.

Note If demand circuit operation is implicitly or explicitly enabled, LSAs are flooded as DoNotAge LSAs on the Open Shortest Path First (OSPF) or OSPF Version 3 (OSPFv3) interface, and will not be re-flooded until the network topology changes.

Use the **no** form of this command to disable flood reduction.

Examples

The following example suppresses periodic LSA refresh for the OSPF interface, ETH3/4, in area 0:

```
[local]Redback(config-ospf)#area 0  
[local]Redback(config-ospf-area)#interface ETH3/4  
[local]Redback(config-ospf-if)#flood-reduction
```

Related Commands

area—OSPF or OSPF3 router configuration mode

interface—OSPF and OSPF3 area configuration mode

router ospf

router ospf3

graceful-restart

graceful-restart [*interval* | **helper** [**strict-checking**]]

no graceful-restart [*interval* | **helper** [**strict-checking**]]

Purpose

Enables graceful restart for the Open Shortest Path First (OSPF) or OSPF Version 3 (OSPFv3) instance. When the OSPF or OSPFv3 instance is restarted, it attempts to restart gracefully, consistent with RFC 3623, *Graceful OSPF Restart*.

Command Mode

OSPF router configuration

OSPF3 router configuration

Syntax Description

<i>interval</i>	Optional. Grace period, in seconds. During this time, the OSPF or OSPFv3 instance attempts to restart gracefully. The range of values is 10 to 900; the default value is 120.
helper	Optional. Enables OSPF helper mode.
strict-checking	Optional. Disables OSPF helper mode on an LSA change.

Default

Graceful restart is disabled.

Usage Guidelines

Use the **graceful-restart** command to enable an OSPF or OSPFv3 instance to attempt to restart gracefully after a planned or unplanned restart (crash). This implies that the forwarding state will be maintained while OSPF or OSPFv3 reestablishes its neighbor adjacencies and recalculate its routes. It also implies that the OSPF or OSPFv3 instance will advertise its intent to restart gracefully to its neighbors. The OSPF or OSPFv3 instance will discontinue graceful restart when all of its prior OSPF or OSPFv3 adjacencies have been established or when the grace period expires.

Use the **no** form of this command to disable graceful restart.

Examples

The following example enables an OSPF instance to restart gracefully, and discontinues graceful restart when it determines graceful restart has been completed successfully, or when the grace period of 60 seconds has expired:

```
[local]Redback(config-ospf)#graceful-restart 60
```

Related Commands

router ospf

router ospf3

hello-interval

hello-interval *interval*

{**no** | **default**} **hello-interval**

Purpose

Configures the interval at which Open Shortest Path First (OSPF) or OSPF Version 3 (OSPFv3) Hello packets are sent out through the specified interface, sham link, or virtual link.

Command Mode

OSPF interface configuration
OSPF sham link configuration
OSPF virtual link configuration
OSPF3 interface configuration

Syntax Description

<i>interval</i>	Interval, in seconds, between Hello packets. The range of values is 1 to 65,535; the default value is 10. This value must be the same for all devices that attempt to establish adjacencies over a shared subnet.

Default

The default interval between Hello packets is 10 seconds for broadcast and point-to-point (P2P) interfaces, and 30 seconds for point-to-multipoint (P2MP) and nonbroadcast multiaccess (NBMA) networks.

Usage Guidelines

Use the **hello-interval** command to configure the interval at which OSPF or OSPFv3 Hello packets are sent out through the specified interface, sham link, or virtual link.

Hello packets are sent at a fixed interval on all interfaces, sham links, and virtual links to establish and maintain neighbor relationships. This interval must be the same on all OSPF or OSPFv3 routers on an IP subnet. The smaller the Hello interval, the faster topological changes are detected; however, a smaller interval results in additional traffic.

The following restrictions apply to the **hello-interval** command:

- After the **fast-hello** command is configured, you cannot use the **hello-interval** command until the **fast-hello** command has been disabled.
- After the **hello-interval** command has been configured, you cannot use the **fast-hello** command until the **hello-interval** command has been disabled.

Use the **no** or **default** form of this command to return the interval to its default setting of 10 seconds.

Examples

The following example sets the interval between Hello packets to 12 seconds:

```
[local]Redback(config-ospf-if)#hello-interval 12
```

Related Commands

authentication

cost

fast-hello

interface—OSPF and OSPF3 area configuration mode

retransmit-interval

router-dead-interval

router-priority

sham-link

transmit-delay

virtual-link

interface

interface {*if-name* | *ip-addr*}

no interface {*if-name* | *ip-addr*}

Purpose

In OSPF area configuration mode, enables Open Shortest Path First (OSPF) routing on a specified interface and enters OSPF interface configuration mode.

In OSPF3 area configuration mode, enables OSPF Version 3 (OSPFv3) routing on a specified interface and enters OSPF3 interface configuration mode.

Command Mode

OSPF area configuration

OSPF3 area configuration

Syntax Description

<i>if-name</i>	Interface name.
<i>ip-addr</i>	IP address of the interface.

Default

None

Usage Guidelines

Use the **interface** command (in OSPF area configuration mode) to enable OSPF routing on a specified interface, and to enter OSPF interface configuration mode.

Use the **interface** command (in OSPF3 area configuration mode) to enable OSPFv3 routing on a specified interface, and to enter OSPF3 interface configuration mode.

OSPF or OSPFv3 routing must be enabled on at least one interface. That interface must already be configured through the **interface** command (in context configuration mode).

An OSPF or OSPFv3 interface can connect to a:

- Broadcast network—Supports more than two attached routers and have the ability to address a single physical message to all attached routers.
- Point-to-point (P2P) network—Joins a single pair of routers.
- Nonbroadcast multi-access (NBMA)—a network topology supporting a full mesh of routers; however, there is no capability for sending a single message to all routers.
- Point-to-multipoint (P2MP) network—Acts as though the nonbroadcast network is a collection of P2P links.
- Loopback interface—An interface that is not bound to any circuit.

Use the **no** form of this command to disable OSPF routing on the specified interface.



Caution Risk of lost or down OSPF or OSPFv3 interfaces. If an interface is configured using an IP address and that IP address is deleted, the corresponding OSPF or OSPFv3 interface is deleted. If an interface is configured using an interface name and that interface name is deleted, the corresponding OSPF or OSPFv3 interface is deleted. However, if an interface is configured using an interface name and its primary IP address is changed, the OSPF or OSPFv3 interface continues normal operation using the new primary IP address. If an OSPF or OSPFv3 interface is configured using an interface name and its primary address is deleted, the OSPF or OSPFv3 interface is forced to the down state. To reduce the risk, avoid deleting an OSPF or OSPFv3 interface's IP address.

Examples

The following example enables OSPF routing on the interface at IP address, 192.30.200.10:

```
[local]Redback(config-ospf-area)#interface 192.30.200.10  
[local]Redback(config-ospf-if)#
```

Related Commands

```
router ospf  
router ospf3
```

log-neighbor-up-down

log-neighbor-up-down

no log-neighbor-up-down

Purpose

Logs an informational message when a neighbor transitions to or from the full adjacency state.

Command Mode

OSPF router configuration

OSPF3 router configuration

Syntax Description

This command has no keywords or arguments.

Default

Transitions are not logged.

Usage Guidelines

Use the **log-neighbor-up-down** command to log an informational message when a neighbor transitions to or from the full adjacency state.

Use the **no** form of this command to disable the logging of messages for neighbor transition events.

Examples

The following example logs neighbor transitions:

```
[local]Redback(config-ospf)#log-neighbor-up-down
```

Related Commands

neighbor

maximum redistribute

maximum redistribute *prefixes* [**retry-interval** *interval*]

no maximum redistribute

Purpose

Sets a maximum limit on the number of routes that can be redistributed into the specified Open Shortest Path First (OSPF) or OSPF Version 3 (OSPFv3) instance.

Command Mode

OSPF router configuration

OSPFv3 router configuration

Syntax Description

<i>prefixes</i>	Maximum number of routes that can be redistributed into the OSPF or OSPFv3 routing instance. The range of values is 1 to 100,000.
retry-interval <i>interval</i>	Optional. Amount of time, in minutes, before OSPF or OSPFv3 attempts to redistribute routes after the maximum prefix value is exceeded. The range of values is 1 to 120.

Default

There is no maximum limit for the number of routes that can be redistributed.

Usage Guidelines

Use the **maximum redistribute** command to set a maximum limit on the number of routes that can be redistributed into the specified OSPF or OSPFv3 instance.

If the maximum number of redistributed prefixes is reached, OSPF or OSPFv3 stops redistributing external routes for the duration specified by the *interval* argument.

Use the **no** form of this command to return to the default setting, which is an unlimited number of routes.

Examples

The following example limits redistribution of routes into the OSPF routing instance, 650 to 5000:

```
[local]Redback(config-ctx)#router ospf 650  
[local]Redback(config-ospf)#maximum redistribute 5000
```

Related Commands

maximum redistribute-quantum
redistribute

maximum redistribute-quantum

maximum redistribute-quantum *prefixes*

no maximum redistribute-quantum

Purpose

Sets a maximum limit on the number of routes that can be redistributed per second into the Open Shortest Path First (OSPF) or OSPF Version 3 (OSPFv3) instance.

Command Mode

OSPF router configuration

OSPFv3 router configuration

Syntax Description

prefixes

Maximum number of routes that can be redistributed per second into the OSPF or OSPFv3 routing instance. The range of values is 1 to 10,000; the default value is 2,000.

Default

The maximum number of routes that can be redistributed per second into the OSPF or OSPFv3 routing instance is 2,000.

Usage Guidelines

Use the **maximum redistribute-quantum** command to set a maximum limit on the number of routes that can be redistributed per second into the OSPF or OSPFv3 routing instance.

Use the **no** form of this command to return the limit to its default value of 2,000 routes per second.

Examples

The following example set the maximum number of routes that can be redistributed per second into the OSPF routing instance 30 to 1000:

```
[local]Redback(config-ctx)#router ospf 30  
[local]Redback(config-ospf)#maximum redistribute-quantum 1000
```

Related Commands

maximum redistribute
redistribute

mpls shortcuts

mpls shortcuts

Purpose

Enables the use of Multiprotocol Label Switching (MPLS) label-switched paths (LSPs) as intra-area next hops.

Command Mode

OSPF router configuration

Syntax Description

This command has no keywords or arguments.

Default

The use of MPLS LSPs is disabled.

Usage Guidelines

Use the **mpls shortcuts** command to enable the use of MPLS LSPs as intra-area next hops.

Examples

The following example enables the use of MPLS LSPs as intra-area next hops:

```
[local]Redback(config-ctx)#router ospf  
[local]Redback(config-ospf)#mpls shortcuts
```

Related Commands

None

mpls traffic-engineering

mpls traffic-engineering

Purpose

Enables Open Shortest Path First (OSPF) advertisement of traffic engineering metrics.

Command Mode

OSPF router configuration

Syntax Description

This command has no keywords or arguments.

Default

The use of Multiprotocol Label Switching (MPLS) traffic engineering is disabled.

Usage Guidelines

Use the **mpls traffic engineering** command to cause OSPF to advertise traffic engineering metrics for OSPF interfaces.

Examples

The following example enables the use of MPLS traffic engineering:

```
[local]Redback(config-ctx)#router ospf  
[local]Redback(config-ospf)#mpls traffic-engineering
```

Related Commands

None

neighbor

neighbor {*ip-addr* | *ipv6-addr*} [**cost** *cost*] [**poll-interval** *interval*] [**router-priority** *priority*]

no neighbor {*ip-addr* | *ipv6-addr*} [**cost** *cost*] [**poll-interval** *interval*] [**router-priority** *priority*]

Purpose

Configures an Open Shortest Path First (OSPF) or OSPF Version 3 (OSPFv3) neighbor.

Command Mode

OSPF interface configuration
OSPF3 interface configuration

Syntax Description

<i>ip-addr</i>	OSPF neighbor IP address in the form <i>A.B.C.D</i> .
<i>ipv6-addr</i>	OSPFv3 neighbor IP Version 6 (IPv6) address in the form <i>A:B:C:D:E:F:G</i> .
cost <i>cost</i>	Optional. Cost to reach the neighbor. This cost overrides the interface cost set through the cost command (in OSPF or OSPF3 interface configuration mode). The range of values is 1 to 65,535; the default value is 1.
poll-interval <i>interval</i>	Optional. Interval, in seconds, at which the neighbor is polled when it is unreachable or down. The range of values is 1 to 65,535; the default value is 120.
router-priority <i>priority</i>	Optional. Priority setting for the neighbor. The range of values is 0 to 255; the default value is 1.

Default

If a cost value is not specified, the value set through the **cost** command is used; otherwise, the cost is 1. The poll interval is 120 seconds; the router priority is 1.

Usage Guidelines

Use the **neighbor** command to configure an OSPF or OSPFv3 neighbor.

You can only use the **router-priority** *priority* construct for nonbroadcast multiaccess (NBMA) networks when designated and backup routers are elected.

Use the **no** form of this command to remove a neighbor configuration.

Examples

The following example sets a cost of 10 for the neighbor at IP address 193.12.3.2:

```
[local]Redback(config-ospf-if)#neighbor 193.12.3.2 cost 10
```

Related Commands

network-type

network-type

network-type {**broadcast** | **non-broadcast** | **point-to-point** | **point-to-multipoint**}

no network-type

Purpose

Configures the Open Shortest Path First (OSPF) or OSPF Version 3 (OSPFv3) network type.

Command Mode

OSPF interface configuration

OSPFv3 interface configuration

Syntax Description

broadcast	Specifies that the interface is attached to a broadcast network.
non-broadcast	Specifies that the interface is attached to a nonbroadcast network.
point-to-point	Specifies that the interface is attached to a point-to-point (P2P) network.
point-to-multipoint	Specifies that the interface is attached to a point-to-multipoint (P2MP) network.

Default

The media type determines the network type; for example, an Ethernet interface defaults to the broadcast type.

Usage Guidelines

Use the **network-type** command to configure an OSPF or OSPFv3 network type. You can specify a:

- Broadcast network—Broadcast networks support multiple routers and have the ability to address a single physical message to all attached routers.
- Nonbroadcast multiaccess (NBMA)—A nonbroadcast network, such as X.25, that simulates an OSPF or OSPFv3 broadcast network.
- P2P network—A P2P network joins a single pair of routers.
- P2MP network—Acts as though the nonbroadcast network is a collection of P2P links.

Use the **no** form of this command to return the network type to its default value.

Examples

The following example configures the network type as a broadcast network:

```
[local]Redback(config-ospf-if)#network-type broadcast
```

Related Commands

neighbor

nssa-range

nssa-range *ip-addr* {*netmask* | /*prefix-length*} [**not-advertise** | **tag** *tag*]

no nssa-range *ip-addr* {*netmask* | /*prefix-length*} [**not-advertise** | **tag** *tag*]

Purpose

Summarizes not-so-stubby-area (NSSA) routes advertised by an area border router (ABR).

Command Mode

OSPF area configuration

OSPF3 area configuration

Syntax Description

<i>ip-addr</i>	IP address in the form <i>A.B.C.D</i> .
<i>netmask</i>	Network mask in the form <i>E.F.G.H</i> .
<i>prefix-length</i>	Prefix length. The range of values is 0 to 32.
not-advertise	Optional. Prevents all routes in the specified range from being advertised in interarea route summarizations.
tag <i>tag</i>	Optional. Route tag included in translated external route summarization Type 5 link-state advertisements (LSAs). An unsigned 32-bit integer, the range of values is 1 to 4,294,967,295; the default value is 0.

Default

Address ranges for NSSA route summarization are not specified.

Usage Guidelines

Use the **nssa-range** command to summarize NSSA routes advertised by an ABR. This command is used for NSSA-translated external route summarization and is only relevant when the router is configured as an ABR.

Use the optional **not-advertise** keyword to prevent the specified route from being advertised in translated external route summarizations.

Use the **no** form of this command to disable route summarization for a particular summary range. All individual routes contained in the summary range are advertised to other areas.

Examples

The following example sends routes that fall into the range 10.1.0.0 255.255.0.0 as a single autonomous system (AS) external advertisement:

```
[local]Redback(config-ospf-area)#nssa-range 10.1.0.0 255.255.0.0
```

Related Commands

area
area-type
default-route
network-type
range

originate-default

originate-default { **always** | **route-map** *map-name* } [**metric** *metric*] [**metric-type** *type*]

no originate-default

Purpose

Originates the default route advertisement in the Open Shortest Path First (OSPF) or OSPF Version 3 (OSPFv3) routing domain.

Command Mode

OSPF router configuration

OSPFv3 router configuration

Syntax Description

always	Always originates a default route.
route-map <i>map-name</i>	Route map name. Originates the default route when all conditions in the specified route map are met and when the route exists in the Route Information Base (RIB).
metric <i>metric</i>	Optional. Metric value for the default route. The range of values is 1 to 16,777,214; the default value is 1.
metric-type <i>type</i>	Optional. External route metric type for a Type 5 default link-state advertisement (LSA). The <i>type</i> argument specifies one of the following metric types: • 1—Specifies a Type 1 metric type. • 2—Specifies a Type 2 metric type.

Default

No default route is originated. When this command is used to originate a default route, the metric value is 1.

Usage Guidelines

Use the **originate-default** command to originate the default route advertisement in the OSPF or OSPFv3 routing domain.

Use the **no** form of this command to remove the default route.

Examples

The following example configures the OSPF instance to originate a default route when there is a route in the RIB for routes matching the `rmap01` route map:

```
[local]Redback(config-ospf)#originate-default route-map rmap01
```

Related Commands

route-map

passive

passive

{no | default} passive

Purpose

Disables the sending and receiving of Open Shortest Path First (OSPF) or OSPF Version 3 (OSPFv3) packets through the interface.

Command Mode

OSPF interface configuration

OSPF3 interface configuration

Syntax Description

This command has no arguments or keywords.

Default

The interface is not a passive interface.

Usage Guidelines

Use the **passive** command to disable normal OSPF or OSPFv3 operations on an interface while still advertising the interface's IP subnet as an intra-area stub network in the OSPF or OSPFv3 routing domain.

Use the **no** or **default** form of this command to return the interface to its default state.

Examples

The following example disables normal OSPF operation on the interface `ospf1`, while still advertising the interface's IP subnet as an intra-area stub network in the OSPF routing domain:

```
[local]Redback(config-ospf-area)#interface ospf1  
[local]Redback(config-ospf-if)#passive
```

Related Commands

interface—OSPF and OSPF3 area configuration mode

range

range {*ip-addr/prefix-length* | *ipv6-addr/prefix-length*} [**not-advertise**]

no range {*ip-addr/prefix-length* | *ipv6-addr/prefix-length*} [**not-advertise**]

Purpose

Summarizes interarea routes advertised by an area border router (ABR).

Command Mode

OSPF area configuration

OSPF3 area configuration

Syntax Description

<i>ip-addr/prefix-length</i>	Specifies the IP address, in the form <i>A.B.C.D</i> , and the prefix length, separated by the slash (/) character. The range of values for the <i>prefix-length</i> argument is 0 to 32.
<i>ipv6-addr/prefix-length</i>	Specifies the IP Version 6 (IPv6) address, in the form <i>A:B:C:D:E:F:G:H</i> , and the prefix length, separated by the slash (/) character. The range of values for the <i>prefix-length</i> argument is 0 to 128.
not-advertise	Optional. Prevents the specified route from being advertised in interarea route summarizations.

Default

Route address ranges for interarea route summarization are not specified.

Usage Guidelines

Use the **range** command to summarize interarea routes advertised by an ABR.

Use the optional **not-advertise** keyword to prevent the specified route from being advertised in route summarizations.

Use the **no** form of this command to disable route summarization for a particular summary range. All individual routes contained in the summary range will be advertised to other areas.

Examples

The following example advertises routes that fall into the range 10.1.0.0 255.255.0.0 in interarea route summaries (one each of the other areas):

```
[local]Redback(config-ospf-area)#range 10.1.0.0 255.255.0.0
```

Related Commands

area
area-type

network-type
nssa-range

redistribute

```
redistribute {bgp asn | connected | isis instance [level-1 | level-2] | nat | ospf instance [external
[type-1 | type-2]] [inter-area] [intra-area] [nssa [type-1 | type-2]] | rip instance | static [dvsrc] |
subscriber [address | static]} [metric metric] [metric-type type] [route-map map-name]
[tag tag]
```

```
no redistribute {bgp asn | connected | isis instance [level-1 | level-2] | nat | ospf instance [external
[type-1 | type-2]] [inter-area] [intra-area] [nssa [type-1 | type-2]] | rip instance | static [dvsrc] |
subscriber [address | static]} [metric metric] [metric-type type] [route-map map-name]
[tag tag]
```

Purpose

Redistribute routes learned from other protocols into the Open Shortest Path First (OSPF) or OSPF Version 3 (OSPFv3) routing instance.

Command Mode

OSPF router configuration
OSPFv3 router configuration

Syntax Description

bgp <i>asn</i>	Border Gateway Protocol (BGP) autonomous system number (ASN). Redistributes routes from the specified BGP autonomous system (AS) into the OSPF or OSPFv3 routing instance. The range of values for the <i>asn</i> argument is 1 to 65,535.
connected	Redistributes routes from directly attached networks into the OSPF or OSPFv3 routing instance.
isis <i>instance</i>	Intermediate System-to-Intermediate System (IS-IS) instance name. Redistribute routes from the specified IS-IS routing instance into the OSPF or OSPFv3 routing instance.
level-1	Optional. Redistributes IS-IS level 1 routes only.
level-2	Optional. Redistributes IS-IS level 2 routes only.
nat	Redistributes network address translation (NAT) routes into the OSPF or OSPFv3 routing instance.
ospf <i>instance</i>	OSPF instance ID. Redistributes routes from another OSPF or OSPFv3 routing instance into the current OSPF or OSPFv3 routing instance. The range of values for the <i>instance</i> argument is 1 to 65,535.
external	Optional. Redistributes only external OSPF or OSPFv3 routes.
type-1	Optional. Redistributes only Type 1 external OSPF or OSPFv3 routes.
type-2	Optional. Redistributes only Type 2 external OSPF or OSPFv3 routes.
inter-area	Optional. Redistributes only interarea OSPF or OSPFv3 routes.

intra-area	Optional. Redistributes only intraarea OSPF or OSPFv3 routes.
nssa	Optional. Redistributes only OSPF or OSPFv3 NSSA routes.
type-1	Optional. Redistributes only OSPF or OSPFv3 NSSA Type 1 routes.
type-2	Optional. Redistributes only OSPF or OSPFv3 NSSA Type 2 routes.
rip <i>instance</i>	Routing Information Protocol (RIP) instance name. Redistributes routes from the specified RIP routing instance into the current OSPF or OSPFv3 routing instance.
static	Redistributes static IP routes into the OSPF or OSPFv3 routing instance. Optional with the subscriber keyword. Redistributes only static subscriber routes into the OSPF routing instance.
dvsvr	Optional. Redistributes the dynamically verified static routing (DVSR) subtype of static routes into the OSPF or OSPFv3 routing instance.
subscriber	Redistributes routes configured within subscriber records into the OSPF or OSPFv3 routing instance.
address	Optional. Redistributes only subscriber address routes into the OSPF or OSPFv3 routing instance.
metric <i>metric</i>	Optional. Cost of the redistributed routes. The range of values is 0 to 16,777,215; the default value is 20.
metric-type <i>type</i>	Optional. Metric type assigned to the redistributed routes. The <i>type</i> argument specifies one of the following metric types: • 1—Specifies a Type 1 metric type. • 2—Specifies a Type 2 metric type.
route-map <i>map-name</i>	Optional. Route map name. Modifies the attributes of redistributed routes using the specified route map.
tag <i>tag</i>	Optional. Route tag used to redistribute routes. An unsigned 32-bit integer, the range of values is 1 to 4,294,967,295; the default value is 0.

Default

Routes learned by other protocols are not distributed into the OSPF or OSPFv3 routing instance.

Usage Guidelines

Use the **redistribute** command to redistribute routes learned from other protocols into the OSPF or OSPFv3 routing instance.

You must enter multiple **redistribute** commands to redistribute routes from several different kinds of routing protocols into the OSPF or OSPFv3 routing instance.

Use the **no** form of this command to disable redistribution of the specified routing protocol or method.

Examples

The following example redistributes RIP into the OSPF routing instance:

```
[local]Redback(config-ospf)#redistribute rip
```

Related Commands

route-map

retransmit-interval

retransmit-interval *interval*

{**no** | **default**} **retransmit-interval**

Purpose

Modifies the interval at which link-state advertisements (LSAs) retransmissions are sent out through the specified interface, sham link, or virtual link.

Command Mode

OSPF interface configuration
 OSPF sham link configuration
 OSPF virtual link configuration
 OSPF3 interface configuration

Syntax Description

<i>interval</i>	Interval, in seconds, at which LSA transmissions are sent. The range of values is 1 to 65,535; the default value is 5.

Default

LSA retransmissions are sent every five seconds.

Usage Guidelines

Use the **retransmit-interval** command to modify the interval at which LSA retransmissions are sent out through the specified interface, sham link, or virtual link.

When a SmartEdge router sends LSAs to neighbors, it expects to receive an acknowledgment packet within a set amount of time. If the SmartEdge router does not receive an acknowledgment, it retransmits the LSA.

Use the **no** or **default** form of this command to return the interval to its default setting.

Examples

The following example configures an OSPF interface to retransmit LSAs every 7 seconds:

```
[local]Redback(config-ospf-if)#retransmit-interval 7
```

Related Commands

authentication	router-priority
cost	sham-link
hello-interval	transmit-delay
interface —OSPF and OSPF3 area configuration mode	virtual-link
router-dead-interval	

router-dead-interval

router-dead-interval *interval*

{**no** | **default**} **router-dead-interval**

Purpose

Modifies the amount of time the Open Shortest Path First (OSPF) or OSPF Version 3 (OSPFv3) process waits to receive a Hello packet from a neighbor before determining that the neighbor is not operational.

Command Mode

OSPF interface configuration
OSPF sham link configuration
OSPF virtual link configuration
OSPF3 interface configuration

Syntax Description

<i>interval</i>	Amount of time, in seconds, that the OSPF or OSPFv3 process waits to receive a Hello packet. The range of values is 1 to 65,535. The value must be the same for all routers on a common network.

Default

The interval is 40 seconds for broadcast and point-to-point (P2P) networks, and 120 seconds for point-to-multipoint (P2MP) and nonbroadcast multiaccess (NBMA) networks.

Usage Guidelines

Use the **router-dead-interval** command to modify the amount of time the OSPF or OSPFv3 process waits to receive a Hello packet from a neighbor before determining that the neighbor is not operational. The router dead interval can be configured on a specific interface, sham link, or virtual link

If a Hello packet is not received within the configured amount of time, the OSPF or OSPFv3 process modifies its topology database to indicate that the neighbor is not operational.

The router dead interval value must be the same for all routers on a common network. The value must be greater than that of the Hello interval to avoid destroying adjacencies when the neighbor router is operational.

The following restrictions apply to the **router-dead-interval** command:

- After the **fast-hello** command is configured, you cannot use the **router-dead-interval** command until the **fast-hello** command has been disabled.
- After the **router-dead-interval** command has been configured, you cannot use the **fast-hello** command until the **router-dead-interval** command has been disabled.

Use the **no** or **default** form of this command to return the interval value to its default setting.

Examples

The following example configures an OSPF interface to wait 60 seconds without receiving a Hello packet from its neighbor before determining that the neighbor is not operational:

```
[local]Redback(config-ospf-if)#router-dead-interval 60
```

Related Commands

- authentication**
- cost**
- hello-interval**
- interface**—OSPF and OSPF3 area configuration mode
- retransmit-interval**
- router-priority**
- sham-link**
- transmit-delay**
- virtual-link**

router-id

router-id *ip-addr*

no router-id

Purpose

Configures a fixed Open Shortest Path First (OSPF) or OSPF Version 3 (OSPFv3) router ID for the SmartEdge router.

Command Mode

OSPF router configuration

OSPFv3 router configuration

Syntax Description

ip-addr IP address of the interface to be used as the router ID.

Default

A router ID is not preconfigured.

Usage Guidelines

Use the **router-id** command to configure a fixed OSPF or OSPFv3 router ID for the SmartEdge router.

OSPF or OSPFv3 uses the router ID to identify the originating router for packets and link-state advertisements (LSAs). If the OSPF or OSPFv3 router ID is not configured, OSPF or OSPFv3 chooses the lowest loopback interface address. If there are no loopback interfaces, OSPF or OSPFv3 chooses the lowest interface address. The default OSPF or OSPFv3 router ID is selected when OSPF or OSPFv3 is started initially or restarted using the **process restart** command (in exec mode). For information on the **process restart** command, see the “Software Operations” chapter in the *Basic System Operations Guide* for the SmartEdge OS.

Use the **no** form of this command to remove a router ID.

Examples

The following example configures the IP address, 193.25.105.83, as the router ID:

```
[local]Redback(config-ospf)#router-id 193.25.105.83
```

Related Commands

router-id—BGP router configuration mode

router-id—context configuration mode

router ospf

router ospf3

router ospf

router ospf *instance*

no router ospf *instance*

Purpose

Configures an Open Shortest Path First (OSPF) routing instance and enters OSPF router configuration mode.

Command Mode

context configuration

Syntax Description

instance Instance ID. The range of values is 1 to 65,535.

Default

OSPF routing is disabled.

Usage Guidelines

Use the **router ospf** command to configure an OSPF routing instance and to enter OSPF router configuration mode.

Use the **no** form of this command to disable OSPF routing.

Examples

The following example configures the OSPF instance, 105, and enters OSPF router configuration mode:

```
[local]Redback(config-ctx)#router ospf 105  
[local]Redback(config-ospf)#
```

Related Commands

router-id

router ospf3

router ospf3

router ospf3 *instance-id*

no router ospf3 *instance-id*

Purpose

Creates an Open Shortest Path First Version 3 (OSPFv3) routing instance and enters OSPF3 router configuration mode.

Command Mode

context configuration

Syntax Description

instance-id Instance ID. The range of values is 1 to 65,535.

Default

OSPFv3 routing is disabled.

Usage Guidelines

Use the **router ospf3** command to create an OSPFv3 routing instance and to enter OSPF3 router configuration mode.

Use the **no** form of this command to disable OSPFv3 routing.

Examples

The following example configures the OSPFv3 instance, 105, and enters OSPF3 router configuration mode:

```
[local]Redback(config-ctx)#router ospf3 105  
[local]Redback(config-ospf3)#
```

Related Commands

router-id
router ospf

router-priority

router-priority *priority*

default router-priority

Purpose

Modifies the Open Shortest Path First (OSPF) or OSPF Version 3 (OSPFv3) preference for the SmartEdge router to act as the designated router on a network.

Command Mode

OSPF interface configuration

OSPFv3 interface configuration

Syntax Description

priority Priority setting. The range of values is 0 to 255; the default value is 1.

Default

The priority value is 1.

Usage Guidelines

Use the **router-priority** command to modify the OSPF or OSPFv3 preference for the SmartEdge router to act as the designated router on a network.

Enter any value greater than or equal to 1 to indicate that the SmartEdge router can act as the designated router. The router with the highest priority is used as the designated router for the network if there is not a designated router already on the network. If two routers have the same priority value, the router with the higher router ID is the designated router for the network; see the **router-id** command.

A value of 0 causes the router to never act as the designated router.

Use the **default** form of this command to return the priority to the default value of 1.

Examples

The following example sets the router priority to 2:

```
[local]Redback(config-ospf-if)#router-priority 2
```

Related Commands

authentication

cost

hello-interval

interface—OSPF and OSPFv3 area configuration mode

retransmit-interval

router-dead-interval

router-id

transmit-delay

sham-link

sham-link *src-addr dest-addr*

no sham-link *src-addr dest-addr*

Purpose

Creates an Open Shortest Path First (OSPF) adjacency tunneled over a Virtual Private Network (VPN) backbone and enters OSPF sham link configuration mode.

Command Mode

OSPF area configuration

Syntax Description

<i>src-addr</i>	Source IP address used as the local endpoint for the sham link. It must be the address of a local loopback interface.
<i>dest-addr</i>	Destination IP address used as the remote endpoint for the sham link.

Default

No OSPF sham links are configured.

Usage Guidelines

Use the **sham-link** command to create an OSPF adjacency tunneled (sham link) over a VPN backbone and enters OSPF sham link configuration mode. Sham links allow the VPN backbone path to be preferred when there are intra-area backdoor links between customer edge (CE) routers in the VPN.

The local connected route corresponding to the source IP address for the sham link must be redistributed into Border Gateway Protocol (BGP) and advertised over the VPN infrastructure to a provider edge (PE) router containing the other end of the sham link.

The route corresponding the remote end of the sham link must be redistributed into the corresponding OSPF instance in the VPN context. VPN routing must be enabled for the OSPF instance.

The cost of the sham link can be configured or will inherit the BGP Multi-Exit Discriminator (MED) from the VPN route.

Use the **no** form of this command to remove the sham link.

For more information on sham links, see the Internet Draft, *OSPF as the PE/CE Protocol in BGP/MPLS VPNs*, draft-rosen-vpns-ospf-bgp-mpls-04.txt.

Examples

The following example configures a sham link with cost 10 in area 0 for the OSPF instance within the VPN context:

```
[local]Redback(config-ospf)#vpn domain-id 1.1.1.1 domain-tag 0xfeedacee
[local]Redback(config-ospf)#area 0.0.0.0
```

```
[local]Redback(config-ospf-area)#sham-link 1.1.1.1 2.2.2.2  
[local]Redback(config-ospf-sham-link)#cost 10  
[local]Redback(config-ospf-sham-link)#exit  
[local]Redback(config-ospf)#redistribute bgp 1000
```

Related Commands

area—OSPF router configuration mode

router ospf

vpn

spf-timers

spf-timers *delay holdtime*

{**no** | **default**} **spf-timers**

Purpose

Configures the delay time between the receipt of a topology change and the start of the Shortest Path First (SPF) calculation, and to configure the hold time between two consecutive SPF calculations.

Command Mode

OSPF router configuration

OSPF3 router configuration

Syntax Description

<i>delay</i>	Delay time, in seconds, between the receipt of a topology change and the start of the SPF calculation. The range of values is 0 to 4,294,967,295; the default value is 5. A value of 0 means that an SPF calculation starts immediately when a topology change occurs.
<i>holdtime</i>	Minimum time, in seconds, between two consecutive SPF calculations. The range of values is 0 to 4,294,967,295; the default value is 10. A value of 0 means that there is no minimum wait time between successive SPF calculations.

Default

The delay is 5 seconds. The hold time is 10 seconds.

Usage Guidelines

Use the **spf-timers** to configure the delay time between the receipt of a topology change and the start of the SPF calculation, and to configure the hold time between two consecutive SPF calculations. Setting the delay and hold time to a low value enables faster switching to an alternate path in the event of failure. However, it also consumes more CPU processing time.

Use the **spf-timers 0 0** command to enable OSPF fast convergence. With OSPF fast convergence, route recalculation occurs as soon as new events arise.

Use the **no** or **default** form of this command to return the delay and hold time values to their default values.

Examples

The following example sets the SPF delay and hold time to 2 and 5:

```
[local]Redback(config-ospf)#spf-timers 2 5
```

Related Commands

None

stub-router

stub-router [**on-startup** *[interval]*] | **bgp-converge-delay** *[interval]* | **strict-bgp-tracking**
no stub-router

Purpose

Configures the router as an Open Shortest Path First (OSPF) or OSPF Version 3 (OSPFv3) stub router.

Command Mode

OSPF router configuration
OSPF3 router configuration

Syntax Description

on-startup	Optional. Sets router as a stub router on startup, and continues until timer expires.
<i>interval</i>	Optional. Timer interval in seconds. The range of values is 10 to 3,600 seconds; the default value is 210 seconds.
bgp-converge-delay	Optional. Sets router as a stub router on startup, and continues until timer expires or the Border Gateway Protocol (BGP) converges.
strict-bgp-tracking	Optional. Sets router as a stub router whenever BGP has not converged. If BGP is not converged or not running, stub router operation remains active. There is no time out for the stub router as long as BGP is not converged.

Default

The router is not configured as a stub router.

Usage Guidelines

Use the **stub router** command to configure the router as an OSPF or OSPFv3 stub router. To avoid transit traffic, a stub router advertises all of its links using the maximum cost of 65,535.

Use the **set-overload-bit** command in IS-IS router configuration mode without any option to indefinitely set the stub router configuration.

Use the **on-startup** keyword if BGP is not configured on the router, or if BGP convergence is not an issue. When the router starts, OSPF or OSPFv3 temporarily sets the stub router configuration to allow the router to reach full functionality, with complete routing information on the router.

Use the **bgp-converge-delay** keyword if BGP is not fully converged, and you want to use the stub router configuration to delay other routers from sending transit traffic through the router until BGP converges. If the BGP converge delay time expires, the stub router configuration is removed, even if BGP has not converged; therefore, you should adjust the BGP converge delay time so that it is appropriate to your network size and the amount information in the BGP routing table.

Use the **strict-bgp-tracking** keyword if BGP is not fully converged, and you want to use the stub router configuration to stop other routers from sending transit traffic through the router to until BGP converges. The stub router configuration is removed only when full BGP convergence is reached.

Use the **no** form of this command to remove the stub router configuration.

Examples

The following example configures the SmartEdge router as an OSPF stub router:

```
[local]Redback(config-ctx)#router ospf  
[local]Redback(config-ospf)#stub-router
```

Related Commands

router-id
set-overload-bit

summary-address

summary-address { *ip-addr/prefix-length* | *ipv6-addr/prefix-length* } [**not-advertise** | **tag tag**]

no summary-address { *ip-addr/prefix-length* | *ipv6-addr/prefix-length* } [**not-advertise** | **tag tag**]

Purpose

Summarizes external routes that are redistributed into the Open Shortest Path First (OSPF) or OSPF Version 3 (OSPFv3) routing domain.

Command Mode

OSPF router configuration
OSPF3 router configuration

Syntax Description

<i>ip-addr/prefix-length</i>	Specifies the IP address, in the form <i>A.B.C.D</i> , and the prefix length, separated by the slash (/) character. The range of values for the <i>prefix-length</i> argument is 0 to 32.
<i>ipv6-addr/prefix-length</i>	Specifies the IP Version 6 (IPv6) address, in the form <i>A:B:C:D:E:F:G:H</i> , and the prefix length, separated by the slash (/) character. The range of values for the <i>prefix-length</i> argument is 0 to 128.
not-advertise	Optional. Suppresses the advertisement of Type 5 link-state advertisements (LSAs) for routes contained in the specified IP address range.
tag tag	Optional. Route tag included in translated external route summarization Type 5 link-state advertisements (LSAs). An unsigned 32-bit integer, the range of values is 1 to 4,294,967,295; the default value is 0.

Default

No external redistributed routes are summarized.

Usage Guidelines

Use the **summary-address** command to summarize external routes that are redistributed into the OSPF or OSPFv3 routing instance.

Use the **no** form of this command to disable route summarization of an IP address block and allow all individual routes to be redistributed into the OSPF or OSPFv3 routing instance.

Examples

The following example advertises a summary of the routes that fall into the 10.0.0.0 255.0.0.0 range:

```
[local]Redback(config-ospf)#summary-address 10.0.0.0 255.0.0.0
```


Related Commands

redistribute—OSPF and OSPF3 router configuration mode

transmit-delay

transmit-delay *delay*

{**no** | **default**} **transmit-delay**

Purpose

Sets a delay value, increasing the age of link-state advertisements (LSAs) sent over the specified interface, sham link, or virtual link.

Command Mode

OSPF interface configuration
 OSPF sham link configuration
 OSPF virtual link configuration
 OSPF3 interface configuration

Syntax Description

<i>delay</i>	Delay, in seconds. The range of values is 1 to 65,535; the default value is 1 second.

Default

No delay value is set. When set, the delay value is one second.

Usage Guidelines

Use the **transmit-delay** command to set a delay value, increasing the age of LSAs sent over the specified interface, sham link, or virtual link.

Before a link-state update packet is advertised, the age of the LSAs in the packet must be increased by a value proportionate to the speed of the interface, sham link, or virtual link; for example, on a very slow interface, sham link, or virtual link, you might set the transmit delay to two seconds to ensure that you do not receive an LSA that is less recent than the copy in the router's link-state database.

Use the **no** or **default** form of this command return the delay value to its default setting.

Examples

The following example sets an OSPF interface transmit delay to 3 seconds:

```
[local]Redback(config-ospf-if)#transmit-delay 3
```

Related Commands

authentication	router-dead-interval
cost	router-priority
hello-interval	sham-link
interface —OSPF and OSPF area configuration mode	virtual-link
retransmit-interval	

virtual-link

virtual-link {*transit-id* | *transit-addr*} *virtual-endpoint-addr*

no virtual-link {*transit-id* | *transit-addr*} *virtual-endpoint-addr*

Purpose

In OSPF area configuration mode, creates an Open Shortest Path First (OSPF) virtual link through the specified transit area and enters OSPF virtual link configuration mode.

In OSPF3 area configuration mode, creates an OSPF Version 3 (OSPFv3) virtual link through the specified transit area and enters OSPF3 virtual link configuration mode.

Command Mode

OSPF area configuration

OSPF3 area configuration

Syntax Description

<i>transit-id</i>	Transit area ID for the virtual link specified as a 32-bit number.
<i>transit-addr</i>	Transit area IP address for the virtual link in the form <i>A.B.C.D</i> .
<i>virtual-endpoint-addr</i>	Router ID of the virtual link endpoint in the form <i>A.B.C.D</i> .

Default

There are no predefined virtual links for the area.

Usage Guidelines

Use the **virtual-link** command in OSPF area configuration mode to create an OSPF virtual link through the specified transit area and enters OSPF virtual link configuration mode.

Use the **virtual-link** command in OSPF3 area configuration mode to create an OSPFv3 virtual link through the specified transit area and enters OSPF3 virtual link configuration mode.

Virtual links can be configured between any two backbone routers that have an interface to a common non-backbone area. Virtual links belong to the backbone. The protocol treats two routers joined by a virtual link as if they were connected by an unnumbered point-to-point backbone network.

Virtual links can only be configured in the backbone area (area ID=0), and the transit area cannot be the backbone area.

Use the **no** form of this command to remove the virtual link.

For more information on OSPF virtual links, see RFC 2328, *OSPF Version 2*.

For more information on OSPFv3 virtual links, see RFC 2740, *OSPF for IPv6*.

Examples

The following example configures a virtual link through area 1, with a virtual link endpoint of 30.30.30.30, and enters OSPF virtual link configuration mode:

```
[local]Redback(config-ospf)#router ospf 1  
[local]Redback(config-ospf)#area 0  
[local]Redback(config-ospf-area)#virtual-link 1 30.30.30.30  
[local]Redback(config-ospf-virt-link)#
```

Related Commands

area—OSPF router configuration mode
authentication
hello-interval
interface—OSPF area configuration mode
retransmit-interval
router-dead-interval
transmit-delay

BFD Configuration

This chapter provides an overview of Bidirectional Forwarding Detection (BFD) and describes the tasks and commands used to configure BFD features through the SmartEdge® OS.

For information about the tasks and commands used to monitor, troubleshoot, and administer BFD, see the “BFD Operations” chapter in the *Routing Protocols Operations Guide* for the SmartEdge OS.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples
- Command Descriptions

Overview

Bidirectional Forwarding Detection (BFD) is a simple Hello protocol that, in many respects, is similar to the detection components of some routing protocols. A pair of routers periodically transmit BFD packets over each path between the two routers, and if a system stops receiving BFD packets after a predefined time interval, some component in that particular bidirectional path to the neighboring router is assumed to have failed.

A path is only declared to be operational when two-way communication has been established between systems.

BFD provides low overhead, short-duration detection of failures in the path between adjacent forwarding engines, including the interfaces, data links, and to the extent possible, the forwarding engines themselves.

The legacy Hello mechanism run by routing protocols does not offer detections of less than one second, and for some applications, more than one second is too long and represents a large amount of data loss at gigabit rates. BFD provides the ability to detect communication failures in less than one second.

Configuration Tasks

Note In this section, the command syntax in the task tables displays only the root command; for the complete command syntax, see the full description for the command in the “Command Descriptions” section.

To configure BFD, perform the tasks described in the following sections:

- Configuring a BFD Neighbor
- Configuring BFD on an Interface
- Enabling or Disabling BFD for a Routing Interface

Configuring a BFD Neighbor

A BFD session is established for each BFD neighbor configured. More than one BFD neighbor can be configured.

To configure a BFD neighbor, perform the tasks described in Table 7-1. Enter all commands in BFD neighbor configuration mode, unless otherwise noted.

Table 7-1 Configure a BFD Neighbor

Task	Root Command	Notes
Create a BFD instance and enter BFD router configuration mode.	router bfd	Enter this command in context configuration mode.
Create a new BFD neighbor, or select an existing one for modification, and enter BFD neighbor configuration mode.	neighbor	Enter this command in BFD router configuration mode.
Specify the detection multiplier value.	detection-multiplier	The negotiated minimum transmit interval (the minimum desired transmit interval agreed upon by both peers) is multiplied by the detection multiplier value to provide the detection time for the transmitting system in asynchronous mode. The detection time is the time it takes to declare a neighbor as down. For example, if the minimum desired transmit interval was negotiated at 10 ms and the detection multiplier is set to 3, then the detection time is 30 ms. Using the detection multiplier adds robustness to BFD by allowing the system to not bring down a neighbor if only one BFD packet is missed.
Specify the minimum required interval, in milliseconds, between received BFD control packets that the system is capable of supporting.	minimum receive-interval	
Specify the minimum desired transmit interval, in milliseconds, used by the local system when transmitting BFD control packets.	minimum transmit-interval	

Configuring BFD on an Interface

Configuring BFD on an interface establishes a separate BFD session for each neighbor on the interface. Neighbors are learned by the client routing protocol (such as Open Shortest Path First [OSPF]) that has BFD detection enabled.

Note BFD clients are routing protocols that use BFD to detect communication failures in less than one second. Currently, OSPF is the only routing protocol supported by BFD.

To configure BFD on an interface, perform the tasks described in Table 7-2. Enter all commands in BFD interface configuration mode, unless otherwise noted.

Table 7-2 Configure BFD on an Interface

Task	Root Command	Notes
Create a BFD instance and enter BFD router configuration mode.	router bfd	Enter this command in context configuration mode.
Enables BFD on a named interface and enters BFD interface configuration mode.	interface	Enter this command in BFD router configuration mode. The interface must already be configured through the interface command (in context configuration mode) before BFD can be enabled on it. For more information about the interface command, see the "Interface Configuration" chapter in the <i>Basic System Configuration Guide</i> for the SmartEdge OS.
Specify the detection multiplier value.	detection-multiplier	The negotiated minimum transmit interval (the minimum desired transmit interval agreed up by both peers) is multiplied by the detection multiplier value to provide the detection time for the transmitting system in asynchronous mode. The detection time is the time it takes to declare a neighbor as down. For example, if the minimum desired transmit interval was negotiated at 10 ms and the detection multiplier is set to 3, then the detection time is 30 ms. Using the detection multiplier adds robustness to BFD by allowing the system to not bring down a neighbor if only one BFD packet is missed.
Specify the minimum required interval, in milliseconds, between received BFD control packets that the system is capable of supporting.	minimum receive-interval	
Specify the minimum desired transmit interval, in milliseconds, used by the local system when transmitting BFD control packets.	minimum transmit-interval	

Enabling or Disabling BFD for a Routing Interface

By default, BFD is enabled for all supported routing instances, but you can only enable or disable BFD for a particular interface within a routing instance. Because BFD is enabled by default, you must first disable BFD before you can enable it to return BFD to its default operating mode.

Note Currently, OSPF is the only routing protocol supported by BFD.

To enable or disable BFD for a routing protocol instance, perform the task described in Table 7-3.

Table 7-3 Enable or Disable BFD for a Routing interface

Task	Root Command	Notes
Enable or disable BFD for a routing interface.	bfd detection	Enter this command in OSPF interface configuration mode. Use the no form of this command to disable BFD for a routing protocol instance.

Configuration Examples

This section provides BFD configuration examples in the following sections:

- BFD Neighbor
- BFD Interface
- Disabling BFD

BFD Neighbor

A BFD session is established for each BFD neighbor configured. More than one BFD neighbor can be configured. The following example configures the BFD neighbor, 192.168.0.24, sets the minimum desired transmit interval to 30 ms, sets the minimum receive interval to 30 ms, and the sets detection multiplier to 4:

```
[local]Redback#configure
[local]Redback(config)#context local
[local]Redback(config-ctx)#router bfd
[local]Redback(config-bfd)#neighbor 192.168.0.24
[local]Redback(config-bfd-nbr)#minimum receive-interval 30
[local]Redback(config-bfd-nbr)#minimum transmit-interval 30
[local]Redback(config-bfd-nbr)#detection-multiplier 4
[local]Redback(config-bfd-nbr)#end
```


BFD Interface

Configuring BFD on an interface establishes a separate BFD session for each neighbor on the interface. Neighbors are learned by the client routing protocol (such as OSPF) that has BFD detection enabled. The following example configures BFD on the interface, `foo`, sets the minimum desired transmit interval to 25 ms, sets the minimum receive interval to 40 ms, and the sets detection multiplier to 2:

```
[local]Redback#configure
[local]Redback(config)#context local
[local]Redback(config-ctx)#router bfd
[local]Redback(config-bfd)#interface foo
[local]Redback(config-bfd-if)#minimum receive-interval 25
[local]Redback(config-bfd-if)#minimum transmit-interval 40
[local]Redback(config-bfd-if)#detection-multiplier 2
[local]Redback(config-bfd-if)#end
```

Disabling BFD

The following example disables BFD on the OSPF interface, `foo`:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#router ospf 15
[local]Redback(config-ospf)#interface foo
[local]Redback(config-ospf-if)#no bfd detection
[local]Redback(config-ospf-if)#
```

Command Descriptions

This section describes the syntax and usage guidelines for the commands used to configure BFD features. The commands are presented in alphabetical order.

bfd detection	minimum transmit-interval
detection-multiplier	neighbor
interface	router bfd
minimum receive-interval	

bfd detection

bfd detection

no bfd detection

Purpose

Enables Bidirectional Forwarding Detection (BFD) for a routing interface.

Command Mode

OSPF interface configuration

Syntax Description

This command has no keywords or arguments.

Default

BFD is enabled.

Usage Guidelines

Use the **bfd detection** command to enable BFD for a routing interface.

By default, BFD is enabled for all supported routing instances, but you can only enable or disable BFD for a particular interface within a routing instance. You must first disable BFD before you can enable it to return BFD to its default operating mode.

Note Currently, Open Shortest Path First (OSPF) is the only routing protocol supported by BFD.

Use the **no** form of this command to disable BFD for a routing protocol interface.

Examples

The following example disables BDF on the OSPF interface, foo:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#router ospf 15
[local]Redback(config-ospf)#interface foo
[local]Redback(config-ospf-if)#no bfd detection
[local]Redback(config-ospf-if)#
```

Related Commands

None

detection-multiplier

detection-multiplier *value*

{**no** | **default**} **detection-multiplier**

Purpose

Specifies the detection multiplier value.

Command Mode

BFD interface configuration

BFD neighbor configuration

Syntax Description

value

Detection multiplier value. The range of values is 1 to 10; the default value is 3.

Default

The default detection multiplier value is 3.

Usage Guidelines

Use the **detection-multiplier** command to specify the detection multiplier value.

The negotiated minimum transmit interval (the minimum desired transmit interval agreed upon by both peers) is multiplied by the detection multiplier value to provide the detection time for the transmitting system in asynchronous mode. The detection time is the time it takes to declare a neighbor as down. For example, if the minimum desired transmit interval was negotiated at 10 ms and the detection multiplier is set to 3, then the detection time is 30 ms. Using the detection multiplier adds robustness to Bidirectional Forwarding Detection (BFD) by allowing the system to not bring down a neighbor if only one BFD packet is missed.

Use the **no** or **default** form of this command to return the detection multiplier value to 3.

Examples

The following example sets the detection multiplier value on the interface, `to_foo`, to 7:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#router bfd
[local]Redback(config-bfd)#interface to_foo
[local]Redback(config-bfd-if)#detection-multiplier 7
[local]Redback(config-bfd-if)#
```

Related Commands

interface

minimum receive-interval

minimum transmit-interval

neighbor

interface

interface {*if-name* | *ip-addr*}

no interface {*if-name* | *ip-addr*}

Purpose

Enables Bidirectional Forwarding Detection (BFD) on a named interface and enters BFD interface configuration mode.

Command Mode

BFD router configuration

Syntax Description

<i>if-name</i>	Interface name.
<i>ip-addr</i>	IP address of the interface, in the form <i>A.B.C.D</i> .

Default

None

Usage Guidelines

Use the **interface** command to enable BFD on a named interface and enter BFD interface configuration mode.

The interface must already be configured through the **interface** command (in context configuration mode) before BFD can be enabled on it. For more information about the **interface** command, see the “Interface Configuration” chapter in the *Basic System Configuration Guide* for the SmartEdge OS.

Use the **no** form of this command to disable BFD on the specified interface.

Examples

The following example enables BFD on the interface, `to_foo`:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#router bfd
[local]Redback(config-bfd)#interface to_foo
[local]Redback(config-bfd-if)#
```

Related Commands

detection-multiplier	neighbor
minimum receive-interval	router bfd
minimum transmit-interval	

minimum receive-interval

minimum receive-interval *interval*

{ **no** | **default** } **minimum receive-interval**

Purpose

Specifies the minimum required interval, in milliseconds, between received Bidirectional Forwarding Detection (BFD) control packets that the system is capable of supporting.

Command Mode

BFD interface configuration

BFD neighbor configuration

Syntax Description

interval

Minimum required receive interval value. The range of values, in milliseconds, is 10 to 60,000; the default value is 1,000.

Default

The default minimum receive interval is 1,000 ms (1 second).

Usage Guidelines

Use the **minimum receive-interval** command to specify the minimum required interval, in milliseconds, between received BFD control packets that the system is capable of supporting.

Use the **no** or **default** form of this command to return the minimum required receive interval to 1,000 ms.

Examples

The following example sets the minimum required receive interval on the interface, `to_foo`, to 30 ms:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#router bfd
[local]Redback(config-bfd)#interface to_foo
[local]Redback(config-bfd-if)#minimum receive-interval 30
[local]Redback(config-bfd-if)#
```

Related Commands

detection-multiplier

interface

minimum transmit-interval

neighbor

minimum transmit-interval

minimum transmit-interval *interval*

{**no** | **default**} **minimum transmit-interval**

Purpose

Specifies the minimum desired transmit interval, in milliseconds, used by the local system when transmitting Bidirectional Forwarding Detection (BFD) control packets.

Command Mode

BFD interface configuration

BFD neighbor configuration

Syntax Description

interval

Minimum desired transmit interval value. The range of values, in milliseconds, is 10 to 60,000; the default value is 1,000.

Default

The default minimum desired transmit interval is 1,000 ms (1 second).

Usage Guidelines

Use the **minimum transmit-interval** command to specify the minimum desired transmit interval, in milliseconds, used by the local system when transmitting BFD control packets.

Use the **no** or **default** form of this command to return the minimum desired transmit interval to 1,000 ms.

Examples

The following example sets the minimum desired transmit interval on the interface, `to_foo`, to 30 ms:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#router bfd
[local]Redback(config-bfd)#interface to_foo
[local]Redback(config-bfd-if)#minimum transmit-interval 30
[local]Redback(config-bfd-if)#
```

Related Commands

detection-multiplier

interface

minimum receive-interval

neighbor

neighbor

neighbor *ip-addr*

no neighbor *ip-addr*

Purpose

Creates a new Bidirectional Forwarding Detection (BFD) neighbor, or selects an existing one for modification, and enters BFD neighbor configuration mode.

Command Mode

BFD router configuration

Syntax Description

ip-addr BFD neighbor IP address, in the form *A.B.C.D*.

Default

No BFD neighbors are configured.

Usage Guidelines

Use the **neighbor** command to create a new BFD neighbor, or select an existing one for modification, and enter BFD neighbor configuration mode.

Use the **no** form of this command to delete a BFD neighbor configuration.

Examples

The following example creates a new BFD neighbor, 10.10.10.1:

```
[local]Redback(config)#context local  
[local]Redback(config-ctx)#router bfd  
[local]Redback(config-bfd)#neighbor 10.10.10.1  
[local]Redback(config-bfd-nbr)#
```

Related Commands

detection-multiplier
interface
minimum receive-interval
minimum transmit-interval
router bfd

router bfd

router bfd

no router bfd

Purpose

Creates a Bidirectional Forwarding Detection (BFD) instance and enters BFD router configuration mode.

Command Mode

context configuration

Syntax Description

This command has no keywords or arguments.

Default

No BFD instances are configured.

Usage Guidelines

Use the **router bfd** command to create a BFD instance and enter BFD router configuration mode.

Use the **no** form of this command to disable the BFD instance.

Examples

The following example creates a BFD instance on the context, `local`, and enters BFD router configuration mode:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#router bfd
[local]Redback(config-bfd)#
```

Related Commands

detection-multiplier
interface

BGP Configuration

This chapter provides an overview of the Border Gateway Protocol (BGP) and describes the tasks and commands used to configure BGP features through the SmartEdge® OS.

For information about the tasks and commands used to monitor, troubleshoot, and administer BGP, see the “BGP Operations” chapter in the *Routing Protocols Operations Guide* for the SmartEdge OS.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples
- Command Descriptions

Overview

BGP is an Exterior Gateway Protocol (EGP) based on distance-vector algorithms, and uses the Transmission Control Protocol (TCP) as its transport protocol. BGP is a protocol between exactly two BGP nodes, or BGP speakers. First, the TCP connection is established and then the two BGP speakers exchange dynamic routing information over the connection. The exchange of messages is a BGP session between BGP peers.

We support multiple BGP features, including those specified in the following IETF drafts and RFCs:

- Base features:
 - Y. Rekhter, T. Li, RFC 1771, *Border Gateway Protocol 4 (BGP-4)*, March 1995
 - Y. Rekhter, T. Li, Internet Draft, *A Border Gateway Protocol 4 (BGP-4)*, draft-ietf-idr-bgp4-12.txt, January 2001
- Route reflection:
 - T. Bates, R. Chandra, E. Chen, RFC 2796, *BGP Route Reflection - An Alternative to Full Mesh IBGP*, April 2000
- Autonomous system confederations:
 - P. Traina, D. McPherson, J. Scudder, RFC 3065, *Autonomous System Confederations for BGP*, February 2001

- Communities attribute:
 - R. Chandra, P. Traina, T. Li, RFC 1997, *BGP Communities Attribute*, August 1996
- MD-5 authentication:
 - A. Heffernan, RFC 2385, *Protection of BGP Sessions via the TCP MD5 Signature Option*, August 1998
- Route-flap damping:
 - C. Villamizar, R. Chandra, R. Govindan, RFC 2439, *BGP Route Flap Damping*, November 1998
- Capabilities advertisement:
 - R. Chandra, J. Scudder, RFC 2842, *Capabilities Advertisement with BGP-4*, May 2000
- Multiprotocol extensions:
 - T. Bates, R. Chandra, D. Katz, Y. Rekhter, RFC 2858, *Multiprotocol Extensions for BGP-4*, June 2000
- Route refresh capability:
 - E. Chen, RFC 2918, *Route Refresh Capability for BGP-4*, September 2000
- Outbound route filtering (ORF) capability:
 - E. Chen, Y. Rekhter, Internet Draft, *Cooperative Route Filtering Capability for BGP-4*, draft-ietf-idr-route-filter-03.txt, April 2001
- Address prefix-based ORF capability:
 - E. Chen, S. Ramachandra, Internet Draft, *Address Prefix Based Outbound Route Filter for BGP-4*, draft-chen-bgp-prefix-orf-02.txt, April 2001
- Graceful restart capability:
 - S. Ramachandra, Y. Rekhter, R. Fernando, J. Scudder, E. Chen, Internet Draft, *Graceful Restart Mechanism for BGP*, draft-ietf-idr-restart-01.txt, July 2001
- Four-byte autonomous system (AS) capability:
 - Q. Vohra, E. Chen, Internet Draft, *BGP Support For Four-Octet AS Number Space*, draft-ietf-idr-as4bytes-03.txt, May 2001

Redback Networks also supports the following additional features:

- Routing policies, including these types of filters:
 - Prefix lists
 - AS path lists
 - Route maps
- BGP route sourcing, including these methods:
 - Redistribution from other routing protocols into the BGP routing domain
 - Origination of BGP routes through the **network** command in BGP address family configuration mode
- Route aggregation through the support of the AS_SET attribute

- Default origination—both conditional and unconditional
- Maximum number of prefixes setting
- Multipath capability for both internal BGP (iBGP) and external BGP (eBGP)
- Peer groups, including these features:
 - Address family-specific grouping
 - Decoupling of peer groups and default origination
- Route-flap statistics for both iBGP and eBGP
- Accounting of routes by these methods:
 - Number of routes sourced
 - Number of routes accepted, active, dampened, and historical from each peer
 - Number of routes advertised to a peer
- Advanced debug facilities, including these features:
 - Per-neighbor based generation of debug messages
 - Storage and display of malformed messages and notification messages.
 - Peer reset history

In-depth information on how BGP is structured, and how it operates, is described in the following sections:

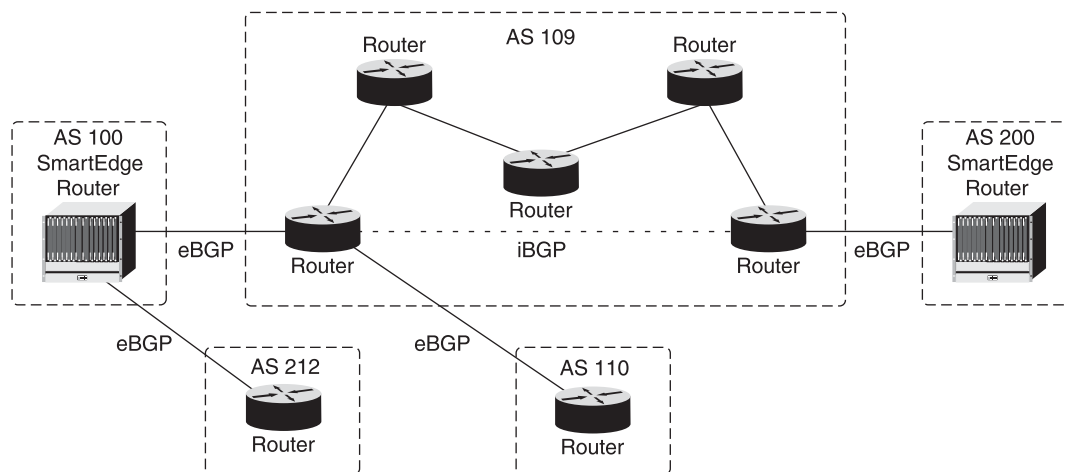
- iBGP and eBGP
- iBGP Route Reflectors
- iBGP Confederations
- Route Aggregation
- MBGP
- Routing Policy Triggered Update
- Non-Intrusive MD5 Password Change

iBGP and eBGP

Routers that belong to the same AS and exchange BGP updates are running internal BGP (iBGP), and routers that belong to different autonomous systems and exchange BGP updates are running external BGP (eBGP).

Figure 8-1 illustrates the concept of autonomous systems and iBGP versus eBGP.

Figure 8-1 iBGP and eBGP Networks



0540

iBGP Route Reflectors

Typically, iBGP speakers must be fully meshed. Any BGP speaker that receives messages from an external router must advertise the routes it receives to all BGP speakers in its autonomous system. However, if a route reflector is configured, although it must have connections to all other BGP speakers in the AS, not all other BGP speakers must be fully meshed. When a BGP speaker in the AS receives messages from an external router, it is sufficient to advertise these routes only to the route reflector, which then re-advertises the routes to all other BGP speakers in the AS.

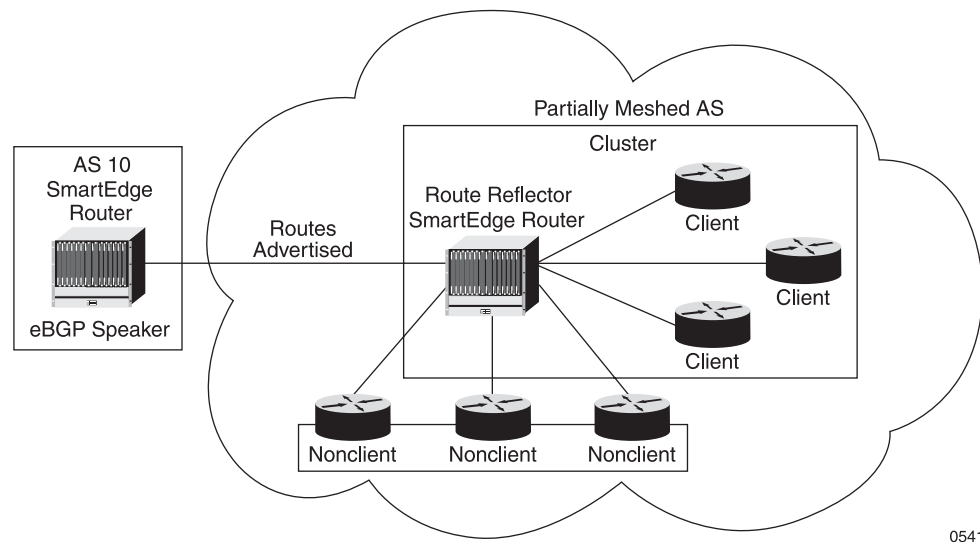
Internal peers of the route reflector are divided into two groups: client peers and nonclient peers. A route reflector reflects routes between these two groups. The route reflector and its client peers form a cluster. Nonclient peers must be fully meshed with each other. Client peers are not required to be fully meshed and do not communicate with BGP speakers outside their cluster. If it is required, peer client-to-peer client route reflection can be disabled.

When the route reflector receives an advertised route:

- Any route from an external BGP speaker is advertised to all peers.
- Any route from a nonclient peer is advertised to all client peers.
- Any route from a client peer is advertised to all peers.

Figure 8-2 shows an example of iBGP networking using route reflection.

Figure 8-2 iBGP Network Using Route Reflection

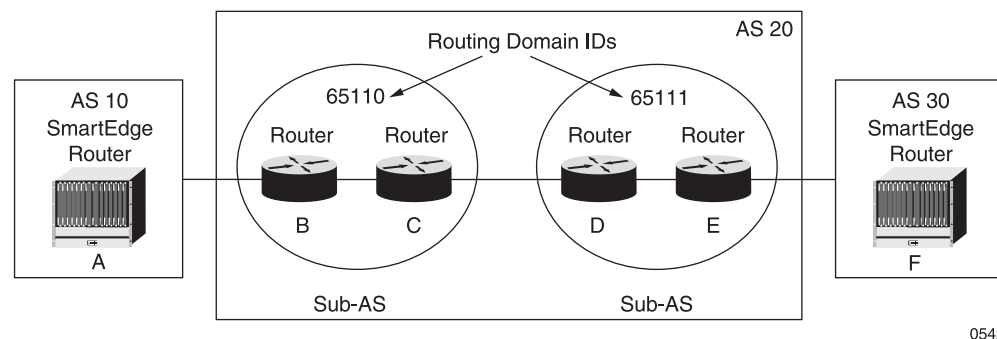


iBGP Confederations

Another way to reduce iBGP mesh is to divide an autonomous system into subautonomous systems grouped by a routing domain identifier. The AS and its subautonomous systems are part of the same confederation. Externally, the confederation looks like a single AS. Each subautonomous system is fully meshed within itself and has a few connections to other subautonomous systems in the confederation.

Neighbors from other subautonomous systems are treated as special eBGP peers. Even though peers in different subautonomous systems engage in eBGP sessions, they exchange routing information as if they were iBGP peers. Specifically, the next-hop, the Multi-Exit Discriminator (MED), and local preference information is preserved, so that a single Interior Gateway Protocol (IGP) is used for all of the subautonomous systems; see Figure 8-3.

Figure 8-3 iBGP Confederation



Route Aggregation

BGP4 supports Classless InterDomain Routing (CIDR). With CIDR, routers use the network prefix to determine the dividing point between the network number and the host number. For example, the range of addresses 128.186.1.0 to 128.186.1.255 can be represented as the network prefix 128.186.1.0/24; the 24 indicates that all addresses in the segment agree in their first 24 bits.

In addition, CIDR does not require a network to be of standard size, as is the case in classful addressing, which provides 8-bit (Class A), 16-bit (Class B), and 24-bit (Class C) network deployment. This flexibility in CIDR enables the creation of arbitrarily sized networks.

Of particular importance is CIDR's ability to lend itself to the concept of route aggregation. The Internet is divided into addressing domains. Within a domain, detailed information is available about all of the networks that reside in the domain. Outside of an addressing domain, however, only the common network prefix is advertised. By allowing a single routing table entry to specify a route to many individual network addresses, aggregation minimizes the size of the routing table. A router cannot aggregate an address if it does not have a more specific route of that address in the BGP routing table. More-specific routes can be injected in the BGP routing table by incoming updates from other autonomous systems.

MBGP

Multiprotocol BGP (MBGP) makes use of multiprotocol extensions to BGP4, as defined in RFC 2283, *Multiprotocol Extensions for BGP-4*, that allow other protocols to use BGP to exchange protocol-specific information.

One of the main advantages of MBGP is the ability to use BGP's scalability and policy control, to easily configure routers to peer with other interdomain routers, exchange multicast source route information, and configure multicast routing policies using familiar BGP commands. MBGP also carries two sets of routes: One set for unicast routing and one set for multicast routing, allowing you to configure separate routing policies for unicast and multicast routes.

Routing Policy Triggered Update

Before Release 2.5, whenever there was a change in an inbound or outbound routing policy, such as a prefix-list, as-path-list, or route-map, for a BGP peer, the **clear bgp neighbor ip-addr soft [in | out]** command had to be manually issued to make the policy change effective. Currently, routing policy changes automatically take effect, and issuing the **clear bgp neighbor ip-addr soft [in | out]** command to update routing policies can cause updates to be unnecessarily sent, so it is not recommended.

To aggregate multiple policy changes, the SmartEdge OS performs the necessary action 15 seconds after a policy change.



Caution Risk of dropped connection. If the remote peer does not support the BGP Route Refresh Capability, an inbound policy change for the peer results in an automatic hard reset of the session. To reduce the risk, ensure that the remote peer supports the BGP Route Refresh Capability.

Non-Intrusive MD5 Password Change

The non-intrusive Message Digest 5 (MD5) password change feature for BGP allows you to change the password for a BGP peer without resetting the BGP session. The following sections describe in detail how the non-intrusive MD5 password change feature is implemented:

- Replace a Password
- Add a New Password
- Delete a Password

Replace a Password

When an old MD5 password is replaced by a new one in a BGP peer configuration, both passwords are allowed to co-exist for authentication until the old password expires. To facilitate a smooth transition from the old to new password, a new configuration can be used to specify the time interval during which the old MD5 password co-exists with the new one.

For a TCP connection that is already established, or is in one of the closing states when an existing password is replaced by a new MD5 password, both password strings co-exist for authentication during the specified time interval before the old MD5 password expires. The old MD5 password continues to be used for authentication until either the password expires, or the remote TCP for the peer uses a new MD5 password.

For a TCP connection that is not yet established, when the old password is replaced, the local TCP immediately uses the new MD5 password.

Note BGP keeps only the latest password string configured and the previous password to be replaced. That is, if a third password is configured before the timer for first (active) password expires, the oldest password is immediately deleted, and the expiration timer is started for the second password.

Add a New Password

This feature does not apply when configuring a new MD5 password for a peer while there is no existing password already configured for the peer. The BGP peer session is reset after the new MD5 password is configured.

Delete a Password

This feature does not apply when explicitly deleting a MD5 password from the BGP peer configuration.

When the current active MD5 password is deleted from the configuration, the old password (if existing) and the current password are both immediately deleted, and the BGP session with the peer is reset.

Note To avoid BGP sessions from being reset when changing a peer MD5 password, we recommend that you do not delete the password from the configuration, and always use the **password** command to implicitly replace the password.

Configuration Tasks

Note In this section, the command syntax in the task tables displays only the root command; for the complete command syntax, see the full description for the command in the “Command Descriptions” section.

To configure BGP, perform the tasks described in the following sections:

- Configuring BGP Routing Instances and Instance Attributes
- Configuring BGP Neighbors and Neighbor Attributes
- Configuring BGP Peer Groups and Peer Group Attributes

Configuring BGP Routing Instances and Instance Attributes

A BGP routing instance enables the SmartEdge router to be a BGP speaker. In addition, many BGP parameters that can affect the global routing process can be configured within a BGP routing instance.

To configure a BGP routing instance and other instance attributes, perform the tasks described in the following sections:

- Configure a BGP Routing Instance
- Configure IPv4 Address Family Attributes for a BGP Routing Instance
- Configure IPv6 Address Family Attributes for a BGP Routing Instance
- Configure Graceful Restart for a BGP Routing Instance
- Configure BGP Route Reflection
- Configure a BGP Confederation

Configure a BGP Routing Instance

To configure a BGP routing instance, perform the tasks described in Table 8-1. Enter all commands in BGP router configuration mode, unless otherwise noted.

Table 8-1 Configure a BGP Routing Instance

Task	Root Command	Notes
Create a BGP routing instance using an autonomous system number (ASN) and enter BGP router configuration mode.	router bgp	Enter this command in context configuration mode.
Allow the comparison of the Multi-Exit Discriminator (MED) for paths from BGP neighbors in different autonomous systems.	bestpath med always-compare	By default, the comparison of the MED is enabled for paths from BGP neighbors in the same autonomous system.
Specify a period of time that must pass before the BGP routing process drops sessions of directly connected external peers once the link used to reach them goes down.	fast-reset	By default, BGP sessions remain connected after the outbound interface goes down. BGP sessions are dropped after the BGP holdtime value, set through the timers command in BGP router configuration mode, is exceeded.

Table 8-1 Configure a BGP Routing Instance (*continued*)

Task	Root Command	Notes
Configure the local preference attribute for the BGP routes.	local-preference	The local preference value is applied to BGP routes that do not have the local-preference attribute assigned to them.
Log BGP neighbor resets.	log-neighbor-changes	
Configure the BGP routing process to use multiple equal-cost best paths for load-balancing outgoing traffic packets.	multi-paths	
Configure a fixed BGP router ID.	router-id	By default, the BGP router ID is the IP address of a loopback interface if one is configured. If a loopback interface is not configured, the interface with the highest IP address is used as the router ID. Peering sessions are reset when the router ID is changed.
Modify keepalive and holdtime timers for all BGP neighbors.	timers	By default, the keepalive timer is set to 60 seconds and the holdtime value is set to 180 seconds.
Configure IPv4 Multicast or Unicast Address Family Attributes	For the complete list of tasks used to configure IPv4 address family attributes, see the "Configure IPv4 Address Family Attributes for a BGP Routing Instance" section.	
Configure IPv6 Unicast Address Family Attributes	For the complete list of tasks used to configure IPv6 address family attributes, see the "Configure IPv6 Address Family Attributes for a BGP Routing Instance" section.	
Configure the BGP graceful restart characteristics.	For the complete list of tasks used to configure BGP graceful restart, see the "Configure Graceful Restart for a BGP Routing Instance" section.	
Configure BGP Route Reflection.	For the complete list of tasks used to configure BGP route reflection, see the "Configure BGP Route Reflection" section.	
Configure BGP confederations.	For the complete list of tasks used to configure BGP confederations, see the "Configure a BGP Confederation" section.	

Configure IPv4 Address Family Attributes for a BGP Routing Instance

To configure the IPv4 address family attributes for a BGP routing instance, perform the tasks described in Table 8-2. Enter all commands in BGP address family configuration mode, unless otherwise noted.

Table 8-2 Configure IPv4 Address Family Attributes for a BGP Routing Instance

Task	Root Command	Notes
Specify the use of standard IP Version 4 (IPv4) multicast or unicast address prefixes for the BGP routing instance, and access BGP address family configuration mode.	address-family ipv4	Enter this command in BGP router configuration mode.
Create an aggregate entry in the BGP database for the BGP address family.	aggregate-address	
Enable eBGP route dampening for the specified BGP address family.	dampening	
Configure the administrative distance values for a BGP address family.	distance	BGP uses distances to compare and prioritize routes. The lower the distance, the more preferred the route.
Enable route-flap statistics accounting for the BGP address family.	flap-statistics	

Table 8-2 Configure IPv4 Address Family Attributes for a BGP Routing Instance (*continued*)

Task	Root Command	Notes
Originate BGP routes that are advertised to peers.	network	
Redistribute routes learned through other protocols into the BGP routing process.	redistribute	
Assign a traffic index to routes installed for a BGP address family.	table-map	Traffic index counters are maintained on interfaces with traffic index accounting enabled. For more information about BGP attribute-based accounting, see the “Configuring BGP Attribute-Based Accounting” section in Chapter 12, “Routing Policy Configuration.”

Configure IPv6 Address Family Attributes for a BGP Routing Instance

To configure the IPv6 address family attributes for a BGP routing instance, perform the tasks described in Table 8-3. Enter all commands in BGP address family configuration mode, unless otherwise noted.

Table 8-3 Configure IPv6 Address Family Attributes for a BGP Routing Instance

Task	Root Command	Notes
Specify the use of standard IP Version 6 (IPv6) unicast address prefixes for the BGP routing instance, and access BGP address family configuration mode.	address-family ipv6 unicast	Enter this command in BGP router configuration mode.
Create an aggregate entry in the BGP database for the BGP address family.	aggregate-address	
Enable eBGP route dampening for the specified BGP address family.	dampening	
Configure the administrative distance values for a BGP address family.	distance	BGP uses distances to compare and prioritize routes. The lower the distance, the more preferred the route.
Enable route-flap statistics accounting for the BGP address family.	flap-statistics	
Originate BGP routes that are advertised to peers.	network	
Redistribute routes learned through other protocols into the BGP routing process.	redistribute	
Assign a traffic index to routes installed for a BGP address family.	table-map	Traffic index counters are maintained on interfaces with traffic index accounting enabled. For more information about BGP attribute-based accounting, see the “Configuring BGP Attribute-Based Accounting” section in Chapter 12, “Routing Policy Configuration.”

Configure Graceful Restart for a BGP Routing Instance

The graceful restart capability can be used by a BGP speaker to indicate its ability to preserve its forwarding state during a BGP restart. The BGP speaker can also convey to peers its intention of generating the end-of-Routing Information Base (RIB) marker upon the completion of its initial routing updates.

To configure the graceful restart characteristics for a BGP routing instance, perform the tasks described in Table 8-4. Enter all commands in BGP router configuration mode.

Table 8-4 Configure Graceful Restart for a BGP Routing Instance

Task	Root Command	Notes
Set the maximum amount of time that it will take for a local BGP peer to come up after it has been reset.	maximum restart-time	
Set the maximum amount of time the local BGP speaker retains routes it has previously received from a remote peer once that remote peer restarts the connection.	maximum retain-time	Any routes that have not been updated by the remote peer are deleted by the local peer after the local peer receives the end-of-RIB marker from the remote peer, or after the timer expires.
Set the maximum delay time for the BGP routing process after a reset has occurred before performing initial best path calculations.	maximum update-delay	Use this feature when all peers do not support a graceful restart, or when a peer may not send an end-of-RIB marker.

Configure BGP Route Reflection

If a BGP route reflector is configured, while it must have connections to all other BGP speakers in the AS, not all other BGP speakers must be fully meshed. When a BGP speaker in the AS receives messages from an external router, it is sufficient to advertise these routes only to the router reflector, which then re-advertises the routes to all other BGP speakers in the AS.

To configure BGP route reflection, perform the tasks described in Table 8-5. Enter all commands in BGP router configuration mode.

Table 8-5 Configure BGP Route Reflection

Task	Root Command	Notes
Enable client-to-client reflection.	client-to-client reflection	By default, routes are reflected between clients of a route reflector.
Disable client-to-client reflection.	client-to-client reflection	Use the no form of this command. Disable client-to-client reflection when you do not want routes that have been learned from one client to be reflected to other clients; for example, when clients are fully meshed.
Assign a separate cluster ID to each route reflector.	cluster-id	Use this command when there is more than one route reflector in a cluster.

Configure a BGP Confederation

To reduce iBGP mesh, you can divide an autonomous system into subautonomous systems grouped by a routing domain identifier. The AS and its subautonomous systems are part of a BGP confederation. Externally, the confederation looks like a single autonomous system.

To configure a BGP confederation, perform the tasks described in Table 8-6. Enter all commands in BGP router configuration mode.

Table 8-6 Configure a BGP Confederation

Task	Root Command	Notes
Configure a BGP confederation.	confederation identifier	
Configure the subautonomous systems that belong to the BGP confederation.	confederation peers	

Configuring BGP Neighbors and Neighbor Attributes

BGP speakers (BGP-enabled routers) that exchange inter-AS routing information are called BGP neighbors. BGP supports two kinds of neighbors: internal and external. Internal neighbors are in the same AS; external neighbors are in different autonomous systems. External neighbors must be adjacent to each other and share the same subnet, while internal neighbors may be located anywhere inside the same autonomous system.

To enable BGP speakers to effectively communicate with each other, each BGP speaker must be configured with information about its BGP neighbors.

To configure a BGP neighbors and other neighbor attributes, perform the tasks described in the following sections:

- Configure a BGP Neighbor
- Configure IPv4 Address Family Attributes for a BGP Neighbor
- Configure IPv6 Address Family Attributes for a BGP Neighbor
- Configure Graceful Restart for a BGP Neighbor

Configure a BGP Neighbor

To configure a BGP neighbor, perform the tasks described in Table 8-7. Enter all commands in BGP neighbor configuration mode, unless otherwise noted.

Table 8-7 Configure a BGP Neighbor

Task	Root Command	Notes
Create a BGP neighbor and access BGP neighbor configuration mode.	neighbor	Enter this command in BGP router configuration mode.
Advertise to a peer that this BGP speaker is willing to accept address prefix-based route filtering from the peer.	accept filter prefix-list	
Modify the minimal interval at which BGP routing updates are sent to the specified neighbor.	advertisement-interval	
Associate a description with the neighbor.	description	
Configure the maximum number of hops used to reach an eBGP neighbor when the neighbor is not directly connected.	ebgp-multihop	This command must be enabled for BGP connections to be established with neighbors that are not directly connected.
Enable the BGP time-to-live (TTL) security check in the kernel for the BGP neighbor.	enforce ttl	For the BGP TTL security check to function correctly, it must be enabled on both ends of an eBGP session. Enabling only one end causes the eBGP session to drop.
Configure the ASN that the BGP routing process uses to peer with the specified eBGP neighbor.	local-as	
Advertise the local peer address as the next-hop address.	next-hop-self	By default, when a BGP neighbor receives BGP routes from an eBGP neighbor, routes are sent to iBGP neighbors without changing the next-hop address.
Configure an encrypted Message Digest 5 (MD5) password for the BGP neighbor.	password	

Table 8-7 Configure a BGP Neighbor (*continued*)

Task	Root Command	Notes
Apply the attributes of a configured BGP peer group to one or more BGP neighbors.	peer-group	You can assign a neighbor can be assigned to a peer group only if the neighbor and the peer group is of the same type—external or internal BGP. If a neighbor belongs to a particular peer group, you cannot configure it to belong to another peer group. You must first explicitly delete the previous peer group membership before reconfiguring the peer membership. Attributes are inherited from the peer group to which a neighbor is assigned. The following BGP neighbor configuration mode commands represent attributes that you cannot customize per neighbor when the neighbor is assigned to a peer group: advertisement-interval, ebgp-multihop, local-as, send community, and timers. Attributes inherited from a peer group that you can customize per neighbor include those set by the following commands: description, password, send prefix, shutdown, and update-source.
Configure the ASN of the eBGP neighbor.	remote-as	
Send the community attribute to the specified eBGP neighbor.	send community	
Advertise to a BGP peer that this BGP speaker would like to send prefixed-based filtering to the peer.	send filter prefix-list	
Enable a BGP router to send MPLS labels with BGP IPv4 routes to a peer BGP router.	send label	You must configure this command on both the local router and the peer router in order for the routers to send IPv4 unicast routes with MPLS labels.
Administratively shut down a BGP session with the specified neighbor.	shutdown	This command temporarily shuts down a BGP session without removing a BGP neighbor from the configuration.
Configure the time interval, in seconds, during which an old MD5 password can co-exist with a new MD5 password for authentication.	timer password	Configuring the password timer interval affects only the BGP peers which have existing MD5 passwords replaced after this configuration is committed.
Modify keepalive and holdtime timers for a specific neighbor.	timers	Values set for a BGP neighbor override the values set for the BGP routing instance.
Specify the IP address of the interface used for BGP peering.	update-source	
Configure IPv4 multicast or unicast address family attributes.	For the complete list of tasks used to configure IPv4 address family attributes, see the "Configure IPv4 Address Family Attributes for a BGP Neighbor" section.	
Configure IPv6 unicast address family attributes.	For the complete list of tasks used to configure IPv6 address family attributes, see the "Configure IPv6 Address Family Attributes for a BGP Neighbor" section.	
Configure the graceful restart characteristics.	For the complete list of tasks used to configure BGP graceful restart, see the "Configure Graceful Restart for a BGP Neighbor" section.	

Configure IPv4 Address Family Attributes for a BGP Neighbor

To configure the IPv4 address family attributes for a BGP neighbor, perform the tasks described in Table 8-8. Enter all commands in BGP neighbor address family configuration mode, unless otherwise noted.

Table 8-8 Configure IPv4 Address Family Attributes for a BGP Neighbor

Task	Root Command	Notes
Specify the use of standard IP Version 4 (IPv4) multicast or unicast address prefixes for the neighbors in the BGP address family, and to access BGP neighbor address family configuration mode.	address-family ipv4	Enter this command in BGP neighbor configuration mode.
Filter BGP routing updates from or to the specified BGP neighbor address family.	as-path-list	
Advertise the default route of the specified address family, even when the default route is not installed in the BGP routing table, to a BGP neighbor.	default-originate	
Specify how the BGP routing process responds when the maximum number of prefixes sent by the BGP neighbor for the specified address family is exceeded.	maximum prefix	
Apply the attributes of a configured BGP peer group to one or more BGP neighbor address families.	peer-group	A BGP neighbor address family can belong to more than one peer group and you can modify it to belong to a different peer group without having to delete the previous peer group association first. Attributes are inherited from the peer group to which a BGP neighbor address family is assigned. The following commands in BGP neighbor address family configuration mode represent attributes that you cannot customize per address family once it is assigned to a peer group: as-path-list out, prefix-list out, remove-private-as, and route-map out. Attributes inherited from a peer group that you can customize per neighbor address family include those set by the following commands: as-path-list in, default-originate, maximum-prefix, prefix-list in, and route-map in.
Filter BGP routes from or to the specified neighbor address family.	prefix-list	
Remove ASNs from routes advertised to the specified BGP neighbor address family.	remove-private-as	
Apply a route map that modifies BGP attributes or filters BGP routes received from or sent to the BGP neighbor.	route-map	
Configure an iBGP neighbor as a route reflector client for a BGP address family.	route-reflector-client	

Configure IPv6 Address Family Attributes for a BGP Neighbor

To configure the IPv6 address family attributes for a BGP neighbor, perform the tasks described in Table 8-9. Enter all commands in BGP neighbor address family configuration mode, unless otherwise noted.

Table 8-9 Configure IPv6 Address Family Attributes for a BGP Neighbor

Task	Root Command	Notes
Specify the use of standard IPv6 unicast address prefixes for the neighbors in the BGP address family, and to access BGP neighbor address family configuration mode.	address-family ipv6 unicast	Enter this command in BGP neighbor configuration mode.
Filter BGP routing updates from or to the specified BGP neighbor address family.	as-path-list	
Advertise the default route of the specified address family, even when the default route is not installed in the BGP routing table, to a BGP neighbor.	default-originate	
Specify how the BGP routing process responds when the maximum number of prefixes sent by the BGP neighbor for the specified address family is exceeded.	maximum prefix	
Apply the attributes of a configured BGP peer group to one or more BGP neighbor address families.	peer-group	A BGP neighbor address family can belong to more than one peer group and you can modify it to belong to a different peer group without having to delete the previous peer group association first. Attributes are inherited from the peer group to which a BGP neighbor address family is assigned. The following commands in BGP neighbor address family configuration mode represent attributes that you cannot customize per address family once it is assigned to a peer group: as-path-list out, prefix-list out, remove-private-as, and route-map out. Attributes inherited from a peer group that you can customize per neighbor address family include those set by the following commands: as-path-list in, default-originate, maximum-prefix, prefix-list in, and route-map in.
Filter BGP routes from or to the specified neighbor address family.	prefix-list	
Remove ASNs from routes advertised to the specified BGP neighbor address family.	remove-private-as	
Apply a route map that modifies BGP attributes or filters BGP routes received from or sent to the BGP neighbor.	route-map	
Configure an iBGP neighbor as a route reflector client for a BGP address family.	route-reflector-client	

Configure Graceful Restart for a BGP Neighbor

The graceful restart capability can be used by a BGP speaker to indicate its ability to preserve its forwarding state during a BGP restart. The BGP speaker can also convey to peers its intention of generating the end-of-Routing Information Base (RIB) marker upon the completion of its initial routing updates.

To configure the graceful restart characteristics for a BGP neighbor, perform the tasks described in Table 8-10. Enter all commands in BGP neighbor configuration mode.

Table 8-10 Configure Graceful Restart for a BGP Neighbor

Task	Root Command	Notes
Set the maximum amount of time after the local BGP speaker has been reset before it attempts to reconnect with the remote peer.	maximum restart-time	
Set the maximum amount of time the local BGP speaker retains routes it has previously received from a remote peer once that remote peer restarts the connection.	maximum retain-time	Any routes that have not been updated by the remote peer are deleted by the local peer after the local peer receives the end-of-RIB marker from the remote peer, or after the timer expires.
Force a BGP neighbor to retain routes from an iBGP peer once the peer has restarted.	retain-ibgp-routes	By default, routes are not retained for an iBGP peer after the peer restarts unless all iBGP peers support a graceful restart; however, in some network topologies, it may be desirable and feasible to retain the routes for an iBGP peer, even if not all iBGP peers support a graceful restart.

Configuring BGP Peer Groups and Peer Group Attributes

BGP peer groups are helpful in cases where many BGP neighbors are configured with the same update policies. Grouping a large number of neighbors into one or more peer groups simplifies modifications to a configuration and makes the BGP update calculation process more efficient. A BGP peer group can be an eBGP or as an iBGP peer group.

To configure a BGP peer groups and other peer group attributes, perform the tasks described in the following sections:

- Configure a BGP Peer Group
- Configure IPv4 Address Family Attributes for a BGP Peer Group
- Configure IPv6 Address Family Attributes for a BGP Peer Group
- Apply Peer Group Attributes

Configure a BGP Peer Group

To configure a BGP peer group, perform the tasks described in Table 8-11. Enter all commands in BGP peer group configuration mode, unless otherwise noted.

Table 8-11 Configure a BGP Peer Group

Task	Root Command	Notes
Configure a BGP peer group, and enter BGP peer group configuration mode.	peer-group	Enter this command in BGP router configuration mode.
Modify the minimal interval at which BGP routing updates are sent to the specified BGP peer group.	advertisement-interval	

Table 8-11 Configure a BGP Peer Group (*continued*)

Task	Root Command	Notes
Associate a description with the peer group.	description	
Configure the maximum number of hops used to reach an eBGP neighbor when the BGP peer group is not directly connected.	ebgp-multihop	This command must be enabled for BGP connections to be established with neighbors that are not directly connected.
Enable the BGP TTL security check in the kernel for the BGP peer group.	enforce ttl	For the BGP TTL security check to function correctly, it must be enabled on both ends of an eBGP session. Enabling only one end causes the eBGP session to drop.
Advertise the local peer address as the next-hop address.	next-hop-self	
Configure an encrypted MD5 password for the BGP peer group.	password	
Send the community attribute to the specified BGP peer group.	send community	
Enable a flapping peer to be temporarily suppressed for a configurable amount of time.	session-dampening	This command is per peer and peer-group based. If the peer is member of a peer group, the command is inherited from the peer-group and can be customized in the peer configuration. The main benefit of this feature is to avoid flapping peers from using system resources, and also to reduce routing churn induced by a flapping peer.
Administratively shut down a BGP session with the specified peer group.	shutdown	This command temporarily shuts down a BGP session without removing a BGP peer group from the configuration.
Modify keepalive and holdtime timers for a peer group.	timers	
Specify the IP address of the interface used for BGP peering.	update-source	By default, when a BGP peer group receives BGP routes from an eBGP peer group, routes are sent to iBGP neighbors without changing the next-hop address.
Configure IPv4 multicast or unicast address family attributes.	For the complete list of tasks used to configure IPv4 address family attributes, see the "Configure IPv4 Address Family Attributes for a BGP Peer Group" section.	
Configure IPv6 unicast address family attributes.	For the complete list of tasks used to configure IPv6 address family attributes, see the "Configure IPv6 Address Family Attributes for a BGP Peer Group" section.	

Configure IPv4 Address Family Attributes for a BGP Peer Group

To configure IPv4 address family attributes for a BGP peer group, perform the tasks described in Table 8-12. Enter all commands in BGP peer group address family configuration mode, unless otherwise noted.

Table 8-12 Configure IPv4 Address Family Attributes for a BGP Peer Group

Task	Root Command	Notes
Specify the use of standard IPv4 multicast or unicast address prefixes for peer groups in the BGP peer groups address family, and enter BGP peer group address family configuration mode.	address-family ipv4	Enter this command in BGP peer group configuration mode.
Filter BGP routing updates from or to the specified BGP neighbor address family.	as-path-list	
Advertise the default route of the specified address family, even when the default route is not installed in the BGP routing table, to a BGP neighbor.	default-originate	
Specify how the BGP address family responds when the maximum number of prefixes sent by the BGP peer group for the specified address family is exceeded.	maximum prefix	
Filter BGP routes from the peer group for the specified address family.	prefix-list	
Remove ASNs from routes advertised to the specified BGP peer group address family.	remove-private-as	
Apply a route map that modifies BGP attributes or filters BGP routes received from or sent to the specified peer group address family.	route-map	
Configure an iBGP peer group as a route reflector client for a BGP address family.	route-reflector-client	

Configure IPv6 Address Family Attributes for a BGP Peer Group

To configure IPv6 address family attributes for a BGP peer group, perform the tasks described in Table 8-13. Enter all commands in BGP peer group address family configuration mode, unless otherwise noted.

Table 8-13 Configure IPv6 Address Family Attributes for a BGP Peer Group

Task	Root Command	Notes
Specify the use of standard IPv6 unicast address prefixes for peer groups in the BGP peer groups address family, and enter BGP peer group address family configuration mode.	address-family ipv6 unicast	Enter this command in BGP peer group configuration mode.
Filter BGP routing updates from or to the specified BGP neighbor address family.	as-path-list	
Advertise the default route of the specified address family, even when the default route is not installed in the BGP routing table, to a BGP neighbor.	default-originate	
Specify how the BGP address family responds when the maximum number of prefixes sent by the BGP peer group for the specified address family is exceeded.	maximum prefix	
Filter BGP routes from the peer group for the specified address family.	prefix-list	
Remove ASNs from routes advertised to the specified BGP peer group address family.	remove-private-as	

Table 8-13 Configure IPv6 Address Family Attributes for a BGP Peer Group (*continued*)

Task	Root Command	Notes
Apply a route map that modifies BGP attributes or filters BGP routes received from or sent to the specified peer group address family.	route-map	
Configure an iBGP peer group as a route reflector client for a BGP address family.	route-reflector-client	

Apply Peer Group Attributes

A BGP neighbor, or BGP neighbor address family, can inherit attributes from the peer group to which a neighbor is assigned. The following BGP neighbor configuration mode commands represent attributes that cannot be customized per neighbor when the neighbor is assigned to a peer group: **advertisement-interval**, **ebgp-multihop**, **local-as**, **send community**, and **timers**. Attributes inherited from a peer group that can be customized per neighbor include those set by the following commands: **description**, **password**, **send prefix**, **shutdown**, and **update-source**.

To apply peer group attributes, perform the tasks described in Table 8-14.

Table 8-14 Apply Peer Group Attributes

Task	Root Command	Notes
Apply peer group attributes to a BGP neighbor.	peer-group	Enter this command in BGP neighbor configuration mode.
Apply peer group attributes to a BGP neighbor address family.	peer-group	Enter this command in BGP peer group configuration mode.

Configuration Examples

This section provides BGP configuration examples in the following sections:

- Basic BGP
- iMBGP Peer
- iMBGP Peer Group
- eMBGP Peer
- eMBGP Peer Group

Basic BGP

The following example show the minimum commands needed to configure BGP:

```
[local]Router_A#config
[local]Router_A(config)#context local
[local]Router_A(config-ctx)#router bgp 64001
[local]Router_A(config-bgp)#router-id 1.1.1.71
[local]Router_A(config-bgp)#address-family ipv4 unicast
[local]Router_A(config-bgp-af)#redistribute static
[local]Router_A(config-bgp-af)#exit
```

```

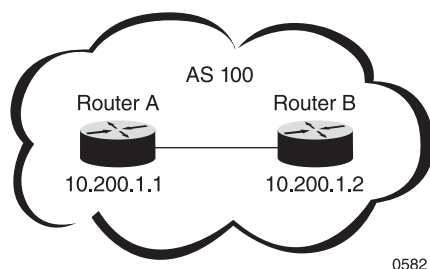
[local]Router_A(config-bgp)#peer-group iBGP internal
[local]Router_A(config-bgp-peer-group)#next-hop-self
[local]Redback(config-bgp-peer-group)#update-source loopback0
[local]Redback(config-bgp-peer-group)#address-family ipv4 unicast
[local]Redback(config-bgp-peer-af)#exit
[local]Redback(config-bgp-peer-group)#exit
[local]Redback(config-bgp)#peer-group customer-routes external
[local]Redback(config-bgp-peer-group)#address-family ipv4 unicast
[local]Redback(config-bgp-peer-af)#route-map rmap1 out
[local]Redback(config-bgp-peer-af)#exit
[local]Redback(config-bgp-peer-group)#exit
[local]Redback(config-bgp)#neighbor 1.1.1.1 internal
[local]Redback(config-bgp-neighbor)#peer-group ibgp
[local]Redback(config-bgp-neighbor)#exit
[local]Redback(config-bgp)#neighbor 2.2.2.2 external
[local]Redback(config-bgp-neighbor)#remote-as 200
[local]Redback(config-bgp-neighbor)#peer-group customer-routes
[local]Redback(config-bgp-neighbor)#address-family ipv4 unicast
[local]Redback(config-bgp-af)#prefix-list bar in
[local]Redback(config-bgp-af)#route-map foo2 in
[local]Redback(config-bgp-af)#exit
[local]Redback(config-bgp-neighbor)#exit
[local]Redback(config-bgp)#neighbor 3.3.3.3 external
[local]Redback(config-bgp-neighbor)#remote-as 300
[local]Redback(config-bgp-neighbor)#address-family ipv4 unicast
[local]Redback(config-bgp-af)#prefix-list bar in
[local]Redback(config-bgp-af)#route-map foo3 out

```

iMBGP Peer

The following example configures two iMBGP peers. Figure 8-4 shows the network topology for the configuration.

Figure 8-4 iMBGP Peer Topology



The configuration for Router_A is as follows:

```

[local]Router_A#config
[local]Router_A(config)#context local
[local]Router_A(config-ctx)#interface lo1 loopback
[local]Router_A(config-if)#ip address 10.200.1.1/32
[local]Router_A(config-if)#exit

```

```
[local]Router_A(config-ctx)#router bgp 100
[local]Router_A(config-bgp)#router-id 10.200.1.1
[local]Router_A(config-bgp)#neighbor 10.200.1.2 internal
[local]Router_A(config-bgp-neighbor)#update-source lo1
[local]Router_A(config-bgp-neighbor)#address-family ipv4 multicast
[local]Router_A(config-bgp-af)#exit
[local]Router_A(config-bgp-neighbor)#exit
[local]Router_A(config-bgp)#exit
[local]Router_A(config-ctx)#ip route 10.200.1.2/32 102.1.1.2
```

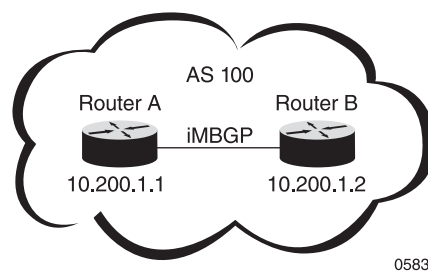
The configuration for Router_B is as follows:

```
[local]Router_B#config
[local]Router_B(config)#context local
[local]Router_B(config-ctx)#interface lo1 loopback
[local]Router_B(config-if)#ip address 10.200.1.2/32
[local]Router_B(config-if)#exit
[local]Router_B(config-ctx)#router bgp 100
[local]Router_B(config-bgp)#router-id 10.200.1.2
[local]Router_B(config-bgp)#neighbor 10.200.1.1 internal
[local]Router_B(config-bgp-neighbor)#update-source lo1
[local]Router_B(config-bgp-neighbor)#address-family ipv4 multicast
[local]Router_B(config-bgp-af)#exit
[local]Router_B(config-bgp-neighbor)#exit
[local]Router_B(config-bgp)#exit
[local]Router_B(config-ctx)#ip route 10.200.1.1/32 102.1.1.1
```

iMBGP Peer Group

The following example configures an iMBGP peer group for two iMBGP peers. Figure 8-5 shows the network topology for the configuration.

Figure 8-5 iMBGP Peer Group Topology



The configuration for Router_A is as follows:

```
[local]Router_A#config
[local]Router_A(config)#context local
[local]Router_A(config-ctx)#interface lo1 loopback
[local]Router_A(config-if)#ip address 10.200.1.1/32
[local]Router_A(config-if)#exit
[local]Router_A(config-ctx)#router bgp 100
[local]Router_A(config-bgp)#router-id 10.200.1.1
```

```
[local]Router_A(config-bgp)#address-family ipv4 multicast
[local]Router_A(config-bgp-af)#exit
[local]Router_A(config-bgp)#peer-group iMBGP internal
[local]Router_A(config-bgp-peer-group)#update-source lo1
[local]Router_A(config-bgp-peer-group)#address-family ipv4 multicast
[local]Router_A(config-bgp-peer-af)#exit
[local]Router_B(config-bgp-peer-group)#exit
[local]Router_A(config-bgp)#neighbor 10.200.1.2 internal
[local]Router_A(config-bgp-neighbor)#peer-group iMBGP
```

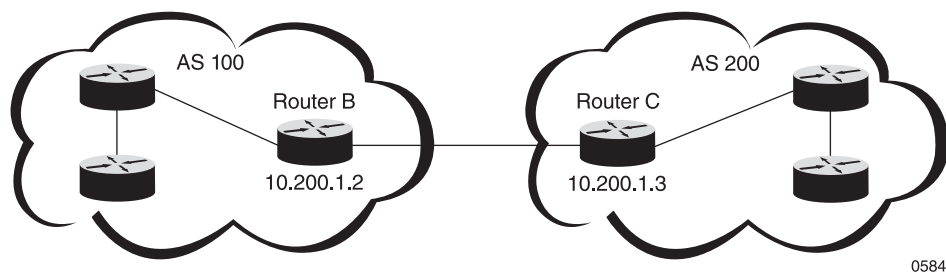
The configuration for Router_B is as follows:

```
[local]Router_B#config
[local]Router_B(config)#context local
[local]Router_B(config-ctx)#interface lo1 loopback
[local]Router_B(config-if)#ip address 10.200.1.2/32
[local]Router_B(config-if)#exit
[local]Router_B(config-ctx)#router bgp 100
[local]Router_B(config-bgp)#router-id 10.200.1.2
[local]Router_B(config-bgp)#address-family ipv4 multicast
[local]Router_B(config-bgp-af)#exit
[local]Router_B(config-bgp)#peer-group iMBGP internal
[local]Router_B(config-bgp-peer-group)#update-source lo1
[local]Router_B(config-bgp-peer-group)#address-family ipv4 multicast
[local]Router_B(config-bgp-peer-af)#exit
[local]Router_B(config-bgp-peer-group)#exit
[local]Router_B(config-bgp)#neighbor 10.200.1.1 internal
[local]Router_B(config-bgp-neighbor)#peer-group iMBGP
```

eMBGP Peer

The following example configures two eMBGP peers. Figure 8-6 shows the network topology for the configuration.

Figure 8-6 eMBGP Peer Network Topology



The configuration for Router_B is as follows:

```
[local]Router_B#config
[local]Router_B(config)#context local
[local]Router_B(config-ctx)#interface lo1 loopback
[local]Router_B(config-if)#ip address 10.200.1.2/32
[local]Router_B(config-if)#exit
```

```
[local]Router_B(config-ctx)#router bgp 100
[local]Router_B(config-bgp)#router-id 10.200.1.2
[local]Router_B(config-bgp)#neighbor 10.200.1.3 external
[local]Router_B(config-bgp-neighbor)#remote-as 200
[local]Router_B(config-bgp-neighbor)#ebgp-multihop 10
[local]Router_B(config-bgp-neighbor)#update-source lo1
[local]Router_B(config-bgp-neighbor)#address-family ipv4 multicast
```

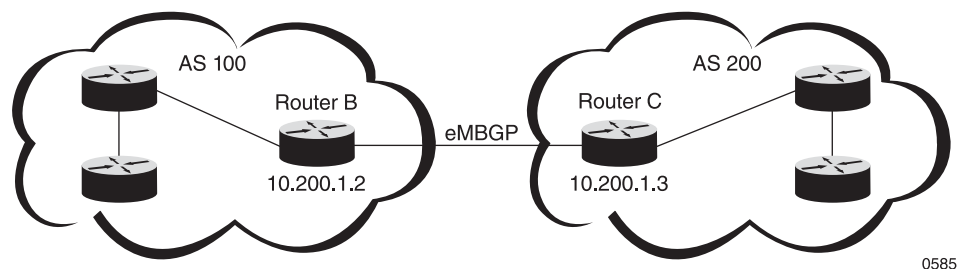
The configuration for Router_C is as follows:

```
[local]Router_C#config
[local]Router_C(config)#context local
[local]Router_C(config-ctx)#interface lo1 loopback
[local]Router_C(config-if)#ip address 10.200.1.3/32
[local]Router_C(config-if)#exit
[local]Router_C(config-ctx)#router bgp 100
[local]Router_C(config-bgp)#router-id 10.200.1.2
[local]Router_C(config-bgp)#neighbor 10.200.1.1 internal
[local]Router_C(config-bgp-neighbor)#remote-as 100
[local]Router_C(config-bgp-neighbor)#ebgp-multihop 10
[local]Router_C(config-bgp-neighbor)#update-source lo1
[local]Router_C(config-bgp-neighbor)#address-family ipv4 multicast
```

eMBGP Peer Group

The following example configures an eMBGP peer group for two eMBGP peers. Figure 8-7 shows the network topology for the configuration.

Figure 8-7 eMBGP Peer Group Network Topology



0585

The configuration for Router_B is as follows:

```
[local]Router_B#config
[local]Router_B(config)#context local
[local]Router_B(config-ctx)#interface lo1 loopback
[local]Router_B(config-if)#ip address 10.200.1.2/32
[local]Router_B(config-if)#exit
[local]Router_B(config-ctx)#router bgp 100
[local]Router_B(config-bgp)#router-id 10.200.1.2
[local]Router_B(config-bgp)#address-family ipv4 multicast
[local]Router_B(config-bgp-af)#exit
[local]Router_B(config-bgp)#peer-group eMBGP external
```

```
[local]Router_B(config-bgp-peer-group)#ebgp-multihop 10
[local]Router_B(config-bgp-peer-group)#update-source lo1
[local]Router_B(config-bgp-peer-group)#address-family ipv4 multicast
[local]Router_B(config-bgp-peer-af)#exit
[local]Router_B(config-bgp-peer-group)#neighbor 10.200.1.3 external
[local]Router_B(config-bgp-neighbor)#remote-as 200
[local]Router_B(config-bgp-neighbor)#peer-group eMBGP
```

The configuration for Router_C is as follows:

```
[local]Router_C#config
[local]Router_C(config)#context local
[local]Router_C(config-ctx)#interface lo1 loopback
[local]Router_C(config-if)#ip address 10.200.1.3/32
[local]Router_C(config-if)#exit
[local]Router_C(config-ctx)#router bgp 200
[local]Router_C(config-bgp)#router-id 10.200.1.3
[local]Router_C(config-bgp)#address-family ipv4 multicast
[local]Router_C(config-bgp-af)#exit
[local]Router_C(config-bgp)#peer-group eMBGP external
[local]Router_C(config-bgp-peer-group)#ebgp-multihop 10
[local]Router_C(config-bgp-peer-group)#update-source lo1
[local]Router_C(config-bgp-peer-group)#address-family ipv4 multicast
[local]Router_C(config-bgp-peer-af)#exit
[local]Router_C(config-bgp-peer-group)#neighbor 10.200.1.2 external
[local]Router_C(config-bgp-neighbor)#remote-as 100
[local]Router_C(config-bgp-neighbor)#peer-group eMBGP
```


Command Descriptions

This section describes the syntax and usage guidelines for the commands used to configure BGP features. The commands are presented in alphabetical order.

accept filter prefix-list	maximum update-delay
address-family ipv4	multi-paths
address-family ipv6 unicast	neighbor
advertisement-interval	network
aggregate-address	next-hop-self
asloop-in	password
as-override	peer-group
as-path-list	prefix-list
bestpath med always-compare	redistribute
client-to-client reflection	remote-as
cluster-id	remove-private-as
confederation identifier	retain-ibgp-routes
confederation peers	route-map
dampening	route-origin
default-originate	router bgp
description	route-reflector-client
distance	router-id
ebgp-multihop	send community
enforce ttl	send ext-community
fast-reset	send filter prefix-list
flap-statistics	send label
local-as	session-dampening
local-preference	shutdown
log-neighbor-changes	table-map
maximum prefix	timer password
maximum restart-time	timers
maximum retain-time	update-source

accept filter prefix-list

accept filter prefix-list

no accept filter prefix-list

Purpose

Advertises to a Border Gateway Protocol (BGP) peer that a BGP speaker can accept address prefix-based route filtering from a peer.

Command Mode

BGP neighbor configuration

Syntax Description

This command has no keywords or arguments.

Default

The command is disabled.

Usage Guidelines

Use the **accept filter prefix-list** command to advertise to a BGP peer that a BGP speaker can accept address prefix-based route filtering from a peer. Use this command to save resources and avoid the generation, transmission, and processing of unnecessary routing updates.

When this command is enabled, and if the BGP peer advertises its preference to send address prefixed-based filtering (through the **send filter prefix-list** command in BGP neighbor configuration mode), the remote peer sends its inbound address prefix-based filtering to the local BGP speaker. The local BGP speaker uses the received address prefix-based filtering along with its local routing policies to determine whether or not routes should be advertised to the peer.

Note This command cannot be enabled on a BGP neighbor that is part of a peer group because this feature cannot be customized for individual members inside of a peer group.

Use the **show bgp neighbor ip-address received prefix-filter** command to display address prefix-based route filtering configuration information.

Use the **no** form of this command to disable a BGP speaker from accepting route filtering from a peer.

For further information, see the Internet Drafts, *Cooperative Route Filtering Capability for BGP-4*, draft-ietf-idr-route-filter-03.txt, and *Address Prefix Based Outbound Route Filter for BGP-4*, draft-chen-bgp-prefix-orf-02.txt.

Examples

The following example enables the router to accept address prefix-based route filtering from the BGP peer at IP address 10.1.1.1:

```
[local]Redback(config-bgp)#neighbor 10.1.1.1 external
[local]Redback(config-bgp-neighbor)#accept filter prefix-list
```

Related Commands

`prefix-list`
`send filter prefix-list`

address-family ipv4

address-family ipv4 {multicast | unicast}
no address-family ipv4 {multicast | unicast}

Purpose

When entered in BGP router configuration mode, specifies the use of standard IP Version 4 (IPv4) multicast or unicast address prefixes for the BGP routing instance and enters BGP address family configuration mode.

When entered in BGP neighbor configuration mode, this command specifies the use of IPv4 multicast or unicast address prefixes for the specified BGP neighbor, and enters BGP neighbor address family configuration mode.

When entered in BGP peer group configuration mode, this command specifies the use of IPv4 multicast or unicast address prefixes for the specified BGP peer group, and enters BGP peer group address family configuration mode.

Command Mode

BGP neighbor configuration
BGP peer group configuration
BGP router configuration

Syntax Description

multicast	Specifies multicast address prefixes.
unicast	Specifies unicast address prefixes.

Default

When entered in BGP router configuration mode, this command has no default setting.

When entered in BGP neighbor configuration mode or BGP peer group configuration mode, address prefixes are set to IPv4 multicast.

Usage Guidelines

Use the **address-family ipv4** command in BGP router configuration mode to specify the use of standard IPv4 unicast or multicast address prefixes for the BGP routing instance, and to enter BGP address family configuration mode. The **aggregate-address**, **dampening**, **flap-statistics**, **network**, and **redistribute** commands are available in BGP address family configuration mode. Routes are sent to BGP neighbors that have corresponding address family attributes.

Use the **address-family ipv4** command in BGP neighbor configuration mode to specify the use of IPv4 unicast or multicast address prefixes for the BGP neighbor, and to enter BGP neighbor address family configuration mode. The commands that configure the routing policies used with neighbors, **as-path-list**, **default-originate**, **prefix-list**, **maximum prefix**, **remove-private-as**, **route-map**, and **route-reflector-client**, are available in BGP neighbor address family configuration mode. To be established a BGP session, you must configure a neighbor with corresponding address family attributes.

Use the **address-family ipv4** command in BGP peer group configuration mode to specify the use of IPv4 multicast or unicast address prefixes, and to enter BGP peer group address family configuration mode. The commands that configure routing policies used with members of a peer group, **as-path-list**, **default-originate**, **prefix-list**, **maximum prefix**, **remove-private-as**, and **route-map**, are available in BGP peer group address family configuration mode.

Use the **no** form of this command to remove BGP address family attributes for the specified BGP instance or neighbor.

Examples

The following example illustrates the BGP routing process running in autonomous system 100. In this example, the network 20.0.0.0/8 advertises BGP routing updates which are sent in unicast mode, while Open Shortest Path First (OSPF) routes are redistributed into the BGP routing domain as multicast routes. The SmartEdge router is a unicast BGP peer with the neighbor at IP address 102.210.210.1 and is a multicast peer with the neighbor at IP address 68.68.68.68. Inbound prefix list **perf1** and outbound route map **map2** are applied in unicast mode to the neighbor at IP address 102.210.210.1.

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#router bgp 100
[local]Redback(config-bgp)#address-family ipv4 unicast
[local]Redback(config-bgp-af)#network 20.0.0.0/8
[local]Redback(config-bgp-af)#exit
[local]Redback(config-bgp)#address-family ipv4 multicast
[local]Redback(config-bgp-af)#redistribute ospf 100
[local]Redback(config-bgp-af)#exit
[local]Redback(config-bgp)#neighbor 102.210.210.1 external
[local]Redback(config-bgp-neighbor)#address-family ipv4 unicast
[local]Redback(config-bgp-af)#prefix-list perf1 in
[local]Redback(config-bgp-af)#route-map map2 out
[local]Redback(config-bgp-af)#exit
[local]Redback(config-bgp-neighbor)#exit
[local]Redback(config-bgp)#neighbor 68.68.68.68 external
[local]Redback(config-bgp-neighbor)#remote-as 300
[local]Redback(config-bgp-neighbor)#address-family ipv4 multicast
```

Related Commands

- as-path-list**
- default-originate**
- maximum prefix**
- network**
- prefix-list**
- redistribute**
- remove-private-as**
- route-map**
- route-reflector-client**

address-family ipv6 unicast

address-family ipv6 unicast

no address-family ipv6 unicast

Purpose

When entered in BGP router configuration mode, specifies the use of IP Version 6 (IPv6) unicast address prefixes for the Border Gateway Protocol (BGP) routing instance and enters BGP address family configuration mode.

When entered in BGP neighbor configuration mode, this command specifies the use of IPv6 unicast address prefixes for the specified BGP neighbor, and enters BGP neighbor address family configuration mode.

When entered in BGP peer group configuration mode, this command specifies the use of IPv6 unicast address prefixes for the specified BGP peer group, and enters BGP peer group address family configuration mode.

Command Mode

BGP neighbor configuration
BGP peer group configuration
BGP router configuration

Syntax Description

This command has no keywords or arguments.

Default

When entered in BGP router configuration mode, this command has no default setting.

When entered in BGP neighbor configuration mode or BGP peer group configuration mode, address prefixes are set to IPv6 unicast.

Usage Guidelines

Use the **address-family ipv6 unicast** command in BGP router configuration mode to specify the use of standard IPv6 unicast address prefixes for the BGP routing instance, and to enter BGP address family configuration mode. Routes are sent to BGP neighbors that have corresponding address family attributes.

Use the **address-family ipv6 unicast** command in BGP neighbor configuration mode to specify the use of IPv6 unicast address prefixes for the BGP neighbor, and to enter BGP neighbor address family configuration mode. To establish a BGP session, you must configure a neighbor with corresponding address family attributes.

Use the **address-family ipv6 unicast** command in BGP peer group configuration mode to specify the use of IPv6 unicast address prefixes, and to enter BGP peer group address family configuration mode.

Use the **no** form of this command to remove BGP address family attributes for the specified BGP instance or neighbor.

Examples

The following example illustrates the BGP routing process running in autonomous system 100. In this example, the network, AF26:3344:ADF7:77B5::2000/128, advertises BGP routing updates which are sent in IPv6 unicast mode.

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#router bgp 100
[local]Redback(config-bgp)#address-family ipv6 unicast
[local]Redback(config-bgp-af)#network AF26:3344:ADF7:77B5::2000/128
[local]Redback(config-bgp-af)#
```

Related Commands

- as-path-list**
- default-originate**
- maximum prefix**
- network**
- prefix-list**
- redistribute**
- remove-private-as**
- route-map**
- route-reflector-client**

advertisement-interval

advertisement-interval *interval*

no advertisement-interval *interval*

Purpose

Modifies the minimum interval at which Border Gateway Protocol (BGP) routing updates are sent to the specified neighbor or members of the specified peer group.

Command Mode

BGP neighbor configuration
BGP peer group configuration

Syntax Description

<i>interval</i>	Minimum interval, in seconds, at which BGP routing updates are sent. The range of values is 1 to 600. For external BGP (eBGP), the default value is 30. For internal BGP (iBGP), the default value is 5.

Default

The default advertisement interval is 30 seconds for eBGP and 5 seconds for iBGP.

Usage Guidelines

Use the **advertisement-interval** command to set the minimum interval at which BGP routing updates are sent to the specified neighbor or members of the specified peer group.

Note This command cannot be enabled if the neighbor belongs to a peer group.

Use the **no** form of this command to restore the advertisement interval to its default value.

Examples

The following example sends unicast routing updates every 60 seconds to the neighbor at IP address 102.210.210.1:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#router bgp 64001
[local]Redback(config-bgp)#neighbor 102.210.210.1 external
[local]Redback(config-bgp-neighbor)#advertisement-interval 60
[local]Redback(config-bgp-neighbor)#address-family ipv4 unicast
[local]Redback(config-bgp-af)#
```


The following example displays output from the **show bgp neighbor** command for the configuration in the previous example:

```
[local]Redback>show bgp neighbor 10.100.1.102
```

```
BGP neighbor: 102.210.210.1, remote AS: 64001, internal link
```

```
Version: 4, router identifier: 102.210.210.1
```

```
State: Established for 00:30:10
```

```
.
```

```
.
```

```
.
```

```
Minimum time between advertisement runs: 60 secs
```

Related Commands

timers

aggregate-address

aggregate-address {*ip-addr/prefix-length* | *ipv6-addr/prefix-length*} [**as-set**]
[**component-map** *map-name*] [**attribute-map** *map-name*]

no aggregate-address {*ip-addr/prefix-length* | *ipv6-addr/prefix-length*} [**as-set**]
[**component-map** *map-name*] [**attribute-map** *map-name*]

Purpose

Creates an aggregate entry in the Border Gateway Protocol (BGP) database for the BGP address family.

Command Mode

BGP address family configuration

Syntax Description

<i>ip-addr/prefix-length</i>	Specifies the IP address, in the form <i>A.B.C.D</i> , and the prefix length, separated by the slash (/) character. The range of values for the <i>prefix-length</i> argument is 0 to 32.
<i>ipv6-addr/prefix-length</i>	Specifies the IP Version 6 (IPv6) address, in the form <i>A:B:C:D:E:F:G:H</i> , and the prefix length, separated by the slash (/) character. The range of values for the <i>prefix-length</i> argument is 0 to 128.
as-set	Optional. Generates autonomous system (AS) set path information.
component-map <i>map-name</i>	Optional. Name of the route map used to select the routes to create an aggregate entry.
attribute-map <i>map-name</i>	Optional. Name of the route map used to set the attribute of the aggregate route.

Default

The command is disabled.

Usage Guidelines

Use the **aggregate-address** command to create an aggregate entry in a unicast or multicast BGP database for the BGP address family. You can implement aggregate routing in BGP by either redistributing an aggregate route into the BGP routing domain or by using this feature.

Use this command with no arguments to create an aggregate entry in the BGP routing table when any more-specific BGP routes that fall into the specified range are available. The origin of the aggregate route is advertised as the local autonomous system.

Use the **as-set** keyword to create an aggregate entry in the BGP routing table and to advertise the origin of the aggregate route as an AS_SET consisting of all elements contained in all paths that are being summarized. Do not use this form of the command when aggregating many paths, because this route must

be continually updated as autonomous system path reachability information for the summarized routes changes.

Use the **no** form of this command to remove an aggregate entry.

Examples

The following example creates an aggregate entry in the BGP routing table as long as there are more-specific routes in the 11.0.0.0/8 address block:

```
[local]Redack(config)#router bgp 64000  
[local]Redack(config-bgp)#address-family ipv4 unicast  
[local]Redack(config-bgp-af)#aggregate-address 11.0.0.0/8
```

Related Commands

network

asloop-in

asloop-in *loop-count*

no asloop-in

Purpose

Disables the AS_PATH loop detection by accepting a route advertisement that contains the local autonomous system number (ASN) in the AS_PATH attribute.

Command Mode

BGP neighbor configuration

Syntax Description

<i>loop-count</i>	Number of times that the local ASN can appear in the AS_PATH attribute. Valid values are 1 to 10.

Default

The AS_PATH loop detection is enabled.

Usage Guidelines

Use the **asloop-in** command to disable the AS_PATH loop detection by accepting a route advertisement that contains the local ASN in the AS_PATH attribute.

Because enabling the **asloop-in** command disables AS_PATH loop detection, it must only be used for specific applications that require this type of behavior, and in situations with strict network control. One application for this command is the Border Gateway Protocol/Multiprotocol Label Switching Virtual Private Network (BGP/MPLS VPN) hub-and-spoke configuration, in which a hub provider edge (PE) router may receive routes containing its own ASN from a hub customer edge (CE) router. To disable AS_PATH loop detection, use the **asloop-in** command on the exporting context of the hub PE router.

Note The **asloop-in** command is useful only when Border Gateway Protocol is used for PE-to-CE routing.

Note For a CE router to send a route advertisement back to the PE router from which the route is learned, the CE router must be configured as a BGP peer with the PE router configured as a member of the peer group. By default, routes are not sent back to the neighbor autonomous system (AS) from where they are received.

Use the **no** form of this command to enable the AS_PATH loop detection.

Examples

The following example enables BGP on a PE router to accept routes with the ASN 100 in the AS_PATH attribute up to 2 times from peer 2.2.2.1:

```
[local]Redback(config)#context local
```

```
[local]Redback(config-ctx)#router bgp 100
[local]Redback(config-bgp)#exit
[local]Redback(config-ctx)#context bar vpn-rd 20.21.22.23:200
[local]Redback(config-ctx)#router bgp vpn
[local]Redback(config-bgp)#address-family ipv4 unicast
[local]Redback(config-bgp-af)#export route-target 300:400
[local]Redback(config-bgp-af)#exit
[local]Redback(config-bgp)#neighbor 2.2.2.1 external
[local]Redback(config-bgp-neighbor)#remote-as 64001
[local]Redback(config-bgp-neighbor)#asloop-in 2
[local]Redback(config-bgp-neighbor)#address-family ipv4 unicast
```

Related Commands

as-override
peer-group

as-override

as-override

no as-override

Purpose

Replaces all occurrences of a peer's autonomous system number (ASN) in the AS_PATH attribute of a route with the local ASN, when advertising the route to the peer.

Command Mode

BGP neighbor configuration

Syntax Description

This command has no keywords or arguments.

Default

The peer's ASN is not replaced by the local ASN.

Usage Guidelines

Use the **as-override** command to replace all occurrences of a peer's ASN in the AS_PATH attribute of a route with the local ASN, when advertising the route to the peer.

When multiple Virtual Private Network (VPN) sites share the same ASN, enabling the AS override feature allows routes originating from an autonomous system (AS) to be accepted by a router residing in the same AS. By default, the receiving router rejects the received route advertisement if the AS_PATH attribute shows that the route originated from its own AS to prevent routing loops.

Note The **as-override** command is useful only when Border Gateway Protocol (BGP) is used for provider edge-to-customer edge (PE-to-CE) routing.

Note Enabling the AS override feature may result in route loops. This feature should only be used for specific applications that require this type of behavior, and in situations with strict network control.

Note The **as-override** command can only be used in VPN contexts.

Use the **no** form of this command to disable the AS override feature.

Examples

The following example replaces all occurrences of ASN 64001 in the AS_PATH attribute with the local router's ASN 100 when advertising the routes to peer 1.1.1.1:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#router bgp 100
[local]Redback(config-ctx)#exit
[local]Redback(config)#context foo vpn-rd 10.11.12.13:100
```

```
[local]Redback(config-ctx)#router bgp vpn  
[local]Redback(config-bgp)#neighbor 1.1.1.1 external  
[local]Redback(config-bgp-neighbor)#remote-as 64001  
[local]Redback(config-bgp-neighbor)#as-override  
[local]Redback(config-bgp-neighbor)#address-family ipv4 unicast
```

Related Commands

asloop-in
route-origin
send label

as-path-list

as-path-list *apl-name* {**in** | **out**}

no as-path-list *apl-name* {**in** | **out**}

Purpose

Filters Border Gateway Protocol (BGP) routing updates from or to the specified BGP neighbor or peer group address family.

Command Mode

BGP neighbor address family configuration
BGP peer group address family configuration

Syntax Description

<i>apl-name</i>	Autonomous system (AS) path list name.
in	Applies the filter to incoming routes from the BGP neighbor.
out	Applies the filter to outgoing routes to the BGP neighbor. This keyword only applies in BGP neighbor address family configuration mode.

Default

None

Usage Guidelines

Use the **as-path-list** command to filter the BGP routing updates from or to the specified BGP neighbor or peer group address family. Use the **in** keyword to filter the BGP incoming routes from the specified BGP neighbor or peer group. Use the **out** keyword to filter outgoing routes to the BGP neighbor or peer group. The content of the filter list is based on the AS path, which is defined through the **as-path-list** command in context configuration mode.

Note The **out** keyword cannot be enabled on a BGP neighbor that is part of a peer group because this feature cannot be customized for individual members inside of a peer group.



Caution Risk of unfiltered routes. If a filter list is applied to a BGP neighbor, but there is no corresponding as path list in context configuration mode, routes are not filtered. To reduce the risk, verify that an AS path list has been configured before applying it to a BGP neighbor.

Currently, AS path list changes automatically take effect, and issuing the **clear bgp neighbor ip-addr soft [in | out]** command in exec mode to update an AS path list can cause updates to be unnecessarily sent; therefore, it is not recommended.

To aggregate multiple policy changes, such as the AS path list, the SmartEdge OS performs the automatic update 15 seconds after any routing policy has changed.

Note If the remote peer does not support the BGP route refresh capability, an inbound policy change for the peer will result in an automatic hard reset of the session.

Use the **no** form of this command to disable the filter.

Examples

The following example permits only unicast routes that originate in AS 101 coming from the BGP neighbor at IP address 102.210.210.1. In addition, the SmartEdge router sends all multicast BGP routes, except for those routes that belong to AS 202, to the neighbor at IP address 68.68.68.68.

```
[local]Redback(config-ctx)#as-path-list filter-101
[local]Redback(config-as-path-list)#permit _101$
[local]Redback(config-as-path-list)#exit
[local]Redback(config-ctx)#as-path-list filter-202
[local]Redback(config-as-path-list)#deny _202_
[local]Redback(config-as-path-list)#permit .*
.
.
.
[local]Redback(config-ctx)#router bgp 100
[local]Redback(config-bgp)#neighbor 102.210.210.1 external
[local]Redback(config-bgp-neighbor)#remote-as 200
[local]Redback(config-bgp-neighbor)#address-family ipv4 unicast
[local]Redback(config-bgp-af)#as-path-list filter-101 in
[local]Redback(config-bgp-af)#exit
[local]Redback(config-bgp-neighbor)#exit
[local]Redback(config-bgp)#neighbor 68.68.68.68 external
[local]Redback(config-bgp-neighbor)#remote-as 300
[local]Redback(config-bgp-neighbor)#address-family ipv4 multicast
[local]Redback(config-bgp-af)#as-path-list filter-202 out
```

Related Commands

address-family ipv4
as-path-list—context configuration mode
neighbor
route-map

bestpath med always-compare

bestpath med always-compare

no bestpath med always-compare

Purpose

Allows the comparison of the Multi-Exit Discriminator (MED) for paths from Border Gateway Protocol (BGP) neighbors in different autonomous systems.

Command Mode

BGP router configuration

Syntax Description

This command has no keywords or arguments.

Default

The command is disabled.

Usage Guidelines

Use the **bestpath med always-compare** command to allow the comparison of the MED for paths from BGP neighbors in different autonomous systems.

The MED is one of the parameters that is considered when selecting the best path among many alternative paths. The path with a lower MED is preferred over a path with a higher MED. By default, MED comparison is done only among paths from the same autonomous system. This command changes the default behavior by allowing comparison of MEDs among paths regardless of the autonomous system from which the paths are received.

Use the **no** form of this command to disable the comparison of the MED for paths from BGP neighbors in different autonomous systems.

Examples

The following example enables the BGP speakers in autonomous system 64001 to compare the MED for paths from BGP neighbors in different autonomous systems:

```
[local]Redback(config)#router bgp 64001  
[local]Redback(config-bgp)#bestpath med always-compare
```

Related Commands

multi-paths

client-to-client reflection

client-to-client reflection

no client-to-client reflection

Purpose

Enables route reflection between clients of a Border Gateway Protocol (BGP) route reflector.

Command Mode

BGP router configuration

Syntax Description

This command has no keywords or arguments.

Default

Routes are reflected from one client to other clients.

Usage Guidelines

Use the **client-to-client reflection** command to enable route reflection between clients of a BGP route reflector.

By default, routes are reflected between clients of a route reflector. Under certain circumstances, a network administrator may not want routes that have been learned from one client to be reflected to other clients. One example is the case where clients are fully meshed. In this case, use the **no client-to-client reflection** command to disable route reflection.

Use the **no** form of this command to disable client-to-client reflection.

Examples

The following example configures the router as a unicast route reflector for neighbors, 102.210.210.1 and 122.101.12.145, and disables client-to-client reflection:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#router bgp 100
[local]Redback(config-bgp)#no client-to-client reflection
[local]Redback(config-bgp)#neighbor 102.210.210.1 external
[local]Redback(config-bgp-neighbor)#address-family ipv4 unicast
[local]Redback(config-bgp-af)#route-reflector-client
[local]Redback(config-bgp-af)#exit
[local]Redback(config-bgp-neighbor)#exit
[local]Redback(config-bgp)#neighbor 122.101.12.145 external
[local]Redback(config-bgp-neighbor)#address-family ipv4 unicast
[local]Redback(config-bgp-af)#route-reflector-client
```

Related Commands

cluster-id

route-reflector-client

cluster-id

cluster-id *ip-addr*

no cluster-id *ip-addr*

Purpose

Assigns a cluster ID if the Border Gateway Protocol (BGP) cluster has more than one route reflector.

Command Mode

BGP router configuration

Syntax Description

ip-addr IP address of the route reflector.

Default

The router ID is used as the cluster ID.

Usage Guidelines

Use the **cluster-id** command to assign a cluster ID if the BGP cluster has more than one route reflector. If this command is not enabled, the router ID is used as the cluster ID.

Together, a route reflector and its clients form a cluster. If there is more than one route reflector in a cluster, all route reflectors in that cluster should be configured with the same ID. A common cluster ID allows a route reflector to recognize updates from other route reflectors in the same cluster, prevents the possibility of a routing loop, and prevents the sending of duplicate updates.

Note Do not configure a cluster ID if the device is not a route reflector.

Use the **no** form of this command to remove a cluster ID.

Examples

The following example configures a cluster ID of 100.25.34.5:

```
[ local ]Redback(config)#context local  
[ local ]Redback(config-ctx)#router bgp 100  
[ local ]Redback(config-bgp)#cluster-id 100.25.34.5
```

Related Commands

client-to-client reflection
route-reflector-client

confederation identifier

confederation identifier {*asn* | *as:nn*}

no confederation identifier {*asn* | *as:nn*}

Purpose

Configures a Border Gateway Protocol (BGP) confederation identifier.

Command Mode

BGP router configuration

Syntax Description

<i>asn</i>	Autonomous system number (ASN). The range of values is 1 to 65,535. The subrange of 64,512 to 65,535 is reserved for private autonomous systems.
<i>as:nn</i>	ASN and a 2-byte number.

Default

No confederation identifier is configured.

Usage Guidelines

Use the **confederation identifier** command to configure a BGP confederation identifier. Use this command in conjunction with the **confederation peers** command in BGP router configuration mode to reduce internal BGP (iBGP) mesh by dividing an autonomous system into subautonomous systems and grouping them into a single confederation.

In the confederation, the subautonomous systems have external BGP (eBGP) connections to each other, but they exchange information as though they were iBGP peers. This means that they preserve next-hop, Multi-Exit Discriminator (MED), and local preference information. Externally, the confederation appears as a single autonomous system, and the confederation identifier is viewed as the ASN.

Use the **no** form of this command to remove a confederation identifier.

Examples

In the following example, the confederation consists of subautonomous systems, 65501, 65502, 65503, and 65504. Externally, there appears to be a single autonomous system with ASN 100.

```
[local]Redback(config-ctx)#router bgp 65501  
[local]Redback(config-bgp)#confederation identifier 100  
[local]Redback(config-bgp)#confederation peers 65502 65503 65504
```

Related Commands

confederation peers

confederation peers

confederation peers {asn... | as:nn...}

no confederation peers {asn... | as:nn...}

Purpose

Configures the subautonomous systems that belong to a Border Gateway Protocol (BGP) confederation.

Command Mode

BGP router configuration

Syntax Description

<i>asn...</i>	One or more autonomous system numbers (ASNs). The range of values is 1 to 65,535. The subrange of 64,512 to 65,535 is reserved for private autonomous systems.
<i>as:nn...</i>	One or more autonomous system numbers (ASNs) and a 2-byte number.

Default

No subautonomous systems are configured.

Usage Guidelines

Use the **confederation peers** command to configure the subautonomous systems that belong to a BGP confederation. Use this command in conjunction with the **confederation identifier** command in BGP router configuration mode to reduce internal BGP (iBGP) mesh. Subautonomous systems are visible within the confederation, but externally.

In the confederation, the subautonomous systems have external BGP (eBGP) connections to each other, but they exchange information as though they were IBGP peers. This means that they preserve next-hop, Multi-Exit Discriminator (MED), and local preference information. Externally, the confederation appears as a single autonomous system, and the confederation identifier is viewed as the ASN.

Use the **no** form of this command to remove an autonomous system from a BGP confederation.

Examples

The following example specifies that autonomous systems, 65501, 65502, 65503, and 65504 belong to a single confederation that is known externally as ASN 100:

```
[local]Redback(config-ctx)#router bgp 65501
[local]Redback(config-bgp)#confederation identifier 100
[local]Redback(config-bgp)#confederation peers 65502 65503 65504
```

Related Commands

confederation identifier

dampening

dampening [*half-life reuse suppress max-suppress* | **route-map** *map-name*] [**persistent**]

no dampening [*half-life reuse suppress max-suppress* | **route-map** *map-name*] [**persistent**]

Purpose

Enables external Border Gateway Protocol (eBGP) route dampening for the specified address family.

Command Mode

BGP address family configuration

Syntax Description

<i>half-life</i>	Optional. Amount of time, in minutes, after which a penalty is decreased. Once a route has been assigned a penalty, the penalty is decreased by half once the half-life period expires. The range of values is 1 to 45; the default value is 15.
<i>reuse</i>	Optional. Value that determines whether a route is unsuppressed and can be reused. When a penalty for a flapping route decreases to the point that it falls below this value, the route is unsuppressed and can be reused. Routes are scanned for reuse every 10 seconds. The range of values is 1 to 20,000; the default value is 750.
<i>suppress</i>	Optional. Value that determines if a route is suppressed. A route is suppressed when its penalty exceeds this limit. The range of values is 1 to 20,000; the default value is 2,000.
<i>max-suppress</i>	Optional. Maximum penalty, in minutes, that can be applied to a route. The range of values is 1 to 20,000; the default value is 4 times the value of the <i>half-life</i> argument. When the half life argument is left at its default value of 15 minutes, the <i>max-suppress</i> value defaults to 60.
route-map <i>map-name</i>	Optional. Route map name. Any set or match conditions, or both, in the specified route map are applied to BGP route dampening.
persistent	Optional. Specifies persistent route dampening, which keeps the dampening statistics for a route across peer resets.

Default

Route dampening is disabled. When enabled, the value for the *half-life* argument is 15 minutes. The value for the *reuse* argument is 750 minutes. The value for the *suppress* argument is 2,000 minutes. The value for the *max-suppress* argument is 4 times the value of the *half-life* argument.

Usage Guidelines

Use the **dampening** command to enable eBGP route dampening for the specified address family.

When a route from a remote peer is withdrawn, the local BGP speaker considers the withdrawn route to be a flap, and assigns a penalty of 1,000 to the route. If the remote peer sends a replacement route, the local BGP speaker assigns a penalty of 500 to the route.

Use the **no** form of this command to disable route dampening for the specified address family.

Examples

The following example enables route dampening:

```
[local]Redback(config)#router bgp 64000  
[local]Redback(config-bgp)#address-family ipv4 multicast  
[local]Redback(config-bgp-af)#dampening
```

Related Commands

flap-statistics
session-dampening

default-originate

default-originate [**route-map** *map-name*]

no default-originate [**route-map** *map-name*]

Purpose

Advertises the default route of the specified address family, even when the default route is not installed in the Border Gateway Protocol (BGP) routing table, to the BGP neighbor.

Command Mode

BGP neighbor address family configuration

BGP peer group address family configuration

Syntax Description

route-map *map-name* Optional. Name of the route map. The match and set conditions of the specified route map are applied before the default route is sent.

Default

No default route is sent to peers.

Usage Guidelines

Use the **default-originate** command to advertise the default route of the specified address family, even when the default route is not installed in the BGP routing table, to the BGP neighbor. The default route, 0.0.0.0/0, is typically sent to a BGP neighbor that does not carry full Internet routes.

If the **route-map** *map-name* keyword construct is not used, or if the specified route map does not include a **match ip address prefix-list** *pl-name* statement, the specified address family unconditionally advertises the default route to the BGP neighbor.

When the **route-map** *map-name* keyword construct is used, and the route map has a **match ip address prefix-list** *pl-name* statement, the specified address family advertises the default route only if the address prefix entry specified in the IP prefix list exists in the routing information base (RIB).

Use the **no** form of this command to avoid sending the default route to neighbors or peer groups.

Examples

The following example sends the unicast default route unconditionally to the neighbor at IP address 102.210.210.1, and only sends it to the neighbor at IP address, 68.68.68.68, when route, 20.0.0.0/8, with the next-hop address, 192.192.192.253:

```
[local]Redback(config-ctx)#route-map map1
[local]Redback(config-route-map)#match ip address prefix-list pref1
[local]Redback(config-route-map)#match ip next-hop prefix-list next-hop-list
[local]Redback(config-route-map)#exit
[local]Redback(config-ctx)#ip prefix-list pref1
[local]Redback(config-prefix-list)#permit 20.0.0.0/8
```

```
[local]Redback(config-prefix-list)#exit
[local]Redback(config-ctx)#ip prefix-list next-hop-list
[local]Redback(config-prefix-list)#permit 192.192.192.253/32
[local]Redback(config-prefix-list)#exit
[local]Redback(config-ctx)#router bgp 100
[local]Redback(config-bgp)#neighbor 102.210.210.1 external
[local]Redback(config-bgp-neighbor)#remote-as 200
[local]Redback(config-bgp-neighbor)#address-family ipv4 unicast
[local]Redback(config-bgp-af)#default-originate
[local]Redback(config-bgp-af)#exit
[local]Redback(config-bgp-neighbor)#exit
[local]Redback(config-bgp)#neighbor 68.68.68.68 external
[local]Redback(config-bgp-neighbor)#remote-as 300
[local]Redback(config-bgp-neighbor)#address-family ipv4 unicast
[local]Redback(config-bgp-af)#default-originate route-map map1
```

Related Commands

route-map

description

description *text*

no description

Purpose

Associates a description with the Border Gateway Protocol (BGP) neighbor or peer group.

Command Mode

BGP neighbor configuration

BGP peer group configuration

Syntax Description

text Description of the neighbor (maximum of 80 characters).

Default

None

Usage Guidelines

Use the **description** command to associate a description with the BGP neighbor or peer group. This command does not affect the BGP connection. It is used as a note in the configuration.

Use the **no** form of this command to remove a description from the configuration. Because there can be only one description for a BGP neighbor or peer group, when you use the **no** form of this command, it is not necessary to include the *text* argument.

Examples

The following example provides the description, Palo Alto BGP Neighbor 12, for the BGP neighbor at IP address, 102.210.210.1:

```
[local]Redback(config-ctx)#router bgp 100  
[local]Redback(config-bgp)#neighbor 102.210.210.1 external  
[local]Redback(config-bgp-neighbor)#description Palo Alto BGP Neighbor 12
```

Related Commands

neighbor

distance

distance *external-distance internal-distance local-distance*

{**no** | **default**} **distance** *external-distance internal-distance local-distance*

Purpose

Configures the administrative distance values for a Border Gateway Protocol (BGP) address family.

Command Mode

BGP address family configuration

Syntax Description

<i>external-distance</i>	Administrative distance for routes external to the autonomous system (AS). The range of values is 1 to 255; the default value is 20.
<i>internal-distance</i>	Administrative distance for routes internal to the AS. The range of values is 1 to 255; the default value is 200.
<i>local-distance</i>	Administrative distance for local routes. The range of values is 1 to 255; the default value is 200.

Default

The external administrative distance is set to 20. The internal and local administrative distances are set to 200.

Usage Guidelines

Use the **distance** command to configure the administrative distance values for a BGP address family. BGP uses distances to compare and prioritize routes. The lower the distance, the more preferred the route.

Use the **no** or **default** form of this command to return the values to their default settings.

Examples

The following example configures the administrative distance for external routes to 120, internal routes to 225 and local routes to 5:

```
[local]Redback(config-bgp-af)#distance 120 225 5
```

Related Commands

None

ebgp-multihop

ebgp-multihop *max-hops*

no ebgp-multihop *max-hops*

Purpose

Configures the maximum number of hops used to reach the external Border Gateway Protocol (eBGP) neighbor when the neighbor or peer group is not directly connected.

Command Mode

BGP neighbor configuration

BGP peer group configuration

Syntax Description

max-hops Maximum number of hops. The range of values is 1 to 255; the default value is 1.

Default

The maximum number of hops is set to 1.

Usage Guidelines

Use the **ebgp-multihop** command to configure the maximum number of hops used to reach the eBGP neighbor when the neighbor or peer group is not directly connected.

Note You must enable this command for BGP connections to be established with neighbors that are not directly connected.

Note You cannot enable this command on a BGP neighbor that is part of a peer group, because this feature cannot be customized for individual members inside of a peer group.

Use the **no** form of this command to restore the maximum number of hops to the default value of 1.

Examples

The following example sets the maximum number of hops to the neighbor at IP address, 12.10.10.1 to 3:

```
[local]Redback(config-ctx)#router bgp 100  
[local]Redback(config-bgp)#neighbor 12.10.10.1 external  
[local]Redback(config-bgp-neighbor)#ebgp-multihop 3
```

Related Commands

enforce ttl
neighbor

enforce ttl

enforce ttl

no enforce ttl

Purpose

Enables Border Gateway Protocol (BGP) time-to-live (TTL) security check in the kernel for the specified BGP neighbor or BGP peer group.

Command Mode

BGP neighbor configuration
BGP peer group configuration

Syntax Description

This command has no keywords or arguments.

Default

BGP TTL security check is not enabled in kernel.

Usage Guidelines

Use the **enforce ttl** command to enable BGP TTL security check in the kernel for the specified BGP neighbor or BGP peer group.

The BGP TTL security check feature can be used instead of, or in conjunction with, the BGP Session Protection via TCP Message Digest 5 (MD5) signature option for external BGP (eBGP); however, the TTL-based security check mechanism is more simple to operate because it does not require the coordination for managing the MD5 keys.



Caution Risk of data loss. Enabling the BGP TTL security check on only one end of an eBGP session causes the session to drop. To reduce the risk, verify that the BGP TTL security check feature is enabled on both ends of the eBGP session.

The BGP TTL security check is designed to protect the BGP infrastructure from CPU-utilization based attacks caused by sending control traffic that appears to be valid control traffic to a BGP session. It protects the BGP infrastructure by setting the value of the TTL field to 255 in outgoing BGP packets, and dropping incoming BGP packets that have TTL values less than the maximum TTL value (255) minus the maximum number of eBGP hops allowed.

For example, if you use the **ebgp-multihop** command to set the maximum number of hops used to reach an eBGP neighbor to two, then you should receive eBGP packets with TTL values of no less than 253 (255 - 2). When the BGP TTL security check is enabled using the **enforce ttl** command, all incoming BGP packets that have a TTL value less than 253 are dropped.

If the **ebgp-multihop** command is not used to set the maximum number of hops, then the default maximum hop value of 1 is used, and the BGP TTL security check drops all incoming BGP packets with TTL values less than 254.

Examples

The following example enables the BGP TTL security check to drop all BGP packets with a TTL value lower than 254 received from BGP neighbor, 10.10.10.20:

```
[local]Redback(config-bgp)#neighbor 10.10.10.20 external  
[local]Redback(config-bgp-neighbor)#enforce ttl
```

Related Commands

ebgp-multihop
neighbor
password
peer-group

fast-reset

fast-reset {*interval* | **confed** *interval*}

no fast-reset

Purpose

Configures the Border Gateway Protocol (BGP) routing process to wait a specified period of time before dropping sessions of directly connected external peers if the link used to reach them goes down.

Command Mode

BGP router configuration

Syntax Description

<i>interval</i>	Interval, in seconds, the BGP routing process waits once an interface has been reset before dropping a connection. The range of values is 1 to 60.
confed	Applies a fast reset only to peers in the associated BGP confederation.

Default

BGP sessions remain connected after the outbound interface goes down. BGP sessions are dropped after the BGP holdtime value, set through the **timers** command in BGP router configuration mode, is exceeded.

Usage Guidelines

Use the **fast-reset** command to configure the BGP routing process to wait a specified period of time before dropping sessions of directly connected external peers if the link used to reach them goes down.

Use the **confed** keyword to apply a fast reset only to peers in the associated BGP confederation.

For faster route convergence, it may be desirable to drop a BGP session faster than the time specified by the holdtime value using the **timers** command.

Use the **no** form of this command to disable the automatic wait period.

Examples

The following example configures the BGP routing process to wait 50 seconds after an interface has been reset before it drops the connection:

```
[local]Redback(config-bgp)#fast-reset 50
```

Related Commands

timers

flap-statistics

flap-statistics

no flap-statistics

Purpose

Enables route-flap statistics accounting for the address family for both internal Border Gateway Protocol (iBGP) and external BGP (eBGP) routing processes.

Command Mode

BGP address family configuration

Syntax Description

This command has no keywords or arguments.

Default

Route-flap statistics accounting is disabled.

Usage Guidelines

Use the **flap-statistics** command to enable route-flap statistics accounting for both iBGP and eBGP routing processes.

This command is useful for determining routing stability and for diagnosing problems. In particular, this command is useful for troubleshooting persistent iBGP routing loops. Use this command if the network is experiencing a high degree of route flapping.

Use the **no** form of this command to disable route-flap statistics accounting.

Examples

The following example enables route-flap statistics accounting:

```
[local]Redback(config-ctx)#router bgp 64001  
[local]Redback(config-bgp)#address-family ipv4 multicast  
[local]Redback(config-bgp-af)#flap-statistics
```

Related Commands

dampening

local-as

local-as {*asn* | *nn:nn*}

no local-as {*asn* | *nn:nn*}

Purpose

Configures the autonomous system number (ASN) that the Border Gateway Protocol (BGP) routing process uses to peer with the specified external BGP (eBGP) neighbor.

Command Mode

BGP neighbor configuration

Syntax Description

<i>asn</i>	ASN in integer format. The range of values is 1 to 65,535. The subrange 64,512 to 65,535 is reserved for private autonomous systems.
<i>nn:nn</i>	ASN in 4-byte integer format, where the first <i>nn</i> indicates the two higher-order bytes and the second <i>nn</i> denotes the two lower-order bytes.

Default

None

Usage Guidelines

Use the **local-as** command to specify the ASN that the BGP routing process uses to peer with the specified eBGP neighbor. Under most circumstances, the BGP routing process peers with neighbors that use the same ASN, which is configured through the **router bgp** command in context configuration mode. The **local-as** command allows the configuration of a different ASN to be used with the specified eBGP neighbor.

Use the **no** form of this command to remove the local ASN.

Examples

The following example configures an ASN of 100 for the SmartEdge router. The SmartEdge router peers with the neighbors at IP address, 102.210.210.1, and IP address, 103.220.220.3, using ASN 100. However, it peers with the neighbor at IP address, 68.68.68.68, using ASN 200.

```
[local]Redback(config-ctx)#router bgp 100
[local]Redback(config-bgp)#neighbor 102.210.210.1 external
[local]Redback(config-bgp-neighbor)#remote-as 500
[local]Redback(config-bgp-neighbor)#address-family ipv4 unicast
[local]Redback(config-bgp-af)#exit
[local]Redback(config-bgp-neighbor)#exit
[local]Redback(config-bgp)#neighbor 103.220.220.3 external
[local]Redback(config-bgp-neighbor)#remote-as 500
[local]Redback(config-bgp-neighbor)#address-family ipv4 unicast
[local]Redback(config-bgp-af)#exit
```

```
[local]Redback(config-bgp-neighbor)#exit  
[local]Redback(config-bgp)#neighbor 68.68.68.68 external  
[local]Redback(config-bgp-neighbor)#remote as-400  
[local]Redback(config-bgp-neighbor)#local-as 200  
[local]Redback(config-bgp-neighbor)#address-family ipv4 unicast
```

Related Commands

neighbor
remote-as
router-id

local-preference

local-preference *pref-num*

no local-preference *pref-num*

Purpose

Configures the value of the local preference number, a value that is applied to Border Gateway Protocol (BGP) routes that do not have the local-preference attribute.

Command Mode

BGP router configuration

Syntax Description

<i>pref-num</i>	Local preference number. The range of values is 0 to 4,294,967,295; the default value is 100.

Default

The default preference is 100.

Usage Guidelines

Use the **local-preference** command to configure the value of the local preference number.

Use the **no** form of this command to restore the default local preference value of 100.

Examples

The following example sets the preference to 300:

```
[local]Redback(config-ctx)#router bgp 100  
[local]Redback(config-bgp)#local-preference 300
```

Related Commands

route-map—context configuration mode
set local-preference

log-neighbor-changes

log-neighbor-changes

no log-neighbor-changes

Purpose

Configures the Border Gateway Protocol (BGP) routing process to log BGP neighbor resets.

Command Mode

BGP router configuration

Syntax Description

This command has no keywords or arguments.

Default

BGP neighbor resets are logged.

Usage Guidelines

Use the **log-neighbor-changes** command to configure the BGP routing process to log BGP neighbor resets. Frequent resets could indicate excessive packet loss or other network problems.

Use the **no** form of this command to ensure that resets are not logged.

Examples

The following example configures the BGP routing process so that BGP neighbor resets are not logged:

```
[local]Redback(config-ctx)#router bgp 100  
[local]Redback(config-bgp)#no log-neighbor-changes
```

Related Commands

neighbor

maximum prefix

maximum prefix *max-prefix* [**threshold** *threshold*] [**downtime** *interval* | **warning-only**]

no maximum prefix *max-prefix* [**threshold** *threshold*] [**downtime** *interval* | **warning-only**]

Purpose

Specifies how the Border Gateway Protocol (BGP) routing process responds when the maximum number of prefixes sent by the BGP neighbor or BGP peer group for the specified address family is exceeded.

Command Mode

BGP neighbor address family configuration
BGP peer group address family configuration

Syntax Description

<i>max-prefix</i>	Maximum number of prefixes that can be sent by the neighbor. The range of values is 1 to 4,294,967,295; the default is an unlimited number of prefixes.
threshold <i>threshold</i>	Optional. Warning that is generated when the specified threshold value, expressed as a percentage, is reached. The range of values is 1 to 100; the default value is 75.
downtime <i>interval</i>	Optional. Interval, in seconds, for which the connection to the neighbor is down once the specified maximum number of prefixes is exceeded. If this keyword construct is not enabled, the connection remains down until the clear bgp ip-address command in exec mode is issued.
warning-only	Optional. Issues a warning to the neighbor once the specified maximum number of prefixes is exceeded. The connection remains intact.

Default

The BGP routing process accepts an unlimited number of prefixes. If you enter this command without any keywords, the BGP session will be torn down once the *max-prefix* argument value is exceeded. The session remains down until the **clear bgp ip-address** command is issued. The threshold is 75.

Usage Guidelines

Use the **maximum prefix** command to specify how the BGP routing process responds when the maximum number of prefixes sent by the BGP neighbor or BGP peer group for the specified address family is exceeded.

Use the **no** form of this command to return the BGP routing process to the default behavior of allowing an unlimited number of routes and to reset the system to the default behavior of dropping the BGP session when the maximum number of prefixes is exceeded.

Examples

The following example allows a maximum number of 10000 unicast routes from the neighbor at IP address 102.210.210.1 and generates a warning after 90% of the routes (9000) are received:

```
[local]Redback(config-ctx)#router bgp 100  
[local]Redback(config-bgp)#neighbor 102.210.210.1 external  
[local]Redback(config-bgp-neighbor)#address-family ipv4 unicast  
[local]Redback(config-bgp-af)#maximum prefix 10000 threshold 90
```

Once 10,000 unicast routes are received, the BGP routing process drops the BGP session. The session remains down until the **clear bgp 102.210.210.1** command in exec mode is issued.

Related Commands

None

maximum restart-time

maximum restart-time *interval*

no maximum restart-time *interval*

Purpose

Sets the maximum amount of time that it will take for a local BGP peer to come up after it has been reset.

Command Mode

BGP neighbor configuration

BGP router configuration

Syntax Description

<i>interval</i>	Maximum time, in seconds, that a remote peer will hold the routes received from a local bgp peer after the local peer has been reset during BGP graceful restart. The range of values is 10 to 180; the default value is 60.

Default

The command is disabled. When enabled, the local BGP speaker attempts to reconnect with the remote peer after 60 seconds, or one minute.

Usage Guidelines

Use the **maximum restart-time** command to set the maximum amount of time that it will take for a local BGP peer to come up after it has been reset.

This graceful restart capability allows a BGP speaker to indicate its ability to preserve its forwarding state during BGP restart.

Use the **no** form of this command to disable a maximum restart time.

Examples

The following example configures the BGP routing process for autonomous system, 64001, to attempt to reconnect with the remote peer within 40 seconds after a reset has occurred:

```
[local]Redback(config-ctx)#router bgp 64001
[local]Redback(config-bgp)#maximum restart-time 40
```

The following example configures the external BGP (eBGP) neighbor, 10.1.1.1, to attempt to reconnect with the remote peer within 45 seconds after a reset has occurred:

```
[local]Redback(config-bgp)#neighbor 10.1.1.1 external
[local]Redback(config-bgp-neighbor)#maximum restart-time 45
```

Related Commands

None

maximum retain-time

maximum retain-time *interval*

no maximum retain-time *interval*

Purpose

Configures the maximum amount of time the local Border Gateway Protocol (BGP) speaker retains routes it previously received from a remote peer once that remote peer restarts the connection.

Command Mode

BGP neighbor configuration

BGP router configuration

Syntax Description

interval

Maximum amount of time, in seconds, that the local BGP speaker retains routes it has previously received from the remote peer. The range of values is 30 to 300; the default value is 180 seconds.

Default

The command is disabled. When enabled, the local BGP speaker retains routes it has previously received from the remote peer for 180 seconds, or three minutes.

Usage Guidelines

Use the **maximum retain-time** command to set the maximum amount of time the local BGP speaker retains routes it previously received from a remote peer once that remote peer restarts the connection.

Any routes that have not been updated by the remote peer are deleted by the local peer after the local peer receives the end-of-routing information base (RIB) marker from the remote peer, or after the timer expires. An end-of-RIB marker from the remote peer indicates that its initial update has been completed.

Use the **no** form of this command to disable the maximum retain time.

Examples

The following example configures the BGP routing process for autonomous system, 64001, to retain routes that have been received from a remote peer once the remote peer restarts the connection for 120 seconds, or two minutes:

```
[local]Redback(config-ctx)#router bgp 64001  
[local]Redback(config-bgp)#maximum retain-time 120
```

The following example configures the external BGP (eBGP) neighbor, 10.1.1.1, to attempt to retain routes from a remote peer once the remote peer restarts the connection for 90 seconds:

```
[local]Redback(config-bgp)#neighbor 10.1.1.1 external  
[local]Redback(config-bgp-neighbor)#maximum retain-time 90
```

Related Commands

`retain-ibgp-routes`

maximum update-delay

maximum update-delay *interval*

no maximum update-delay *interval*

Purpose

Sets the maximum delay time for the Border Gateway Protocol (BGP) routing process after a reset has occurred before performing initial best-path calculations.

Command Mode

BGP router configuration

Syntax Description

interval

Maximum amount of time, in seconds, that the BGP routing process waits after reset before performing initial best-path calculations. The range of values is 1 to 300.

Default

The command is disabled.

Usage Guidelines

Use the **maximum update-delay** command to set the maximum delay time for the BGP routing process after a reset has occurred before performing initial best-path calculations.

This feature is useful in the case where not all peers support a graceful restart, and in the case where a peer may not send an end-of-Routing Information Base (RIB) marker. Best-path calculations are performed after all peers have send an end-of-RIB marker, or when the timer expires.

Use the **no** form of this command to disable the maximum delay time.

Examples

The following example configures the BGP routing process for autonomous system, 64001, to wait 60 seconds, or one minute, after a reset has occurred before performing initial best-path calculations:

```
[local]Redback(config-ctx)#router bgp 64001  
[local]Redback(config-bgp)#maximum update-delay 60
```

Related Commands

maximum restart-time

multi-paths

```
multi-paths {external path-num [internal path-num] | internal path-num [external path-num] }  
{no | default} multi-paths {external path-num [internal path-num] | internal path-num  
[external path-num] }
```

Purpose

Configures the Border Gateway Protocol (BGP) routing process to use multiple equal-cost best paths for load-balancing outgoing BGP traffic packets.

Command Mode

BGP router configuration

Syntax Description

external <i>path-num</i>	External BGP (eBGP) equal-cost paths. Optional when internal BGP (iBGP) equal-cost paths are specified. The <i>path-num</i> argument specifies the maximum number of equal-cost best paths. The range of values is 1 to 8; the default value is 1.
internal <i>path-num</i>	eBGP equal-cost paths. Optional when eBGP equal-cost paths are specified. The <i>path-num</i> argument specifies the maximum number of equal-cost best paths. The range of values is 1 to 8; the default value is 1.

Default

The command is disabled.

Usage Guidelines

Use the **multi-paths** command to configure the BGP routing process to use multiple equal-cost BGP best paths for load-balancing outgoing traffic packets.

Use the **external** keyword to balance loads among equal-cost paths from different eBGP neighbors that reside in a single autonomous system (AS). For eBGP, equal-cost means that each path shares the same weight, local preference, AS path length, origin type, and Multi-Exit Discriminator (MED) attributes. If one of these attributes is different, the path is not considered to be an equal-cost path. In addition, the eBGP paths uses originate from the same AS.

Use the **internal** keyword to balance loads among equal-cost paths from different iBGP neighbors. For iBGP, equal-cost means that each path shares the same weight, local preference, AS path length, origin type, and MED attributes. In addition, each path must share the same Interior Gateway Protocol (IGP) metric to the next hop.

Use the **no** or **default** form of this command to restore the default setting.

Examples

The following example load-balances outgoing traffic packets between 2 eBGP paths and 5 iBGP paths:

```
[local]Redback(config)#router bgp 64001  
[local]Redback(config-bgp)#multi-paths external 2 internal 5
```

Related Commands

multi-paths eibgp
neighbor

neighbor

neighbor {*ip-addr* | *ipv6-addr*} {**external** | **internal**}

no neighbor *ip-addr* {**external** | **internal**}

Purpose

Configures a Border Gateway Protocol (BGP) neighbor and enters BGP neighbor configuration mode.

Command Mode

BGP router configuration

Syntax Description

<i>ip-addr</i>	BGP neighbor IP address in the form <i>A.B.C.D</i> .
<i>ipv6-addr</i>	BGP neighbor IP Version 6 (IPv6) address in the form <i>A:B:C:D:E:F:G</i> .
external	Identifies the peer as an external BGP (eBGP) neighbor.
internal	Identifies the peer as an internal BGP (iBGP) neighbor.

Default

There are no preconfigured neighbors.

Usage Guidelines

Use the **neighbor** command to configure a BGP neighbor and enter BGP neighbor configuration mode. If you enter the **external** keyword, you must also enable the **remote-as** command in BGP neighbor configuration mode. If you enter the **internal** keyword, the **remote-as** command is not needed.

When the **neighbor** command is issued, the address family for that neighbor defaults to unicast. For an IP Version 4 (IPv4) address family, you can modify this setting through the **address-family ipv4** command in BGP neighbor configuration mode.

Use the **no** form of this command to remove a configured BGP neighbor.

Examples

The following example configures an eBGP neighbor at IP address, 102.210.210.1, and enters BGP neighbor configuration mode:

```
[local]Redback(config-ctx)#router bgp 100
[local]Redback(config-bgp)#neighbor 102.210.210.1 external
[local]Redback(config-bgp-neighbor)#
```

The following example configures an iBGP neighbor at IPv6 address, 28FF:AA12:0DB8:85A3::2000, and enters BGP neighbor configuration mode:

```
[local]Redback(config-ctx)#router bgp 100
[local]Redback(config-bgp)#neighbor 28FF:AA12:0DB8:85A3::2000 internal
[local]Redback(config-bgp-neighbor)#
```

Related Commands

`address-family ipv4`
`remote-as`
`send community`
`send ext-community`

network

network {*ip-addr/prefix-length* | *ipv6-addr/prefix-length*} [**route-map** *map-name*]

no network {*ip-addr/prefix-length* | *ipv6-addr/prefix-length*} [**route-map** *map-name*]

Purpose

Originates Border Gateway Protocol (BGP) routes that are advertised to peers for the BGP address family.

Command Mode

BGP address family configuration

Syntax Description

<i>ip-addr/prefix-length</i>	Specifies the IP address, in the form <i>A.B.C.D</i> , and the prefix length, separated by the slash (/) character. The range of values for the <i>prefix-length</i> argument is 0 to 32.
<i>ipv6-addr/prefix-length</i>	Specifies the IP Version 6 (IPv6) address, in the form <i>A:B:C:D:E:F:G:H</i> , and the prefix length, separated by the slash (/) character. The range of values for the <i>prefix-length</i> argument is 0 to 128.
route-map <i>map-name</i>	Optional. Route map conditions to apply to the prefix.

Default

No routes are specified.

Usage Guidelines

Use the **network** command to originate BGP routes that are advertised to peers.

Use the **route-map** *map-name* construct to apply a route map to modify the BGP attributes of these routes. Routes specified in the **network** command must exist in the routing table to generate those routes into BGP.

Use the **no** form of this command to remove routes.

Examples

The following example advertises unicast route 120.34.56.0/24 to unicast BGP neighbors. Multicast route 40.0.0.0/8 is advertised to multicast BGP neighbors using a metric of 100. The two **ip route** commands in context configuration mode statically add these routes to the routing table.

```
[local]Redback(config-ctx)#ip route 40.0.0.0/8 null0
[local]Redback(config-ctx)#ip route 120.34.56.0/24 null0
[local]Redback(config-ctx)#route-map map1
[local]Redback(config-route-map)#set metric 100
[local]Redback(config-route-map)#exit
[local]Redback(config-ctx)#router bgp 100
[local]Redback(config-bgp)#address-family ipv4 unicast
[local]Redback(config-bgp-af)#network 120.34.56.0/24
```



```
[local]Redback(config-bgp-af)#exit  
[local]Redback(config-bgp)#address-family ipv4 multicast  
[local]Redback(config-bgp-af)#network 40.0.0.0/8 route-map map1
```

Related Commands

aggregate-address
redistribute
route-map

next-hop-self

next-hop-self

no next-hop-self

Purpose

Advertises the local peer address as the next-hop address for all external Border Gateway Protocol (eBGP) routes sent to the specified neighbor or peer group.

Command Mode

BGP neighbor configuration
BGP peer group configuration

Syntax Description

This command has no keywords or arguments.

Default

The command is disabled.

Usage Guidelines

Use the **next-hop-self** command to advertise the local peer address as the next-hop address for all eBGP routes sent to the specified BGP neighbor or peer group. This command disables the sending of third-party next-hop information to peers.

By default, when it receives BGP routes from an eBGP neighbor, the BGP routing process forwards eBGP routes to its internal BGP (iBGP) neighbors without changing the next-hop address; this is still the case if the eBGP neighbors are on the same subnet as the local BGP speaker.

When you enable the **next-hop-self** command, the BGP routing process changes the next-hop address, advertising the local peer address as the next-hop address for all received eBGP routes.

Use the **no** form of this command to restore the default behavior of sending third-party next-hop information to peers.

Examples

The following example ensures that all updates destined for the neighbor at IP address, 10.100.1.102, advertise this SmartEdge router as the next hop:

```
[local]Redback(config-ctx)#router bgp 64001  
[local]Redback(config-bgp)#neighbor 10.100.1.102 external  
[local]Redback(config-bgp-neighbor)#remote-as 64001  
[local]Redback(config-bgp-neighbor)#next-hop-self
```

The following example provides output from the **show bgp neighbor** command where the neighbor views the SmartEdge router as the next hop for all received routes:

```
[local]Redback>show bgp neighbor 10.100.1.102
```

```
BGP neighbor: 10.100.1.102, remote AS: 64001, internal link  
Version: 4, router identifier: 10.100.1.102  
State: Established for 00:41:01  
.  
.  
.  
Next hop set to self (next-hop-self)  
.  
.  
.  
Prefixes: advertised 99877, accepted 2, active 2
```

Related Commands

neighbor
update-source

password

password *password*

no password

Purpose

Configures an encrypted Message Digest 5 (MD5) password for the Border Gateway Protocol (BGP) neighbor or peer group.

Command Mode

BGP neighbor configuration
BGP peer group configuration

Syntax Description

password Alphanumeric string consisting of up to 80 characters.

Default

None

Usage Guidelines

Use the **password** command to assign an encrypted MD5 password for the BGP neighbor or peer group.

Note For a BGP session to be established, the MD5 password must be the same on both the router and its neighbor.

Use the **no** form of this command to remove an assigned password from the BGP neighbor or peer group.

Examples

The following example assigns the password `secret` to the external BGP (eBGP) neighbor at IP address `10.10.1.1`:

```
[local]Redback(config-bgp)#neighbor 10.10.1.1 external  
[local]Redback(config-bgp-neighbor)#password secret
```

Related Commands

enforce ttl
neighbor
timer password

peer-group

```
peer-group group-name {external | internal}  
no peer-group group-name {external | internal}
```

Purpose

Configures a Border Gateway Protocol (BGP) peer group and defines the peer group as external BGP (eBGP) or internal BGP (iBGP), or applies the attributes of a configured peer group to a BGP neighbor or BGP neighbor address family.

Command Mode

BGP neighbor address family configuration
BGP neighbor configuration
BGP router configuration

Syntax Description

<i>group-name</i>	Name of the peer group.
external	Configures an eBGP peer group.
internal	Configures an iBGP peer group.

Default

There are no preconfigured peer groups. Once a peer group is configured, it is enabled.

Usage Guidelines

Use the **peer-group** command to configure a BGP peer group and define the peer group as eBGP or iBGP, or to apply the attributes of a configured peer group to a BGP neighbor or BGP neighbor address family.

Peer groups are helpful in cases where many BGP neighbors are configured with the same outbound update policies. Grouping a large number of neighbors into one or more peer groups simplifies modifications to a configuration, and more importantly, makes BGP update generation more efficient. The use of peer groups is strongly recommended when there are a large number of peers.

Use the **peer-group** command in BGP router configuration mode to create a peer group name, and to enter BGP peer group configuration mode, where attributes can be configured for the specified peer group.

You can apply attributes to BGP neighbors or address families. Attributes that are not configurable for peer groups are those set by the following commands in BGP neighbor configuration mode: **accept prefix-filter**, **local-as**, and **remote-as**.

Use the **peer-group** command in BGP neighbor configuration mode to apply the characteristics of a peer group to one or more BGP neighbors. A neighbor can be assigned to a peer group only if the neighbor and the peer group is of the same type—external or internal BGP. If a neighbor belongs to a particular peer group, it cannot be configured to belong to another peer group. The previous peer group membership must first be explicitly deleted before the peer membership can be reconfigured.

Attributes are inherited from the peer group to which a neighbor is assigned. The following BGP neighbor configuration mode commands represent attributes that cannot be customized per neighbor when the neighbor is assigned to a peer group: **advertisement-interval**, **ebgp-multihop**, **local-as**, **send community**, and **timers**. Attributes inherited from a peer group that can be customized per neighbor include those set by the following commands: **description**, **password**, **send prefix**, **shutdown**, and **update-source**.

Use the **peer-group** command in BGP neighbor address family configuration mode to apply the characteristics of a peer group to one or more BGP neighbor address families. A BGP neighbor address family can belong to more than one peer group and can be modified to belong to a different peer group without having to delete the previous peer group association first.

Attributes are inherited from the peer group to which a BGP neighbor address family is assigned. The following commands in BGP neighbor address family configuration mode represent attributes that cannot be customized per address family once it is assigned to a peer group: **as-path-list out**, **prefix-list out**, **remove-private-as**, and **route-map out**. Attributes inherited from a peer group that can be customized per neighbor address family include those set by the following commands: **as-path-list in**, **default-originate**, **maximum-prefix**, **prefix-list in**, and **route-map in**.

By default, a configured peer group is automatically enabled. To disable a peer group, enter the **shutdown** command in BGP peer group configuration mode.

Use the **no** form of this command to remove a peer group.

Examples

The following example assigns the BGP neighbor at IP address 10.1.1.1 to the peer group pgrp-101. The BGP neighbor at IP address 10.1.1.1 inherits all of its configuration from peer group pgrp-101. The configuration also assigns the BGP neighbor at IP address 10.2.2.2 to the peer group pgrp-200. The BGP neighbor at IP address 10.2.2.2 inherits all outbound routing policies and the properties of the **remove-private-AS** command from peer group pgrp-200, but does not inherit the group's inbound policies or description information.

```
[local]Redback(config-ctx)#router bgp 101
[local]Redback(config-bgp)#peer-group pgrp-101 internal
[local]Redback(config-bgp-peer-group)#description config IBGP neighbors
[local]Redback(config-bgp-peer-group)#password encrypted 8F733D8CD3F98AE0
[local]Redback(config-bgp-peer-group)#update-source interface1
[local]Redback(config-bgp-peer-group)#next-hop-self
[local]Redback(config-bgp-peer-group)#address-family ipv4 unicast
[local]Redback(config-bgp-peer-af)#maximum prefix 20000
[local]Redback(config-bgp-peer-af)#exit
[local]Redback(config-bgp-peer-group)#exit
[local]Redback(config-bgp)#peer-group pgrp-200 external
[local]Redback(config-bgp-peer-group)#ebgp-multihop 10
[local]Redback(config-bgp-peer-group)#address-family ipv4 unicast
[local]Redback(config-bgp-peer-af)#as-path-list aspath-in in
[local]Redback(config-bgp-peer-af)#as-path-list aspath-out out
[local]Redback(config-bgp-peer-af)#remove-private-AS
[local]Redback(config-bgp-peer-af)#exit
[local]Redback(config-bgp-peer-group)#exit
[local]Redback(config-bgp)#neighbor 10.1.1.1 internal
[local]Redback(config-bgp-neighbor)#peer-group pgrp-101
[local]Redback(config-bgp-neighbor)#address-family ipv4 unicast
```

```
[local]Redback(config-bgp-neighbor)#exit
[local]Redback(config-bgp)#neighbor 10.2.2.2 external
[local]Redback(config-bgp-neighbor)#peer-group pgrp-200
[local]Redback(config-bgp-neighbor)#remote-as 200
[local]Redback(config-bgp-neighbor)#description neighbor at corpA
[local]Redback(config-bgp-neighbor)#address-family ipv4 unicast
[local]Redback(config-bgp-af)#as-path-list as-in in
[local]Redback(config-bgp-af)#as-path-list as-out out
[local]Redback(config-bgp-af)#route-map rmap-out out
```

Related Commands

neighbor

prefix-list

prefix-list *pl-name* {**in** | **out**}

no prefix-list *pl-name* {**in** | **out**}

Purpose

Filters Border Gateway Protocol (BGP) routes from or to the neighbor address family or peer group address family.

Command Mode

BGP neighbor address family configuration
BGP peer group address family configuration

Syntax Description

<i>pl-name</i>	Name of the prefix list.
in	Applies the prefix list to incoming updates from the neighbor.
out	Applies the prefix list to outgoing updates to the neighbor. This keyword can only be applied in BGP neighbor address family configuration mode.

Default

There are no preconfigured prefix lists.

Usage Guidelines

Use the **prefix-list** command to filter BGP routes from or to the neighbor address family or peer group address family. Use this command in conjunction with the **ip prefix-list** command in context configuration mode, which creates the conditions of the filter.

Use the **in** keyword to filter incoming BGP routes from the specified neighbor or peer. Use the **out** keyword to filter outgoing BGP routes to the specified neighbor.

Note You cannot enable the **out** keyword on a BGP neighbor that is part of a peer group, because this feature cannot be customized for individual members inside of a peer group.

Currently, prefix list changes automatically take effect, and issuing the **clear bgp neighbor ip-addr soft [in | out]** command in exec mode to update a prefix list can cause updates to be unnecessarily sent; therefore, it is not recommended.

To aggregate multiple policy changes, such as the prefix list, the SmartEdge OS performs the automatic update 15 seconds after any routing policy has changed.

Note If the remote peer does not support the BGP route refresh capability, an inbound policy change for the peer will result in an automatic hard reset of the session.

Use the **no** form of this command to remove the application of a prefix list.

Examples

The following example denies incoming unicast BGP routes 10.0.0.0/8 (and more-specific routes) from the unicast neighbor at IP address 102.210.210.1. Outgoing multicast BGP routes 204.16.16.0/24 can be sent to the multicast neighbor at IP address 68.68.68.68:

```
[local]Redback(config-ctx)#ip prefix-list prefix-101
[local]Redback(config-prefix-list)#deny 10.0.0.0/8 le 32
[local]Redback(config-prefix-list)#permit 0.0.0.0/0 le 32
[local]Redback(config-prefix-list)#exit
[local]Redback(config-ctx)#ip prefix-list prefix-202
[local]Redback(config-prefix-list)#permit 204.16.16.0/24
.
.
.
[local]Redback(config-ctx)#router bgp 100
[local]Redback(config-bgp)#neighbor 102.210.210.1 external
[local]Redback(config-bgp-neighbor)#remote-as 200
[local]Redback(config-bgp-neighbor)#address-family ipv4 unicast
[local]Redback(config-bgp-af)#prefix-list prefix-101 in
[local]Redback(config-bgp-af)#exit
[local]Redback(config-bgp-neighbor)#exit
[local]Redback(config-bgp)#neighbor 68.68.68.68 external
[local]Redback(config-bgp-neighbor)#remote-as 300
[local]Redback(config-bgp-neighbor)#address-family ipv4 multicast
[local]Redback(config-bgp-af)#prefix-list prefix-202 out
```

Related Commands

address-family ipv4

ip prefix-list—context configuration mode

redistribute

```
redistribute {connected | isis instance [level-1 | level-2] | nat | ospf instance [internal | [external]
[nssa-external] | rip instance | static [dvsr] | subscriber [address | static]}
[route-map map-name]
```

```
no redistribute {connected | isis instance [level-1 | level-2] | nat | ospf instance [internal | [external]
[nssa-external] | rip instance | static [dvsr] | subscriber [address | static]}
[route-map map-name]
```

Purpose

Redistributes routes learned through other routing protocols into the Border Gateway Protocol (BGP) routing domain.

Command Mode

BGP address family configuration

Syntax Description

connected	Redistributes routes from directly attached networks into the BGP routing domain.
isis <i>instance</i>	Intermediate System-to-Intermediate System (IS-IS) instance name. Redistributes routes from the specified IS-IS routing instance into the BGP routing domain.
level-1	Optional. Specifies IS-IS level 1 routing.
level-2	Optional. Specifies IS-IS level 2 routing.
nat	Redistributes network address translation (NAT) routes into the BGP routing domain.
ospf <i>instance</i>	Open Shortest Path First (OSPF) instance ID. Redistributes routes from the specified OSPF routing instance into the BGP routing domain. The range of values is 1 to 65,535.
internal	Optional. Redistributes OSPF internal routes into the BGP routing domain.
external	Optional. Redistributes OSPF external routes into the BGP routing domain.
nssa-external	Optional. Redistributes not-so-stubby-area (NSSA) routes into the BGP routing domain.
rip <i>instance</i>	Routing Information Protocol (RIP) instance name. Redistributes routes from the specified RIP routing instance into the BGP routing domain.
static	Redistributes static routes into the BGP routing domain. Optional with the subscriber keyword. Redistributes only static subscriber routes into the BGP routing domain.
dvsr	Optional. Redistributes the dynamically verified static routing (DVSR) subtype of static routes into the BGP routing domain.

subscriber	Redistributes routes configured within subscriber records into the BGP routing domain.
address	Optional. Redistributes only subscriber address routes into the BGP routing domain.
route-map <i>map-name</i>	Optional. Route map name. Applies a previously configured route map. If this option is not specified, all routes from the specified protocol are redistributed with their default attributes into the BGP routing domain.

Default

Routes learned by other protocols are not distributed into the BGP routing domain.

Usage Guidelines

Use the **redistribute** command to redistribute routes learned through other routing protocols into the BGP routing domain. Redistributed routes are advertised to all BGP neighbors for the address family.

Note The default route, 0.0.0.0, is not redistributed. Use the **network** command in BGP address family configuration mode to advertise the default route.

You must enter multiple **redistribute** commands to redistribute routes from several different kinds of routing protocols into the BGP routing domain.

Use the **no** form of this command to disable the specified type of route redistribution.

Examples

The following example redistributes external OSPF routes from OSPF instance 100 into the BGP routing domain as unicast routes. The static route 192.200.201.0/24 is redistributed into the BGP routing domain as unicast routes with the community attribute of 100:100.

```
[local]Redback(config-ctx)#route-map static-to-bgp
[local]Redback(config-route-map)#ip address prefix-list static-to-bgp-prefix
[local]Redback(config-route-map)#set community 100:100
[local]Redback(config-route-map)#exit
[local]Redback(config-ctx)#ip prefix-list static-to-bgp-prefix
[local]Redback(config-prefix-list)#permit 192.200.201.0/24
.
.
.
[local]Redback(config-ctx)#router bgp 100
[local]Redback(config-bgp)#address-family ipv4 unicast
[local]Redback(config-bgp-af)#redistribute ospf 100 external
[local]Redback(config-bgp-af)#redistribute static route-map static-to-bgp
```

Related Commands

address-family ipv4
aggregate-address

network
route-map—context configuration mode

remote-as

remote-as {*asn* | *nn:nn*}

no remote-as {*asn* | *nn:nn*}

Purpose

Configures the autonomous system number (ASN) of the external Border Gateway Protocol (eBGP) neighbor.

Command Mode

BGP neighbor configuration

Syntax Description

<i>asn</i>	ASN in integer format. The range of values is 1 to 65,535. The subrange of 64,512 to 65,535 is reserved for private ASNs.
<i>nn:nn</i>	ASN in 4-byte integer format, where the first <i>nn</i> indicates the two higher-order bytes and the second <i>nn</i> denotes the two lower-order bytes.

Default

None

Usage Guidelines

Use the **remote-as** command to configure the ASN of the eBGP neighbor.

Use the **no** form of this command to remove the ASN.

Examples

The following example assigns ASN 4001 to the eBGP neighbor at IP address 102.201.2.45:

```
[local]Redback(config-ctx)#router bgp 100  
[local]Redback(config-bgp)#neighbor 102.201.2.45 external  
[local]Redback(config-bgp-neighbor)#remote-as 4001
```

Related Commands

local-as
neighbor
router-id

remove-private-as

remove-private-as

no remove-private-as

Purpose

Removes private autonomous system numbers (ASNs) from routes that are advertised to the Border Gateway Protocol (BGP) neighbor address family or peer group address family.

Command Mode

BGP neighbor address family configuration

BGP peer group address family configuration

Syntax Description

This command has no keywords or arguments.

Default

The ASNs are not removed.

Usage Guidelines

Use the **remove-private-as** command to remove private ASNs from routes that are advertised to the BGP neighbor address family or peer group address family.

Use the **no** form of this command to send private ASNs.

Examples

The following example advertises BGP unicast routes to the neighbor at IP address 102.21.2.45. Any ASNs contained in these routes are removed.

```
[local]Redback(config-ctx)#router bgp 100  
[local]Redback(config-bgp)#neighbor 102.201.2.45 external  
[local]Redback(config-bgp-neighbor)#address-family ipv4 unicast  
[local]Redback(config-bgp-af)#remote-as 200  
[local]Redback(config-bgp-af)#remove-private-as
```

Related Commands

address-family ipv4

retain-ibgp-routes

retain-ibgp-routes

{no | default} retain-ibgp-routes

Purpose

Forces the Border Gateway Protocol (BGP) neighbor to retain routes from an internal BGP (iBGP) peer when the peer has restarted, provided the peer supports a graceful restart.

Command Mode

BGP neighbor configuration

Syntax Description

This command has no keywords or arguments.

Default

The command is disabled.

Usage Guidelines

Use the **retain-ibgp-routes** command to force the BGP neighbor to retain routes from an iBGP peer when the peer has restarted, provided the peer supports a graceful restart.

By default, routes are not retained for an iBGP peer after the peer restarts unless all iBGP peers support a graceful restart. However, in some network topologies, it may be desirable and feasible to retain the routes for an iBGP peer, even if not all iBGP peers support a graceful restart.

Use the **no** or **default** form of this command to disable this feature.

Examples

The following example forces the BGP neighbor, 10.1.1.1, to retain routes from an iBGP peer once the peer has restarted, provided the peer supports a graceful restart:

```
[local]Redback(config-bgp)#neighbor 10.1.1.1 internal  
[local]Redback(config-bgp-neighbor)#retain-ibgp-routes
```

Related Commands

maximum retain-time

route-map

route-map *map-name* {**in** | **out**}

no route-map *map-name* {**in** | **out**}

Purpose

Applies a route map that modifies Border Gateway Protocol (BGP) attributes or filters BGP routes received from or sent to the BGP neighbor or peer group.

Command Mode

BGP neighbor address family configuration

BGP peer group address family configuration

Syntax Description

<i>map-name</i>	Name of the route map.
in	Applies the route map to incoming BGP routes sent from the BGP neighbor.
out	Applies the route map to outgoing BGP routes sent to the BGP neighbor.

Default

A route map is not applied to a BGP neighbor.

Usage Guidelines

Use the **route-map** command to apply a route map that modifies BGP attributes or to filter BGP routes sent to or received from the BGP neighbor or peer group. Use the **in** keyword to modify attributes or filter incoming routes received from the neighbor or peer group. Use the **out** keyword to modify attributes or filter outgoing routes sent to the neighbor.

Use the **route-map** command in context configuration mode to determine the attribute modifications and filtering conditions of the applied route map.

Currently, route map changes automatically take effect, and issuing the **clear bgp neighbor ip-addr soft [in | out]** command in exec mode to update a route map can cause updates to be unnecessarily sent; therefore, it is not recommended.

To aggregate multiple policy changes, such as the route map, the SmartEdge OS performs the automatic update 15 seconds after any routing policy has changed.

Note If the remote peer does not support the BGP route refresh capability, an inbound policy change for the peer will result in an automatic hard reset of the session.

Use the **no** form of this command to remove a route map.

Examples

The following example denies unicast BGP routes 10.0.0.0/8 (and more-specific routes) sent from the unicast BGP neighbor at IP address 102.210.210.1. All other routes to this neighbor have the community attribute set to 100:14499. Only multicast BGP routes 204.16.16.0/24 are sent to the multicast BGP neighbor at IP address 68.68.68.68.

```
[local]Redback(config-ctx)#route-map rmap-20 deny 10
[local]Redback(config-route-map)#match ip address prefix-list prefix-deny-10
[local]Redback(config-route-map)#exit
[local]Redback(config-ctx)#route-map rmap-20 permit 20
[local]Redback(config-route-map)#set community 100:14499
[local]Redback(config-route-map)#exit
[local]Redback(config-ctx)#route-map rmap-30 permit 10
[local]Redback(config-route-map)#match ip address prefix-list prefix-permit-300
[local]Redback(config-route-map)#exit
[local]Redback(config-ctx)#ip prefix-list prefix-deny-10
[local]Redback(config-prefix-list)#permit 10.0.0.0/8 le 32
[local]Redback(config-prefix-list)#exit
[local]Redback(config-ctx)#ip prefix-list prefix-permit-300
[local]Redback(config-prefix-list)#permit 204.16.16.0/24
[local]Redback(config-prefix-list)#exit
.
.
.
[local]Redback(config-ctx)#router bgp 100
[local]Redback(config-bgp)#neighbor 102.210.210.1 external
[local]Redback(config-bgp-neighbor)#remote-as 200
[local]Redback(config-bgp-af)#exit
[local]Redback(config-bgp-neighbor)#address-family ipv4 unicast
[local]Redback(config-bgp-neighbor)#route-map rmap-200 in
[local]Redback(config-bgp-af)#exit
[local]Redback(config-bgp-neighbor)#exit
[local]Redback(config-bgp)#neighbor 68.68.68.68 external
[local]Redback(config-bgp-neighbor)#remote-as 300
[local]Redback(config-bgp-neighbor)#send community
[local]Redback(config-bgp-neighbor)#address-family ipv4 multicast
[local]Redback(config-bgp-af)#route-map rmap-300 out
```

Related Commands

- address-family ipv4
- default-originate
- local-as
- match ip address prefix-list
- redistribute
- route-map—context configuration mode

route-origin

route-origin *ext-com*

no route-origin

Purpose

Identifies the specific site from where a route has originated.

Command Mode

BGP address family configuration

Syntax Description

ext-com

Site of origin extended community value used to uniquely identify a site within internally connected multiple Virtual Private Network (VPN) sites. The site of origin extended community value can be expressed in either of the following formats:

- *asn:nnnn*, where *asn* is the autonomous system number and *nnnn* is a 32-bit integer.
- *ip-addr:nn*, where *ip-addr* is the IP address in the form *A.B.C.D* and *nn* is a 16-bit integer.

Default

No site of origin is specified.

Usage Guidelines

Use the **route-origin** command identify the specific site from where a route has originated.

When routes are received by a provider edge (PE) router, the route's route-origin attribute is checked against the route origin associated with the VPN for the receive site. Received routes are rejected if the route origin values are the same. This prevents the readvertisement of routes back to their originating sites.

Note The **route-origin** command is useful only when BGP is used for PE-to-customer edge (CE) routing.

Use the **no** form of this command to remove the route-origin attribute from a route.

Examples

The following example configures routes originating from context **foo** to carry route origin **100:300** as part of the extended community attribute when they are advertised to other PE routers:

```
[local]Redback(config)#context foo vpn-rd 10.11.12.13:100
[local]Redback(config-ctx)#router bgp vpn
[local]Redback(config-bgp)#address-family ipv4 unicast
[local]Redback(config-bgp-af)#route-origin 100:300
[local]Redback(config-bgp-af)#export route-target 10.11.12.13:100
[local]Redback(config-bgp-af)#import route-target 100:100 10.11.12.13:100
```

Related Commands

as-override

router bgp

router bgp {*asn* | *nn:nn*}

no router bgp {*asn* | *nn:nn*}

Purpose

Configures a Border Gateway Protocol (BGP) routing instance using an autonomous system number (ASN) and enters BGP router configuration mode.

Command Mode

context configuration

Syntax Description

<i>asn</i>	ASN in integer format. The range of values is 1 to 65,535. The subrange of 64,512 to 65,535 is reserved for private ASNs.
<i>nn:nn</i>	ASN in 4-byte integer format, where the first <i>nn</i> indicates the two higher-order bytes and the second <i>nn</i> denotes the two lower-order bytes.

Default

BPG routing is not enabled.

Usage Guidelines

Use the **router bgp** command to configure a BGP routing instance using an ASN, and to enter BGP configuration mode.

Use the **no** form of this command to disable the BGP routing instance.

Examples

The following example enables BGP routing for ASN 321 and enters BGP router configuration mode:

```
[local]Redback(config)#context local  
[local]Redback(config-ctx)#router bgp 321  
[local]Redback(config-bgp)#
```

Related Commands

router-id

route-reflector-client

route-reflector-client

no route-reflector-client

Purpose

Configures the internal Border Gateway Protocol (iBGP) neighbor (or peer group) as a route reflector client for the BGP address family.

Command Mode

BGP neighbor address family configuration
BGP peer group address family configuration

Syntax Description

This command has no keywords or arguments.

Default

The neighbor is not configured as a route reflector client.

Usage Guidelines

Use the **route-reflector-client** command to configure the iBGP neighbor (or peer group) for the specified address family as a route reflector client. No other configuration is required for an iBGP neighbor to act as a route reflector client.

Together, a route reflector and its clients form a cluster. If there is more than one route reflector in a cluster, all route reflectors in that cluster should be configured with the same ID through the **cluster-id** command. If there is no cluster ID, the router ID is used.

Note This command cannot be enabled on a BGP neighbor that is part of a peer group because this feature cannot be customized for individual members inside of a peer group.

Use the **no** form of this command to remove the route reflector client specification from the iBGP neighbor.

Examples

The following example configures the iBGP neighbor at IP address, 102.210.210.1, as a route reflector client for the unicast address family:

```
[local]Redback(config-ctx)#router bgp 100  
[local]Redback(config-bgp)#neighbor 102.210.210.1 external  
[local]Redback(config-bgp-neighbor)#remote-as 100  
[local]Redback(config-bgp-neighbor)#address-family ipv4 unicast  
[local]Redback(config-bgp-af)#route-reflector-client
```

Related Commands

`address-family ipv4`
`client-to-client reflection`
`cluster-id`

router-id

router-id *ip-addr*

no router-id *ip-addr*

Purpose

Configures a fixed Border Gateway Protocol (BGP) router ID for the SmartEdge router.

Command Mode

BGP router configuration

Syntax Description

ip-addr IP address of the SmartEdge router.

Default

The router ID is the IP address of a loopback interface, if one is configured. If a loopback interface is not configured, the interface with the highest IP address is used as the router ID.

Usage Guidelines

Use the **router-id** command to configure a fixed BGP router ID for the SmartEdge router.



Caution Risk of dropped connection. When you change a router ID, any active peering sessions using the current router ID are dropped. To reduce the risk, avoid changing the router ID when peering sessions are actively running.

Use the **no** form of this command to remove the fixed router ID.

Examples

The following example configures a fixed BGP router ID of 10.10.1.1:

```
[local]Redback(config-ctx)#router bgp 64001  
[local]Redback(config-bgp)#router-id 10.1.1.1
```

Related Commands

router bgp

router-id—context configuration mode

router-id—OSPF configuration mode

send community

send community

no send community

Purpose

Specifies that the community attribute is sent to the specified external Border Gateway Protocol (eBGP) neighbor or peer group.

Command Mode

BGP neighbor configuration

BGP peer group configuration

Syntax Description

This command has no keywords or arguments.

Default

The community attribute is not sent to the eBGP neighbor or peer group. The community attribute is always sent to internal BGP (iBGP) peers.

Usage Guidelines

Use the **send community** command to specify that the community attribute is sent to the specified eBGP neighbor or peer group.

Note This command is used only with eBGP neighbors or peer groups. The community attribute is always sent to iBGP peers.

Note You cannot enable this command on a BGP neighbor that is part of a peer group, because this feature cannot be customized for individual members inside of a peer group.

Use the **no** form of this command to restore the default behavior of not sending the community attribute to eBGP neighbors.

Examples

The following example sends the community attribute to the eBGP neighbor at IP address 123.45.34.2:

```
[local]Redback(config-ctx)#router bgp 100  
[local]Redback(config-bgp)#neighbor 123.45.34.2 external  
[local]Redback(config-bgp-neighbor)#remote as-200  
[local]Redback(config-bgp-neighbor)#send community
```

Related Commands

match community-list
neighbor
send ext-community

send filter prefix-list
send label
set community

send ext-community

send ext-community

no send ext-community

Purpose

Specifies that the extended community attribute is sent to the specified external Border Gateway Protocol (eBGP) neighbor or peer group.

Command Mode

BGP neighbor configuration
BGP peer group configuration

Syntax Description

This command has no keywords or arguments.

Default

The extended community attribute is not sent to the eBGP neighbor or peer group. The extended community attribute is always sent to internal BGP (iBGP) peers.

Usage Guidelines

Use the **send ext-community** command to specify that the extended community attribute is sent to the specified eBGP neighbor or peer group.

Note This command is used only with eBGP neighbors or peer groups. The extended community attribute is always sent to iBGP peers.

Note You cannot enable this command on a BGP neighbor that is part of a peer group, because this feature cannot be customized for individual members inside of a peer group.

Use the **no** form of this command to restore the default behavior of not sending the extended community attribute to eBGP neighbors.

Examples

The following example sends the extended community attribute to the eBGP neighbor at IP address 123.45.34.2:

```
[local]Redback(config-ctx)#router bgp 100  
[local]Redback(config-bgp)#neighbor 123.45.34.2 external  
[local]Redback(config-bgp-neighbor)#remote as-200  
[local]Redback(config-bgp-neighbor)#send ext-community
```


Related Commands

- `match ext-community-list`
- `neighbor`
- `send community`
- `send filter prefix-list`
- `send label`
- `set ext-community`

send filter prefix-list

send filter prefix-list

no send filter prefix-list

Purpose

Advertises to a Border Gateway Protocol (BGP) peer that a BGP speaker can send prefixed-based filtering to a peer.

Command Mode

BGP neighbor configuration

Syntax Description

This command has no keywords or arguments.

Default

The command is disabled.

Usage Guidelines

Use the **send filter prefix-list** command to advertise to a BGP peer that a BGP speaker can send address prefix-based route filtering to a peer.

When this command is enabled, and if the BGP peer advertises its willingness to accept address prefixed-based filtering (through the **accept filter prefix-list** command in BGP neighbor configuration mode), this local BGP speaker sends its inbound address prefix-based filtering to the remote peer. The remote peer uses the received address prefix-based filtering along with its local routing policies to determine whether or not routes should be advertised to the peer.

Use this command to save resources and avoid the generation, transmission, and processing of unnecessary routing updates.

Note This command cannot be enabled on a BGP neighbor that is part of a peer group because this feature cannot be customized for individual members inside of a peer group.

Use the **show bgp neighbor ip-addr received prefix-filter** command to display address prefix-based route filtering configuration information.

Use the **no** form of this command to disable a BGP speaker from accepting route filtering from a peer.

For further information, see the Internet Drafts, *Cooperative Route Filtering Capability for BGP-4*, draft-ietf-idr-route-filter-03.txt, and *Address Prefix Based Outbound Route Filter for BGP-4*, draft-chen-bgp-prefix-orf-02.txt.

Examples

The following example enables the external BGP (eBGP) speaker at IP address, 10.1.1.1, to send outbound route filters to BGP peers:

```
[local]Redback(config-bgp)#neighbor 10.1.1.1 external  
[local]Redback(config-bgp-neighbor)#send filter prefix-list
```

Related Commands

- accept filter prefix-list**
- neighbor**
- prefix-list**
- send community**
- send ext-community**
- send label**

send label

send label

no send label

Purpose

Enables a Border Gateway Protocol (BGP) router to send Multiprotocol Label Switching (MPLS) labels with BGP IP Version 4 (IPv4) routes to a peer BGP router.

Command Mode

BGP address family configuration

Syntax Description

This command has no keywords or arguments.

Default

BGP routers distribute BGP IPv4 unicast routes without MPLS labels.

Usage Guidelines

Use the **send label** command to enable a BGP router to send MPLS labels with BGP IPv4 routes to a peer BGP router.

Note You must configure this command on both the local router and the peer router in order for the routers to send IPv4 unicast routes with MPLS labels.

One application for this command is the BGP/MPLS Virtual Private Network (VPN) Carrier Supporting Carrier configuration. The user must configure this command on the provider edge (PE) and customer edge (CE) routers between the super carrier and the ISP carrier.

This command has the following restrictions:

- If the **send label** command is configured for a peer that is already up, the BGP session with that peer will be automatically reset to make the configuration effective.
- The **send label** command is only used with the IPv4 unicast address family, and is available only in an eBGP peer configuration.

Use the **no** form of this command to disable the BGP router from sending MPLS labels with IPv4 unicast routes.

Examples

The following example enables the local router to send MPLS labels along with BGP IPv4 unicast routes to peer 1.1.1.1:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#router bgp 100
[local]Redback(config-bgp)#neighbor 1.1.1.1 external
```

```
[local]Redback(config-bgp-neighbor)#send label  
[local]Redback(config-bgp-neighbor)#address-family ipv4 unicast  
[local]Redback(config-bgp-af)#
```

Related Commands

```
neighbor  
send community  
send ext-community  
send filter prefix-list
```

session-dampening

session-dampening [*half-life reuse suppress max-suppress-time*]

no session-dampening

Purpose

Enables a flapping peer to be temporarily suppressed for a configurable amount of time.

Command Mode

BGP neighbor configuration
BGP peer group configuration

Syntax Description

<i>half-life</i>	Optional. Time, in minutes, after which a penalty is decreased. Once the session has been assigned a penalty, the penalty is decreased by half after the half-life period. The process of reducing the penalty occurs every 5 seconds. The range of values for the half-life period is 1 to 45; the default value is 15.
<i>reuse</i>	Optional. Value that determines whether a session is unsuppressed and can be reused. When a penalty for a flapping peer decreases to the point that it falls below this value, the session is unsuppressed and can be reused. Sessions are scanned for reuse every 5 seconds. The range of values is 1 to 20,000; the default value is 1,500.
<i>suppress</i>	Optional. Value that determines if a session is suppressed. A session is suppressed when its penalty exceeds this limit. The range of values is 1 to 20,000; the default value is 3,000.
<i>max-suppress-time</i>	Optional. Maximum time (in minutes) a session can be denied to open. The range of values is 1 to 255; the default value is four times the <i>half-life</i> argument. If the half life value is allowed to default, the <i>maximum-suppress</i> value defaults to 60.

Default

Session dampening is disabled.

Usage Guidelines

Use the **session-dampening** command to enables a flapping peer to be temporarily suppressed for a configurable amount of time.

This command is per peer and peer-group based. If the peer is member of a peer group, the command is inherited from the peer-group and can be customized in the peer configuration.

The main benefit of this feature is to avoid flapping peers from using system resources, and also to reduce routing churn induced by a flapping peer.

A message is logged when a session is dampened and undampened.

Note Session dampening is different from route dampening. Session dampening dampens peers when it is reset, and route dampening dampens routes from a peer in established states.

Use the **no** form of this command to disable session dampening.

Examples

The following example enables session dampening with a half life of 5 minutes, a reuse value of 1000, a suppress value of 4000, and a maximum suppress time of 10 minutes:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#router bgp 100
[local]Redback(config-bgp)#peer-group pi internal
[local]Redback(config-bgp-peer-group)#session-dampening 5 1000 4000 10
```

Related Commands

dampening
flap-statistics

shutdown

shutdown

no shutdown

Purpose

Administratively shuts down the Border Gateway Protocol (BGP) session with the specified neighbor or peer group.

Command Mode

BGP neighbor configuration
BGP peer group configuration

Syntax Description

This command has no keywords or arguments.

Default

None

Usage Guidelines

Use the **shutdown** command to administratively shut down the BGP session with the specified neighbor or peer group. This command is useful to temporarily shut down a session without removing the BGP neighbor from the SmartEdge router configuration.

Use the **no** form of this command to restore the BGP session between the SmartEdge router and the specified neighbor.

Examples

The following example administratively shuts down the BGP session with the neighbor at IP address 10.100.3.2:

```
[local]Redback(config-ctx)#router bgp 64001  
[local]Redback(config-bgp)#neighbor 10.100.3.2 external  
[local]Redback(config-bgp-neighbor)#shutdown
```

Related Commands

neighbor

table-map

table-map *map-name*

no table-map *map-name*

Purpose

Assigns a traffic index to routes installed for a Border Gateway Protocol (BGP) address family.

Command Mode

BGP address family configuration

Syntax Description

<i>map-name</i>	Name of the route map.

Default

A table map is not applied to a BGP address family.

Usage Guidelines

Use the **table-map** command to assign a traffic index to routes installed for a BGP address family.

Traffic index counters are maintained on interfaces with traffic index accounting enabled. Traffic indices are associated with BGP routes based on route-maps matching on BGP attributes. When IP packets are received on an interface with traffic index accounting enabled, and the route lookup for the packet's destination IP address corresponds to a BGP route with a traffic index assigned, the corresponding byte and packet counters are incremented. For more information, see the **set traffic-index** and **traffic-index accounting** commands.

Use the **route-map** command in BGP neighbor address family configuration mode and BGP peer group address family configuration mode to determine the attribute modifications and filtering conditions of the applied route map.

Use the **no** form of this command to remove the table map.

Examples

The following example assigns a traffic index to routes installed for a BGP address family using the `bgp-accounting` route map:

```
[local]Redback(config-ctx)#router bgp 64001
[local]Redback(config-bgp)#address-family ipv4 unicast
[local]Redback(config-bgp-af)#table-map bgp-accounting
```

Related Commands

route-map
set traffic-index

traffic-index accounting

timer password

timer password *interval*

no timer password *interval*

Purpose

Configures the time interval, in seconds, during which an old Message Digest 5 (MD5) password can co-exist with a new MD5 password for authentication.

Command Mode

BGP router configuration

Syntax Description

<i>interval</i>	Interval, in seconds, during which the new and old MD5 passwords co-exist. The range of values is 1 to 3,600.

Default

The timer interval is set to 1,800 seconds.

Usage Guidelines

Use the **timer password** command to configure the time interval, in seconds, during which an old MD5 password can co-exist with a new MD5 password for authentication. Configuring the password timer interval affects only the Border Gateway Protocol (BGP) peers which have existing MD5 passwords replaced after this configuration is committed.

Examples

The following example allows new MD5 passwords for BGP peers to co-exist with the password being replaced for 300 seconds (five minutes):

```
[local]Redback(config-ctx)#router bgp 1000  
[local]Redback(config-bgp)#timer password 300
```

Related Commands

password—BGP neighbor configuration mode

timers

timers **keepalive** *interval* **holdtime** *interval*
no timers **keepalive** *interval* **holdtime** *interval*

Purpose

Modifies Border Gateway Protocol (BGP) timers for the routing instance, neighbor, or peer group.

Command Mode

BGP neighbor configuration
BGP peer group configuration
BGP router configuration

Syntax Description

keepalive <i>interval</i>	Interval, in seconds, at which the BGP routing process sends keepalive messages. The range of values is 1 to 65,535; the default value is 60.
holdtime <i>interval</i>	Interval, in seconds, after which, if the BGP routing process has not received a keepalive message, it considers the neighbor to be unavailable. The range of values is 3 to 65,535; the default value is 180.

Default

The keepalive time is 60 seconds. The holdtime is 180 seconds.

Usage Guidelines

Use the **timers** command in BGP router configuration mode to modify keepalive and holdtime timers for all BGP neighbors.

Use the **timers** command in BGP neighbor configuration mode to modify keepalive and holdtime timers for a specific neighbor. Values set for a BGP neighbor override the values set for the BGP routing instance.

Use the **timers** command in BGP peer group configuration mode to modify keepalive and holdtime timers for a peer group.

Note If a neighbor is part of a peer group, and you try to apply this command in BGP neighbor configuration mode, the timer conditions are not applied to the neighbor. Use the **timers** command in BGP peer group configuration mode instead.

Use the **no** form of this command to restore timer settings to their default values.

Examples

The following example sets the keepalive period to 45 seconds and the holdtime to 135 seconds for only the neighbor at IP address 123.45.34.2:

```
[local]Redback(config-ctx)#router bgp 100  
[local]Redback(config-bgp)#neighbor 123.45.34.2 external  
[local]Redback(config-bgp-neighbor)#timers keepalive 45 holdtime 135
```

Related Commands

advertisement-interval
fast-reset

update-source

update-source *if-name*

no update-source

Purpose

Specifies the IP address of the interface used for Border Gateway Protocol (BGP) peering.

Command Mode

BGP neighbor configuration

BGP peer group configuration

Syntax Description

<i>if-name</i>	Name of the interface used to bring up the BGP session.

Default

The SmartEdge router brings up BGP sessions using any interface.

Usage Guidelines

Use the **update-source** command to assign the interface used to bring up a BGP session with the specified neighbor or peer group.

Use the **no** form of this command to bring up BGP sessions using any interface.

Examples

The following example configures loopback0 as the interface used to bring up BGP sessions with the neighbor at IP address 123.45.34.2:

```
[local]Redback(config-ctx)#router bgp 100  
[local]Redback(config-bgp)#neighbor 123.45.34.2 external  
[local]Redback(config-bgp-neighbor)#remote-as 200  
[local]Redback(config-bgp-neighbor)#update-source loopback0
```

Related Commands

neighbor

BGP/MPLS VPN Configuration

This chapter provides an overview of the Border Gateway Protocol/Multiprotocol Label Switching Virtual Private Network (BGP/MPLS VPN) and describes the tasks and commands used to configure BGP/MPLS VPN features through the SmartEdge[®] OS.

For information about the tasks and commands used to monitor, troubleshoot, and administer BGP/MPLS VPNs, see the “BGP/MPLS VPN Operations” chapter in the *Routing Protocols Operations Guide* for the SmartEdge OS.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples
- Command Descriptions

Overview

The following sections provide an overview of BGP/MPLS VPN concepts:

- Virtual Private Networks
- VPN Topology
- Packet Labels
- Multiple VPN Contexts
- VPN-IPv4 Address Family
- Route Distribution Among PE Routers by BGP
- PE-to-CE Route Distribution
- Route Target Attribute
- Site of Origin Attribute
- BGP/MPLS VPN over GRE
- GRE over MPLS

- Carrier of Carriers
- Multihop eBGP Label Redistribution

Virtual Private Networks

In its most general definition, a Virtual Private Network (VPN) is a network in which customer connectivity among multiple remote sites is deployed across a shared central infrastructure, yet still provides the same access or security as a private network.

More specifically, a BGP/MPLS VPN is a collection of policies, and these policies control connectivity among a set of sites. A customer site is connected to the service provider network, often called a backbone, by one or more ports, where the service provider associates each port with a VPN context.

BGP/MPLS VPN allows you to implement a wide range of policies; for example, within a given VPN, you can allow every site to have a direct route to every other site (full mesh), or you can restrict certain pairs of sites from having direct routes to each other (partial mesh).

VPN Topology

A typical BGP/MPLS VPN topology consists of multiple customer sites connected to a central service provider site. Customer edge (CE) routers provide customer access to the service provider network over a data link to one or more provider edge (PE) routers. The CE routers establish an adjacency with their directly connected PE routers, and after the adjacency is established, the CE routers advertise their site's local VPN routes to the PE router and learn remote VPN routes from the PE router.

PE routers can exchange routing information with CE routers using static routing, Routing Information Protocol Version 2 (RIPv2), Open Shortest Path First (OSPF), or Border Gateway Protocol (BGP). PE routers maintain VPN routing information for the VPNs to which they are directly attached.

Packet Labels

With BGP/MPLS VPNs, there are typically two labels in a packet: an Interior Gateway Protocol (IGP) label (tunnel label) and a VPN label. The IGP label is used in delivering the packet from an ingress PE router to the egress PE router, where the CE router is attached. The VPN label is used by the egress PE router to deliver the packet out of the interface connected to the proper CE router.

Multiple VPN Contexts

PE routers maintain a separate VPN context for each VPN connection. Each customer connection, such as Frame Relay permanent virtual circuit (PVC), Asynchronous Transfer Mode (ATM) PVC, or virtual LAN (VLAN), is mapped to a specific VPN context. Multiple ports on a PE router can be associated with a single VPN context; however, it is the ability of PE routers to maintain multiple VPN contexts that supports the per-VPN segregation of routing information.

PE routers advertise VPN routes learned from CE routers using internal Border Gateway Protocol (iBGP). PE routers can maintain iBGP sessions to route reflectors as an alternative to a full mesh of iBGP sessions. Deploying multiple route reflectors enhances network scalability because it eliminates the need for any single network component to maintain all VPN routes.

MPLS is used to forward VPN data traffic across the provider's backbone, the ingress PE router functions as the ingress label edge router (LER), and the egress PE router functions as the egress LER.

VPN-IPv4 Address Family

VPN customers often manage their own networks and use private IP addresses. If globally unique IP addresses are not used, the same IP Version 4 (IPv4) address can be used to identify different systems in different VPNs; however, BGP assumes that each IPv4 address it carries is globally unique, so routing problems can occur. BGP/MPLS VPNs solves this problem by converting duplicate IP addresses into globally unique addresses by using VPN-IPv4 address families.

MBGP extensions allow BGP to carry routes from multiple address families. A VPN-IPv4 address is a 12-byte quantity, beginning with an 8-byte route distinguisher (RD), and ending with a 4-byte IPv4 address. If two VPNs use the same IPv4 address prefix, the PE routers translate these into unique VPN-IPv4 address prefixes, which ensures that if the same address is used in two different VPNs, it is possible to install two completely different routes to that address, one for each VPN.

Note The RD contains no information about the origin of the route, or about the set of VPNs to which the route is to be distributed. The purpose of the RD is to allow you to create distinct routes to a common IPv4 address prefix.

A PE router must be configured to associate routes that lead to particular CE router with a particular RD. The PE router can be configured to associate all routes leading to the same CE router with the same RD, or it can be configured to associate different routes with different RDs, even if they lead to the same CE router.

Route Distribution Among PE Routers by BGP

PE routers can distribute VPN-IPv4 routes to each other by means of an iBGP connection. When a PE router distributes a VPN-IPv4 route using BGP, it uses its own address as the BGP next hop. It also assigns and distributes an MPLS label. When the PE router processes a received packet that has this label at the top of the stack, the PE router pops the stack, and sends the packet directly to the site from to which the route leads. This usually means that it just sends the packet to the CE router from which it learned the route.

The MPLS label that is distributed by the PE router requires a label-switched path (LSP) between the router that installs a route and the BGP next hop of that route. That is, an MPLS LSP must be configured for VPN route distribution to operate.

PE-to-CE Route Distribution

PE routers attached to a particular VPN must learn the addresses from that VPN. The PE router translates these addresses into VPN-IPv4 addresses using a configured RD. The PE router then uses the VPN-IPv4 routes as input to BGP.

Possible CE-to-PE distribution methods include:

1. Static routing can be used.
2. CE and PE routers can be Routing Information Protocol (RIP) peers, and the CE router can use RIP to tell the PE router the set of address prefixes which are reachable at the CE router's site.
3. CE and PE routers can be OSPF peers. If the CE routers at the customer site contain more than one OSPF area, the PE-to-CE connection should be in area 0, and the CE and PE routers should be configured as area border routers (ABRs). If the CE routers at the customer site only contain a single OSPF area, then the PE-to-CE connection can be in that area, or area 0.
4. CE and PE routers can be BGP peers, and the CE router can use eBGP to tell the PE router the set of address prefixes, which are at the CE router's site.

Route Target Attribute

When a VPN-IPv4 route is created by a PE router, it is associated with one or more BGP extended community route target attributes. The route target attribute identifies a collection of sites to which a PE router distributes routes. A PE router uses this attribute to constrain the import of remote routes into its routing tables.

Before accepting routes that have been distributed by another PE router, each VPN context on a PE router is configured with an import route target policy. A PE router can only add a VPN-IPv4 route to a routing table for the VPN if the route target attribute carried with the route matches one of the import route targets on the PE router for the VPN.

Site of Origin Attribute

The site of origin attribute uniquely identifies the site from which the PE router learned the route. All routes learned from a particular site must be assigned the same site of origin attribute, even if a site has multiple connections to a single PE router, or is connected to multiple PE routers. Distinct site of origin attributes must be used for distinct sites.

The site of origin attribute is used to avoid routing loops in situations where multiple VPN sites using the AS override feature are internally connected.

BGP/MPLS VPN over GRE

Encapsulating packets via Generic Routing Encapsulation (GRE) from an ingress PE router to an egress PE router is called soft GRE tunneling. Soft GRE tunnels are not Interior Gateway Protocol (IGP) visible links, and routing adjacencies are not supported across these tunnels. As a result, soft GRE tunnels have little in common with traditional (hard) GRE tunnels. The tunnel exists only in the sense of GRE encapsulation and decapsulation.

Only the ingress PE router and the egress PE router need to support the soft GRE functionality, and the PE routers can span over multiple autonomous systems.

Using soft GRE tunnels to transport MPLS-encapsulated packets is called BGP/MPLS VPN over GRE, and is used to offer BGP/MPLS VPN service when a portion of a network does not have label switching enabled. BGP/MPLS VPN over GRE does not require preconfiguration of the remote GRE endpoint. These endpoints are the BGP next-hop addresses of the VPN routes and are learned dynamically via BGP.

GRE over MPLS

GRE over MPLS provides a way to establish a GRE tunnel over an MPLS LSP, allowing you to run applications, such as multicast, over the GRE tunnel. For GRE to work properly over MPLS, VPN contexts must be configured at both ends of the GRE tunnel.

To configure GRE over MPLS, you must perform the following tasks:

1. Configure BGP/MPLS VPN at both ends of the GRE tunnel.
2. Configure the GRE tunnel in the local VPN context. The tunnel remote IP address for the GRE tunnel must be an IP address in the remote VPN context.

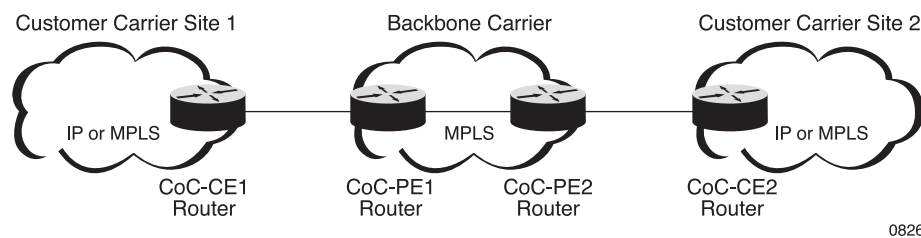
For a detailed GRE over MPLS configuration example, see the “Configuration Examples” section.

Carrier of Carriers

The carrier of carriers (CoC) feature provides a way for a service provider to use a segment of another service provider's backbone network to transport traffic between two geographically separated networks. The service provider that uses CoC to connect its two networks is called the customer carrier, and the service provider that provides a segment of its backbone network is called the backbone carrier.

The BGP/MPLS VPN implementation of the CoC feature uses eBGP to distribute MPLS labels in IPv4 unicast routes between customer carrier CE routers and backbone carrier PE routers. The backbone carrier uses MPLS to route traffic across its backbone network. The customer carrier can use either IP or MPLS routing in its networks. Figure 9-1 displays the network topology for a typical BGP/MPLS VPN CoC configuration.

Figure 9-1 Typical BGP/MPLS VPN CoC Network Topology



Note If a non-SmartEdge router is used as a CoC-PE or CoC-CE router, that router must support IPv4 BGP label distribution. For more information about IPv4 label distribution, see RFC 3107, *Carrying Label Information in BGP-4*.

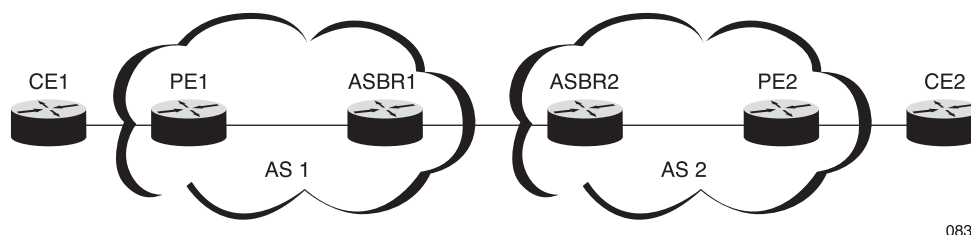
The BGP/MPLS VPN CoC implementation adheres to the following rules:

- All routers within the customer carrier network must be fully meshed using iBGP peering.
- The Label Distribution Protocol (LDP) must be enabled on the backbone link between the CoC-PE routers in the backbone network. In addition, the CoC-PE routers must be fully meshed using iBGP peering within the autonomous system.
- The two customer carrier networks being connected through the backbone carrier must have the same autonomous system numbers (ASNs).
- Within the customer carrier autonomous system, in addition to the iBGP peering on the backbone links between the PE and CoC-CE routers, an IGP, such as OSPF or IS-IS, must be enabled. By default, the loopback interface IP address is used as both the router ID and iBGP peering address, so it must be reachable.
- For better scalability on the links in a backbone carrier network, only the iBGP routes from the customer carrier networks are sent across the backbone carrier network.

Multihop eBGP Label Redistribution

The multihop eBGP label redistribution feature enables you to configure a VPN network that redistributes labeled IPv4 VPN routes between source and destination autonomous systems using eBGP redistribution of labeled IPv4 routes from a local autonomous system (AS) to a neighboring AS. Figure 9-2 displays the network topology for a typical multihop eBGP label redistribution configuration.

Figure 9-2 Typical Multihop eBGP Label Redistribution Network Topology



The ASBRs do not maintain or distribute IPv4 VPN routes. Instead, each ASBR must maintain labeled IPv4 routes to the PE routers within its AS. The routers use eBGP to distribute the routes to other autonomous systems. ASBRs in any transit AS must also use eBGP to forward the labeled routes. This creates a label-switched path from the ingress PE router to the egress PE router, allowing PE routers in different autonomous systems establish multihop eBGP connections to each other, and exchange VPN-IPv4 routes over those connections.

Configuration Tasks

Note In this section, the command syntax in the task tables displays only the root command; for the complete command syntax, see the full description for the command in the “Command Descriptions” section.

To configure BGP/MPLS VPNs, perform the tasks described in the following sections:

- Configuring a VPN-IPv4 Address Family for BGP Sessions Between PE Routers
- Creating a New VPN Context
- Configuring a BGP Routing Instance in a VPN Context
- Configuring Multipath Load Balancing in a BGP/MPLS VPN
- Configuring the Next-Hop Reachability Check for VPN Routes
- Configuring Route Targets
- Configuring PE-to-CE Routing
- Identifying the Specific Site from Where a Route Has Originated
- Enabling Soft GRE Tunneling

Configuring a VPN-IPv4 Address Family for BGP Sessions Between PE Routers

To configure a VPN-IPv4 address family for BGP sessions between PE routers, perform the tasks described in Table 9-1. The Notes column lists the configuration mode in which you enter commands.

Table 9-1 Configure a VPN-IPv4 Address Family for BGP Sessions Between PE Routers

Task	Root Command	Notes
Configure a BGP routing instance in the local context, and access BGP configuration mode.	router bgp	Enter this command in context configuration mode. For detailed information about this command, see Chapter 8, “BGP Configuration.”
Enable VPN-IPv4 prefixes for a BGP routing instance and enter BGP address family configuration mode.	address-family ipv4 vpn	Enter this command in BGP configuration mode. This command cannot be used in non-local contexts.
Enable VPN-IPv4 prefixes for a specified BGP neighbor in an iBGP session, and to access BGP neighbor address family configuration mode.	address-family ipv4 vpn	Enter this command in BGP neighbor configuration mode. This command cannot be used in non-local contexts.
Enable VPN-IPv4 prefixes for a specified BGP peer group, and to enter BGP peer group address family configuration mode.	address-family ipv4 vpn	Enter this command in BGP peer group configuration mode. This command cannot be used in non-local contexts.

Creating a New VPN Context

To configure a new VPN context, perform the tasks described in Table 9-2. Enter all commands in global configuration mode.

Table 9-2 Configure a New VPN Context

Task	Root Command	Notes
Enable the multiple context feature.	service multiple-contexts	For more information about the service multiple-contexts command, see the “Context Configuration” chapter in the <i>Basic System Configuration Guide</i> for the SmartEdge OS.
Create a new VPN context, and enter context configuration mode.	context vpn-rd	You cannot create new contexts on the system unless you have enabled the multiple context feature using the service multiple-contexts in global configuration mode. Entering the full context vpn-rd command is required to configure a VPN context. Entering the command without the vpn-rd portion creates a context that will not be recognized as VPN-enabled.

Configuring a BGP Routing Instance in a VPN Context

To configure a BGP routing instance in a VPN context, perform the task described in Table 9-3. Enter the command in context configuration mode.

Table 9-3 Configure a BGP Routing Instance in a VPN Context

Task	Root Command	Notes
Configure a BGP routing instance in a VPN context, and enter BGP configuration mode.	router bgp vpn	A BGP instance is always required within a VPN context for the following reasons: • Customer routes must be distributed into BGP so they can be advertised across the iBGP sessions that connect PE routers. Customer routes can be distributed into BGP either statically or from other active routing protocols. • Route targets must also be configured within BGP address family configuration mode. BGP does not function properly in a VPN context until it is first configured in the local context. Even though an ASN is not used when configuring a BGP instance in a VPN context, this instance uses the ASN from the BGP instance in the local context for peering with CE routers. When configuring BGP peering sessions within a VPN context, only external neighbor sessions can be configured, because peering in a VPN context must only be configured with CE routers. Also, the only permitted address family is IPv4 unicast, and peer groups cannot be configured.

Configuring Multipath Load Balancing in a BGP/MPLS VPN

To configure multipath load balancing in a BGP/MPLS VPN, perform the task described in Table 9-4. Enter the command in BGP router configuration mode.

Table 9-4 Configure Multipath Load Balancing in a BGP/MPLS VPN

Task	Root Command	Notes
Configure multipath load balancing using both eBGP and iBGP equal-cost paths in a BGP/MPLS VPN.	multi-paths eibgp	

Configuring the Next-Hop Reachability Check for VPN Routes

To configure the next-hop reachability check for VPN routes, perform the task described in Table 9-5. Enter the command in BGP router configuration mode.

Table 9-5 Configure the Next-Hop Reachability Check for VPN Routes

Task	Root Command	Notes
Require the next hop of a BGP VPN path to be reachable through an MPLS LSP or a tunnel in order for a VPN route to be considered active.	next-hop-on-lsp	Use the no form of this command to enable a BGP VPN path to be considered active without requiring the next hop of a VPN path to be reachable through an MPLS LSP or a tunnel. One common application for this command is when configuring a BGP route reflector that is not part of an MPLS network, but is used to reflect BGP VPN routes to its clients within that MPLS network. In this configuration, the next hops of the VPN paths may not be reachable through an MPLS LSP or a tunnel from the route reflector's point of view. To solve the problem, use the no form of the this command command to disable the LSP or tunnel reachability check for the next hops, and therefore allow the BGP route reflector to correctly select the best paths and reflect the best paths to its clients.

Configuring Route Targets

To configure route targets, perform the tasks described in Table 9-6. Enter all commands in BGP address family configuration mode.

Table 9-6 Configure Route Targets

Task	Root Command	Notes
Create a list of export route target extended communities for a specified VPN context.	export route-target	You can add multiple target communities on the same line, or you can issue the command multiple times with a single target as the parameter. Export route targets are sent as extended community attributes to other PE routers. An export route map can be configured instead of a single target community value to give finer control over exported BGP routes. A route map allows you to filter routes or change attributes such as the export route target based on policy requirements. A route map may only be used when a target community value has not yet been configured. This command can only be used in VPN contexts.
Create a list of import route target extended communities for a specified VPN context.	import route-target	You can add multiple target communities on the same line, or you can issue the command multiple times with a single target as the parameter. BGP routes learned from other PE routers that carry a specific route target extended community are imported into all VPN contexts configured with that extended community as an import route target. This command can only be used in VPN contexts.
Enable automatic BGP route target community filtering.	route-target filter	This command configures the local router, if it is not configured as a route reflector, to ignore all VPN routes received that are not imported into any VPN context. You can control the number of IPv4 VPN routes that the local autonomous system border router (ASBR) advertise to the remote ASBR by configuring a community for exportable routes on the inbound interface of the PE router, and configuring a community based filter on the outbound interface of the local ASBR to advertise only routes that match the community.

Configuring PE-to-CE Routing

To configure PE-to-CE routing, perform the tasks described in Table 9-7. Enter all commands in BGP router configuration mode, unless otherwise noted.

Table 9-7 Configure PE-to-CE Routing

Task	Root Command	Notes
Disable the AS_PATH loop detection by accepting a route advertisement that contains the local ASN in the AS_PATH attribute.	asloop-in	Because enabling the asloop-in command disables AS_PATH loop detection, it must only be used for specific applications that require this type of behavior, and in situations with strict network control; for example, the BGP/MPLS VPN hub-and-spoke configuration, in which a hub PE router may receive routes containing its own ASN from a hub CE router. To disable AS_PATH loop detection, use the asloop-in command on the exporting context of the hub PE router. The asloop-in command is useful only when BGP is used for PE-to-CE routing. For a CE router to send a route advertisement back to the PE router from which the route is learned, the CE router must be configured as a BGP peer with the PE router configured as a member of the peer group. By default, routes are not sent back to the neighbor AS from where they are received.
Replace all occurrences of a peer's ASN in the AS_PATH attribute of a route with the local ASN, when advertising the route to the peer.	as-override	When multiple VPN sites share the same ASN, enabling the AS override feature allows routes originating from an AS to be accepted by a router residing in the same AS. By default, the receiving router rejects the received route advertisement if the AS_PATH attribute shows that the route originated from its own AS to prevent routing loops. The as-override command is useful only when BGP is used for PE-to-CE routing. Enabling the AS override feature may result in route loops. This feature should only be used for specific applications that require this type of behavior, and in situations with strict network control. The as-override command can only be used in VPN contexts.
Enable an OSPF instance within a VPN context to treat redistributed BGP routes as VPN routes.	vpn	When a CE site is connected to multiple areas, the CE router's connection to a PE router should be in area 0 to allow correct handling of summary link-state advertisements (LSAs). The vpn command is useful only when OSPF is used for PE-to-CE routing.

Identifying the Specific Site from Where a Route Has Originated

To identify the specific site from where a route has originated, perform the task described in Table 9-8. Enter the command in BGP address family configuration mode.

Table 9-8 Identify the Specific Site from Where a Route Has Originated

Task	Root Command	Notes
Identify the specific site from where a route has originated.	route-origin	When routes are received by a PE router, the route's route-origin attribute is checked against the route origin associated with the VPN for the receive site. Received routes are rejected if the route origin values are the same. This prevents the readvertisement of routes back to their originating sites. This command is useful only when BGP is used for PE-to-CE routing.

Enabling Soft GRE Tunneling

To enabling soft GRE tunneling, perform the task described in Table 9-9. Enter the command in context configuration mode.

Table 9-9 Enable Soft GRE Tunneling

Task	Root Command	Notes
Enable soft GRE tunneling on the specified context.	ip soft-gre	Using soft GRE tunnels to transport MPLS-encapsulated packets is called BGP/MPLS VPN over GRE, and is used to offer BGP/MPLS VPN service when a portion of a network does not have label switching enabled. BGP/MPLS VPN over GRE does not require a preconfiguration of the remote GRE endpoint. These endpoints are the BGP next-hop addresses of the VPN routes and are learned dynamically via BGP.

Configuration Examples

This section provides BGP/MPLS VPN configuration examples in the following sections:

- Backbone Connectivity
- PE-to-CE Route Distribution
- Different BGP/MPLS VPN Topologies
- GRE over MPLS
- BGP/MPLS VPN over GRE
- New BGP Commands for BGP/MPLS VPN
- CoC
- Multihop eBGP Label Redistribution

Backbone Connectivity

The backbone connectivity must be configured in the local context.

An IGP, such as OSPF, IS-IS, or LDP must be enabled on backbone links. By default the loopback interface IP address is used as both the router ID and LDP transport address, so it needs to be reachable. Furthermore, MPLS switching must be enabled on the backbone links.

The following configuration allows two routers carry BGP routes for VPN-IPv4 unicast addresses. A VPN-IPv4 unicast address is an 8 to 12 byte quantity, beginning with an 8-byte RD and ending with an IPv4 address.

Note A VPN-IPv4 address family must be configured for the BGP PE peers. IPv4 unicast and multicast address families can be enabled for the same peers if needed.

The configuration for the PE1 router is as follows:

```
[local]PE1#config
[local]PE1(config)#context local
[local]PE1(config-ctx)#interface loop1 loopback
```

```
[local]PE1(config-if)#ip address 1.1.1.1/32
[local]PE1(config-if)#isis router isis-backbone
[local]PE1(config-if)#isis passive-interface
[local]PE1(config-ctx)#interface backbone1
[local]PE1(config-if)#ip address 2.2.2.1/24
[local]PE1(config-if)#isis router isis-backbone
[local]PE1(config-ctx)#router isis
[local]PE1(config-isis)#net 49.2222.0010.0100.1001.00
[local]PE1(config-ctx)#router mpls 1
[local]PE1(config-mpls)#interface backbone1
[local]PE1(config-ctx)#router ldp
[local]PE1(config-ldp)#interface backbone1
[local]PE1(config-ctx)#router bgp 100
[local]PE1(config-bgp)#neighbor 1.1.1.2 internal
[local]PE1(config-bgp-neighbor)#update-source loop1
[local]PE1(config-bgp-neighbor)#next-hop-self
[local]PE1(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE1(config)#port pos 6/1
[local]PE1(config-port)#bind interface backbone1 local
[local]PE1(config-port)#no shutdown
[local]PE1(config-port)#end
```

The configuration for the PE2 router is as follows:

```
[local]PE2#config
[local]PE2(config)#context local
[local]PE2(config-ctx)#interface loop1 loopback
[local]PE2(config-if)#ip address 1.1.1.2/32
[local]PE2(config-if)#isis router isis-backbone
[local]PE2(config-if)#isis passive-interface
[local]PE2(config-ctx)#interface backbone1
[local]PE2(config-if)#ip address 2.2.2.2/24
[local]PE2(config-if)#isis router isis-backbone
[local]PE2(config-ctx)#router isis
[local]PE2(config-isis)#net 49.2222.0010.0100.1002.00
[local]PE2(config-ctx)#router mpls 1
[local]PE2(config-mpls)#interface backbone1
[local]PE2(config-ctx)#router ldp
[local]PE2(config-ldp)#interface backbone1
[local]PE2(config-ctx)#router bgp 100
[local]PE2(config-bgp)#neighbor 1.1.1.1 internal
[local]PE2(config-bgp-neighbor)#update-source loop1
[local]PE2(config-bgp-neighbor)#next-hop-self
[local]PE2(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE2(config)#port pos 6/1
[local]PE2(config-port)#bind interface backbone1 local
[local]PE2(config-port)#no shutdown
[local]PE2(config-port)#end
```

PE-to-CE Route Distribution

PE-to-CE route distribution can be configured using any of the following techniques:

- VPN Using Static Routing
- VPN Using RIP
- VPN Using OSPF
- VPN Using eBGP

Note This section does not include the configuration for the backbone connectivity in the local context.

Note You must configure the **service multiple-context** command in order to configure a VPN context.

VPN Using Static Routing

The configuration for the PE router is as follows:

```
[local]PE#config
[local]PE(config)#service multiple-context
[local]PE(config)#context VPN1 vpn-rd 1.1.1.1:101
[local]PE(config-ctx)#interface 12/1
[local]PE(config-if)#ip address 10.10.1.1/24
[local]PE(config-ctx)#router bgp vpn
[local]PE(config-bgp)#address-family ipv4 unicast
[local]PE(config-bgp-af)#export route-target 100:101
[local]PE(config-bgp-af)#import route-target 100:101
[local]PE(config-bgp-af)#redistribute static
[local]PE(config-bgp-af)#redistribute connected
[local]PE(config-ctx)#ip route 192.1.1.0/24 10.10.1.2
[local]PE(config)#port ethernet 12/1
[local]PE(config-port)#bind interface 12/1 VPN1
[local]PE(config-port)#no shutdown
[local]PE(config-port)#end
```

The configuration for the CE router is as follows:

```
[local]CE#config
[local]CE(config)#context local
[local]CE(config-ctx)#interface loop1 loopback
[local]CE(config-if)#ip address 192.1.1.2/32
[local]CE(config-ctx)#interface 2/2
[local]CE(config-if)#ip address 10.10.1.2/24
[local]CE(config-ctx)#ip route 0.0.0.0/0 10.10.1.1
[local]CE(config)#port ethernet 2/2
[local]CE(config-port)#bind interface 2/2 local
[local]CE(config-port)#no shutdown
[local]CE(config-port)#end
```

VPN Using RIP

The configuration for the PE router is as follows:

```
[local]PE#config
[local]PE(config)#service multiple-context
[local]PE(config)#context VPN1 vpn-rd 1.1.1.1:101
[local]PE(config-ctx)#interface 12/1
[local]PE(config-if)#ip address 10.1.1.1/24
[local]PE(config-if)#rip router CE
[local]PE(config-ctx)#router rip CE
[local]PE(config-rip)#redistribute bgp 100
[local]PE(config-ctx)#router bgp vpn
[local]PE(config-bgp)#address-family ipv4 unicast
[local]PE(config-bgp-af)#export route-target 100:101
[local]PE(config-bgp-af)#import route-target 100:101
[local]PE(config-bgp-af)#redistribute rip CE
[local]PE(config-bgp-af)#redistribute connected
[local]PE(config)#port ethernet 12/1
[local]PE(config-port)#bind interface 12/1 VPN1
[local]PE(config-port)#no shutdown
[local]PE(config-port)#end
```

The configuration for the CE router is as follows:

```
[local]CE#config
[local]CE(config)#context local
[local]CE(config-ctx)#interface 2/2
[local]CE(config-if)#ip address 10.1.1.2/24
[local]CE(config-ctx)#router rip PE
[local]CE(config-rip)#redistribute connected
[local]CE(config)#port ethernet 2/2
[local]CE(config-port)#bind interface 2/2 local
[local]CE(config-port)#no shutdown
[local]CE(config-port)#end
```

VPN Using OSPF

The configuration for the PE router is as follows:

```
[local]PE#config
[local]PE(config)#service multiple-context
[local]PE(config)#context VPN1 vpn-rd 1.1.1.1:101
[local]PE(config-ctx)#interface 12/1
[local]PE(config-if)#ip address 10.1.1.1/24
[local]PE(config-ctx)#router ospf 1
[local]PE(config-ospf)#vpn domain-id 5.5.5.5 domain-tag 0x00000001 local-as 100
[local]PE(config-ospf)#area 0.0.0.0
[local]PE(config-ospf)#interface 12/1
[local]PE(config-ospf-interface)#cost 100
[local]PE(config-ospf)#redistribute bgp 100
[local]PE(config-ctx)#router bgp vpn
[local]PE(config-bgp)#address-family ipv4 unicast
```

```
[local]PE(config-bgp-af)#export route-target 100:101
[local]PE(config-bgp-af)#import route-target 100:101
[local]PE(config-bgp-af)#redistribute connected
[local]PE(config-bgp-af)#redistribute ospf
[local]PE(config)#port ethernet 12/1
[local]PE(config-port)#bind interface 12/1 VPN1
[local]PE(config-port)#no shutdown
[local]PE(config-port)#end
```

The configuration for the CE router is as follows:

```
[local]CE#config
[local]CE(config)#context local
[local]CE(config-ctx)#interface 2/2
[local]CE(config-if)#ip address 10.1.1.2/24
[local]CE(config-ctx)#router ospf 1
[local]CE(config-ospf)#area 0.0.0.0
[local]CE(config-ospf)#interface 2/2
[local]CE(config-ospf-interface)#cost 100
[local]CE(config)#port ethernet 2/2
[local]CE(config-port)#bind interface 2/2 local
[local]CE(config-port)#no shutdown
[local]CE(config-port)#end
```

VPN Using eBGP

The configuration for the PE router is as follows:

```
[local]PE#config
[local]PE(config)#service multiple-context
[local]PE(config)#context VPN1 vpn-rd 1.1.1.1:101
[local]PE(config-ctx)#interface 12/1
[local]PE(config-if)#ip address 10.1.1.1/24
[local]PE(config-ctx)#router bgp vpn
[local]PE(config-bgp)#address-family ipv4 unicast
[local]PE(config-bgp-af)#export route-target 100:101
[local]PE(config-bgp-af)#import route-target 100:101
[local]PE(config-bgp)#neighbor 10.1.1.2 external
[local]PE(config-bgp-neighbor)#remote-as 200
[local]PE(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE(config)#port ethernet 12/1
[local]PE(config-port)#bind interface 12/1 VPN1
[local]PE(config-port)#no shutdown
[local]PE(config-port)#end
```

The configuration for the CE router is as follows:

```
[local]CE#config
[local]CE(config)#context local
[local]CE(config-ctx)#interface 2/2
[local]CE(config-if)#ip address 10.1.1.2/24
[local]CE(config-ctx)#router bgp 200
[local]CE(config-bgp)#address-family ipv4 unicast
```

```
[local]CE(config-bgp)#neighbor 10.1.1.1 external
[local]CE(config-bgp-neighbor)#remote-as 100
[local]CE(config-bgp-neighbor)#address-family ipv4 unicast
[local]CE(config)#port ethernet 2/2
[local]CE(config-port)#bind interface 2/2 local
[local]CE(config-port)#no shutdown
[local]CE(config-port)#end
```

Different BGP/MPLS VPN Topologies

Configuration examples for different BGP/MPLS VPN topologies are provided in the following sections:

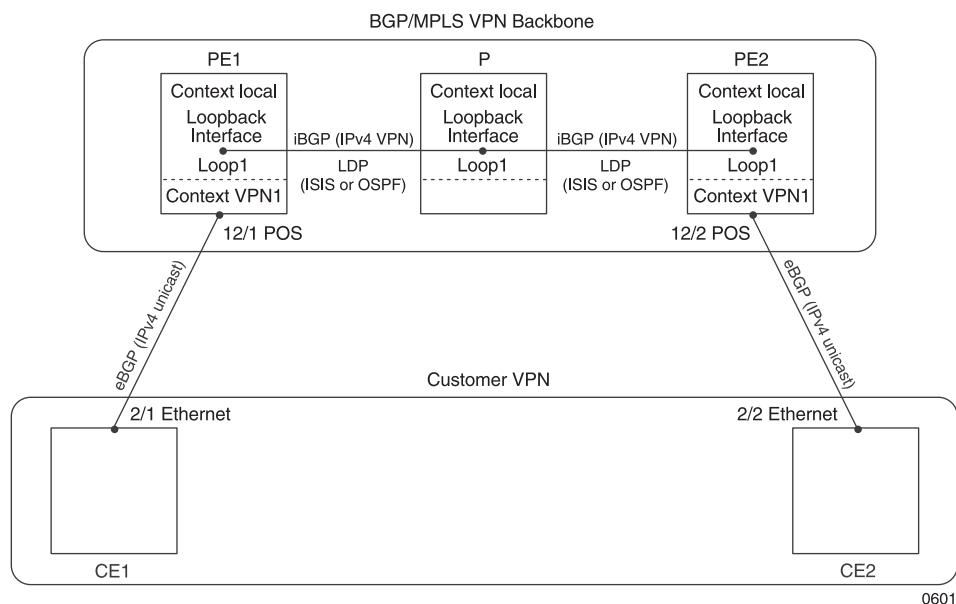
- Typical BGP/MPLS VPN
- Local Import
- Hub-and-Spoke

Note The examples shown in this section all assume eBGP is used for PE-to-CE router connectivity.

Typical BGP/MPLS VPN

The following example configures a typical BGP/MPLS VPN network configuration. Figure 9-3 shows the network topology for the configuration.

Figure 9-3 Typical BGP/MPLS VPN



The configuration for the CE1 router is as follows:

```
[local]CE1#config
[local]CE1(config)#context local
[local]CE1(config-ctx)#interface 2/2
[local]CE1(config-if)#ip address 10.1.1.2/24
[local]CE1(config-ctx)#router bgp 200
```

```
[local]CE1(config-bgp)#address-family ipv4 unicast
[local]CE1(config-bgp)#neighbor 10.1.1.1 external
[local]CE1(config-bgp-neighbor)#remote-as 100
[local]CE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]CE1(config)#port ethernet 2/2
[local]CE1(config-port)#bind interface 2/2 local
[local]CE1(config-port)#no shutdown
[local]CE1(config-port)#end
```

The configuration for the PE1 router is as follows:

```
[local]PE1#config
[local]PE1(config)#service multiple-context
[local]PE1(config)#context local
[local]PE1(config-ctx)#interface loop1 loopback
[local]PE1(config-if)#ip address 1.1.1.2/32
[local]PE1(config-if)#isis router isis-backbone
[local]PE1(config-if)#isis passive-interface
[local]PE1(config-ctx)#interface backbone1
[local]PE1(config-if)#ip address 2.2.2.1/24
[local]PE1(config-if)#isis router isis-backbone
[local]PE1(config-ctx)#router isis
[local]PE1(config-isis)#net 49.2222.0010.0100.1001.00
[local]PE1(config-ctx)#router mpls 1
[local]PE1(config-mpls)#interface backbone1
[local]PE1(config-ctx)#router ldp
[local]PE1(config-ldp)#interface backbone1
[local]PE1(config-ctx)#router bgp 100
[local]PE1(config-bgp)#address-family ipv4 vpn
[local]PE1(config-bgp-af)#redistribute connected
[local]PE1(config-bgp)#neighbor 1.1.1.1 internal
[local]PE1(config-bgp-neighbor)#update-source loop1
[local]PE1(config-bgp-neighbor)#next-hop-self
[local]PE1(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE1(config)#context VPN1 vpn-rd 1.1.1.2:100
[local]PE1(config-ctx)#interface 12/1
[local]PE1(config-if)#ip address 10.1.1.1/24
[local]PE1(config-ctx)#router bgp vpn
[local]PE1(config-bgp)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#export route-target 100:101
[local]PE1(config-bgp-af)#import route-target 100:101
[local]PE1(config-bgp-af)#redistribute connected
[local]PE1(config-bgp)#neighbor 10.1.1.2 external
[local]PE1(config-bgp-neighbor)#remote-as 200
[local]PE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE1(config)#port ethernet 12/1
[local]PE1(config-port)#bind interface 12/1 VPN1
[local]PE1(config-port)#no shutdown
[local]PE1(config)#port pos 6/1
[local]PE1(config-port)#bind interface backbone1 local
[local]PE1(config-port)#no shutdown
[local]PE1(config-port)#end
```

The configuration for the P router is as follows:

```
[local]P#config
[local]P(config)#context local
[local]P(config-ctx)#interface loop1 loopback
[local]P(config-if)#ip address 1.1.1.2/32
[local]P(config-if)#isis router isis-backbone
[local]P(config-if)#isis passive-interface
[local]P(config-ctx)#interface backbone1
[local]P(config-if)#ip address 2.2.2.2/24
[local]P(config-if)#isis router isis-backbone
[local]P(config-ctx)#router isis
[local]P(config-isis)#net 49.2222.0010.0100.1002.00
[local]P(config-ctx)#router mpls 1
[local]P(config-mpls)#interface backbone1
[local]P(config-ctx)#router ldp
[local]P(config-ldp)#interface backbone1
[local]P(config-ctx)#router bgp 100
[local]P(config-bgp)#neighbor 1.1.1.1 internal
[local]P(config-bgp-neighbor)#update-source loop1
[local]P(config-bgp-neighbor)#next-hop-self
[local]P(config-bgp-neighbor)#address-family ipv4 vpn
[local]P(config-bgp-af)#route-reflector-client
[local]P(config-bgp)#neighbor 1.1.1.3 internal
[local]P(config-bgp-neighbor)#update-source loop1
[local]P(config-bgp-neighbor)#next-hop-self
[local]P(config-bgp-neighbor)#address-family ipv4 vpn
[local]P(config-bgp-af)#route-reflector-client
[local]P(config)#port pos 6/1
[local]P(config-port)#bind interface backbone1 local
[local]P(config-port)#no shutdown
[local]P(config-port)#end
```

The configuration for the PE2 router is as follows:

```
[local]PE2#config
[local]PE2(config)#service multiple-context
[local]PE2(config)#context local
[local]PE2(config-ctx)#interface loop1 loopback
[local]PE2(config-if)#ip address 1.1.1.3/32
[local]PE2(config-if)#isis router isis-backbone
[local]PE2(config-if)#isis passive-interface
[local]PE2(config-ctx)#interface backbone1
[local]PE2(config-if)#ip address 2.2.2.3/24
[local]PE2(config-if)#isis router isis-backbone
[local]PE2(config-ctx)#router isis
[local]PE2(config-isis)#net 49.2222.0010.0100.1003.00
[local]PE2(config-ctx)#router mpls 1
[local]PE2(config-mpls)#interface backbone1
[local]PE2(config-ctx)#router ldp
[local]PE2(config-ldp)#interface backbone1
[local]PE2(config-ctx)#router bgp 100
```



```

[local]PE2(config-bgp)#neighbor 1.1.1.2 internal
[local]PE2(config-bgp-neighbor)#update-source loop1
[local]PE2(config-bgp-neighbor)#next-hop-self
[local]PE2(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE2(config)#context VPN1 vpn-rd 1.1.1.3:100
[local]PE2(config-ctx)#interface 12/2
[local]PE2(config-if)#ip address 11.1.1.1/24
[local]PE2(config-ctx)#router bgp vpn
[local]PE2(config-bgp)#address-family ipv4 unicast
[local]PE2(config-bgp-af)#export route-target 100:101
[local]PE2(config-bgp-af)#import route-target 100:101
[local]PE2(config-bgp-af)#redistribute connected
[local]PE2(config-bgp)#neighbor 11.1.1.2 external
[local]PE2(config-bgp-neighbor)#remote-as 300
[local]PE2(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE2(config)#port ethernet 12/2
[local]PE2(config-port)#bind interface 12/2 VPN1
[local]PE2(config-port)#no shutdown
[local]PE2(config)#port pos 6/1
[local]PE2(config-port)#bind interface backbone1 local
[local]PE2(config-port)#no shutdown
[local]PE2(config-port)#end

```

The configuration for the CE2 router is as follows:

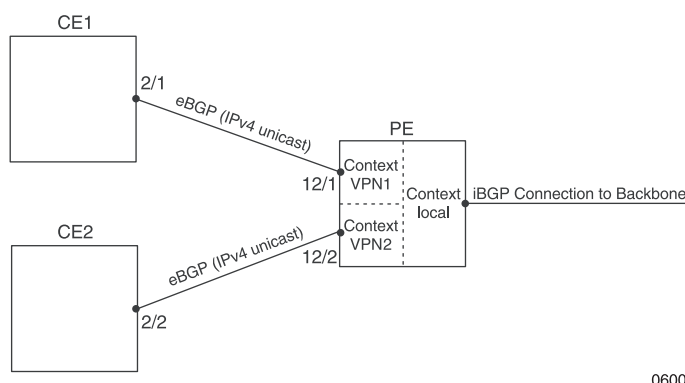
```

[local]CE2#config
[local]CE2(config)#context local
[local]CE2(config-ctx)#interface 2/2
[local]CE2(config-if)#ip address 11.1.1.2/24
[local]CE2(config-ctx)#router bgp 300
[local]CE2(config-bgp)#address-family ipv4 unicast
[local]CE2(config-bgp)#neighbor 11.1.1.2 external
[local]CE2(config-bgp-neighbor)#remote-as 100
[local]CE2(config-bgp-neighbor)#address-family ipv4 unicast
[local]CE2(config)#port ethernet 2/2
[local]CE2(config-port)#bind interface 2/2 local
[local]CE2(config-port)#no shutdown
[local]CE2(config-port)#end

```

Local Import

Two CE routers that belong to the same VPN site, and are also connected to the same PE router, are usually configured to be in the same VPN context on the PE router; however, local import can be used if the two CE routers have different import or export policies. The following example configures a local import network configuration. Figure 9-4 shows the network topology for the configuration.

Figure 9-4 Local Import Network Topology

The configuration for the CE1 router is as follows:

```
[local]CE1#config
[local]CE1(config)#context local
[local]CE1(config-ctx)#interface 2/1
[local]CE1(config-if)#ip address 10.1.1.2/24
[local]CE1(config-ctx)#router bgp 200
[local]CE1(config-bgp)#address-family ipv4 unicast
[local]CE1(config-bgp)#neighbor 10.1.1.1 external
[local]CE1(config-bgp-neighbor)#remote-as 100
[local]CE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]CE1(config)#port ethernet 2/1
[local]CE1(config-port)#bind interface 2/1 local
[local]CE1(config-port)#no shutdown
[local]CE1(config-port)#end
```

The configuration for the CE1 router is as follows:

```
[local]CE1#config
[local]CE1(config)#service multiple-context
[local]CE1(config)#context local
[local]CE1(config-ctx)#interface loop1 loopback
[local]CE1(config-if)#ip address 1.1.1.1/32
[local]CE1(config-if)#isis router isis-backbone
[local]CE1(config-if)#isis passive-interface
[local]CE1(config-ctx)#interface backbone1
[local]CE1(config-if)#ip address 2.2.2.1/24
[local]CE1(config-if)#isis router isis-backbone
[local]CE1(config-ctx)#router isis
[local]CE1(config-isis)#net 49.2222.0010.0100.1001.00
[local]CE1(config-ctx)#router mpls 1
[local]CE1(config-mpls)#interface backbone1
[local]CE1(config-ctx)#router ldp
[local]CE1(config-ldp)#interface backbone1
[local]CE1(config-ctx)#router bgp 100
[local]CE1(config-bgp)#neighbor 1.1.1.2 internal
[local]CE1(config-bgp-neighbor)#update-source loop1
[local]CE1(config-bgp-neighbor)#next-hop-self
```

```

[local]CE1(config-bgp-neighbor)#address-family ipv4 vpn
[local]CE1(config)#context VPN1 vpn-rd 1:1
[local]CE1(config-ctx)#interface 12/1
[local]CE1(config-if)#ip address 10.1.1.1/24
[local]CE1(config-ctx)#router bgp vpn
[local]CE1(config-bgp)#address-family ipv4 unicast
[local]CE1(config-bgp-af)#export route-target 100:101 100:102
[local]CE1(config-bgp-af)#import route-target 100:101 100:102
[local]CE1(config-bgp-af)#redistribute connected
[local]CE1(config-bgp)#neighbor 10.1.1.2 external
[local]CE1(config-bgp-neighbor)#remote-as 200
[local]CE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]CE1(config)#context vpn1 vpn-rd 1:1
[local]CE1(config-ctx)#interface 12/2
[local]CE1(config-if)#ip address 11.1.1.1/24
[local]CE1(config-ctx)#router bgp vpn
[local]CE1(config-bgp)#address-family ipv4 unicast
[local]CE1(config-bgp-af)#export route-target 100:101 100:103
[local]CE1(config-bgp-af)#import route-target 100:101 100:103
[local]CE1(config-bgp-af)#redistribute connected
[local]CE1(config-bgp)#neighbor 11.1.1.2 external
[local]CE1(config-bgp-neighbor)#remote-as 300
[local]CE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]CE1(config)#port ethernet 12/1
[local]CE1(config-port)#bind interface 12/1 VPN1
[local]CE1(config-port)#no shutdown
[local]CE1(config)#port ethernet 12/2
[local]CE1(config-port)#bind interface 12/2 VPN1
[local]CE1(config-port)#no shutdown
[local]CE1(config)#port pos 6/1
[local]CE1(config-port)#bind interface backbone1 local
[local]CE1(config-port)#no shutdown
[local]CE1(config-port)#end

```

The configuration for the CE2 router is as follows:

```

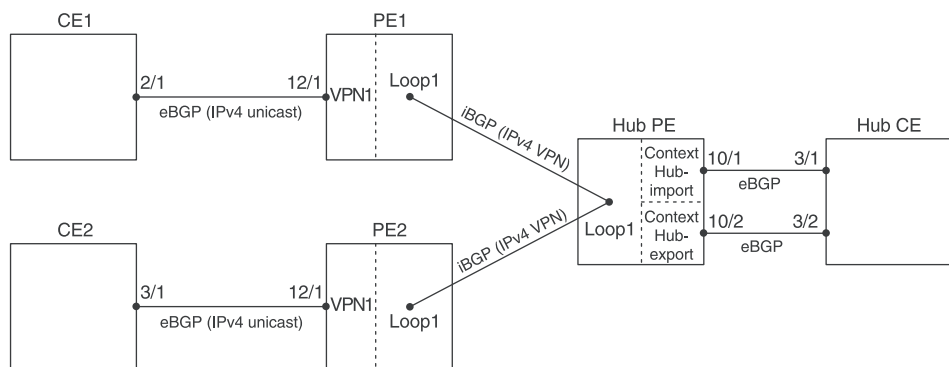
[local]CE2#config
[local]CE2(config)#context local
[local]CE2(config-ctx)#interface 2/2
[local]CE2(config-if)#ip address 11.1.1.2/24
[local]CE2(config-ctx)#router bgp 300
[local]CE2(config-bgp)#address-family ipv4 unicast
[local]CE2(config-bgp)#neighbor 11.1.1.1 external
[local]CE2(config-bgp-neighbor)#remote-as 100
[local]CE2(config-bgp-neighbor)#address-family ipv4 unicast
[local]CE2(config)#port ethernet 2/2
[local]CE2(config-port)#bind interface 2/2 local
[local]CE2(config-port)#no shutdown
[local]CE2(config-port)#end

```

Hub-and-Spoke

Hub-and-Spoke topology allows all spoke sites to send their traffic towards a central site location for various different reasons; for example, authentication. The following example configures a Hub-and-Spoke network with two spoke sites and one hub site. Figure 9-5 shows the network topology for the configuration.

Figure 9-5 Hub and Spoke Network Topology



0599

The configuration for the CE1 router is as follows:

```
[local]CE1#config
[local]CE1(config)#context local
[local]CE1(config-ctx)#interface 2/1
[local]CE1(config-if)#ip address 10.1.1.2/24
[local]CE1(config-ctx)#router bgp 200
[local]CE1(config-bgp)#address-family ipv4 unicast
[local]CE1(config-bgp)#neighbor 10.1.1.1 external
[local]CE1(config-bgp-neighbor)#remote-as 100
[local]CE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]CE1(config)#port ethernet 2/1
[local]CE1(config-port)#bind interface 2/1 local
[local]CE1(config-port)#no shutdown
[local]CE1(config-port)#end
```

The configuration for the PE1 router is as follows:

```
[local]PE1#config
[local]PE1(config)#service multiple-context
[local]PE1(config)#context local
[local]PE1(config-ctx)#interface loop1 loopback
[local]PE1(config-if)#ip address 1.1.1.1/32
[local]PE1(config-if)#isis router isis-backbone
[local]PE1(config-if)#isis passive-interface
[local]PE1(config-ctx)#interface backbone1
[local]PE1(config-if)#ip address 2.2.2.1/24
[local]PE1(config-if)#isis router isis-backbone
[local]PE1(config-ctx)#router isis
[local]PE1(config-isis)#net 49.2222.0010.0100.1001.00
[local]PE1(config-ctx)#router mpls 1
```

```

[local]PE1(config-mpls)#interface backbone1
[local]PE1(config-ctx)#router ldp
[local]PE1(config-ldp)#interface backbone1
[local]PE1(config-ctx)#router bgp 100
[local]PE1(config-bgp)#neighbor 1.1.1.2 internal
[local]PE1(config-bgp-neighbor)#update-source loop1
[local]PE1(config-bgp-neighbor)#next-hop-self
[local]PE1(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE1(config)#context VPN1 vpn-rd 1.1.1.2:101
[local]PE1(config-ctx)#interface 12/1
[local]PE1(config-if)#ip address 10.1.1.1/24
[local]PE1(config-ctx)#router bgp vpn
[local]PE1(config-bgp)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#export route-target 1:1
[local]PE1(config-bgp-af)#import route-target 2:2
[local]PE1(config-bgp-af)#redistribute connected
[local]PE1(config-bgp)#neighbor 10.1.1.2 external
[local]PE1(config-bgp-neighbor)#remote-as 200
[local]PE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE1(config)#port ethernet 12/1
[local]PE1(config-port)#bind interface 12/1 local
[local]PE1(config-port)#no shutdown
[local]PE1(config)#port pos 6/1
[local]PE1(config-port)#bind interface backbone1 local
[local]PE1(config-port)#no shutdown
[local]PE1(config-port)#end

```

The configuration for the Hub PE router is as follows:

```

[local]PE#config
[local]PE(config)#service multiple-context
[local]PE(config)#context local
[local]PE(config-ctx)#interface loop1 loopback
[local]PE(config-if)#ip address 1.1.1.1/32
[local]PE(config-if)#isis router isis-backbone
[local]PE(config-if)#isis passive-interface
[local]PE(config-ctx)#interface backbone1
[local]PE(config-if)#ip address 2.2.2.2/24
[local]PE(config-if)#isis router isis-backbone
[local]PE(config-ctx)#router isis
[local]PE(config-isis)#net 49.2222.0010.0100.1002.00
[local]PE(config-ctx)#router mpls 1
[local]PE(config-mpls)#interface backbone1
[local]PE(config-ctx)#router ldp
[local]PE(config-ldp)#interface backbone1
[local]PE(config-ctx)#router bgp 100
[local]PE(config-bgp)#address-family ipv4 unicast
[local]PE(config-bgp)#neighbor 1.1.1.2 internal
[local]PE(config-bgp-neighbor)#update-source loop1
[local]PE(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE(config-bgp)#neighbor 1.1.1.3 internal
[local]PE(config-bgp-neighbor)#update-source loop1

```

```
[local]PE(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE(config)#context HUB-import vpn-rd 1.1.1.1:1
[local]PE(config-ctx)#interface 10/1
[local]PE(config-if)#ip address 8.1.1.1/24
[local]PE(config-ctx)#router bgp vpn
[local]PE(config-bgp)#address-family ipv4 unicast
[local]PE(config-bgp-af)#import route-target 1:1
[local]PE(config-bgp-af)#redistribute connected
[local]PE(config-bgp)#neighbor 8.1.1.2 external
[local]PE(config-bgp-neighbor)#remote-as 400
[local]PE(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE(config)#context HUB-export vpn-rd 1.1.1.1:2
[local]PE(config-ctx)#interface 10/2
[local]PE(config-if)#ip address 9.1.1.1/24
[local]PE(config-ctx)#router bgp vpn
[local]PE(config-bgp)#address-family ipv4 unicast
[local]PE(config-bgp-af)#export route-target 2:2
[local]PE(config-bgp-af)#redistribute connected
[local]PE(config-bgp)#neighbor 9.1.1.2 external
[local]PE(config-bgp-neighbor)#remote-as 400
[local]PE(config-bgp-neighbor)#asloop-in 2
[local]PE(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE(config)#port ethernet 10/1
[local]PE(config-port)#bind interface 10/1 HUB-import
[local]PE(config-port)#no shutdown
[local]PE(config)#port ethernet 10/2
[local]PE(config-port)#bind interface 10/2 HUB-export
[local]PE(config-port)#no shutdown
[local]PE(config)#port pos 6/1
[local]PE(config-port)#bind interface backbone1 local
[local]PE(config-port)#no shutdown
[local]PE(config-port)#end
```

Note The Hub PE router must have two connections to the Hub CE router, one connection in the import context, and another in the export context. Additionally, the Hub PE router's exporting route target must be configured as an import route target on all spoke PE routers, and export route targets on the spoke PE routers must also be configured as import route targets on the Hub PE router. In this Hub-and-Spoke example, all spoke sites export 1:1 to the hub site, and hub site exports 2:2 to all spoke sites.

The configuration for the Hub CE router is as follows:

```
[local]CE#config
[local]CE(config)#context local
[local]CE(config-ctx)#interface 3/1
[local]CE(config-if)#ip address 8.1.1.2/24
[local]CE(config-ctx)#interface 3/2
[local]CE(config-if)#ip address 9.1.1.2/24
[local]CE(config-ctx)#router bgp 400
[local]CE(config-bgp)#address-family ipv4 unicast
[local]CE(config-bgp)#peer-group HUB-pgrp external
[local]CE(config-peer-group)#address-family ipv4 unicast
[local]CE(config-bgp)#neighbor 8.1.1.1 external
```

```
[local]CE(config-bgp-neighbor)#remote-as 100
[local]CE(config-bgp-neighbor)#address-family ipv4 unicast
[local]CE(config-bgp)#neighbor 9.1.1.1 external
[local]CE(config-bgp-neighbor)#remote-as 100
[local]CE(config-bgp)#peer-group HUB-pgrp
[local]CE(config)#port ethernet 3/1
[local]CE(config-port)#bind interface 3/1 local
[local]CE(config-port)#no shutdown
[local]CE(config)#port ethernet 3/2
[local]CE(config-port)#bind interface 3/2 local
[local]CE(config-port)#no shutdown
[local]CE(config-port)#end
```

Note A peer group must be configured for the eBGP peers on the Hub CE router to send back advertisements received from the Hub PE router. By default, routes will not be advertised back to the Hub PE router.

The configuration for the PE2 router is as follows:

```
[local]PE2#config
[local]PE2(config)#service multiple-context
[local]PE2(config)#context local
[local]PE2(config-ctx)#interface loop1 loopback
[local]PE2(config-if)#ip address 1.1.1.3/32
[local]PE2(config-if)#isis router isis-backbone
[local]PE2(config-if)#isis passive-interface
[local]PE2(config-ctx)#interface backbone1
[local]PE2(config-if)#ip address 2.2.2.3/24
[local]PE2(config-if)#isis router isis-backbone
[local]PE2(config-ctx)#router isis
[local]PE2(config-isis)#net 49.2222.0010.0100.1003.00
[local]PE2(config-ctx)#router mpls 1
[local]PE2(config-mpls)#interface backbone1
[local]PE2(config-ctx)#router ldp
[local]PE2(config-ldp)#interface backbone1
[local]PE2(config-ctx)#router bgp 100
[local]PE2(config-bgp)#neighbor 1.1.1.1 internal
[local]PE2(config-bgp-neighbor)#update-source loop1
[local]PE2(config-bgp-neighbor)#next-hop-self
[local]PE2(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE2(config)#context VPN1 vpn-rd 1.1.1.3:101
[local]PE2(config-ctx)#interface 12/1
[local]PE2(config-if)#ip address 11.1.1.1/24
[local]PE2(config-ctx)#router bgp vpn
[local]PE2(config-bgp)#address-family ipv4 unicast
[local]PE2(config-bgp-af)#export route-target 1:1
[local]PE2(config-bgp-af)#import route-target 2:2
[local]PE2(config-bgp-af)#redistributed connected
[local]PE2(config-bgp)#neighbor 11.1.1.2 external
[local]PE2(config-bgp-neighbor)#remote-as 300
[local]PE2(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE2(config)#port ethernet 12/1
[local]PE2(config-port)#bind interface 12/1 VPN1
```

```
[local]PE2(config-port)#no shutdown
[local]PE2(config)#port pos 6/1
[local]PE2(config-port)#bind interface backbone1 local
[local]PE2(config-port)#no shutdown
[local]PE2(config-port)#end
```

The configuration for the CE2 router is as follows:

```
[local]CE2#config
[local]CE2(config)#context local
[local]CE2(config-ctx)#interface 3/1
[local]CE2(config-if)#ip address 11.1.1.2/24
[local]CE2(config-ctx)#router bgp 300
[local]CE2(config-bgp)#address-family ipv4 unicast
[local]CE2(config-bgp)#neighbor 11.1.1.1 external
[local]CE2(config-bgp-neighbor)#remote-as 100
[local]CE2(config-bgp-neighbor)#address-family ipv4 unicast
[local]CE2(config)#port ethernet 3/1
[local]CE2(config-port)#bind interface 3/1 local
[local]CE2(config-port)#no shutdown
[local]CE2(config-port)#end
```

GRE over MPLS

GRE over MPLS provides a way to establish a GRE tunnel over an MPLS LSP, allowing you to run applications, such as multicast, over the GRE tunnel. The following example configures BGP/MPLS VPNs on routers PE1 and PE2. The GRE tunnel, tun1, is created over MPLS by specifying the GRE peer relationship on both ends of the tunnel, which are represented by routers PE1 and PE2. For each GRE peer relationship specified, the remote IP address must be an IP address in the remote VPN context.

The configuration for the PE1 router is as follows:

```
[local]PE1(config)#context local
[local]PE1(config-ctx)#interface lo1 loopback
[local]PE1(config-if)#ip address 2.2.2.2/32
[local]PE1(config-ctx)#interface toP
[local]PE1(config-if)#ip address 10.1.1.2/30
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#router ospf 1
[local]PE1(config-ospf)#area 0.0.0.0
[local]PE1(config-ospf-area)#interface lo1
[local]PE1(config-ospf-interface)#passive
[local]PE1(config-ospf-area)#interface toP
[local]PE1(config-ospf-area)#exit
[local]PE1(config-ospf)#exit
[local]PE1(config-ctx)#router mpls 1
[local]PE1(config-mpls)#no propagate ttl ip-to-mpls
[local]PE1(config-mpls)#exit
[local]PE1(config-ctx)#router rsvp
[local]PE1(config-rsvp)#interface toP
[local]PE1(config-rsvp-if)#lsp lsp1
[local]PE1(config-rsvp-lsp)#ingress 2.2.2.2
```



```

[local]PE1(config-rsvp-lsp)#egress 3.3.3.3
[local]PE1(config-rsvp-lsp)#exit
[local]PE1(config-rsvp-if)#exit
[local]PE1(config-rsvp)#exit
[local]PE1(config-ctx)#router bgp 100
[local]PE1(config-bgp)#neighbor 3.3.3.3 internal
[local]PE1(config-bgp-neighbor)#update-source lo1
[local]PE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE1(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE1(config-bgp-neighbor)#exit
[local]PE1(config-bgp)#exit
[local]PE1(config-ctx)#exit
[local]PE1(config)#context vpn1 vpn-rd 2.2.2.2:1
[local]PE1(config-ctx)#no ip domain-lookup
[local]PE1(config-ctx)#interface gre1
[local]PE1(config-if)#ip address 30.1.1.1/30
[local]PE1(config-ctx)#interface toCE1
[local]PE1(config-if)#ip address 100.1.1.1/24
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#router bgp vpn
[local]PE1(config-bgp)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#export route-target 100:1
[local]PE1(config-bgp-af)#import route-target 100:1
[local]PE1(config-bgp-af)#redistribute connected
[local]PE1(config-bgp-af)#exit
[local]PE1(config-bgp)#exit
[local]PE1(config-ctx)#gre-peer name tun1 remote 100.2.1.1 local 100.1.1.1
[local]PE1(config-ctx)#end

[local]PE1(config-port)#no shutdown
[local]PE1(config-port)#end

```

The configuration for the PE2 router is as follows:

```

[local]PE2(config)#context local
[local]PE2(config-ctx)#interface loop loopback
[local]PE2(config-if)#ip address 3.3.3.3/32
[local]PE2(config-ctx)#interface toP
[local]PE2(config-if)#ip address 10.1.2.2/30
[local]PE2(config-if)#exit
[local]PE2(config-ctx)#router ospf 1
[local]PE2(config-ospf)#area 0.0.0.0
[local]PE2(config-ospf-area)#interface loop
[local]PE2(config-ospf-interface)#passive
[local]PE2(config-ospf-area)#interface toP
[local]PE2(config-ospf-area)#exit
[local]PE2(config-ospf)#exit
[local]PE2(config-ctx)#router mpls 1
[local]PE2(config-mpls)#no propagate ttl ip-to-mpls
[local]PE2(config-mpls)#exit
[local]PE2(config-ctx)#router rsvp
[local]PE2(config-rsvp)#interface toP

```

```

[local]PE2(config-rsvp-if)#lsp lsp1 signaled
[local]PE2(config-rsvp-lsp)#ingress 3.3.3.3
[local]PE2(config-rsvp-lsp)#egress 2.2.2.2
[local]PE2(config-rsvp-lsp)#exit
[local]PE2(config-rsvp-if)#exit
[local]PE2(config-rsvp)#exit
[local]PE2(config-ctx)#router bgp 100
[local]PE2(config-bgp)#neighbor 2.2.2.2 internal
[local]PE2(config-bgp-neighbor)#update-source loop
[local]PE2(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE2(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE2(config-bgp-neighbor)#exit
[local]PE2(config-bgp)#exit
[local]PE2(config-ctx)#exit
[local]PE2(config)#context vpn1 vpn-rd 3.3.3.3:1
[local]PE2(config-ctx)#no ip domain-lookup
[local]PE2(config-ctx)#interface gre1
[local]PE2(config-if)#ip address 30.1.1.2/30
[local]PE2(config-ctx)#interface toCE1
[local]PE2(config-if)#ip address 100.2.1.1/24
[local]PE2(config-if)#exit
[local]PE2(config-ctx)#router bgp vpn
[local]PE2(config-bgp)#address-family ipv4 unicast
[local]PE2(config-bgp-af)#export route-target 100:1
[local]PE2(config-bgp-af)#import route-target 100:1
[local]PE2(config-bgp-af)#redistribute connected
[local]PE2(config-bgp-af)#exit
[local]PE2(config-bgp)#exit
[local]PE2(config-ctx)#gre-peer name tun1 remote 100.1.1.1 local 100.2.1.1
[local]PE2(config-gre-peer)#end

[local]PE2(config-port)#no shutdown
[local]PE2(config-port)#end

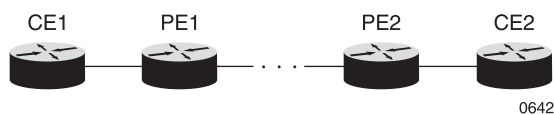
```

BGP/MPLS VPN over GRE

BGP/MPLS VPN over GRE provides a way to offer BGP/MPLS VPN service when a portion of a network does not have label switching enabled. For BGP/MPLS VPN over GRE to work, the PE routers must know how to handle GRE and label packets, and they must have MPLS enabled on the interface that receives GRE and label packets from the backbone.

Figure 9-6 shows the network topology for this BGP/MPLS VPN over GRE configuration example where both PE routes are within the same AS.

Figure 9-6 Basic BGP/MPLS VPN over GRE Network Topology



The configuration for the PE1 router is as follows:

```
[local]PE1(config)#context local
[local]PE1(config-ctx)#interface loop loopback
[local]PE1(config-if)#ip address 1.1.1.1/32
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#interface to_backbone
[local]PE1(config-if)#ip address 15.3.1.1/24
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#interface t0
[local]PE1(config-if)#ip address 50.50.51.2/24
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#router mpls 1
[local]PE1(config-mpls)#interface to_backbone
[local]PE1(config-mpls)#exit
[local]PE1(config-ctx)#router bgp 100
[local]PE1(config-bgp)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#redistribute connected
[local]PE1(config-bgp-af)#exit
[local]PE1(config-bgp)#neighbor 2.2.2.2 internal
[local]PE1(config-bgp-neighbor)#update-source loop
[local]PE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE1(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE1(config-bgp-neighbor)#exit
[local]PE1(config-bgp)#exit
[local]PE1(config-ctx)#ip soft-gre source 1.1.1.1
[local]PE1(config-ctx)#exit
[local]PE1(config)#context vpn0 vpn-rd 100:200
[local]PE1(config-ctx)#interface to_ce1
[local]PE1(config-if)#ip address 10.31.0.2/24
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#router bgp vpn
[local]PE1(config-bgp)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#export route-target 4134:4000
[local]PE1(config-bgp-af)#import route-target 4134:4000
[local]PE1(config-bgp-af)#redistribute connected
[local]PE1(config-bgp-af)#exit
[local]PE1(config-bgp)#neighbor 10.31.0.1 external
[local]PE1(config-bgp-neighbor)#remote-as 4001
[local]PE1(config-bgp-neighbor)#update-source to_ce1
[local]PE1(config-bgp-neighbor)#address-family ipv4 unicast
```

The configuration for the PE2 router is as follows:

```
[local]PE2(config)#context local
[local]PE2(config-ctx)#interface loop loopback
[local]PE2(config-if)#ip address 2.2.2.2/32
[local]PE2(config-if)#exit
[local]PE2(config-ctx)#interface to_backbone
[local]PE2(config-if)#ip address 16.3.1.1/24
[local]PE2(config-if)#exit
[local]PE2(config-ctx)#router mpls 1
[local]PE2(config-mpls)#interface to_backbone
```

```
[local]PE2(config-mpls)#exit
[local]PE2(config-ctx)#router bgp 100
[local]PE2(config-bgp)#address-family ipv4 unicast
[local]PE2(config-bgp-af)#redistribute connected
[local]PE2(config-bgp-af)#exit
[local]PE2(config-bgp)#neighbor 1.1.1.1 internal
[local]PE2(config-bgp-neighbor)#update-source loop
[local]PE2(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE2(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE2(config-bgp-neighbor)#exit
[local]PE2(config-bgp)#exit
[local]PE2(config-ctx)#ip soft-gre source 2.2.2.2
[local]PE2(config-ctx)#exit
[local]PE2(config)#context vpn0 vpn-rd 100:300
[local]PE2(config-ctx)#interface to_ce2
[local]PE2(config-if)#ip address 10.11.0.2/24
[local]PE2(config-if)#exit
[local]PE2(config-ctx)#router bgp vpn
[local]PE2(config-bgp)#address-family ipv4 unicast
[local]PE2(config-bgp-af)#export route-target 4134:4000
[local]PE2(config-bgp-af)#import route-target 4134:4000
[local]PE2(config-bgp-af)#redistribute connected
[local]PE2(config-bgp-af)#exit
[local]PE2(config-bgp)#neighbor 10.11.0.1 external
[local]PE2(config-bgp-neighbor)#remote-as 4001
[local]PE2(config-bgp-neighbor)#update-source to_ce2
[local]PE2(config-bgp-neighbor)#address-family ipv4 unicast
```

If BGP/MPLS VPN service spans multiple autonomous systems, there are two ways to exchange VPN routes between the VPN sites across the autonomous systems:

1. Configure eBGP peering between the autonomous system border routers (ASBRs), enable a VPN address family between the PE router and ASBR, and enable a VPN address family between the ASBRs. That is, within each AS, both IPv4 unicast and VPN routes are exchanged, and ASBRs are used to exchange VPN routes for interdomain routing.
2. Configure multihop eBGP peering between the PE routers, and enable VPN address family between the PE routers to exchange VPN routes. The ASBR and PE routers on the backbone exchange only IPv4 unicast routes.

For both methods, the next-hop-unchanged option must be configured on the ASBRs in the VPN address family for the peer that is peering with the other ASBR to preserve the (next-hop, label) pair.

New BGP Commands for BGP/MPLS VPN

Some BGP/MPLS VPN-related commands should only be used for specific situations. The following sections provide configuration examples that illustrate the correct use of these VPN-related commands.

- Using the asloop-in Command
- Using the as-override Command
- Using the route-origin Command

Using the asloop-in Command

The **asloop-in** command is used to disable the AS_PATH loop detection by accepting a route advertisement which contains the local AS number in AS_PATH.

This command is useful for Hub-and-Spoke network topologies where routes containing a hub PE router's ASN can be advertised to the same hub PE router as route advertisements are forwarded from one spoke to another.

This command should be configured for the hub CE neighbor in the export context on the hub PE router.

The configuration for the hub PE router is as follows:

```
[local]PE#config
[local]PE(config)#context HUB-export vpn-rd 1.1.1.1:2
[local]PE(config-ctx)#interface 10/2
[local]PE(config-if)#ip address 9.1.1.1/24
[local]PE(config-ctx)#router bgp vpn
[local]PE(config-bgp)#address-family ipv4 unicast
[local]PE(config-bgp-af)#export route-target 2:2
[local]PE(config-bgp)#neighbor 9.1.1.2 external
[local]PE(config-bgp-neighbor)#remote-as 400
[local]PE(config-bgp-neighbor)#asloop-in 2
[local]PE(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE(config)#port ethernet 10/2
[local]PE(config-port)#bind interface 10/2 HUB-export
[local]PE(config-port)#no shutdown
[local]PE(config-port)#end
```

Using the as-override Command

The **as-override** command is used to replace all occurrences of the peer's ASN in the AS_PATH attribute with the local ASN when advertising the route to the peer.

Assuming that both VPN sites for the CE1 and CE2 routers use the ASN 200, the **as-override** command must be configured for the CE peers on the PE routers before the route advertisements can be accepted by the CE routers at both sites.

Note Backbone connectivity in the local context is not shown in the following example.

The configuration for the CE1 router is as follows:

```
[local]CE1#config
[local]CE1(config)#context local
[local]CE1(config-ctx)#interface 2/1
[local]CE1(config-if)#ip address 10.1.1.2/24
[local]CE1(config-ctx)#router bgp 200
[local]CE1(config-bgp)#address-family ipv4 unicast
[local]CE1(config-bgp)#neighbor 10.1.1.1 external
[local]CE1(config-neighbor)#remote-as 100
[local]CE1(config-neighbor)#address-family ipv4 unicast
[local]CE1(config)#port ethernet 2/1
[local]CE1(config-port)#bind interface 2/1 local
[local]CE1(config-port)#no shutdown
[local]CE1(config-port)#end
```

The configuration for the PE1 router is as follows:

```
[local]PE1#config
[local]PE1(config)#service multiple-context
[local]PE1(config)#context VPN1 vpn-rd 1.1.1.2:101
[local]PE1(config-ctx)#interface 12/1
[local]PE1(config-if)#ip address 10.1.1.1/24
[local]PE1(config-ctx)#router bgp vpn
[local]PE1(config-bgp)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#export route-target 1:1
[local]PE1(config-bgp-af)#import route-target 2:2
[local]PE1(config-bgp)#neighbor 10.1.1.2 external
[local]PE1(config-bgp-neighbor)#remote-as 200
[local]PE1(config-bgp-neighbor)#as-override
[local]PE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE1(config)#port ethernet 12/1
[local]PE1(config-port)#bind interface 12/1 VPN1
[local]PE1(config-port)#no shutdown
[local]PE1(config-port)#end
```

The configuration for the PE2 router is as follows:

```
[local]PE2#config
[local]PE2(config)#service multiple-context
[local]PE2(config)#context local
[local]PE2(config-ctx)#interface loop1 loopback
[local]PE2(config-if)#ip address 1.1.1.3/32
[local]PE2(config-ctx)#router bgp 100
[local]PE2(config-bgp)#neighbor 1.1.1.1 internal
[local]PE2(config-bgp-neighbor)#update-source loop1
[local]PE2(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE2(config)#context VPN1 vpn-rd 1.1.1.3:101
[local]PE2(config-ctx)#interface 12/1
[local]PE2(config-if)#ip address 11.1.1.1/24
[local]PE2(config-ctx)#router bgp vpn
[local]PE2(config-bgp)#address-family ipv4 unicast
[local]PE2(config-bgp-af)#export route-target 1:1
[local]PE2(config-bgp-af)#import route-target 2:2
[local]PE2(config-bgp)#neighbor 11.1.1.2 external
[local]PE2(config-bgp-neighbor)#remote-as 200
[local]PE2(config-bgp-neighbor)#as-override
[local]PE2(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE2(config)#port ethernet 12/1
[local]PE2(config-port)#bind interface 12/1 VPN1
[local]PE2(config-port)#no shutdown
[local]PE2(config-port)#end
```

The configuration for the CE2 router is as follows:

```
[local]CE2#config
[local]CE2(config)#context local
[local]CE2(config-ctx)#interface 3/1
```

```
[local]CE2(config-if)#ip address 11.1.1.2/24
[local]CE2(config-ctx)#router bgp 200
[local]CE2(config-bgp)#address-family ipv4 unicast
[local]CE2(config-bgp)#neighbor 11.1.1.1 external
[local]CE2(config-bgp-neighbor)#remote-as 100
[local]CE2(config-bgp-neighbor)#address-family ipv4 unicast
[local]CE2(config)#port ethernet 3/1
[local]CE2(config-port)#bind interface 3/1 local
[local]CE2(config-port)#no shutdown
[local]CE2(config-port)#end
```

Using the route-origin Command

In the case of multiple sites sharing the same ASN, using an ASN alone is no longer adequate for AS loop detection. To prevent the readvertisement of routes back to its originating site, use the **route-origin** command to identify the site from where the routes originated.

The configuration for the PE1 router is as follows:

```
[local]PE1#config
[local]PE1(config)#context VPN1 vpn-rd 1.1.1.2:101
[local]PE1(config-ctx)#router bgp vpn
[local]PE1(config-bgp)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#route-origin 100:300
[local]PE1(config-bgp-af)#export route-target 1:1
[local]PE1(config-bgp-af)#import route-target 2:2
[local]PE1(config-bgp-af)#redistribute connected
[local]PE1(config-bgp)#neighbor 10.1.1.2 external
[local]PE1(config-bgp-neighbor)#remote-as 200
[local]PE1(config-bgp-neighbor)#as-override
[local]PE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#end
```

The configuration for the PE2 router is as follows:

```
[local]PE2#config
[local]PE2(config)#context VPN1 vpn-rd 1.1.1.3:101
[local]PE2(config-ctx)#router bgp vpn
[local]PE2(config-bgp)#address-family ipv4 unicast
[local]PE2(config-bgp-af)#route-origin 100:400
[local]PE2(config-bgp-af)#export route-target 1:1
[local]PE2(config-bgp-af)#import route-target 2:2
[local]PE2(config-bgp-af)#redistribute connected
[local]PE2(config-bgp)#neighbor 11.1.1.2 external
[local]PE2(config-bgp-neighbor)#remote-as 200
[local]PE2(config-bgp-neighbor)#as-override
[local]PE2(config-bgp-neighbor)#address-family ipv4 unicast
```

CoC

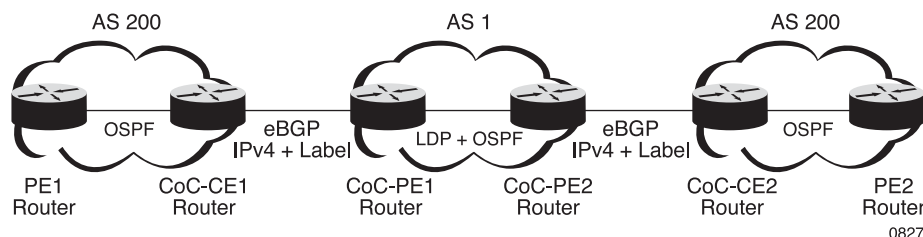
CoC provides a way for a service provider to use a segment of another service provider's backbone network to transport traffic between two geographically separated networks. The service provider that uses CoC to connect its two networks is called the customer carrier, and the service provider that provides a segment of its backbone network is called the backbone carrier.

The BGP/MPLS VPN implementation of the CoC feature uses eBGP to distribute MPLS labels in IPv4 unicast routes between customer carrier CE routers and backbone carrier PE routers. The backbone carrier uses MPLS to route traffic across its backbone network. The customer carrier can use either IP or MPLS routing in its networks.

Figure 9-7 shows the network topology for this BGP/MPLS VPN CoC configuration example, where:

- The customer carrier CE routers (CoC-CE1 and CoC-CE2) are eBGP-peered to the backbone carrier PE routers (CoC-PE1 and CoC-PE2).
- OSPF is enabled in the customer carrier networks.
- LDP and OSPF are enabled in the backbone carrier network.
- The ASN for both customer carrier networks is 200.
- The customer carrier networks only provide IP services.

Figure 9-7 BGP/MPLS VPN Carrier of Carriers Network Topology



The configuration for the PE1 router is as follows:

```
[local]PE1#config
[local]PE1(config)#service multiple-contexts
[local]PE1(config)#context local
[local]PE1(config-ctx)#no ip domain-lookup
[local]PE1(config-ctx)#interface lo1 loopback
[local]PE1(config-if)#ip address 5.5.5.5/32
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#interface to-CoC-CE1
[local]PE1(config-if)#ip address 50.1.1.1/24
[local]PE1(config-if)#exit
[local]PE1(config)#router ospf 1
[local]PE1(config-ospf)#area 0.0.0.0
[local]PE1(config-ospf-area)#interface lo1
[local]PE1(config-ospf-if)#exit
[local]PE1(config-ospf-area)#interface to-CoC-CE1
[local]PE1(config-ospf-if)#exit
[local]PE1(config-ospf-area)#exit
[local]PE1(config-ospf)#exit
```



```

[local]PE1(config-ctx)#router bgp 200
[local]PE1(config-bgp)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#exit
[local]PE1(config-bgp)#peer-group ibgp-peers internal
[local]PE1(config-bgp-peer-group)#advertisement-interval 1
[local]PE1(config-bgp-peer-group)#update-source lo1
[local]PE1(config-bgp-peer-group)#next-hop-self
[local]PE1(config-bgp-peer-group)#address-family ipv4 unicast
[local]PE1(config-bgp-peer-af)#exit
[local]PE1(config-bgp-peer-group)#exit
[local]PE1(config-bgp)#neighbor 3.3.3.3 internal
[local]PE1(config-bgp-neighbor)#peer-group ibgp-peers
[local]PE1(config-bgp-neighbor)#exit
[local]PE1(config-bgp)#neighbor 4.4.4.4 internal
[local]PE1(config-bgp-neighbor)#peer-group ibgp-peers
[local]PE1(config-bgp-neighbor)#exit
[local]PE1(config-bgp)#neighbor 6.6.6.6 internal
[local]PE1(config-bgp-neighbor)#peer-group ibgp-peers
[local]PE1(config-bgp-neighbor)#exit
[local]PE1(config-bgp)#exit
[local]PE1(config-ctx)#exit
[local]PE1(config)#card ether-12-port 3
[local]PE1(config)#port ethernet 3/10
[local]PE1(config-port)#no shutdown
[local]PE1(config-port)#bind interface to-CoC-CE1 local
[local]PE1(config-port)#end

```

The configuration for the CoC-CE1 router is as follows:

```

[local]CoC-CE1#config
[local]CoC-CE1(config)#context local
[local]CoC-CE1(config-ctx)#no ip domain-lookup
[local]CoC-CE1(config-ctx)#interface lo1 loopback
[local]CoC-CE1(config-if)#ip address 3.3.3.3/32
[local]CoC-CE1(config-if)#exit
[local]CoC-CE1(config-ctx)#interface to-CoC-PE1
[local]CoC-CE1(config-if)#ip address 20.1.1.2/24
[local]CoC-CE1(config-if)#exit
[local]CoC-CE1(config-ctx)#interface to-PE1
[local]CoC-CE1(config-if)#ip address 50.1.1.2/24
[local]CoC-CE1(config-if)#exit
[local]CoC-CE1(config-ctx)#router ospf 1
[local]CoC-CE1(config-ospf)#area 0.0.0.0
[local]CoC-CE1(config-ospf-area)#interface lo1
[local]CoC-CE1(config-ospf-if)#exit
[local]CoC-CE1(config-ospf-area)#interface to-PE1
[local]CoC-CE1(config-ospf-if)#exit
[local]CoC-CE1(config-ospf-area)#exit
[local]CoC-CE1(config-ospf)#redistribute bgp 200 route-map redist-to-ospf
[local]CoC-CE1(config-ospf)#exit
[local]CoC-CE1(config-ctx)#ip prefix-list backbone-addr
[local]CoC-CE1(config-prefix-list)#seq 10 permit 20.1.1.1/32

```

```
[local]CoC-CE1(config-prefix-list)#exit
[local]CoC-CE1(config-ctx)#as-path-list zero-aspath-list
[local]CoC-CE1(config-as-path-list)#seq 10 permit ^$
[local]CoC-CE1(config-as-path-list)#exit
[local]CoC-CE1(config-ctx)#community-list permit-111
[local]CoC-CE1(config-community-list)#seq 10 permit 200:111
[local]CoC-CE1(config-community-list)#exit
[local]CoC-CE1(config-ctx)#route-map from-backbone-only permit 10
[local]CoC-CE1(config-route-map)#set community no-advertise
[local]CoC-CE1(config-route-map)#exit
[local]CoC-CE1(config-ctx)#route-map redistrib-to-bgp permit 10
[local]CoC-CE1(config-route-map)#set community 200:111
[local]CoC-CE1(config-route-map)#exit
[local]CoC-CE1(config-ctx)#route-map redistrib-to-ospf permit 10
[local]CoC-CE1(config-route-map)#match ip next-hop prefix-list backbone-addr
[local]CoC-CE1(config-route-map)#match route-type external
[local]CoC-CE1(config-route-map)#exit
[local]CoC-CE1(config-ctx)#route-map to-backbone-only permit 10
[local]CoC-CE1(config-route-map)#match as-path-list zero-aspath-list
[local]CoC-CE1(config-route-map)#match community-list permit-111
[local]CoC-CE1(config-route-map)#exit
[local]CoC-CE1(config-ctx)#route-map to-ibgp-peers deny 10
[local]CoC-CE1(config-route-map)#match as-path-list zero-aspath-list
[local]CoC-CE1(config-route-map)#match community-list permit-111
[local]CoC-CE1(config-route-map)#exit
[local]CoC-CE1(config-ctx)#route-map to-ibgp-peers permit 20
[local]CoC-CE1(config-route-map)#exit
[local]CoC-CE1(config-ctx)#router mpls 1
[local]CoC-CE1(config-mpls)#interface to-CoC-PE1
[local]CoC-CE1(config-mpls-if)#exit
[local]CoC-CE1(config-mpls)#exit
[local]CoC-CE1(config-ctx)#router bgp 200
[local]CoC-CE1(config-bgp)#address-family ipv4 unicast
[local]CoC-CE1(config-bgp-af)#redistribute connected route-map redistrib-to-bgp
[local]CoC-CE1(config-bgp-af)#redistribute ospf 1 route-map redistrib-to-bgp
[local]CoC-CE1(config-bgp-af)#exit
[local]CoC-CE1(config-bgp)#peer-group ibgp-peers internal
[local]CoC-CE1(config-bgp-peer-group)#update-source lo1
[local]CoC-CE1(config-bgp-peer-group)#address-family ipv4 unicast
[local]CoC-CE1(config-bgp-peer-af)#route-map to-ibgp-peers out
[local]CoC-CE1(config-bgp-peer-af)#exit
[local]CoC-CE1(config-bgp-peer-group)#exit
[local]CoC-CE1(config-bgp)#neighbor 4.4.4.4 internal
[local]CoC-CE1(config-bgp-neighbor)#peer-group ibgp-peers
[local]CoC-CE1(config-bgp-neighbor)#exit
[local]CoC-CE1(config-bgp)#neighbor 5.5.5.5 internal
[local]CoC-CE1(config-bgp-neighbor)#peer-group ibgp-peers
[local]CoC-CE1(config-bgp-neighbor)#exit
[local]CoC-CE1(config-bgp)#neighbor 6.6.6.6 internal
[local]CoC-CE1(config-bgp-neighbor)#peer-group ibgp-peers
[local]CoC-CE1(config-bgp-neighbor)#exit
[local]CoC-CE1(config-bgp)#neighbor 20.1.1.1 external
```

```

[local]CoC-CE1(config-bgp-neighbor)#remote-as 1
[local]CoC-CE1(config-bgp-neighbor)#advertisement-interval 1
[local]CoC-CE1(config-bgp-neighbor)#send label
[local]CoC-CE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]CoC-CE1(config-bgp-af)#route-map from-backbone-only in
[local]CoC-CE1(config-bgp-af)#route-map to-backbone-only out
[local]CoC-CE1(config-bgp-af)#exit
[local]CoC-CE1(config-bgp-neighbor)#exit
[local]CoC-CE1(config-bgp)#exit
[local]CoC-CE1(config-ctx)#exit
[local]CoC-CE1(config)#card ether-12-port 2
[local]CoC-CE1(config)#port ethernet 2/4
[local]CoC-CE1(config-port)#no shutdown
[local]CoC-CE1(config-port)#bind interface to-CoC-PE1 local
[local]CoC-CE1(config-port)#exit
[local]CoC-CE1(config)#port ethernet 2/10
[local]CoC-CE1(config-port)#no shutdown
[local]CoC-CE1(config-port)#bind interface to-PE1 local
[local]CoC-CE1(config-port)#end

```

The configuration for the CoC-PE1 router is as follows:

```

[local]CoC-PE1#config
[local]CoC-PE1(config)#service multiple-contexts
[local]CoC-PE1(config)#context local
[local]CoC-PE1(config-ctx)#no ip domain-lookup
[local]CoC-PE1(config-ctx)#interface lo1 loopback
[local]CoC-PE1(config-if)#ip address 1.1.1.1/32
[local]CoC-PE1(config-if)#exit
[local]CoC-PE1(config-ctx)#interface to-CoC-PE2
[local]CoC-PE1(config-if)#ip address 193.4.4.1/16
[local]CoC-PE1(config-if)#exit
[local]CoC-PE1(config-ctx)#router ospf 1
[local]CoC-PE1(config-ospf)#area 0.0.0.0
[local]CoC-PE1(config-ospf-area)#interface lo1
[local]CoC-PE1(config-ospf-if)#exit
[local]CoC-PE1(config-ospf)#interface to-CoC-PE2
[local]CoC-PE1(config-ospf-if)#exit
[local]CoC-PE1(config-ospf)#exit
[local]CoC-PE1(config-ctx)#router mpls 1
[local]CoC-PE1(config-mpls)#interface to-CoC-PE2
[local]CoC-PE1(config-mpls-if)#exit
[local]CoC-PE1(config-mpls)#exit
[local]CoC-PE1(config-ctx)#router ldp
[local]CoC-PE1(config-ldp)#interface to-CoC-PE2
[local]CoC-PE1(config-ldp)#exit
[local]CoC-PE1(config-ctx)#router bgp 1
[local]CoC-PE1(config-bgp)#address-family ipv4 unicast
[local]CoC-PE1(config-bgp-af)#exit
[local]CoC-PE1(config-bgp)#address-family ipv4 vpn
[local]CoC-PE1(config-bgp-af)#no route-target filter
[local]CoC-PE1(config-bgp-af)#exit

```

```
[local]CoC-PE1(config-bgp)#neighbor 2.2.2.2 internal
[local]CoC-PE1(config-bgp-neighbor)#advertisement-interval 1
[local]CoC-PE1(config-bgp-neighbor)#update-source lo1
[local]CoC-PE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]CoC-PE1(config-bgp-af)#exit
[local]CoC-PE1(config-bgp-neighbor)#address-family ipv4 vpn
[local]CoC-PE1(config-bgp-af)#exit
[local]CoC-PE1(config-bgp-neighbor)#exit
[local]CoC-PE1(config-bgp)#exit
[local]CoC-PE1(config-ctx)#exit
[local]CoC-PE1(config)#context vpn1 vpn-rd 2:2
[local]CoC-PE1(config-ctx)#no ip domain-lookup
[local]CoC-PE1(config-ctx)#interface to-CoC-CE1
[local]CoC-PE1(config-if)#ip address 20.1.1.1/16
[local]CoC-PE1(config-if)#exit
[local]CoC-PE1(config-ctx)#router mpls 1
[local]CoC-PE1(config-mpls)#interface to-CoC-CE1
[local]CoC-PE1(config-mpls-if)#label-space context-name local
[local]CoC-PE1(config-mpls-if)#exit
[local]CoC-PE1(config-mpls)#exit
[local]CoC-PE1(config-ctx)#router bgp vpn
[local]CoC-PE1(config-bgp)#address-family ipv4 unicast
[local]CoC-PE1(config-bgp-af)#export route-target 2:2
[local]CoC-PE1(config-bgp-af)#import route-target 2:2
[local]CoC-PE1(config-bgp-af)#exit
[local]CoC-PE1(config-bgp)#neighbor 20.1.1.2 external
[local]CoC-PE1(config-bgp-neighbor)#remote-as 200
[local]CoC-PE1(config-bgp-neighbor)#advertisement-interval 1
[local]CoC-PE1(config-bgp-neighbor)#as-override
[local]CoC-PE1(config-bgp-neighbor)#send label
[local]CoC-PE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]CoC-PE1(config-bgp-af)#exit
[local]CoC-PE1(config-bgp-neighbor)#exit
[local]CoC-PE1(config-bgp)#exit
[local]CoC-PE1(config-ctx)#exit
[local]CoC-PE1(config)#card ether-12-port 3
[local]CoC-PE1(config)#port ethernet 3/1
[local]CoC-PE1(config-port)#no shutdown
[local]CoC-PE1(config-port)#bind interface to-CoC-PE2 local
[local]CoC-PE1(config-port)#exit
[local]CoC-PE1(config)#port ethernet 3/2
[local]CoC-PE1(config-port)#no shutdown
[local]CoC-PE1(config-port)#bind interface to-CoC-CE1 vpn1
[local]CoC-PE1(config-port)#end
```

The configuration for the CoC-PE2 router is as follows:

```
[local]CoC-PE2#config
[local]CoC-PE2(config)#service multiple-contexts
[local]CoC-PE2(config)#context local
[local]CoC-PE2(config-ctx)#no ip domain-lookup
[local]CoC-PE2(config-ctx)#interface lo1 loopback
```

```
[local]CoC-PE2(config-if)#ip address 2.2.2.2/32
[local]CoC-PE2(config-if)#exit
[local]CoC-PE2(config-ctx)#interface to-CoC-PE1
[local]CoC-PE2(config-if)#ip address 193.4.5.2/16
[local]CoC-PE2(config-if)#exit
[local]CoC-PE2(config-ctx)#router ospf 1
[local]CoC-PE2(config-ospf)#area 0.0.0.0
[local]CoC-PE2(config-ospf-area)#interface lo1
[local]CoC-PE2(config-ospf-if)#exit
[local]CoC-PE2(config-ospf-area)#interface to-CoC-PE1
[local]CoC-PE2(config-ospf-if)#exit
[local]CoC-PE2(config-ospf-area)#exit
[local]CoC-PE2(config-ospf)#exit
[local]CoC-PE2(config-ctx)#router mpls 1
[local]CoC-PE2(config-mpls)#interface to-CoC-PE1
[local]CoC-PE2(config-mpls-if)#exit
[local]CoC-PE2(config-mpls)#exit
[local]CoC-PE2(config-ctx)#router ldp
[local]CoC-PE2(config-ldp)#interface to-CoC-PE1
[local]CoC-PE2(config-ldp)#exit
[local]CoC-PE2(config-ctx)#router bgp 1
[local]CoC-PE2(config-bgp)#address-family ipv4 unicast
[local]CoC-PE2(config-bgp-af)#exit
[local]CoC-PE2(config-bgp)#address-family ipv4 vpn
[local]CoC-PE2(config-bgp-af)#no route-target filter
[local]CoC-PE2(config-bgp-af)#exit
[local]CoC-PE2(config-bgp)#neighbor 1.1.1.1 internal
[local]CoC-PE2(config-bgp-neighbor)#advertisement-interval 1
[local]CoC-PE2(config-bgp-neighbor)#update-source lo1
[local]CoC-PE2(config-bgp-neighbor)#address-family ipv4 unicast
[local]CoC-PE2(config-bgp-af)#exit
[local]CoC-PE2(config-bgp-neighbor)#address-family ipv4 vpn
[local]CoC-PE2(config-bgp-af)#exit
[local]CoC-PE2(config-bgp-neighbor)#exit
[local]CoC-PE2(config-bgp)#exit
[local]CoC-PE2(config-ctx)#exit
[local]CoC-PE2(config)#context vpn1 vpn-rd 2:2
[local]CoC-PE2(config-ctx)#no ip domain-lookup
[local]CoC-PE2(config-ctx)#interface to-CoC-CE2
[local]CoC-PE2(config-if)#ip address 30.1.1.1/24
[local]CoC-PE2(config-if)#exit
[local]CoC-PE2(config-ctx)#router mpls 1
[local]CoC-PE2(config-mpls)#interface to-CoC-CE2
[local]CoC-PE2(config-mpls-if)#label-space context-name local
[local]CoC-PE2(config-mpls-if)#exit
[local]CoC-PE2(config-mpls)#exit
[local]CoC-PE2(config-ctx)#router bgp vpn
[local]CoC-PE2(config-bgp)#address-family ipv4 unicast
[local]CoC-PE2(config-bgp-af)#export route-target 2:2
[local]CoC-PE2(config-bgp-af)#import route-target 2:2
[local]CoC-PE2(config-bgp-af)#exit
[local]CoC-PE2(config-bgp)#neighbor 30.1.1.2 external
```

```
[local]CoC-PE2(config-bgp-neighbor)#remote-as 200
[local]CoC-PE2(config-bgp-neighbor)#advertisement-interval 1
[local]CoC-PE2(config-bgp-neighbor)#as-override
[local]CoC-PE2(config-bgp-neighbor)#send label
[local]CoC-PE2(config-bgp-neighbor)#address-family ipv4 unicast
[local]CoC-PE2(config-bgp-af)#exit
[local]CoC-PE2(config-bgp-neighbor)#exit
[local]CoC-PE2(config-bgp)#exit
[local]CoC-PE2(config-ctx)#exit
[local]CoC-PE2(config)#card ether-12-port 2
[local]CoC-PE2(config)#port ethernet 2/2
[local]CoC-PE2(config-port)#no shutdown
[local]CoC-PE2(config-port)#bind interface to-CoC-CE2 vpn1
[local]CoC-PE2(config-port)#exit
[local]CoC-PE2(config)#port ethernet 2/6
[local]CoC-PE2(config-port)#no shutdown
[local]CoC-PE2(config-port)#bind interface to-CoC-PE1 local
[local]CoC-PE2(config-port)#end
```

The configuration for the CoC-CE2 router is as follows:

```
[local]CoC-CE2#config
[local]CoC-CE2(config)#service multiple-contexts
[local]CoC-CE2(config)#context local
[local]CoC-CE2(config-ctx)#interface lo1 loopback
[local]CoC-CE2(config-if)#ip address 4.4.4.4/32
[local]CoC-CE2(config-if)#exit
[local]CoC-CE2(config-ctx)#interface to-CoC-PE2
[local]CoC-CE2(config-if)#ip address 30.1.1.2/24
[local]CoC-CE2(config-if)#exit
[local]CoC-CE2(config-ctx)#interface to-PE2
[local]CoC-CE2(config-if)#ip address 60.1.1.1/24
[local]CoC-CE2(config-if)#exit
[local]CoC-CE2(config-ctx)#router ospf 1
[local]CoC-CE2(config-ospf)#area 0.0.0.0
[local]CoC-CE2(config-ospf-area)#interface lo1
[local]CoC-CE2(config-ospf-if)#exit
[local]CoC-CE2(config-ospf-area)#interface to-PE2
[local]CoC-CE2(config-ospf-if)#exit
[local]CoC-CE2(config-ospf-area)#redistribute bgp 200 route-map redist-to-ospf
[local]CoC-CE2(config-ospf-area)#exit
[local]CoC-CE2(config-ospf)#exit
[local]CoC-CE2(config-ctx)#ip prefix-list backbone-addr
[local]CoC-CE2(config-prefix-list)#seq 10 permit 30.1.1.1/32
[local]CoC-CE2(config-prefix-list)#exit
[local]CoC-CE2(config-ctx)#as-path-list zero-aspath-list
[local]CoC-CE2(config-as-path-list)#seq 10 permit ^$
[local]CoC-CE2(config-as-path-list)#exit
[local]CoC-CE2(config-ctx)#community-list permit-111
[local]CoC-CE2(config-community-list)#seq 10 permit 200:111
[local]CoC-CE2(config-community-list)#exit
[local]CoC-CE2(config-ctx)#route-map from-backbone-only permit 10
```

```

[local]CoC-CE2(config-route-map)#set community no-advertise
[local]CoC-CE2(config-route-map)#exit
[local]CoC-CE2(config-ctx)#route-map redistrib-to-bgp permit 10
[local]CoC-CE2(config-route-map)#set community 200:111
[local]CoC-CE2(config-route-map)#exit
[local]CoC-CE2(config-ctx)#route-map redistrib-to-ospf permit 10
[local]CoC-CE2(config-route-map)#match ip next-hop prefix-list backbone-addr
[local]CoC-CE2(config-route-map)#match route-type external
[local]CoC-CE2(config-route-map)#exit
[local]CoC-CE2(config-ctx)#route-map to-backbone-only permit 10
[local]CoC-CE2(config-route-map)#match as-path-list zero-aspath-list
[local]CoC-CE2(config-route-map)#match community-list permit-111
[local]CoC-CE2(config-route-map)#exit
[local]CoC-CE2(config-ctx)#route-map to-ibgp-peers deny 10
[local]CoC-CE2(config-route-map)#match as-path-list zero-aspath-list
[local]CoC-CE2(config-route-map)#match community-list permit-111
[local]CoC-CE2(config-route-map)#exit
[local]CoC-CE2(config-ctx)#route-map to-ibgp-peers permit 20
[local]CoC-CE2(config-route-map)#exit
[local]CoC-CE2(config-ctx)#router mpls 1
[local]CoC-CE2(config-mpls)#interface to-CoC-PE2
[local]CoC-CE2(config-mpls-if)#exit
[local]CoC-CE2(config-mpls)#exit
[local]CoC-CE2(config-ctx)#router bgp 200
[local]CoC-CE2(config-bgp)#address-family ipv4 unicast
[local]CoC-CE2(config-bgp-af)#redistribute connected route-map redistrib-to-bgp
[local]CoC-CE2(config-bgp-af)#redistribute static
[local]CoC-CE2(config-bgp-af)#redistribute ospf 1 route-map redistrib-to-bgp
[local]CoC-CE2(config-bgp-af)#exit
[local]CoC-CE2(config-bgp)#peer-group ibgp-peers internal
[local]CoC-CE2(config-bgp-peer-group)#update-source lo1
[local]CoC-CE2(config-bgp-peer-group)#address-family upv4 unicast
[local]CoC-CE2(config-bgp-peer-af)#route-map to-ibgp-peers out
[local]CoC-CE2(config-bgp-peer-af)#exit
[local]CoC-CE2(config-bgp-peer-group)#exit
[local]CoC-CE2(config-bgp)#neighbor 3.3.3.3 internal
[local]CoC-CE2(config-bgp-neighbor)#peer-group ibgp-peers
[local]CoC-CE2(config-bgp-neighbor)#exit
[local]CoC-CE2(config-bgp)#neighbor 5.5.5.5 internal
[local]CoC-CE2(config-bgp-neighbor)#peer-group ibgp-peers
[local]CoC-CE2(config-bgp-neighbor)#exit
[local]CoC-CE2(config-bgp)#neighbor 6.6.6.6 internal
[local]CoC-CE2(config-bgp-neighbor)#peer-group ibgp-peers
[local]CoC-CE2(config-bgp-neighbor)#exit
[local]CoC-CE2(config-bgp)#neighbor 30.1.1.1 external
[local]CoC-CE2(config-bgp-neighbor)#remote-as 1
[local]CoC-CE2(config-bgp-neighbor)#description EBGp to CoC-PE2
[local]CoC-CE2(config-bgp-neighbor)#advertisement-interval 1
[local]CoC-CE2(config-bgp-neighbor)#send label
[local]CoC-CE2(config-bgp-neighbor)#address-family ipv4 unicast
[local]CoC-CE2(config-bgp-af)#route-map from-backbone-only in
[local]CoC-CE2(config-bgp-af)#route-map to-backbone-only out

```

```
[local]CoC-CE2(config-bgp-af)#exit
[local]CoC-CE2(config-bgp-neighbor)#exit
[local]CoC-CE2(config-bgp)#exit
[local]CoC-CE2(config-ctx)#exit
[local]CoC-CE2(config)#card ether-12-port 2
[local]CoC-CE2(config)#port ethernet 2/1
[local]CoC-CE2(config-port)#no shutdown
[local]CoC-CE2(config-port)#bind interface to-CoC-PE2 local
[local]CoC-CE2(config-port)#exit
[local]CoC-CE2(config)#port ethernet 2/7
[local]CoC-CE2(config-port)#no shutdown
[local]CoC-CE2(config-port)#bind interface to-PE2 local
[local]CoC-CE2(config-port)#end
```

The configuration for the PE2 router is as follows:

```
[local]PE2#config
[local]PE2(config)#service multiple-contexts
[local]PE2(config)#context local
[local]PE2(config-ctx)#no ip domain-lookup
[local]PE2(config-ctx)#interface lo1 loopback
[local]PE2(config-if)#ip address 6.6.6.6/32
[local]PE2(config-if)#exit
[local]PE2(config-ctx)#interface to-CoC-CE2
[local]PE2(config-if)#ip address 60.1.1.2/24
[local]PE2(config-if)#exit
[local]PE2(config)#router ospf 1
[local]PE2(config-ospf)#area 0.0.0.0
[local]PE2(config-ospf-area)#interface lo1
[local]PE2(config-ospf-if)#exit
[local]PE2(config-ospf-area)#interface to-CoC-CE2
[local]PE2(config-ospf-if)#exit
[local]PE2(config-ospf-area)#exit
[local]PE2(config-ospf)#exit
[local]PE2(config-ctx)#router bgp 200
[local]PE2(config-bgp)#address-family ipv4 unicast
[local]PE2(config-bgp-af)#exit
[local]PE1(config-bgp)#peer-group ibgp-peers internal
[local]PE1(config-bgp-peer-group)#advertisement-interval 1
[local]PE1(config-bgp-peer-group)#update-source lo1
[local]PE1(config-bgp-peer-group)#next-hop-self
[local]PE1(config-bgp-peer-group)#address-family ipv4 unicast
[local]PE1(config-bgp-peer-af)#exit
[local]PE1(config-bgp-peer-group)#exit
[local]PE2(config-bgp)#neighbor 3.3.3.3 internal
[local]PE2(config-bgp-neighbor)#peer-group ibgp-peers
[local]PE2(config-bgp-neighbor)#exit
[local]PE2(config-bgp)#neighbor 4.4.4.4 internal
[local]PE2(config-bgp-neighbor)#peer-group ibgp-peers
[local]PE2(config-bgp-neighbor)#exit
[local]PE2(config-bgp)#neighbor 5.5.5.5 internal
[local]PE2(config-bgp-neighbor)#peer-group ibgp-peers
```



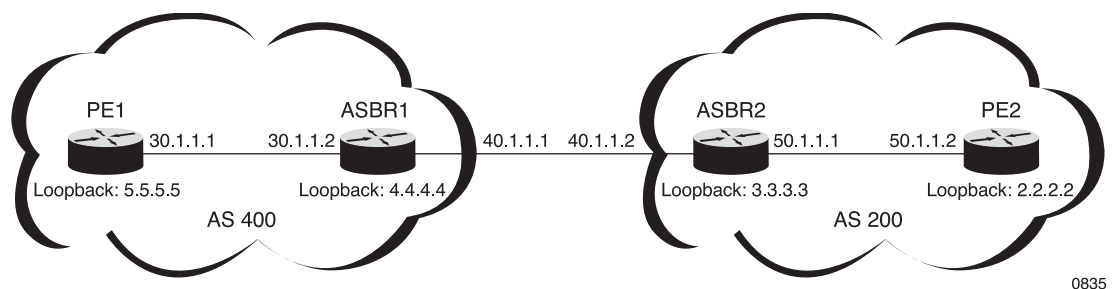
```
[local]PE2(config-bgp-neighbor)#exit
[local]PE2(config-bgp)#exit
[local]PE2(config-ctx)#exit
[local]PE2(config)#card ether-12-port 3
[local]PE2(config)#port ethernet 3/10
[local]PE2(config-port)#no shutdown
[local]PE2(config-port)#bind interface to-CoC-CE2 local
[local]PE2(config-port)#end
```

Multihop eBGP Label Redistribution

Figure 9-8 shows the network topology for this multihop eBGP label redistribution configuration example, where:

- The PE1 router is configured to have the ASBR1 router as its iBGP neighbor and the PE2 router as its eBGP neighbor. It maintains and distributes labeled IPv4 routes with ASBR1 and IPv4 VPN routes with the PE2 router.
- The ASBR1 router is configured to have the PE1 router as its iBGP neighbor and the ASBR2 router as its eBGP neighbor. It maintains a labeled IPv4 route to the PE1 router and exchanges labeled IPV4 VPN routes with the ASBR2 router using eBGP.
- The ASBR2 router is configured to have the PE2 router as its iBGP neighbor and the ASBR1 router as its eBGP neighbor. It maintains a labeled IPv4 route to the PE2 router and exchanges labeled IPV4 VPN routes with the ASBR1 router using eBGP.
- The PE2 router is configured to have the ASBR2 router as its iBGP neighbor and the PE1 router as its eBGP neighbor. It maintains and distributes labeled IPv4 routes with ASBR2 and IPv4 VPN routes with the PE1 router.

Figure 9-8 Multihop eBGP Label Redistribution Network Topology



Note To preserve VPN label next-hop information across the autonomous systems, the next-hop information for IPv4 VPN routes must not be changed on the local PE router when advertising to the remote PE router through multihop eBGP peering.

The configuration for the PE1 router is as follows:

```
[local]PE1#config
[local]PE1(config)#service multiple-contexts
[local]PE1(config)#context local
[local]PE1(config-ctx)#interface 3/10
[local]PE1(config-if)#ip address 30.1.1.1/24
```

```
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#interface lo1 loopback
[local]PE1(config-if)#ip address 5.5.5.5/32
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#router ospf 1
[local]PE1(config-ospf)#area 0.0.0.0
[local]PE1(config-ospf-area)#interface 3/10
[local]PE1(config-ospf-if)#exit
[local]PE1(config-ospf-area)#interface lo1
[local]PE1(config-ospf-if)#exit
[local]PE1(config-ospf)#exit
[local]PE1(config-ctx)#router mpls 1
[local]PE1(config-mpls)#interface 3/10
[local]PE1(config-mpls-if)#exit
[local]PE1(config-mpls)#exit
[local]PE1(config-ctx)#router ldp
[local]PE1(config-ldp)#interface 3/10
[local]PE1(config-ldp)#exit
[local]PE1(config-ctx)#router bgp 400
[local]PE1(config-bgp)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#exit
[local]PE1(config-bgp)#address-family ipv4 vpn
[local]PE1(config-bgp-af)#exit
[local]PE1(config-bgp)#neighbor 2.2.2.2 external
[local]PE1(config-bgp-neighbor)#remote-as 200
[local]PE1(config-bgp-neighbor)#advertisement-interval 1
[local]PE1(config-bgp-neighbor)#ebgp-multihop 10
[local]PE1(config-bgp-neighbor)#update-source lo1
[local]PE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#exit
[local]PE1(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE1(config-bgp-af)#next-hop-unchanged
[local]PE1(config-bgp-af)#exit
[local]PE1(config-bgp-neighbor)#exit
[local]PE1(config-bgp)#neighbor 4.4.4.4 internal
[local]PE1(config-bgp-neighbor)#advertisement-interval 1
[local]PE1(config-bgp-neighbor)#update-source lo1
[local]PE1(config-bgp-neighbor)#send label
[local]PE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#exit
[local]PE1(config-bgp-neighbor)#exit
[local]PE1(config-bgp)#exit
[local]PE1(config-ctx)#exit
[local]PE1(config)#context vpn1 vpn-rd 2:2
[local]PE1(config-ctx)#interface lo1 loopback
[local]PE1(config-if)#ip address 55.55.55.55/32
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#router bgp vpn
[local]PE1(config-bgp)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#export route-target 2:2
[local]PE1(config-bgp-af)#import route-target 2:2
[local]PE1(config-bgp-af)#redistribute connected
```

```

[local]PE1(config-bgp-af)#redistribute static
[local]PE1(config-bgp-af)#exit
[local]PE1(config-bgp)#exit
[local]PE1(config-ctx)#exit
[local]PE1(config)#card ether-12-port 3
[local]PE1(config)#port ethernet 3/10
[local]PE1(config-port)#no shutdown
[local]PE1(config-port)#bind interface 3/10 local
[local]PE1(config-port)#end

```

The configuration for the ASBR1 router is as follows:

```

[local]ASBR1#config
[local]ASBR1(config)#service multiple-contexts
[local]ASBR1(config)#context local
[local]ASBR1(config-ctx)#no ip domain-lookup
[local]ASBR1(config-ctx)#interface 3/2
[local]ASBR1(config-if)#ip address 30.1.1.2/24
[local]ASBR1(config-if)#exit
[local]ASBR1(config-ctx)#interface 3/4
[local]ASBR1(config-if)#ip address 40.1.1.1/24
[local]ASBR1(config-if)#exit
[local]ASBR1(config-ctx)#interface lo1 loopback
[local]ASBR1(config-if)#ip address 4.4.4.4/32
[local]ASBR1(config-if)#exit
[local]ASBR1(config-ctx)#router ospf 1
[local]ASBR1(config-ospf)#area 0.0.0.0
[local]ASBR1(config-ospf-area)#interface lo1
[local]ASBR1(config-ospf-if)#exit
[local]ASBR1(config-ospf-area)#interface 3/2
[local]ASBR1(config-ospf-if)#exit
[local]ASBR1(config-ospf-area)#exit
[local]ASBR1(config-ospf)#exit
[local]ASBR1(config-ctx)#router mpls 1
[local]ASBR1(config-mpls)#interface 3/2
[local]ASBR1(config-mpls-if)#exit
[local]ASBR1(config-mpls)#interface 3/4
[local]ASBR1(config-mpls-if)#exit
[local]ASBR1(config-mpls)#exit
[local]ASBR1(config-ctx)#router ldp
[local]ASBR1(config-ldp)#interface 3/2
[local]ASBR1(config-ldp)#exit
[local]ASBR1(config-ctx)#router bgp 400
[local]ASBR1(config-bgp)#address-family ipv4 unicast
[local]ASBR1(config-bgp-af)#redistribute ospf 1
[local]ASBR1(config-bgp-af)#exit
[local]ASBR1(config-bgp)#neighbor 5.5.5.5 internal
[local]ASBR1(config-bgp-neighbor)#advertisement-interval 1
[local]ASBR1(config-bgp-neighbor)#update-source lo1
[local]ASBR1(config-bgp-neighbor)#next-hop-self
[local]ASBR1(config-bgp-neighbor)#send label
[local]ASBR1(config-bgp-neighbor)#address-family ipv4 unicast

```

```
[local]ASBR1(config-bgp-af)#exit
[local]ASBR1(config-bgp-neighbor)#exit
[local]ASBR1(config-bgp)#neighbor 40.1.1.2 external
[local]ASBR1(config-bgp-neighbor)#remote-as 200
[local]ASBR1(config-bgp-neighbor)#advertisement-interval 1
[local]ASBR1(config-bgp-neighbor)#send label
[local]ASBR1(config-bgp-neighbor)#address-family ipv4 unicast
[local]ASBR1(config-bgp-af)#exit
[local]ASBR1(config-bgp-neighbor)#exit
[local]ASBR1(config-bgp)#exit
[local]ASBR1(config-ctx)#exit
[local]ASBR1(config)#card ether-12-port 3
[local]ASBR1(config)#port ethernet 3/2
[local]ASBR1(config-port)#no shutdown
[local]ASBR1(config-port)#bind interface 3/2 local
[local]ASBR1(config-port)#exit
[local]ASBR1(config)#port ethernet 3/4
[local]ASBR1(config-port)#no shutdown
[local]ASBR1(config-port)#bind interface 3/4 local
[local]ASBR1(config-port)#end
```

The configuration for the ASBR2 router is as follows:

```
[local]ASBR2#config
[local]ASBR2(config)#service multiple-contexts
[local]ASBR2(config)#context local
[local]ASBR2(config-ctx)#no ip domain-lookup
[local]ASBR2(config-ctx)#interface 3/2
[local]ASBR2(config-if)#ip address 40.1.1.2/24
[local]ASBR2(config-if)#exit
[local]ASBR2(config-ctx)#interface 3/4
[local]ASBR2(config-if)#ip address 50.1.1.1/24
[local]ASBR2(config-if)#exit
[local]ASBR2(config-ctx)#interface lo1 loopback
[local]ASBR2(config-if)#ip address 3.3.3.3/32
[local]ASBR2(config-if)#exit
[local]ASBR2(config-ctx)#router ospf 1
[local]ASBR2(config-ospf)#area 0.0.0.0
[local]ASBR2(config-ospf-area)#interface lo1
[local]ASBR2(config-ospf-if)#exit
[local]ASBR2(config-ospf-area)#interface 3/4
[local]ASBR2(config-ospf-if)#exit
[local]ASBR2(config-ospf-area)#exit
[local]ASBR2(config-ospf)#exit
[local]ASBR2(config-ctx)#router mpls 1
[local]ASBR2(config-mpls)#interface 3/2
[local]ASBR2(config-mpls-if)#exit
[local]ASBR2(config-mpls)#interface 3/4
[local]ASBR2(config-mpls-if)#exit
[local]ASBR2(config-mpls)#exit
[local]ASBR2(config-ctx)#router ldp
[local]ASBR2(config-ldp)#interface 3/4
```

```

[local]ASBR2(config-ldp)#exit
[local]ASBR2(config-ctx)#router bgp 400
[local]ASBR2(config-bgp)#address-family ipv4 unicast
[local]ASBR2(config-bgp-af)#redistribute ospf 1
[local]ASBR2(config-bgp-af)#exit
[local]ASBR2(config-bgp)#neighbor 2.2.2.2 internal
[local]ASBR2(config-bgp-neighbor)#advertisement-interval 1
[local]ASBR2(config-bgp-neighbor)#update-source lo1
[local]ASBR2(config-bgp-neighbor)#next-hop-self
[local]ASBR2(config-bgp-neighbor)#send label
[local]ASBR2(config-bgp-neighbor)#address-family ipv4 unicast
[local]ASBR2(config-bgp-af)#exit
[local]ASBR2(config-bgp-neighbor)#exit
[local]ASBR2(config-bgp)#neighbor 40.1.1.1 external
[local]ASBR2(config-bgp-neighbor)#remote-as 200
[local]ASBR2(config-bgp-neighbor)#advertisement-interval 1
[local]ASBR2(config-bgp-neighbor)#send label
[local]ASBR2(config-bgp-neighbor)#address-family ipv4 unicast
[local]ASBR2(config-bgp-af)#exit
[local]ASBR2(config-bgp-neighbor)#exit
[local]ASBR2(config-bgp)#exit
[local]ASBR2(config-ctx)#exit
[local]ASBR2(config)#card ether-12-port 3
[local]ASBR2(config)#port ethernet 3/2
[local]ASBR2(config-port)#no shutdown
[local]ASBR2(config-port)#bind interface 3/2 local
[local]ASBR2(config-port)#exit
[local]ASBR2(config)#port ethernet 3/4
[local]ASBR2(config-port)#no shutdown
[local]ASBR2(config-port)#bind interface 3/4 local
[local]ASBR2(config-port)#end

```

The configuration for the PE2 router is as follows:

```

[local]PE2#config
[local]PE2(config)#service multiple-contexts
[local]PE2(config)#context local
[local]PE2(config-ctx)#interface 3/10
[local]PE2(config-if)#ip address 50.1.1.2/24
[local]PE2(config-if)#exit
[local]PE2(config-ctx)#interface lo1 loopback
[local]PE2(config-if)#ip address 2.2.2.2/32
[local]PE2(config-if)#exit
[local]PE2(config-ctx)#router ospf 1
[local]PE2(config-ospf)#area 0.0.0.0
[local]PE2(config-ospf-area)#interface 3/10
[local]PE2(config-ospf-if)#exit
[local]PE2(config-ospf-area)#interface lo1
[local]PE2(config-ospf-if)#exit
[local]PE2(config-ospf)#exit
[local]PE2(config-ctx)#router mpls 1
[local]PE2(config-mpls)#interface 3/10

```

```
[local]PE2(config-mpls-if)#exit
[local]PE2(config-mpls)#exit
[local]PE2(config-ctx)#router ldp
[local]PE2(config-ldp)#interface 3/10
[local]PE2(config-ldp)#exit
[local]PE2(config-ctx)#router bgp 400
[local]PE2(config-bgp)#address-family ipv4 unicast
[local]PE2(config-bgp-af)#exit
[local]PE2(config-bgp)#address-family ipv4 vpn
[local]PE2(config-bgp-af)#exit
[local]PE2(config-bgp)#neighbor 5.5.5.5 external
[local]PE2(config-bgp-neighbor)#remote-as 200
[local]PE2(config-bgp-neighbor)#advertisement-interval 1
[local]PE2(config-bgp-neighbor)#ebgp-multihop 10
[local]PE2(config-bgp-neighbor)#update-source lo1
[local]PE2(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE2(config-bgp-af)#exit
[local]PE2(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE2(config-bgp-af)#next-hop-unchanged
[local]PE2(config-bgp-af)#exit
[local]PE2(config-bgp-neighbor)#exit
[local]PE2(config-bgp)#neighbor 3.3.3.3 internal
[local]PE2(config-bgp-neighbor)#advertisement-interval 1
[local]PE2(config-bgp-neighbor)#update-source lo1
[local]PE2(config-bgp-neighbor)#send label
[local]PE2(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE2(config-bgp-af)#exit
[local]PE2(config-bgp-neighbor)#exit
[local]PE2(config-bgp)#exit
[local]PE2(config-ctx)#exit
[local]PE2(config)#context vpn1 vpn-rd 2:2
[local]PE2(config-ctx)#interface lo1 loopback
[local]PE2(config-if)#ip address 55.55.55.55/32
[local]PE2(config-if)#exit
[local]PE2(config-ctx)#router bgp vpn
[local]PE2(config-bgp)#address-family ipv4 unicast
[local]PE2(config-bgp-af)#export route-target 2:2
[local]PE2(config-bgp-af)#import route-target 2:2
[local]PE2(config-bgp-af)#redistribute connected
[local]PE2(config-bgp-af)#redistribute static
[local]PE2(config-bgp-af)#exit
[local]PE2(config-bgp)#exit
[local]PE2(config-ctx)#exit
[local]PE2(config)#card ether-12-port 3
[local]PE2(config)#port ethernet 3/10
[local]PE2(config-port)#no shutdown
[local]PE2(config-port)#bind interface 3/10 local
[local]PE2(config-port)#end
```

Command Descriptions

This section describes the syntax and usage guidelines for the commands used to configure BGP/MPLS VPN features. The commands are presented in alphabetical order.

address-family ipv4 vpn
context vpn-rd
export route-target
import route-target
ip soft-gre

multi-paths eibgp
next-hop-on-lsp
router bgp vpn
route-target filter
vpn

address-family ipv4 vpn

address-family ipv4 vpn

Purpose

When entered in BGP configuration mode, enables VPN-IPv4 prefixes for a Border Gateway Protocol (BGP) routing instance and enters BGP address family configuration mode.

When entered in BGP neighbor configuration mode, enables VPN-IPv4 prefixes for a specified BGP neighbor and enters BGP neighbor address family configuration mode.

When entered in BGP peer group configuration mode, enables VPN-IPv4 prefixes for a specified BGP peer group and enters BGP peer group address family configuration mode.

Command Mode

BGP neighbor configuration
BGP peer group configuration
BGP router configuration

Syntax Description

This command has no keywords or arguments.

Default

None

Usage Guidelines

Use the **address-family ipv4 vpn** command in BGP configuration mode to specify the use of VPN-IPv4 prefixes for a BGP routing instance, and to enter BGP address family configuration mode.

Use the **address-family ipv4 vpn** command in BGP neighbor configuration mode to specify the use of VPN-IPv4 prefixes for a BGP neighbor in an internal BGP (iBGP) session, and to enter BGP neighbor address family configuration mode.

Use the **address-family ipv4 vpn** command in BGP peer group configuration mode to specify the use of VPN-IPv4 prefixes for a specified BGP peer group, and to enter BGP peer group address family configuration mode.

Note The **address-family ipv4 vpn** command cannot be used in non-local contexts.

Examples

The following example specifies the use of route flap statistics collection for VPN-IPv4 prefixes, and enables the address family for the BGP neighbor, 102.210.210.1:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#router bgp 100
[local]Redback(config-bgp)#address-family ipv4 vpn
[local]Redback(config-bgp-af)#flap-statistics
[local]Redback(config-bgp-af)#exit
```



```
[local]Redback(config-bgp)#neighbor 102.210.210.1 internal  
[local]Redback(config-bgp-neighbor)#address-family ipv4 vpn
```

Related Commands

- as-path-list**
- flap-statistics**
- remove-private-as**
- route-map**
- route-reflector-client**
- table-map**

context vpn-rd

context *ctx-name* **vpn-rd** *route-distinguisher*

Purpose

Creates a new Virtual Private Network (VPN) context, or specifies an existing VPN context, and enters context configuration mode.

Command Mode

global configuration

Syntax Description

<i>ctx-name</i>	Name of a new or existing context.
<i>route-distinguisher</i>	VPN route distinguisher, which can be expressed in either of the following formats: <i>asn:nnnn</i>, where <i>asn</i> is the autonomous system number and <i>nnnn</i> is a 32-bit integer. <i>ip-addr:nn</i>, where <i>ip-addr</i> is the IP address in the form <i>A.B.C.D</i> and <i>nn</i> is a 16-bit integer.

Default

None. A route distinguisher must be configured for a VPN context to be functional.

Usage Guidelines

Use the **context vpn-rd** command to create a new VPN context, or specify an existing VPN context, and enter context configuration mode. You cannot create new contexts on the system unless you have enabled the multiple context feature using the **service multiple-contexts** in global configuration mode. For information on the **service multiple-contexts** command, see the “Context Configuration” chapter in the *Basic System Configuration Guide* for the SmartEdge OS.

Entering the full **context vpn-rd** command is required to configure a VPN context. Entering the command without the **vpn-rd** portion creates a context that will not be recognized as VPN-enabled.

Note Each VPN context only supports one route distinguisher, and the route distinguisher must conform to the format specified in Internet Draft, *BGP/MPLS VPNs*, draft-ietf-ppvpn-rfc2547bis-01.txt.

Note An existing non-VPN context cannot be configured as a VPN context. You must delete the existing non-VPN context, and recreate it as a VPN context. Likewise, a VPN context cannot be configured as a non-VPN context. You must delete the existing VPN context, and recreate it as a non-VPN context.

Note This command is also documented in the “Context Configuration” chapter in the *Basic System Configuration Guide* for the SmartEdge OS.

Examples

The following example configures a VPN context, `vpncontext`, with the route distinguisher `701:3`:

```
[local]Redback(config)#context vpncontext vpn-rd 701:3  
[local]Redback(config-ctx)#
```

Related Commands

router bgp vpn

export route-target

export route-target {*ext-com* | **route-map** *route-map*}

Purpose

Creates a list of export route target extended communities for a specified Virtual Private Network (VPN) context.

Command Mode

BGP address family configuration

Syntax Description

<i>ext-com</i>	Route target extended community value that is added to the export target list. The route target extended community value can be expressed in either of the following formats: • <i>asn:nnnn</i>, where <i>asn</i> is the autonomous system number and <i>nnnn</i> is a 32-bit integer. • <i>ip-addr:nn</i>, where <i>ip-addr</i> is the IP address in the form <i>A.B.C.D</i> and <i>nn</i> is a 16-bit integer.
route-map <i>route-map</i>	Name of the route map used for this VPN context.

Default

None. A VPN context has no export route targets unless this command is used.

Usage Guidelines

Use the **export route-target** command to create a list of export route target extended communities for a specified VPN context. You can add multiple target communities on the same line, or you can issue the command multiple times with a single target as the parameter. Export route targets are sent as extended community attributes to other provider edge (PE) routers.

An export route map can be configured instead of a single target community value to give finer control over exported Border Gateway Protocol (BGP) routes. A route map allows you to filter routes or change attributes such as the export route target based on policy requirements. A route map may only be used when a target community value has not yet been configured.

Note The **export route-target** command can only be used in VPN contexts.

Examples

The following example configures the export route targets, 701:3 and 192.168.1.2:5:

```
[local]Redback(config)#context vpncontext vpn-rd 701:3
[local]Redback(config-ctx)#router bgp vpn
[local]Redback(config-bgp)#address-family ipv4 unicast
[local]Redback(config-bgp-af)#export route-target 701:3 192.168.1.2:5
```

The following example configures an export route map, customer-export-map:

```
[local]Redback(config)#context vpncontext vpn-rd 701:3
[local]Redback(config-ctx)#route map customer-export-map permit 10
[local]Redback(config-route-map)#match as-path foo
[local]Redback(config-route-map)#set ext-community RT:701:3
[local]Redback(config-route-map)#exit
[local]Redback(config-ctx)#route map customer-export-map permit 20
[local]Redback(config-route-map)#set ext-community RT:701:3
[local]Redback(config-route-map)#exit
[local]Redback(config-ctx)#router bgp vpn
[local]Redback(config-bgp)#address-family ipv4 unicast
[local]Redback(config-bgp-af)#export route-target route-map customer-export-map
```

Related Commands

- import route-target**
- route-map**
- route-target filter**

import route-target

import route-target *ext-com*

Purpose

Creates a list of import route target extended communities for a specified Virtual Private Network (VPN) context.

Command Mode

BGP address family configuration

Syntax Description

ext-com

Route target extended community value that is added to the import target list. The route target extended community value can be expressed in either of the following formats:

- *asn:nnnn*, where *asn* is the autonomous system number and *nnnn* is a 32-bit integer.
- *ip-addr:nn*, where *ip-addr* is the IP address in the form *A.B.C.D* and *nn* is a 16-bit integer.

Default

None. A VPN context has no import route targets unless this command is used.

Usage Guidelines

Use the **import route-target** command to create a list of import route target extended communities for a specified VPN context. You can add multiple target communities on the same line, or you can issue the command multiple times with a single target as the parameter. BGP routes learned from other provider edge (PE) routers that carry a specific route target extended community are imported into all VPN contexts configured with that extended community as an import route target.

Import route targets are used to filter routes from other provider edge (PE) routers before importing the routes into a VPN context.

Note The **import route-target** command can only be used in VPN contexts.

Examples

The following example configures the two import route targets, 701:3 and 192.168.1.2:5:

```
[local]Redback(config)#context vpncontext vpn-rd 701:3
[local]Redback(config-ctx)#router bgp vpn
[local]Redback(config-bgp)#address-family ipv4 unicast
[local]Redback(config-bgp-af)#import route-target 701:3 192.168.1.2:5
```

Related Commands

`export route-target`
`route-target filter`

ip soft-gre

ip soft-gre [**source** *src-addr*]

no ip soft-gre [**source** *src-addr*]

Purpose

Enables soft-Generic Routing Encapsulation (GRE) tunneling on the specified context.

Command Mode

context configuration

Syntax Description

source <i>src-addr</i>	Optional. Source address for the soft GRE tunnel. The IP address is in the form <i>A.B.C.D</i> .

Default

soft GRE tunneling is disabled.

Usage Guidelines

Use the **ip soft-gre** command to enable soft GRE tunneling on the specified context.

Encapsulating packets via Generic Routing Encapsulation (GRE) from an ingress provider edge (PE) router to an egress PE router is called soft GRE tunneling. Soft GRE tunnels are not Interior Gateway Protocol (IGP) visible links, and routing adjacencies are not supported across these tunnels. As a result, soft GRE tunnels have little in common with traditional (hard) GRE tunnels. The tunnel exists only in the sense of GRE encapsulation and decapsulation.

Only the ingress PE router and the egress PE router need to support the soft GRE functionality, and the PE routers can span over multiple autonomous systems.

Using soft GRE tunnels to transport Multiprotocol Label Switching (MPLS)-encapsulated packets is called Border Gateway Protocol/MPLS Virtual Private Network (BGP/MPLS VPN) over GRE, and is used to offer BGP/MPLS VPN service when a portion of a network does not have label switching enabled. BGP/MPLS VPN over GRE does not require pre-configuration of the remote GRE endpoint. These endpoints are the BGP next-hop addresses of the VPN routes, and are learned dynamically via BGP.

Note The **ip soft-gre** command is also documented in Chapter 14, “L2VPN Configuration,” where it is used to enable Layer 2 Virtual Private Network (L2VPN) over GRE.

Use the **no** form of this command to disable soft GRE on the specified context.

Examples

The following example enables soft GRE in the `local` context:

```
[local]Redback(config)#context local  
[local]Redback(config-ctx)#ip soft-gre
```


Related Commands

None

multi-paths eibgp

multi-paths eibgp *path-num*

{**no** | **default**} **multi-paths eibgp**

Purpose

Configures multipath load balancing using a mixture of both external Border Gateway Protocol (eBGP) and internal BGP (iBGP) equal-cost paths in a BGP/Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN).

Command Mode

BGP router configuration

Syntax Description

path-num Maximum number of equal-cost paths to use when balancing the traffic load. The range of values is 1 to 8.

Default

The command is disabled.

Usage Guidelines

Use the **multi-paths eibgp** command to configure multipath load balancing using a mixture of both eBGP and iBGP equal-cost paths in a BGP/MPLS VPN.

Note If the **multi-paths** command (in BGP router configuration mode) is used with the **external** or **internal** keyword to configure the maximum number of pure eBGP or pure iBGP (not a mixture of eBGP and iBGP) equal-cost paths for load balancing, then the number of eBGP or iBGP paths within the mixture of eBGP and iBGP multipaths can not exceed the corresponding limits specified for pure eBGP multipath or pure iBGP multipath respectively.

For more information about the **multi-paths** command, see Chapter 8, “BGP Configuration.”

Use the **no** or **default** form of this command to disable Multipath load balancing.

Examples

The following example configures multipath load balancing among any combination of up to 7 eBGP and iBGP equal cost paths:

```
[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config)#router bgp
[local]Redback(config-bgp)#multi-paths eibgp 7
[local]Redback(config-bgp)#
```

Related Commands

`multi-paths`

next-hop-on-lsp

next-hop-on-lsp

no next-hop-on-lsp

Purpose

Requires the next hop of a Border Gateway Protocol (BGP) Virtual Private Network (VPN) path to be reachable through a Multiprotocol Label Switching (MPLS) label-switched path (LSP) or a tunnel in order for a VPN route to be considered active.

Command Mode

BGP router configuration

Syntax Description

This command has no keywords or arguments.

Default

The next hop of a BGP VPN path must be reachable through an MPLS LSP or a tunnel in order for the VPN route to be considered active.

Usage Guidelines

Use the **next-hop-on-lsp** command to require the next hop of a BGP VPN path to be reachable through an MPLS LSP or a tunnel, in order for a VPN route to be considered active.

Use the **no** form of this command to enable a BGP VPN path to be considered active without requiring the next hop of a VPN path to be reachable through an MPLS LSP or a tunnel.

One common application for this command is configuring a BGP route reflector that is not part of an MPLS network, but is used to reflect BGP VPN routes to its clients within that MPLS network. In this configuration, the next hops of the VPN paths may not be reachable through an MPLS LSP or a tunnel from the route reflector's point of view. To solve the problem, use the **no** form of this command to disable the LSP or tunnel reachability check for the next hops, and therefore allow the BGP route reflector to correctly select the best paths and reflect the best paths to its clients.

Examples

The following example enables the sending of BGP VPN routes when the next hop is not resolved or reachable:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#router bgp
[local]Redback(config-bgp)#next-hop-on-lsp
[local]Redback(config-bgp)#
```

Related Commands

None

router bgp vpn

router bgp vpn

Purpose

Configures a Border Gateway Protocol (BGP) routing instance in a Virtual Private Network (VPN) context and enters BGP configuration mode.

Command Mode

context configuration

Syntax Description

This command has no keywords or arguments.

Default

None

Usage Guidelines

Use the **router bgp vpn** command to configure a BGP routing instance in a VPN context, and enter BGP configuration mode. A BGP instance is always required within a VPN context for the following reasons:

1. Customer routes must be distributed into BGP so they can be advertised across the iBGP sessions that connect provider edge (PE) routers. Customer routes can be distributed into BGP either statically or from other active routing protocols.
2. Route targets must also be configured within BGP address family configuration mode.

BGP does not function properly in a VPN context until it is first configured in the local context. Even though an autonomous system number (ASN) is not used when configuring a BGP instance in a VPN context, this instance uses the ASN from the BGP instance in the local context for peering with customer edge (CE) routers.

When configuring BGP peering sessions within a VPN context, only external neighbor sessions can be configured, because peering in a VPN context must only be configured with CE routers. Furthermore, the only permitted address family is IPv4 unicast, and peer groups cannot be configured.

Examples

The following example configures a BGP routing instance within a VPN context, and redistributes static routes from a customer into BGP:

```
[local]Redback(config)#context vpncontext vpn-rd 701:3
[local]Redback(config-ctx)#router bgp vpn
[local]Redback(config-bgp)#address-family ipv4 unicast
[local]Redback(config-bgp-af)#redistribute static
```

The following example configures a BGP peering session with a CE router:

```
[local]Redback(config)#context vpncontext vpn-rd 701:3  
[local]Redback(config-ctx)#router bgp vpn  
[local]Redback(config-bgp)#neighbor 205.1.2.2 external  
[local]Redback(config-bgp-neighbor)#remote-as 100  
[local]Redback(config-bgp-neighbor)#address-family ipv4 unicast
```

Related Commands

context vpn-rd
router-id

route-target filter

route-target filter

no route-target filter

Purpose

Enables automatic Border Gateway Protocol (BGP) route target community filtering.

Command Mode

BGP address family configuration

Syntax Description

This command has no keywords or arguments.

Default

Denies all incoming IP Version 4 (IPv4) Virtual Private Network (VPN) routes that are not imported into any VPN context, if the local router is not configured as a route reflector.

Usage Guidelines

Use the **route-target filter** command to enable automatic BGP route target community filtering. This command configures the local router, if it is not configured as a route reflector, to ignore all VPN routes received that are not imported into any VPN context.

Note For BGP route target filtering to work properly, you must first use the **address-family ipv4 vpn** command to specify the use of VPN-IPv4 prefixes for the BGP instance.

You can control the number of IPv4 VPN routes that the local autonomous system border router (ASBR) advertise to the remote ASBR by configuring a community for exportable routes on the inbound interface of the provider edge (PE) router, and configuring a community based filter on the outbound interface of the local ASBR to advertise only routes that match the community.

Use the **no** form of this command to allow the local router to accept all BGP IPv4 VPN routes. Accepting all IPv4 VPN routes is the desired behavior for a router configured as an ASBR for inter-AS VPNs.

Examples

The following example configures a local router to accept all received IPv4 VPN routes:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#router bgp 100
[local]Redback(config-bgp)#address-family ipv4 vpn
[local]Redback(config-bgp-af)#no route-target filter
```

Related Commands

address-family ipv4 vpn
export route-target

import route-target

vpn

vpn [**domain-id** *ip-addr*] {**domain-tag** *tag-name* | **local-as** *asn*}

no vpn

Purpose

Enables an Open Shortest Path First (OSPF) instance within a Virtual Private Network (VPN) context to treat redistributed Border Gateway Protocol (BGP) routes as VPN routes.

Command Mode

OSPF router configuration

Syntax Description

domain-id <i>ip-addr</i>	Optional. Domain ID value. Used to determine whether redistributed BGP routes should be treated as VPN routes and be handled differently than an OSPF instance configured within a VPN context; the default value is 0.
domain-tag <i>tag-name</i>	Domain tag. Used for type 5 link-state advertisements (LSAs) corresponding to redistributed BGP routes within the VPN domain. Either the <i>tag-name</i> or <i>asn</i> argument must be specified.
local-as <i>asn</i>	Autonomous system number (ASN), 2-byte. Used to formulate the tag for type 5 LSAs corresponding to redistributed BGP routes with the same VPN. Either the <i>tag-name</i> or <i>asn</i> argument must be specified, but the <i>tag-name</i> argument overrides the use of the <i>asn</i> argument to formulate the tag.

Default

OSPF VPN treatment of routes is disabled.

Usage Guidelines

Use the **vpn** command to enable an OSPF instance within a VPN context to treat redistributed BGP routes as VPN routes.

When a customer edge (CE) site is connected to multiple areas, the CE router's connection to a provider edge (PE) router should be in area 0 to allow correct handling of summary LSAs.

Note The **vpn** command is useful only when OSPF is used for PE-to-CE routing.

Use the **no** form of this command to disable the OSPF VPN treatment of routes.

Examples

The following example configures an OSPF instance within a VPN context to treat redistributed BGP routes with domain IDs equal to 1.1.1.1 as VPN routes:

```
[local]Redback(config-ospf)#vpn domain-id 1.1.1.1 domain-tag 0xfeedacee
```

Related Commands

`context vpn-rd`
`router ospf`
`sham-link`

IS-IS Configuration

This chapter provides an overview of Intermediate System-to-Intermediate System (IS-IS) routing, describes the tasks and commands used to configure IS-IS features through the SmartEdge® OS.

For information about the tasks and commands used to monitor, troubleshoot, and administer IS-IS, see the “IS-IS Operations” chapter in the *Routing Protocols Operations Guide* for the SmartEdge OS.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples
- Command Descriptions

Overview

IS-IS is an Interior Gateway Protocol (IGP) that uses link-state information to make routing decisions.

IS-IS is defined in ISO 10589, *Intermediate System to Intermediate System Intra-Domain Routing Exchange Protocol for Use in Conjunction with the Protocol for Providing the Connectionlessmode Network Service (ISO 8473)*, ISO DP 10589, February 1990, and RFC 1195, *Use of OSI IS-IS for Routing in TCP/IP and Dual Environments*.

Further overview information on IS-IS is described in the following sections:

- Supported IS-IS Features
- IS-IS Packets

Supported IS-IS Features

SmartEdge routers support IS-IS as an IP routing protocol. The implementation also includes:

- Level-1 and level-2 IP routing
- Passive interface
- Point-to-point (P2P) and LAN interface

- Unnumbered interface—For information about the **ip unnumbered** command, which enables IP processing on a point-to-point interface without assigning it an explicit IP address, see the “Interface Configuration” chapter in the *Basic System Configuration Guide* for the SmartEdge OS.
- P2P-over-LAN extension with unnumbered interface
- External route redistribution with a route map policy
- Level-1 to level-2 and level-2 to level-1 route leaking with prefix-list policy
- Multitopology IS-IS extension
- Interface block of link-state protocol (LSP) data unit flooding
- Three-way handshaking on point-to-point
- Graceful restart of IS-IS
- Summary address
- Manual triggering of IS-IS events
- Hash-Based Message Authentication Code-Message Digest 5 (HMAC-MD5) and simple authentication
- Dynamic hostname
- Multiprotocol Label Switching (MPLS) traffic engineering within IS-IS routing
- Traffic engineering wide metric extension
- Support for multiple contexts
- Support for multiple instances within a context
- Set over-load bit with bgp strict-tracking capability
- Periodic partial sequence number protocol data units (PSNPs) on point-to-point connections
- Periodic complete sequence number protocol data units (CSNPs) on point-to-point connections
- LSP receive-only interface
- Extensive **show** and **debug** commands

IS-IS Packets

IS-IS standards refer to packets as protocol data units (PDUs). IS-IS uses four types of PDUs to exchange routing information with neighbors:

- IS-IS Hello (IIH) PDUs
- LSPs
- CSNPs
- PSNPs

See ISO 10589 for detailed definitions of and information about these PDU types.

Configuration Tasks

Note In this section, the command syntax in the task tables displays only the root command; for the complete command syntax, see the full description for the command in the “Command Descriptions” section.

To configure IS-IS, perform the tasks described in the following sections:

- Configuring an IS-IS Instance
- Configuring an IS-IS LSP
- Configuring IS-IS SPF Calculations
- Configuring an IS-IS Interface
- Configuring IS-IS Hello Packets
- Configuring IS-IS Interface LSPs
- Configuring IS-IS Interface Metrics

Configuring an IS-IS Instance

To configure an IS-IS instance, perform the tasks described in Table 10-1. Enter all commands in IS-IS router configuration mode, unless otherwise noted.

Table 10-1 Configure an IS-IS Instance

Task	Root Command	Notes
Create an IS-IS instance.	router isis	Enter this command in context configuration mode. A context can have multiple IS-IS instances. No more than one instance of IS-IS can operate on a single interface. The no router isis command removes the IS-IS instance and all related configuration settings, which is different from deleting the last network entity title (NET). Deleting the last NET disables the IS-IS instance while preserving all configuration information.
The network entity title (NET) defined for each IS-IS instance contains the IS-IS area information and the router ID information. To define the NET for an IS-IS instance.	net	The NET defined for each IS-IS instance contains the IS-IS area information and the router ID information.
Enable only one IS-IS routing level.	is type	By default, both IS-IS routing levels, level-1 and level-2, are enabled.
Enable an address family for the IS-IS instance, and to access IS-IS address family configuration mode.	address-family	The address-family command is used to configure multipotology IS-IS routing. The multipotology IS-IS feature can generate multiple address families (topologies) for IS-IS; for example, one for IPv4 unicast network, and another for IPv4 multicast network. In order for an interface to participate in the routing for an address family, that address family must be enabled both at the instance level and at the interface level. If the IPv4 unicast address family is not desired, you must explicitly disable it using the no address-family command in IS-IS router configuration mode.

Table 10-1 Configure an IS-IS Instance (*continued*)

Task	Root Command	Notes
Enable the advertisement of short or wide metrics, and migration of existing traditional IS-IS networks, into the new scheme on a per-level basis.	metric-style	By default, IS-IS runs with wide metric styles enabled. Use the wide keyword to set the metric style back to the default. The wide-style metric can be enabled when traffic engineering capabilities or metrics longer than 63 are preferred. With the exception of devices in transition mode, all devices in the area must apply the same metric style; otherwise the IP topology becomes partitioned.
Redistribute IP routes learned through external route sources into the IS-IS routing instance.	redistribute	IS-IS can import routes from one or more external route sources including OSPF, RIP, BGP, STATIC, CONNECTED, and from other IS-IS instances. By default, the imported routes are redistributed into the level-2 routing process. The metrics of the external routes are set to zero if not specified. The metric type is internal if not specified as external. Currently, this command is only available for address family IPv4 unicast.
Configure route leaking between levels.	interarea-distribute	Redistributing routes between the IS-IS levels is called route leaking. Route leaking is automatically done from level-1 into level-2. The route leaking from level-2 into level-1 must be explicitly configured with a prefix-list. The leaked routes from level-2 into level-1 is possible in wide metric-style only. Make sure all the routers in the level-1 area can process wide metric-style. Currently, this command is only available for address family IPv4 unicast.
Configure IS-IS authentication at the IS-IS instance level.	authentication	IS-IS authentication is used to check authentication information on incoming IS-IS packets, or to attach authentication information to outgoing packets. There are two types of IS-IS authentication, simple and HMAC-MD5. HMAC-MD5 is more secure and we highly recommend it. Authentication can be configured at the IS-IS router configuration mode level, or at the interface configuration mode level. The interface authentication settings overwrite the router authentication settings for the IS-IS interface-related PDUs on that interface. Authentication at the IS-IS instance level controls the authentication scheme for the entire IS-IS instance on the router. Careful planning is necessary to ensure a smooth rollout of IS-IS authentication across a network. Use a secure channel to configure the passwords. We recommend that you choose HMAC-MD5 because it is highly secure.
Specify multiple summary addresses.	summary-address	IS-IS summary addresses can be used at the redistribution boundary to reduce routing information in the destination IS-IS domain or area. This redistribution boundary includes redistribution of external routes or between IS-IS levels. By default, the summary address is applied to the level-2 domain only. Currently, this command is only available for address family IPv4 unicast.
Change the IS-IS distance.	distance	The distance is used to specify a routing source preference. IS-IS uses the default distance of 115.
Configure a dynamic hostname for an IS-IS instance.	dynamic-hostname	Unless you use this command to specify a different hostname, the hostname of the IS-IS instance is the name specified through the system hostname command in global configuration mode.

Table 10-1 Configure an IS-IS Instance (*continued*)

Task	Root Command	Notes
Enable MPLS traffic engineering within IS-IS routing.	traffic-engineering	Enabling traffic engineering allows IS-IS LSPs to carry traffic engineering information on IS-IS interfaces, and can be enabled on either IS-IS level-1, level-2, or both level-1 and level-2 routing. Resource Reservation Protocol (RSVP) must be configured on the interface for IS-IS traffic engineering information to be included in its LSP for the link. An IS-IS metric style of wide or transition must be used for traffic engineering to take effect. The global router-id command in context configuration mode must be configured for the IS-IS LSP to carry the specified IP address of the router ID interface.
Configure the IS-IS attached bit preferences in L1 LSPs.	attached-bit	Routers in an IS-IS L1 area exchange information within the L1 area. For IP destinations not found in the prefixes in the L1 database, the L1 router must forward packets to the nearest router that is in both IS-IS L1 and L2 with the attached bit set in its L1 LSP.
Change the router's default number of multiple equal-cost IS-IS paths for load balancing of outgoing traffic packets.	maximum paths	The SmartEdge router load balances among the number of paths you specify with the paths argument if, in the routing table, they are the best paths among paths provided by all running routing protocols.
Limit the number of routes that can be redistributed into the IS-IS instance you are configuring.	maximum redistribute	If the maximum number of redistributed prefixes is reached, IS-IS stops redistributing external routes for the duration specified by the retry-interval interval construct.
Set the overload bit so that other devices do not use the SmartEdge router to forward traffic.	set-overload-bit	Other routers can still forward traffic to IP networks advertised by the SmartEdge router.
Enable fast convergence for an IS-IS instance.	fast-convergence	IS-IS fast convergence enables networks to offer high availability IP services to their customers by: • Responding to important network events, such as a backbone link down. • Quickly propagating the information to the entire domain. • Quickly calculating new routing information based on a network topology change, which minimizes the possibility of data packet loss in the network. This fast response not only affects the local router that has the link status change, but also the entire IS-IS routing domain. IS-IS fast convergence response is adaptive to the frequency of network events. It reacts quickly when there is a sudden network change, but it slows down when there are persistent topology changes to offer IS-IS routing stability.
Configure an IS-IS LSP.	For the complete list of tasks used to configure IS-IS LSP, see the "Configuring an IS-IS LSP" section.	
Configure IS-IS SPF calculations.	For the complete list of tasks used to configure IS-IS SPF calculations, see the "Configuring IS-IS SPF Calculations" section.	

Configuring an IS-IS LSP

To configure an IS-IS LSP, perform the tasks described in Table 10-2. Enter all commands in IS-IS router configuration mode.

Table 10-2 Configure an IS-IS LSP

Task	Root Command	Notes
Modify the length of time that IS-IS LSPs can live before timing out.	lsp max-lifetime	In the case of large networks, use this command in conjunction with the lsp refresh-interval command in IS-IS router configuration mode. Longer-lived LSPs allow for less flooding and higher stability. The value set by the lsp max-lifetime command should be at least 60 seconds longer than the value set through the lsp refresh-interval command, and should also be longer than the value set through the lsp gen-interval command.
Control how frequently an LSP can be regenerated for the IS-IS instance.	lsp refresh-interval	In the case of large networks, use this command in conjunction with the lsp max-lifetime command in IS-IS router configuration mode. Longer-lived LSPs allow for less flooding and higher stability. This value should be at least 60 seconds less than the value set through the lsp max-lifetime command, and should also be less than the value set through the lsp gen-interval command. This LSP refresh interval also determines the IS-IS periodical Shortest Path First (SPF) calculations on the system.
Control how frequently an LSP can be regenerated with new content.	lsp gen-interval	Decreasing the frequency at which an LSP can be regenerated with new content can stabilize a network at the cost of slower convergence. New versions of LSPs with updated content are generated less often and produce less load on the network than the load caused by flooding and route recomputation. Typically, the value set by the lsp gen-interval command should be lower than the values set through the lsp max-lifetime and lsp refresh-interval commands in IS-IS router configuration mode.

Configuring IS-IS SPF Calculations

To configure IS-IS SPF calculations, perform the tasks described in Table 10-3. Enter all commands in IS-IS router configuration mode.

Table 10-3 Configure IS-IS SPF Calculations

Task	Root Command	Notes
Modify the delay time between an event that triggers an SPF calculation and the calculation itself.	spf holddown	The purpose of the delay is to prevent immediate successive recalculations when computation triggers, such as new LSPs, occur in bursts as they often do. Because SPF calculations are performed when the topology changes, increasing this value offloads the processor, especially in large topologies, but slows down the convergence of the network.
Configure the minimum interval between SPF calculations.	spf interval	Increasing this value also offloads the processor, especially in large topologies, but slows down the convergence of the network.

Configuring an IS-IS Interface

To configure an IS-IS interface, perform the tasks described in Table 10-4. Enter all commands in IS-IS interface configuration mode, unless otherwise noted.

Table 10-4 Configure an IS-IS Interface

Task	Root Command	Notes
Enable IS-IS routing on the interface, and to access IS-IS interface configuration mode.	interface	Enter this command in IS-IS router configuration mode. Only one IS-IS instance can be running on an interface.
Configure the IS-IS designated router priority setting for the specified LAN interface.	priority	A priority value determines which router on a network is the first router chosen for sending and receiving traffic. The priority value is advertised in Hello packets. The router with the highest priority becomes the Designated Intermediate System (DIS). In IS-IS, there is no backup designated router. If a router is set to priority 0, it has a smaller chance of becoming the DIS, but it may not be prevented from becoming the DIS. When a router with a higher priority becomes available on the network, it takes over as the current DIS. In the case of equal priorities, the highest medium access control (MAC) address breaks the tie.
Enable an address family for the IS-IS interface, and to access IS-IS interface address family configuration mode.	address-family	The address-family command is used to configure multipotology IS-IS routing. The multipotology IS-IS feature can generate multiple address families (topologies) for IS-IS; for example, one for IPv4 unicast network, and another for IPv4 multicast network. In order for an interface to participate in the routing for an address family, that address family must be enabled both at the instance level and at the interface level. If the IPv4 unicast address family is not desired, you must explicitly disable it using the no address-family command in IS-IS router configuration mode.
Configure the type of IS-IS circuit on the interface.	circuit type	
Configure the IS-IS interface maximum transmit unit (MTU) size independent of the IP interface MTU size.	circuit mtu	
Enable periodic CSNPs to be sent on a P2P interface.	csnp periodic-on-ptp	Sending periodic CSNPs on point-to-point interfaces can increase the stability of the network, especially when flooding topology has been heavily pruned.
Configure the interval at which CSNPs are sent over the interface.	csnp interval	CSNPs contain a list of all LSPs in the database. An IS-IS system receiving CSNPs can compare this information with its own LSP database to determine whether it and the CSNP transmitter have synchronized LSP databases. CSNP packets are sent over LAN interfaces every 10 seconds unless you use this command to modify the interval. A shorter interval allows faster convergence; however, it increases bandwidth and CPU usage, and might add to instability in the network. In addition to saving bandwidth and CPU usage, a longer interval can increase overall network stability.
Enable optional IS-IS checksums on the interface.	optional-checksums	
Configure IS-IS instance to advertise the interface's IP addresses without actively running IS-IS on the interface (IS-IS passive mode).	passive-interface	When an IS-IS interface is configured in passive mode, IS-IS packets are sent and no adjacency is formed on the interface. IS-IS advertises the interface's IP address in its LSPs.

Table 10-4 Configure an IS-IS Interface (*continued*)

Task	Root Command	Notes
Configure IS-IS Hello packets.	For the complete list of tasks used to configure IS-IS Hello packets, see the “Configuring IS-IS Hello Packets” section.	
Configure IS-IS interface LSPs.	For the complete list of tasks used to configure IS-IS interface LSPs, see the “Configuring IS-IS Interface LSPs” section.	
Configure IS-IS interface metrics.	For the complete list of tasks used to configure IS-IS interface metrics, see the “Configuring IS-IS Interface Metrics” section.	

Configuring IS-IS Hello Packets

To configure IS-IS Hello packets, perform the tasks described in Table 10-5. Enter all commands in IS-IS interface configuration mode.

Table 10-5 Configure IS-IS Hello Packets

Task	Root Command	Notes
Configure the size of IS-IS Hello packets sent via the interface.	hello padding	
Modify the interval at which IS-IS Hello packets are sent via the interface.	hello interval	A shorter interval allows faster convergence; however, it increases bandwidth and CPU usage, and might add to instability in the network. In addition to saving bandwidth and CPU usage, a longer interval, especially when used in conjunction with a higher Hello multiplier can increase overall network stability. You can configure the Hello interval independently for level-1 and level-2, except on serial point-to-point (P2P) interfaces. Tuning the Hello interval and Hello multiplier on point-to-point interfaces is more useful than on LAN interfaces. Under link flapping, network churn, or heavy traffic congestion can cause Hello packet transmission or processing to be delayed, or packets to be dropped. Setting the Hello hold time too low can cause IS-IS adjacencies to flap, which can cause network instability. Use the millisecond or adaptive-millisecond keyword only on some P2P interfaces where the fast detection of lost adjacencies is required.
Determine how many IS-IS Hello packets can be missed by a neighbor before the SmartEdge router declares that the adjacency is down.	hello multiplier	The advertised holdtime in IS-IS Hello packets is the value of the <i>multiplier</i> argument multiplied by the value of the <i>seconds</i> argument set through the isis hello interval command in interface configuration mode. The Hello multiplier can be configured independently for level 1 and level 2, except on serial P2P interfaces. The level-1 and level-2 keywords are used on multiaccess networks or LAN interfaces. The Hello multiplier and the Hello interval can be different between different devices in one area.

Configuring IS-IS Interface LSPs

To configure an IS-IS instance, perform the tasks described in Table 10-6. Enter all commands in IS-IS interface configuration mode.

Table 10-6 Configure IS-IS Interface LSPs

Task	Root Command	Notes
Control the pace at which LSPs are flooded on the interface to IS-IS neighbors.	lsp interval	In dense-meshed IS-IS network topologies with a large number of devices and IS-IS neighbors, LSP flooding is the key scaling factor. Ensure that devices are not overloaded by LSPs from neighbors.
Prevent LSPs from being flooded on the interface.	lsp block-flooding	This command is typically used for point-to-point IS-IS interfaces. When a network topology has many redundant connections among IS-IS devices, LSPs can be flooded excessively inside the network, costing extra CPU cycles and bandwidth consumption. This feature is especially useful in a large, fully meshed IS-IS topology.
Configure how long the system should wait for an acknowledgment from the neighbor before sending an IS-IS LSP.	lsp retransmit-interval	The number of seconds should be greater than the expected round-trip delay between any two devices on the attached network. This command has no effect on LAN interfaces. On P2P links, the <i>interval</i> argument can be increased to enhance network stability. The retransmission interval can be larger for serial lines. More neighbors and paths over which LSPs are flooded allow for a longer interval.
Prevent the specified interface from forwarding LSPs.	lsp receive-only-mode	This command is used for internal lab test situations only and is relevant only for a stub IS-IS area where the goal is to import the network routing information from the operational network without exporting lab environment routing information into the operational network. After enabling IS-IS on an interface using the interface command in IS-IS router configuration mode, a delay in entering the lsp receive-only-mode command can result in lab routing information leaking into the operational network. To reduce the risk, immediately enter the lsp receive-only-mode command after enabling IS-IS on an interface using the interface command in IS-IS router configuration mode.

Configuring IS-IS Interface Metrics

To configure IS-IS interface metrics, perform the tasks described in Table 10-7. Enter all commands in IS-IS interface configuration mode.

Table 10-7 Configure IS-IS Interface Metrics

Task	Root Command	Notes
Configure the common IS-IS interface metric for the interface.	metric	Enter this command in IS-IS interface configuration mode. Metric values are determined by circuit distance, load-sharing requirements, and other traffic engineering factors.
Configure the IS-IS interface metric for a specific address family.	metric	Enter this command in IS-IS interface address family configuration mode. Metric values are determined by circuit distance, load-sharing requirements, and other traffic engineering factors. Address family IPv4 unicast always uses the common IS-IS interface metric. The metric command is not available for address family IPv4 unicast.

Configuration Examples

This section contains IS-IS configuration examples in the following subsections:

- Basic IS-IS
- Two Routers Using IS-IS for Routing Information Exchange
- IS-IS P2P-over-LAN Circuit
- Three Routers Using IS-IS for Routing Information Exchange
- Basic Multitopology IS-IS

Basic IS-IS

For IS-IS to work, you must configure one or more IS-IS instances in context configuration mode, and enable IS-IS for the interface. Although multiple instances can be configured in a context, only one can be enabled per interface. Use the **router isis** command in context configuration mode to create an IS-IS instance and enter IS-IS router configuration mode where you can configure parameters for the instance. Use the **isis router** command in interface configuration mode to enable a specific IS-IS instance for the interface. In order for IS-IS to exchange routing information with other routers, you must also assign a network entity title (NET).

The implementation of IS-IS supported by SmartEdge routers starts only on demand. One of two triggers starts IS-IS: the **router isis instance** command in context configuration mode, or the **isis router instance** command in interface configuration mode.

The following example illustrates a basic IS-IS configuration on a SmartEdge router. In this configuration, IS-IS is running in the local context with a single instance. The NET assigned to the router is 47.0001.1111.2222.3333.00. The 1111.2222.3333 portion is the system ID of the router, and it has to be unique within the entire IS-IS domain or area. The Ethernet interface, **first-isis-intf**, is configured to run the IS-IS instance, **my-backbone**. An IP address has to be assigned on the interface or an unnumbered interface is used.

```
[local]Redback(config)#context local

[local]Redback(config-ctx)#interface first-isis-intf
[local]Redback(config-if)#ip address 10.1.1.1/24
[local]Redback(config)#exit

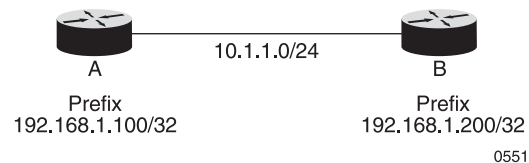
[local]Redback(config-ctx)#router isis my-backbone
[local]Redback(config-isis)#net 47.0001.1111.2222.3333.00
[local]Redback(config-isis)#interface first-isis-intf
[local]Redback(config-isis-if)#exit
[local]Redback(config-isis)#exit
[local]Redback(config-ctx)#exit

[local]Redback(config)#port ethernet 14/2
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface first-isis-intf local
```

Two Routers Using IS-IS for Routing Information Exchange

The following example illustrates two routers configuring IS-IS for routing information exchange; Figure 10-1 shows the topology.

Figure 10-1 Two Routers Exchanging Routing Information



In this example, router A and router B have an Ethernet connection to one another. Both routers run IS-IS level-1 routing and exchange route information with each other. Router A learns router B's loopback address of 192.168.1.200/32, and router B learns router A's loopback address of 192.168.1.100/32. Two different mechanisms are used to export each router's internal IP routes to its neighbors. Router A configures the IS-IS passive-interface to export the prefix 192.168.1.100/32; router B uses the redistribution of connected routes method to export prefix 192.168.1.200/32.

The configuration for Router_A is as follows:

```
[local]Router_A(config)#context local
[local]Router_A(config-ctx)#interface router-A-id loopback
[local]Router_A(config-if)#ip address 192.168.1.100/32
[local]Router_A(config-if)#exit

[local]Router_A(config-ctx)#interface first-isis-intf
[local]Router_A(config-if)#ip address 10.1.1.1/24
[local]Router_A(config-if)#exit

[local]Router_A(config-ctx)#router isis my-backbone
[local]Router_A(config-isis)#net 47.0001.1111.2222.3333.00
[local]Router_A(config-isis)#is type level-1

[local]Router_A(config-isis)#interface router-A-id
[local]Router_A(config-isis-if)#passive-interface
[local]Router_A(config-isis-if)#exit
[local]Router_A(config-isis)#interface first-isis-intf
[local]Router_A(config-isis-if)#exit
[local]Router_A(config-isis)#exit
[local]Router_A(config-ctx)#exit
[local]Router_A(config)#

[local]Router_A(config)#port ethernet 14/2
[local]Router_A(config-port)#no shutdown
[local]Router_A(config-port)#bind interface first-isis-intf local
```

The configuration for Router_B is as follows:

```
[local]Router_B(config)#context local
[local]Router_B(config-ctx)#interface router-B-id loopback
[local]Router_B(config-if)#ip address 192.168.1.200/32
[local]Router_B(config-if)#exit
```

```
[local]Router_B(config-ctx)#interface eth-10-1
[local]Router_B(config-if)#ip address 10.1.1.2/24
[local]Router_B(config-if)#exit

[local]Router_B(config-ctx)#router isis my-backbone
[local]Router_B(config-isis)#net 47.0001.0001.0002.0003.00
[local]Router_B(config-isis)#is type level-1
[local]Router_B(config-isis)#address-family ipv4 unicast
[local]Router_B(config-isis-af)#redistribute connected level-1
[local]Router_B(config-isis-af)#exit

[local]Router_B(config-isis)#interface router-B-id
[local]Router_B(config-isis-if)#passive-interface
[local]Router_B(config-isis-if)#exit
[local]Router_B(config-isis)#interface eth-10-1
[local]Router_B(config-isis-if)#exit
[local]Router_B(config-isis)#exit
[local]Router_B(config-ctx)#exit
[local]Router_B(config)#

[local]Router_B(config)#port ethernet 10/1
[local]Router_B(config-port)#no shutdown
[local]Router_B(config-port)#bind interface eth-10-1 local
```

IS-IS P2P-over-LAN Circuit

The following example configures an IS-IS point-to-point over LAN (P2P-over-LAN) circuit with an unnumbered interface. For detailed information about p2p-over-lan, see the Internet Draft, *draft-shen-isis-ospf-p2p-over-lan-01.txt*.

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#interface lo0 loopback
[local]Redback(config-if)#ip address 10.1.1.1/32
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface to-core2 p2p
[local]Redback(config-if)#ip unnumbered lo0
[local]Redback(config-if)#exit

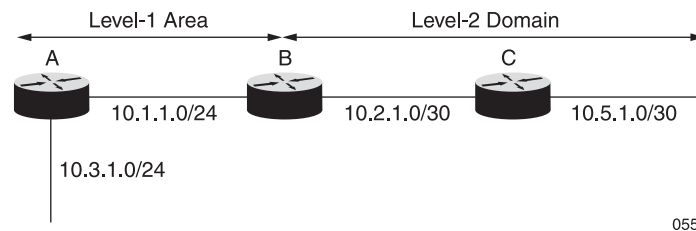
[local]Redback(config-ctx)#router isis my-backbone
[local]Redback(config-isis)#net 47.0001.1111.2222.3333.00
[local]Redback(config-isis)#interface to-core2
[local]Redback(config-isis-if)#exit
[local]Redback(config-isis)#exit
[local]Redback(config-ctx)#exit

[local]Redback(config)#port ethernet 14/2
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface to-core2 local
[local]Redback(config-port)#exit
```

Three Routers Using IS-IS for Routing Information Exchange

The following example has three routers using IS-IS for routing information exchange; Figure 10-2 shows the topology.

Figure 10-2 Three Routers Exchanging Routing Information



0552

Router A and router B are in the same Point of Presence (PoP). Router B is a backbone router connected to remote backbone router C. Router A is an edge router running two IS-IS instances and redistributes routes from one IS-IS instance to the other. Router B leaks level-2 routes into the level-1 area.

The configuration for Router_A is as follows:

```
[local]Router_A#configure
[local]Router_A(config)#context local
[local]Router_A(config-ctx)#interface toCoreRouter
[local]Router_A(config-if)#ip address 10.1.1.1/24
[local]Router_A(config-if)#exit
[local]Router_A(config-ctx)#interface toSubArea
[local]Router_A(config-if)#ip address 10.3.1.1/24
[local]Router_A(config-if)#exit

[local]Router_A(config-ctx)#router isis edge
[local]Router_A(config-isis)#is type level-1
[local]Router_A(config-isis)#net 47.0001.1111.2222.3333.00
[local]Router_A(config-isis)#authentication key-chain keys type hmac-md5
[local]Router_A(config-isis)#address-family ipv4 unicast
[local]Router_A(config-isis-af)#redistribute isis subArea level-1 route-map rtMap1
[local]Router_A(config-isis-af)#exit
[local]Router_A(config-isis)#interface toCoreRouter
[local]Router_A(config-isis-if)#exit
[local]Router_A(config-isis)#exit

[local]Router_A(config-ctx)#router isis subArea
[local]Router_A(config-isis)#is type level-1
[local]Router_A(config-isis)#net 47.0003.1000.2000.3000.00
[local]Router_A(config-isis)#interface toSubArea
[local]Router_A(config-isis-if)#exit
[local]Router_A(config-isis)#exit

[local]Router_A(config-ctx)#ip prefix-list prefixList
[local]Router_A(config-prefix-list)#permit 200.0.0.0/8 le 32
[local]Router_A(config-prefix-list)#permit 100.16.1.0/24 le 32
[local]Router_A(config-ctx)#key-chain keys key-id 1
```

```
[local]Router_A(config-key-chain)#key-string monday
[local]Router_A(config-key-chain)#exit
[local]Router_A(config-ctx)#route-map rtMap1 permit 10
[local]Router_A(config-route-map)#match ip address prefix-list prefixList
[local]Router_A(config-route-map)#set metric 4
[local]Router_A(config-route-map)#exit
[local]Router_A(config-ctx)#exit
[local]Router_A(config)#port ethernet 12/1
[local]Router_A(config-port)#no shutdown
[local]Router_A(config-port)#bind interface toCoreRouter local
[local]Router_A(config-port)#exit
[local]Router_A(config)#port ethernet 10/3
[local]Router_A(config-port)#no shutdown
[local]Router_A(config-port)#bind interface toSubArea local
[local]Router_A(config-port)#exit
[local]Router_A(config)#exit
```

The configuration for Router_B is as follows:

```
[local]Router_B#configure
[local]Router_B(config)#context local
[local]Router_B(config-ctx)#interface toBackbone
[local]Router_B(config-if)#ip address 10.2.1.1/30
[local]Router_B(config-if)#exit
[local]Router_B(config-ctx)#interface toEdge
[local]Router_B(config-if)#ip address 10.1.1.2/24
[local]Router_B(config-if)#exit

[local]Router_B(config-ctx)#router isis core
[local]Router_B(config-isis)#is type level-1
[local]Router_B(config-isis)#net 47.0001.0001.0002.0003.00
[local]Router_B(config-isis)#authentication key-chain keys type hmac-md5
[local]Router_B(config-isis)#address-family ipv4 unicast
[local]Router_B(config-isis-af)#interarea-distribute l2-to-l1 prefix-list prefixList
[local]Router_B(config-isis-af)#exit
[local]Router_B(config-isis)#interface toBackbone
[local]Router_B(config-isis-if)#circuit type level-2-only
[local]Router_B(config-isis-if)#exit
[local]Router_B(config-isis)#interface toEdge
[local]Router_B(config-isis-if)#circuit type level-1
[local]Router_B(config-isis-if)#exit
[local]Router_B(config-isis)#exit

[local]Router_B(config-ctx)#ip prefix-list prefixList
[local]Router_B(config-prefix-list)#permit 100.0.0.0/8 le 32
[local]Router_B(config-prefix-list)#permit 150.16.1.0/16 le 32
[local]Router_B(config-ctx)#key-chain keys key-id 1
[local]Router_B(config-key-chain)#key-string monday
[local]Router_B(config-key-chain)#exit
[local]Router_B(config-ctx)#exit
[local]Router_B(config)#port ethernet 12/1
[local]Router_B(config-port)#no shutdown
```



```
[local]Router_B(config-port)#bind interface toEdge local
[local]Router_B(config-port)#exit
[local]Router_B(config)#port pos 1/1
[local]Router_B(config-port)#no shutdown
[local]Router_B(config-port)#bind interface toBackbone local
[local]Router_B(config-port)#exit
[local]Router_B(config)#exit
```

The configuration for Router_C is as follows:

```
[local]Router_C#configure
[local]Router_C(config)#context local
[local]Router_C(config-ctx)#interface toPop
[local]Router_C(config-if)#ip address 10.2.1.2/30
[local]Router_C(config-if)#exit
[local]Router_C(config-ctx)#interface toSanFrancisco
[local]Router_C(config-if)#ip address 10.5.1.2/30
[local]Router_C(config-if)#exit

[local]Router_C(config-ctx)#router isis backbone
[local]Router_C(config-isis)#is type level-2-only
[local]Router_C(config-isis)#net 49.0002.1234.aaaa.bbbb.00
[local]Router_C(config-isis)#authentication key-chain keys type hmac-md5
[local]Router_C(config-isis)#interface toPop
[local]Router_C(config-isis-if)#exit
[local]Router_C(config-isis)#interface toSanFrancisco
[local]Router_C(config-isis-if)#exit
[local]Router_C(config-isis)#exit

[local]Router_C(config-ctx)#ip prefix-list prefixList
[local]Router_C(config-prefix-list)#permit 100.0.0.0/8 le 32
[local]Router_C(config-prefix-list)#permit 150.16.1.0/16 le 32
[local]Router_C(config-ctx)#key-chain keys key-id 1
[local]Router_C(config-key-chain)#key-string monday
[local]Router_C(config-key-chain)#exit
[local]Router_C(config-ctx)#exit
[local]Router_C(config)#port pos 5/2
[local]Router_C(config-port)#no shutdown
[local]Router_C(config-port)#bind interface toPop local
[local]Router_C(config-port)#exit
[local]Router_C(config)#port pos 9/2
[local]Router_C(config-port)#no shutdown
[local]Router_C(config-port)#bind interface toSanFrancisco local
[local]Router_C(config-port)#exit
[local]Router_C(config)#exit
```

Basic Multitopology IS-IS

The following example enables the IPv4 unicast and IPv4 multicast address families in the IS-IS instance `isis1`, enables the IPv4 unicast and IPv4 multicast address families on the `fa4/1` interface, enables the IPv4 unicast address family only on the `fa4/2` interface, and enables IPv4 multicast only on the `fa4/3` interface:

```
[local]Redback(config-ctx)#router isis isis1
[local]Redback(config-isis)#address-family ipv4 unicast
[local]Redback(config-isis-af)#exit
[local]Redback(config-isis)#address-family ipv4 multicast
[local]Redback(config-isis-af)#exit
[local]Redback(config-isis)#interface fa4/1
[local]Redback(config-isis-if)#address-family ipv4 unicast
[local]Redback(config-isis-if-af)#exit
[local]Redback(config-isis-if)#address-family ipv4 multicast
[local]Redback(config-isis-if-af)#exit
[local]Redback(config-isis-if)#exit
[local]Redback(config-isis)#interface fa4/2
[local]Redback(config-isis-if)#address-family ipv4 unicast
[local]Redback(config-isis-if-af)#exit
[local]Redback(config-isis-if)#exit
[local]Redback(config-isis)#interface fa4/3
[local]Redback(config-isis-if)#no address-family ipv4 unicast
[local]Redback(config-isis-if)#address-family ipv4 multicast
[local]Redback(config-isis-if-af)#exit
[local]Redback(config-isis-if)#exit
```

Command Descriptions

This section describes the syntax and usage guidelines for the commands used to configure IS-IS features. The commands are presented in alphabetical order.

address-family	lsp max-lifetime
attached-bit	lsp receive-only-mode
authentication	lsp refresh-interval
circuit mtu	lsp retransmit-interval
circuit type	maximum paths
csnp interval	maximum redistribute
csnp periodic-on-ntp	metric
distance	metric-style
dynamic-hostname	net
fast-convergence	optional-checksums
hello interval	passive-interface
hello multiplier	priority
hello padding	redistribute
interarea-distribute	router isis
interface	set-overload-bit
is type	spf holddown
lsp block-flooding	spf interval
lsp gen-interval	summary-address
lsp interval	traffic-engineering

address-family

address-family ipv4 {multicast | unicast}
no address-family ipv4 {multicast | unicast}

Purpose

Configures multitopology Intermediate System-to-Intermediate System (IS-IS) routing.

When entered in IS-IS router configuration mode, enables an address family for the IS-IS instance, and enters IS-IS address family configuration mode.

When entered in IS-IS interface configuration mode, enables an address family for the IS-IS interface, and enters IS-IS interface address family configuration mode.

Command Mode

IS-IS interface configuration
 IS-IS router configuration

Syntax Description

ipv4	Specifies the use of IP Version 4 (IPv4) address family.
multicast	Specifies the multicast subfamily to enable multicast topology. Disables the multicast topology when used in the no form of this command.
unicast	Specifies the unicast subfamily to enable unicast topology. Disables the unicast topology when used in the no form of this command.

Default

When an IS-IS instance is created, IPv4 unicast address family is enabled on the IS-IS instance.

When IS-IS is enabled on an interface, IPv4 unicast address family is enabled on the interface.

Usage Guidelines

Use the **address-family** command to configure multitopology IS-IS routing. The multitopology IS-IS feature can generate multiple address families (topologies) for IS-IS; for example, one for IPv4 unicast network, and another for IPv4 multicast network.

Multitopology IS-IS routing is useful in situations where multiple address families are needed; for example, with multitopology IS-IS routing enabled, the reverse path forwarding (RPF) checks used by the multicast routing protocol can use its own Interior Gateway Protocol (IGP) routing table instead of using the unicast routing table.

Use the **address-family** command in IS-IS interface configuration mode to enable an address family on an interface. In order for an interface to participate in the routing for an address family, that address family must be enabled both at the instance level and at the interface level.

Use the **address-family** command in IS-IS router configuration mode to enable an address family on an instance.

Note If the IPv4 unicast address family is not desired, you must explicitly disable it using the **no address-family** command in IS-IS router configuration mode.

Use the **no** form of this command in IS-IS interface configuration mode to disable an address family on an ISIS interface.

Use the **no** form of this command in IS-IS router configuration mode to disable an address family on an IS-IS instance.

For more information on multitopology IS-IS, see the Internet Draft, *M-ISIS: Multi Topology Routing in IS-IS*, draft-ietf-isis-wg-multi-topology-06.txt.

Examples

The following example enables the IPv4 unicast and IPv4 multicast address families in the IS-IS instance **isis1**, enables the IPv4 unicast and IPv4 multicast address families on the **fa4/1** interface, enables the IPv4 unicast address family only on the **fa4/2** interface, and enables IPv4 multicast only on the **fa4/3** interface:

```
[local]Redback(config-ctx)#router isis isis1
[local]Redback(config-isis)#address-family ipv4 unicast
[local]Redback(config-isis-af)#exit
[local]Redback(config-isis)#address-family ipv4 multicast
[local]Redback(config-isis-af)#exit
[local]Redback(config-isis)#interface fa4/1
[local]Redback(config-isis-if)#address-family ipv4 unicast
[local]Redback(config-isis-if-af)#exit
[local]Redback(config-isis-if)#address-family ipv4 multicast
[local]Redback(config-isis-if-af)#exit
[local]Redback(config-isis-if)#exit
[local]Redback(config-isis)#interface fa4/2
[local]Redback(config-isis-if)#address-family ipv4 unicast
[local]Redback(config-isis-if-af)#exit
[local]Redback(config-isis-if)#exit
[local]Redback(config-isis)#interface fa4/3
[local]Redback(config-isis-if)#no address-family ipv4 unicast
[local]Redback(config-isis-if)#address-family ipv4 multicast
[local]Redback(config-isis-if-af)#exit
[local]Redback(config-isis-if)#exit
```

Related Commands

- interarea-distribute**
- metric**
- redistribute**
- summary-address**

attached-bit

attached-bit {**ignore** | **never-set**}

no attached-bit {**ignore** | **never-set**}

Purpose

Configures the Intermediate System-to-Intermediate System (IS-IS) attached bit preferences in level 1 (L1) link-state protocol data units (LSPs).

Command Mode

IS-IS router configuration

Syntax Description

ignore	Configures IS-IS L1 routing to ignore the attached bit in LSPs. The IS-IS L1 router does not install a default route towards level 2 (L2) gateways.
never-set	Configures the IS-IS router to not set the attached bit in its L1 LSP, even if it is L2 attached.

Default

The ignore and never set preferences are both disabled.

Usage Guidelines

Use the **attached-bit** command to configure the IS-IS attached bit preferences in L1 LSPs.

Routers in an IS-IS L1 area exchange information within the L1 area. For IP destinations not found in the prefixes in the L1 database, the L1 router must forward packets to the nearest router that is in both IS-IS L1 and L2 with the attached bit set in its L1 LSP.

Use the **ignore** keyword on an IS-IS L1 router when route leaking is enabled on the IS-IS L2 gateways. When the **ignore** keyword is specified, the router ignores the attached bit on incoming L1 LSPs, and no default route is installed for the router that has the attached bit set in its LSP.

Use the **never-set** keyword on an L1L2 router when route leaking is enabled on the router. When the **never-set** keyword is specified, the router does not set the attached bit in its L1 LSP.

Use the **no** form of this command to disable a configured attached bit preference. You must include either the **ignore** or **never-set** keyword to disable each preference separately.

Examples

The following example configures an L1 router to ignore the attached bits from incoming L1 LSPs:

```
[local]Redback(config-ctx)#router isis ip-backbone
[local]Redback(config-isis)#attached-bit ignore
```

Related Commands

is type

authentication

authentication [**level-1** | **level-2**] **key-chain** *key-chain-name* [**type** {**hmac-md5** | **simple**}]
[**lsp-only**] [**no-check**]

no authentication {**level-1** | **level-2**} **key-chain** *key-chain-name* [**type** {**hmac-md5** | **simple**}]
[**lsp-only**] [**no-check**]

Purpose

Configures Intermediate System-to-Intermediate System (IS-IS) routing packet authentication using the simple or Hash-Based Message Authentication Code-Message Digest 5 (HMAC-MD5) authentication scheme for the IS-IS interface or IS-IS instance.

Command Mode

IS-IS interface configuration

IS-IS router configuration

Syntax Description

level-1	Optional, except in the no form of this command. Sets authentication for level 1 routing.
level-2	Optional, except in the no form of this command. Sets authentication for level 2 routing.
key-chain <i>key-chain-name</i>	Name of the key chain used for authentication.
type	Optional. Specifies that a type of authentication follows.
hmac-md5	Specifies HMAC-MD5 authentication.
simple	Specifies simple authentication.
lsp-only	Optional. If specified, only IS-IS link-state protocol data units (LSPs) are authenticated. Otherwise, IS-IS Hello (IIH), partial sequence number protocol data units (PSNPs), complete sequence number protocol data units (CSNPs), and LSPs are authenticated.
no-check	Optional. Causes the SmartEdge router to use authentication when sending packets, but not to check the packets it receives. This function is used during the transition period so that both devices can turn on authentication without a flag day.

Default

Authentication is not enabled. When you enter this command without specifying either level 1 or level 2 routing, authentication is set for both levels of IS-IS routing. If no authentication type is specified, HMAC-MD5 is used.

Usage Guidelines

Use the **authentication** command in IS-IS interface configuration mode to configure IS-IS routing packet authentication using the simple or HMAC-MD5 authentication scheme for an IS-IS interface.

Use the **authentication** command in IS-IS router configuration mode to configure IS-IS routing packet authentication using the simple or HMAC-MD5 authentication scheme for an IS-IS instance. To use a different key for a specific interface, use the **authentication** command in IS-IS interface configuration mode.

IS-IS authentication increases the network routing security. This command authenticates all IS-IS packets on the IS-IS interface or IS-IS instance.

The **key-chain** *key-chain-name* construct is provided because a key chain is required for simple and MD5 authentication schemes. A key chain provides a method for centrally managing keys and supports automatic key rollover. For information on the **key-chain key-id** command, see the “Key Chain Configuration” chapter in the *IP Services and Security Configuration Guide* for the SmartEdge OS.



Caution Risk of insecure IS-IS authentication. Careful planning is necessary to ensure a smooth rollout of IS-IS authentication across a network. To reduce the risk, and because HMAC-MD5 is highly secure, we strongly recommend using a secure channel to configure the passwords.

Use the **no** form of this command to disable authentication. In the **no** form, you must include either the **level-1** keyword, the **level-2** keyword, or the **key-chain** *key-chain-name* construct.

Examples

The following example applies key chain, key06, to the IS-IS interface, fa4/1, using simple authentication:

```
[local]Redback(config-ctx)#router isis ip-backbone
[local]Redback(config-isis)#interface fa4/1
[local]Redback(config-isis-if)#authentication key-chain key06 type simple
```

The following example applies key chain, key06, to the IS-IS instance, isis01, using HMAC-MD5 authentication:

```
[local]Redback(config-ctx)#router isis isis01
[local]Redback(config-isis)#authentication key-chain key06 type hmac-md5
```

Related Commands

None

circuit mtu

circuit mtu *size*

no circuit mtu

Purpose

Configures the Intermediate System-to-Intermediate System (IS-IS) interface maximum transmission unit (MTU) size independent of the IP interface MTU size.

Command Mode

IS-IS interface configuration

Syntax Description

size MTU size. The range of values is 256 to 9,198.

Default

None

Usage Guidelines

Use the **circuit mtu** command to configure the IS-IS interface MTU size independent of the IP interface MTU size. This configuration command decouples the IS-IS packet MTU and IP packet MTU, if needed, because IS-IS link-state packets must be flooded over all the IS-IS interfaces without link fragmentation. You can use this command to ensure that the maximum size of link-state packets are be transmitted to all the neighbors while ensuring that IP packets delivery remains efficient.

Use the **no** form of this command to use the same MTU size for the IS-IS interface and the IP interface.

Examples

The following IS-IS interface configuration shows an IS-IS running over Ethernet. Not all the routers on this Ethernet LAN can handle IS-IS packets over 1,500 bytes, and this Ethernet interface MTU is above 1,500 bytes, thus the user sets the IS-IS MTU different from the IP interface MTU.

```
[local]Redback(config-ctx)#router isis ip-backbone
[local]Redback(config-isis)#interface ge10/1
[local]Redback(config-isis-if)#circuit mtu 1500
```

Related Commands

None

circuit type

circuit type {**level-1** | **level-1-2** | **level-2-only**}

no circuit type

Purpose

Configures the type of Intermediate System-to-Intermediate System (IS-IS) adjacency on the interface.

Command Mode

IS-IS interface configuration

Syntax Description

level-1	Establishes level 1 adjacencies on the interface.
level-1-2	Establishes level 1 and 2 adjacencies with neighbors that are configured for both levels and that share a common area. Level 2 adjacencies are established for neighbors that do not have a common area.
level-2-only	Establishes level 2 adjacencies on the interface.

Default

The circuit type is level 1 and level 2.

Usage Guidelines

Use the **circuit type** command to configure the type of IS-IS adjacency on the interface.

Use the **no** form of this command to restore the setting to the default type of level 1 and level 2.

Examples

The following example configures the circuit type to **level-2** for the **fa4/1** interface running the **ip-backbone** IS-IS instance. Level 1 Hello packets are not sent on the **fa4/1** interface.

```
[local]Redback(config-ctx)#router isis ip-backbone
[local]Redback(config-isis)#interface fa4/1
[local]Redback(config-isis-if)#circuit type level-2-only
```

Related Commands

is type

csnp interval

csnp interval *seconds* [**level-1** | **level-2**]

no csnp interval

Purpose

Configures the interval at which complete sequence number protocol data units (CSNPs) are sent over the interface.

Command Mode

IS-IS interface configuration

Syntax Description

<i>seconds</i>	Interval of time, in seconds, between transmission of CSNPs on multiaccess networks. The range of values is 1 to 65,535; the default value is 10 seconds.
level-1	Optional. Configures the CSNP interval for level 1 independently.
level-2	Optional. Configures the CSNP interval for level 2 independently.

Default

CSNP packets are sent over LAN interfaces every 10 seconds. CSNPs are not sent over point-to-point (P2P) interfaces. When you enter this command without specifying either IS-IS level 1 or level 2 routing, CSNPs are sent at the same interval for both IS-IS levels.

Usage Guidelines

Use the **csnp interval** command to configure the interval at which CSNPs are sent over the interface. By default, CSNP packets are sent over LAN interfaces every 10 seconds. To enable the sending of CSNP packets on P2P interfaces, use the **csnp periodic-on-ptp** command in IS-IS interface configuration mode.

CSNPs contain a list of all link-state protocol data unit (LSP) packets in the database. An IS-IS system receiving CSNPs can compare this information with its own LSP database to determine whether it and the CSNP transmitter have synchronized LSP databases.

A shorter interval allows faster convergence; however, it increases bandwidth and CPU usage, and can add to instability in the network. In addition to saving bandwidth and CPU usage, a longer interval can increase overall network stability.

Use the **no** form of this command to restore the default interval at which CSNPs are sent over the interface.

Examples

The following example configures the CSNP interval on the fa4/1 interface at 15 seconds for IS-IS level-1 routing only:

```
[local]Redback(config-ctx)#router isis ip-backbone
[local]Redback(config-isis)#interface fa4/1
[local]Redback(config-isis-if)#csnp interval 15 level-1
```

Related Commands

`csnp periodic-on-ntp`

csnp periodic-on-ntp

csnp periodic-on-ntp

no csnp periodic-on-ntp

Purpose

Enables periodic complete sequence number protocol data units (CSNPs) to be sent on the point-to-point (P2P) interface.

Command Mode

IS-IS interface configuration

Syntax Description

This command has no keywords or arguments.

Default

The command is disabled.

Usage Guidelines

Use the **csnp periodic-on-ntp** command to enable periodic CSNPs to be sent on a P2P interface. Sending periodic CSNPs on P2P interfaces can increase the stability of the network, especially when flooding topology has been heavily pruned.

Use the **csnp interval** command in IS-IS interface configuration mode to modify the interval at which CSNPs are sent over the interface.

Use the **no** form of this command to disable the sending of CSNPs on a P2P interface.

Examples

The following example enables the sending of periodic CSNPs for IS-IS level-1 only on the fa4/1 interface:

```
[local]Redback(config-ctx)#router isis ip-backbone
[local]Redback(config-isis)#interface fa4/1
[local]Redback(config-isis-if)#csnp periodic-on-ntp level-1
```

Related Commands

csnp interval

lsp block-flooding

distance

distance *distance*

no distance

Purpose

Defines the administrative distance for an Intermediate System-to-Intermediate System (IS-IS) instance.

Command Mode

IS-IS router configuration

Syntax Description

distance

Administrative distance. The range of values is 1 to 255; the default value is 115.

Default

The default administrative distance is 115.

Usage Guidelines

Use the **distance** command to define the administrative distance for an IS-IS instance.

Administrative distance specifies how desirable a route obtained from IS-IS is as compared to the same route obtained from another protocol.

Table 10-8 lists the default distance for each variety of route sources.

Table 10-8 Default Distances Per-Route Source

Route Source	Default Distance
connected	0
EBGP	20
OSPF	110
IS-IS	115
RIP	120
IBGP	200

Use the **no** form of this command to reset the distance value to the default value of 115.

Examples

The following example modifies the administrative distance for the `isis_2` IS-IS instance to 19:

```
[local]Redback(config-ctx)#router isis isis_2
[local]Redback(config-isis)#distance 19
```

Related Commands

None

dynamic-hostname

dynamic-hostname [**display** | **router-name**]

no dynamic-hostname

Purpose

Configures a hostname for an Intermediate System-to-Intermediate System (IS-IS) instance.

Command Mode

IS-IS router configuration

Syntax Description

display	Optional. Displays the dynamic hostname mapping when any form of the show isis command in exec mode is used.
router-name	Optional. Displays the dynamic hostname for this IS-IS instance.

Default

If this command is not enabled, the name specified through the **system hostname** command in global configuration mode is used.

Usage Guidelines

Use the **dynamic-hostname** command to configure a hostname for an IS-IS instance.

Use the optional **display** keyword to enable dynamic hostname mapping for all **show isis** commands in exec mode.

By default, the hostname of the IS-IS instance is the name specified through the **system hostname** command in global configuration mode. Use the optional **router-name** keyword to allow a different hostname to be advertised for the IS-IS instance. This feature is useful when there are multiple IS-IS instances in that each IS-IS instance can display a different hostname. For information on the **system hostname** command, see the “Basic System Configuration” chapter in the *Basic System Configuration Guide* for the SmartEdge OS.

Use the **no** form of this command to revert to the system hostname or remove dynamic hostname mapping used with **show isis** commands.

Examples

The following example configures dynamic-hostname mapping for the **isis_2** IS-IS instance:

```
[local]Redback(config-ctx)#router isis isis_2
[local]Redback(config-isis)#dynamic-hostname display
```

Related Commands

None

fast-convergence

fast-convergence

no fast-convergence

Purpose

Enables fast convergence for an Intermediate System-to-Intermediate System (IS-IS) instance.

Command Mode

IS-IS router configuration

Syntax Description

This command has no keywords or arguments.

Default

Fast convergence is enabled for all instances of IS-IS routers.

Usage Guidelines

Use the **fast-convergence** command to enable fast convergence for an IS-IS instance.

IS-IS fast convergence enables networks to offer high availability IP services to their customers by:

- Responding to important network events, such as a backbone link down.
- Quickly propagating the information to the entire domain.
- Quickly calculating new routing information based on a network topology change, which minimizes the possibility of data packet loss in the network.

This fast response not only affects the local router that has the link status change, but also the entire IS-IS routing domain.

IS-IS fast convergence response is adaptive to the frequency of network events. It reacts quickly when there is a sudden network change, but it slows down when there are persistent topology changes to offer IS-IS routing stability.

Use the **no** form of this command to disable fast convergence for an IS-IS instance.

Examples

The following example enables fast convergence on the IS-IS instance, `ip-backbone`:

```
[local]Redback(config-ctx)#router isis ip-backbone  
[local]Redback(config-isis)#fast-convergence
```

Related Commands

router isis

hello interval

hello interval {*seconds* [**level-1** | **level-2**] | {**adaptive-millisecond** | **millisecond**} *milliseconds*}
no hello interval

Purpose

Modifies the interval at which Intermediate System-to-Intermediate System (IS-IS) Hello packets are sent on the interface.

Command Mode

IS-IS interface configuration

Syntax Description

<i>seconds</i>	Amount of time, in seconds, after which Hello packets are sent on the interface. The range of values is 1 to 65,535; the default value is 10.
level-1	Optional. Configures the Hello interval for IS-IS level 1 independently.
level-2	Optional. Configures the Hello interval for IS-IS level 2 independently.
adaptive-millisecond	Configures the Hello interval in the sub-second mode, and allows the Hello hold time to be adaptively adjusted when the link or network is under flapping or is unstable.
millisecond	Configures the Hello interval in the sub-second mode.
<i>milliseconds</i>	Amount of time, in 100 millisecond increments, after which Hello packets are sent on the interface. The range of values is 200 to 800 milliseconds.

Default

Hello packets are sent on the interface every 10 seconds. When you enter this command without specifying either IS-IS level 1 or level 2 routing, Hello packets are sent at the same rate for both levels.

Usage Guidelines

Use the **hello interval** command to modify the interval at which IS-IS Hello packets are sent on the interface.

A shorter interval allows faster convergence; however, it increases bandwidth and CPU usage, and might add to instability in the network. In addition to saving bandwidth and CPU usage, a longer interval, especially when used in conjunction with a higher Hello multiplier can increase overall network stability. To modify the Hello multiplier, use the **hello multiplier** command in IS-IS interface configuration mode.

You can configure the Hello interval independently for level 1 and level 2, except on serial point-to-point (P2P) interfaces. Tuning the Hello interval and Hello multiplier on P2P interfaces is more useful than on LAN interfaces.

Use the **millisecond** or **adaptive-millisecond** keyword to specify the sub-second IS-IS Hello interval. The minimum hold time, which is limited by IS-IS protocol, is one second. The hold time advertised by the Hello packets is the product of the Hello interval and the Hello multiplier rounded up to the nearest second. If the adaptive millisecond is configured on the interface, then the hold time can adaptively increase under the condition of adjacency flapping or network instability. The adaptive Hello hold time advertised by the Hello packets is double the regular hold time if the adjacencies over the interface has bounced three times in a 180-second period, and is limited by the hold time of 16 seconds.

The adaptive hold time can be reset to the original hold time value by issuing the **clear isis adaptive-holdtime** command in exec mode on the interface.



Caution Risk of data loss. Under link flapping, network churn, or heavy traffic congestion can cause Hello packet transmission or processing to be delayed, or packets to be dropped. Setting the Hello hold time too low can cause IS-IS adjacencies to flap, which can cause network instability. To reduce the risk, use the **millisecond** or **adaptive-millisecond** keyword only on some point-to-multipoint interfaces, where the fast detection of lost adjacencies is required. If you use the **adaptive-millisecond** keyword, and if the network churns cause IS-IS adjacencies to flap because the hold time is too small, the hold time on the interface is adaptively backed off to a safer region, to avoid network instability.

Use the **no** form of this command to restore the default Hello packet interval.

Examples

The following example configures the fa4/1 interface to send Hello packets every 20 seconds for IS-IS level-2 routing:

```
[local]Redback(config-ctx)#router isis ip-backbone
[local]Redback(config-isis)#interface fa4/1
[local]Redback(config-isis-if)#hello interval 20 level-2
```

Related Commands

hello multiplier

hello multiplier

hello multiplier *multiplier* [**level-1** | **level-2**]

no hello multiplier

Purpose

Determines how many Intermediate System-to-Intermediate System (IS-IS) Hello packets can be missed by a neighbor before the SmartEdge router declares that the adjacency is down.

Command Mode

IS-IS interface configuration

Syntax Description

<i>multiplier</i>	Number of IS-IS Hello packets a neighbor can miss. The range of values is 3 to 1,000; the default value is 3.
level-1	Optional. Configures the Hello multiplier independently for level 1 adjacencies independently.
level-2	Optional. Configures the Hello multiplier independently for level 2 adjacencies independently.

Default

The Hello multiplier is 3. When you enter this command without specifying either IS-IS level 1 or level 2 routing, the Hello multiplier value is the same for both levels.

Usage Guidelines

Use the **hello multiplier** command to determine how many IS-IS Hello packets can be missed by a neighbor before the SmartEdge router declares that the adjacency is down.

The advertised holdtime in IS-IS Hello packets is the value of the *multiplier* argument multiplied by the value of the *seconds* argument set through the **hello interval** command in IS-IS interface configuration mode.

The Hello multiplier can be configured independently for level 1 and level 2, except on serial point-to-point interfaces. The **level-1** and **level-2** keywords are used on multiaccess networks or LAN interfaces. The Hello multiplier and the Hello interval can be different between different devices in one area.

Use the **no** form of this command to restore the default multiplier.

Examples

The following example configures the neighbor to determine that an adjacency has gone down after 5 Hello packets are missed:

```
[local]Redback(config-ctx)#router isis ip-backbone
[local]Redback(config-isis)#interface fa4/1
[local]Redback(config-isis-if)#hello multiplier 5 level-2
```

Related Commands

`hello interval`

hello padding

hello padding { **always** | **first-only** | **never** }

no hello padding

Purpose

Configures the size of Intermediate System-to-Intermediate System (IS-IS) Hello packets sent on the interface.

Command Mode

IS-IS interface configuration

Syntax Description

always	Specifies that Hello packets should always be padded up to a maximum transmission unit (MTU) size. This is the default behavior.
first-only	Specifies that only the initial Hello packets are padded up to the MTU size.
never	Specifies that Hello packets are not padded to an MTU size.

Default

By default, first-only Hello packets are padded up to the MTU size.

Usage Guidelines

Use the **hello padding** command to configure the size of IS-IS Hello packets sent on the interface.

Use the **always** keyword if permanent checking of an MTU size in both directions is preferred and bandwidth is not important. Use the **first-only** keyword to balance between ensuring MTU integrity and saving bandwidth. Use the **never** keyword to allow for maximum bandwidth efficiency with no MTU integrity protection.

Use the **no** form of this command to restore the default.

Examples

The following example pads Hello packets up to the MTU size until the adjacency is established in both directions:

```
[ local]Redback(config-ctx)#router isis ip-backbone
[ local]Redback(config-isis)#interface fa4/1
[ local]Redback(config-isis-if)#hello padding first-only
```

Related Commands

None

interarea-distribute

interarea-distribute {l1-to-l2 | l2-to-l1} [**prefix-list** *pl-name*]

no interarea-distribute {l1-to-l2 | l2-to-l1}

Purpose

Distributes routes from one level of an Intermediate System-to-Intermediate System (IS-IS) to another.

Command Mode

IS-IS address family configuration

Syntax Description

l1-to-l2	Distributes routes from level 1 into level 2. By default, level 1 routes are distributed in to level 2.
l2-to-l1	Distributes routes from level 2 into level 1. By default, level 2 routes are not distributed into level 1.
prefix-list <i>pl-name</i>	Optional. Name of the prefix list that is to be applied.

Default

Level 1 routes are distributed into level 2. Level 2 routes are not distributed into level 1.

Usage Guidelines

Use the **interarea-distribute** command to distribute routes from one level of IS-IS to another. This distribution is also known as route leaking. If scalability is a concern, you can apply a prefix list and its routing policies to filter which routes are leaked from one level to another.

Note Currently, this command is only available for address family IPv4 unicast.

A prerequisite for level 2 to level 1 route leaking is that all devices inside level 1 have the capability of calculating routes based on IS-IS-wide metrics.

Use the **no** form of this command to disable distribution of routes between IS-IS levels.

Examples

The following configuration distributes level 2 routes into level 1 if the routes match 23.4.5.0 for the prefix length 24 and above. All the other routes are not distributed into level 1.

```
[local]Redback(config-ctx)#router isis second_tag
[local]Redback(config-isis)#address-family ipv4 unicast
[local]Redback(config-isis-af)#interarea-distribute l2-to-l1 prefix-list sys2
[local]Redback(config-isis-af)#exit
[local]Redback(config-isis)#exit
[local]Redback(config-ctx)#ip prefix-list sys2 permit 23.4.5.0/24 ge 25
```

Related Commands

address-family
metric-style
redistribute
summary-address

interface

interface *if-name*

no interface *if-name*

Purpose

Enables Intermediate System-to-Intermediate System (IS-IS) routing on the interface and enters IS-IS interface configuration mode.

Command Mode

IS-IS router configuration

Syntax Description

if-name Name of the interface on which IS-IS is to be enabled.

Default

None

Usage Guidelines

Use the **interface** command to enable IS-IS routing on the interface and enter IS-IS interface configuration mode. To activate IS-IS on the interface, you must also assign a network entity title (NET) through the **net** command in IS-IS router configuration mode and bind the interface to a valid, activated port using the **bind interface** command in port configuration mode. For information on the **bind interface** command, see the “Bindings Configuration” chapter in the *Ports, Circuits, and Tunnels Configuration Guide* for the SmartEdge OS.

Note Only one IS-IS instance can be running on an interface.

Use the **no** form of this command to disable IS-IS routing on the interface.

Examples

The following example enables the IS-IS instance, `ip-backbone`, on the `fa4/1` interface. A NET of `49.0003.0003.0003.00` is assigned to the instance and the `fa4/1` interface is bound to an Ethernet port in the `local` context.

```
[local]Redback(config-ctx)#router isis ip-backbone
[local]Redback(config-isis)#net 49.0003.0003.0003.00
[local]Redback(config-isis)#interface fa4/1
[local]Redback(config-isis-if)#exit
[local]Redback(config-isis)#exit
[local]Redback(config-ctx)#exit
[local]Redback(config)#port ethernet 7/1
[local]Redback(config-port)#bind interface fa4/1 local
```

Related Commands

net
router isis

is type

is type {**level-1** | **level-1-2** | **level-2-only**}

no is type

Purpose

Configures the Intermediate System-to-Intermediate System (IS-IS) routing level used by the SmartEdge router for the specified IS-IS instance.

Command Mode

IS-IS router configuration

Syntax Description

level-1	Specifies that the SmartEdge router operates only in the level 1 area.
level-1-2	Specifies that the SmartEdge router participates in both IS-IS level 1 and level 2 routing.
level-2-only	Specifies that the SmartEdge router operates in level 2 only.

Default

The SmartEdge router participates in both level 1 and level 2 routing.

Usage Guidelines

Use the **is type** command to configure the IS-IS routing level used by the SmartEdge router for the specified IS-IS instance.

Use the **level-1** keyword to specify level 1 routing. All other destinations are routed to the closest device running either level 2 or both levels. If the wide-style metric is enabled with the **metric-style** command, routes can be advertised from level 2 areas into the level 1 area, and devices running level 1 can select the best level 2 device on a per-destination basis.

Use the **level-1-2** keyword to specify both level 1 and level 2 routing. The database and Shortest Path First (SPF) computation for each level is independent. When the wide-metric style is enabled with the **metric-style** command, the router can advertise and summarize level 1 routes into level 2 areas and vice versa.

Use the **level-2-only** keyword to specify level 2 routing.

Use the **no** form of this command to restore the SmartEdge router to the default behavior of participating in both level 1 and level 2 routing.

Examples

The following example configures the SmartEdge router for IS-IS level-2-only routing:

```
[local]Redback(config-ctx)#router isis ip-backbone
[local]Redback(config-isis)#is type level-2-only
```

Related Commands

metric-style

lsp block-flooding

lsp block-flooding [**level-1** | **level-2**]

no lsp block-flooding [**level-1** | **level-2**]

Purpose

Prevents intermediate link-state protocol data units (LSPs) from being flooded out through the Intermediate System-to-Intermediate System (IS-IS)-enabled interface.

Command Mode

IS-IS interface configuration

Syntax Description

level-1	Optional. Enables block flooding on IS-IS level 1 routing independently.
level-2	Optional. Enables block flooding on IS-IS level 2 routing independently.

Default

LSPs are flooded over IS-IS-enabled interfaces. When you enter this command without specifying either level 1 or level 2 routing, LSPs are flooded on both ISIS levels 1 and 2.

Usage Guidelines

Use the **lsp block-flooding** command to prevent LSPs from being flooded out through the IS-IS-enabled interface. When a network topology has many redundant connections among IS-IS devices, LSPs can be flooded excessively inside the network, costing extra CPU cycles and bandwidth consumption. This feature is especially useful in a large, fully-meshed IS-IS topology.

Note This command is typically used for point-to-point (P2P) IS-IS interfaces.

Note Avoid blocking some LSPs completely.

Use the **no** form of this command to restore to the default behavior of flooding LSPs on the interface.

Examples

The following example blocks LSP flooding on level 1 only for the fa4/1 interface running the IS-IS instance ip-backbone:

```
[local]Redback(config-ctx)#router isis ip-backbone
[local]Redback(config-isis)#interface oc48-4/1
[local]Redback(config-isis-if)#lsp block-flooding level-1
```

Related Commands

lsp interval
lsp retransmit-interval

lsp gen-interval

lsp gen-interval *interval* [**level-1** | **level-2**]

no lsp gen-interval

Purpose

Controls how frequently a link-state protocol data unit (LSP) can be regenerated with new content for the Intermediate System-to-Intermediate System (IS-IS) instance.

Command Mode

IS-IS router configuration

Syntax Description

<i>interval</i>	Frequency, in seconds, at which an LSP can be regenerated with new content. The range of values is 1 to 120; the default value is 10.
level-1	Optional. Sets the frequency at which an LSP can be regenerated for level 1 independently.
level-2	Optional. Sets the frequency at which an LSP can be regenerated for level 2 independently.

Default

An LSP can be regenerated every 10 seconds.

Usage Guidelines

Use the **lsp gen-interval** command to control how frequently an LSP can be regenerated with new content for the IS-IS instance.

Decreasing the frequency at which an LSP can be regenerated with new content can stabilize a network at the cost of slower convergence. New versions of LSPs with updated content are generated less often and produce less load on the network than the load caused by flooding and route recomputation. Typically, the value set by the **lsp gen-interval** command should be lower than the values set through the **lsp max-lifetime** and **lsp refresh-interval** commands in IS-IS router configuration mode.

Use the **no** form of this command to restore the default.

Examples

The following example sets the LSP regeneration frequency for IS-IS **level-1** to 30 seconds:

```
[local]Redback(config-ctx)#router isis ip-backbone
[local]Redback(config-isis)#lsp gen-interval 30 level-1
```

Related Commands

lsp max-lifetime

lsp refresh-interval

lsp interval

lsp interval *interval*

no lsp interval

Purpose

Controls the pace at which link-state protocol data unit (LSP) transmissions are flooded on the interface to Intermediate System-to-Intermediate System (IS-IS) neighbors.

Command Mode

IS-IS interface configuration

Syntax Description

<i>interval</i>	Interval, in milliseconds, between successive LSPs. The range of values is 10 to 65,535; the default value is 33.

Default

The minimum delay time is set to 33 milliseconds.

Usage Guidelines

Use the **lsp interval** command to control the pace at which LSPs are flooded on the interface to IS-IS neighbors. In dense-meshed IS-IS network topologies with a large number of devices and IS-IS neighbors, LSP flooding is the key scaling factor. Ensure that devices are not overloaded by LSPs from neighbors.

Use the **no** form of this command to restore the default, minimum delay value.

Examples

The following example configures the SmartEdge router to transmit LSPs every 100 milliseconds (10 packets per second) on the `serial1/1` interface:

```
[local]Redback(config-ctx)#router isis ip-backbone
[local]Redback(config-isis)#interface serial1/1
[local]Redback(config-isis-if)#lsp interval 100
```

Related Commands

lsp block-flooding
lsp retransmit-interval

lsp max-lifetime

lsp max-lifetime *lifetime*

no lsp max-lifetime

Purpose

Modifies the length of time that Intermediate System-to-Intermediate System (IS-IS) link-state protocol data units (LSPs) can live on the network before timing out.

Command Mode

IS-IS router configuration

Syntax Description

<i>lifetime</i>	Maximum lifetime, in seconds, of an LSP. The range of values is 120 to 65,535; the default value is 1,200.

Default

The maximum lifetime of an LSP is 1,200 seconds.

Usage Guidelines

Use the **lsp max-lifetime** command to modify the length of time LSPs can live on the network before timing out. Use this command in conjunction with the **lsp refresh-interval** command in the case of large networks. Longer-lived LSPs allow for less flooding and higher stability.

The value set by the **lsp max-lifetime** command should be at least 60 seconds more than the value set through the **lsp refresh-interval** command, and should also be more than the value set through the **lsp gen-interval** command.

Use the **no** form of this command to restore the default maximum lifetime value of 1,200 seconds.

Examples

The following example sets the maximum lifetime for LSPs to 900 seconds, which is 300 seconds more than the LSP refresh interval:

```
[local]Redback(config-isis)#lsp refresh-interval 600
[local]Redback(config-isis)#lsp max-lifetime 900
```

Related Commands

lsp gen-interval
lsp refresh-interval

lsp receive-only-mode

lsp receive-only-mode

no lsp receive-only-mode

Purpose

Prevents the specified Intermediate System-to-Intermediate System (IS-IS) interface from forwarding link-state protocol data units (LSPs).

Command Mode

IS-IS interface configuration

Syntax Description

This command has no keywords or arguments.

Default

None

Usage Guidelines

Use the **lsp receive-only-mode** command to prevent the specified IS-IS interface from forwarding LSPs



Caution Risk of leaked routing information. This command is used for internal lab test situations only and is relevant only for a stub IS-IS area where the goal is to import the network routing information from the operational network without exporting lab environment routing information into the operational network. After enabling IS-IS on an interface using the **interface** command in IS-IS router configuration mode, a delay in entering the **lsp receive-only-mode** command can result in lab routing information leaking into the operational network. To reduce the risk, immediately enter the **lsp receive-only-mode** command after enabling IS-IS on an interface using the **interface** command in IS-IS router configuration mode.

Use the **no** form of this command to reestablish forwarding of LSPs.

Examples

The following example prevents the IS-IS interface, `isis1`, on a lab router from forwarding LSPs:

```
[local]Redback(config-ctx)#router isis ip-backbone
[local]Redback(config-isis)#interface isis1
[local]Redback(config-isis-if)#lsp receive-only-mode
```

Related Commands

interface—IS-IS router configuration mode
lsp block-flooding
passive-interface

lsp refresh-interval

lsp refresh-interval *interval*

no lsp refresh-interval

Purpose

Controls how frequently a link-state protocol data units (LSPs) can be regenerated for the Intermediate System-to-Intermediate System (IS-IS) instance.

Command Mode

IS-IS router configuration

Syntax Description

<i>interval</i>	Frequency, in seconds, with which an LSP can be regenerated. The range of values is 30 to 65,535; the default value is 900.

Default

LSPs can be regenerated every 900 seconds.

Usage Guidelines

Use the **lsp refresh-interval** command to control how frequently an LSP can be regenerated for the specified IS-IS instance.

Use this command in conjunction with the **lsp max-lifetime** command in the case of large networks. Longer-lived LSPs allow for less flooding and higher stability. This value should be at least 60 seconds less than the value set through the **lsp max-lifetime** command, and should also be less than the value set through the **lsp gen-interval** command. This LSP refresh interval also determines the IS-IS periodical Shortest Path First (SPF) calculations on the system.

Use the **no** form of this command to restore the default.

Examples

The following example sets the LSP refresh interval to 600 seconds, which is 300 seconds less than the maximum lifetime value:

```
[local]Redback(config-isis)#lsp refresh-interval 600
[local]Redback(config-isis)#lsp max-lifetime 900
```

Related Commands

lsp gen-interval
lsp max-lifetime

lsp retransmit-interval

lsp retransmit-interval *interval*

no lsp retransmit-interval

Purpose

Configures the length of time the system should wait for an acknowledgment from the neighbor before resending Intermediate System-to-Intermediate System (IS-IS) link-state protocol data units (LSPs).

Command Mode

IS-IS interface configuration

Syntax Description

<i>interval</i>	Interval, in seconds, between LSP retransmissions. The range of values is 0 to 65,535; the default value is 5.

Default

The retransmission interval is five seconds.

Usage Guidelines

Use the **lsp retransmit-interval** command to configure how long the system should wait for an acknowledgment from the neighbor before resending an IS-IS LSP. The number of seconds should be greater than the expected round-trip delay between any two devices on the attached network.

This command has no effect on LAN interfaces. On point-to-point links, the *interval* argument can be increased to enhance network stability. The retransmission interval can be larger for serial lines. More neighbors and paths over which LSPs are flooded allow for a longer interval.

Use the **no** form of this command to restore the default retransmission interval of five seconds.

Examples

The following example configures the pos11/1 interface to retransmit LSPs every 10 seconds:

```
[local]Redback(config-ctx)#router isis ip-backbone
[local]Redback(config-isis)#interface pos11/1
[local]Redback(config-isis-if)#lsp retransmit-interval 10
```

Related Commands

lsp block-flooding

lsp interval

maximum paths

maximum paths *paths*

{**no** | **default**} **maximum paths**

Purpose

Changes the router's default number of multiple equal-cost Intermediate System-to-Intermediate System (IS-IS) paths for load balancing of outgoing traffic packets.

Command Mode

IS-IS router configuration

Syntax Description

<i>paths</i>	Maximum number of equal-cost paths used as the best paths. The range of values is 1 to 8.

Default

The maximum number of equal-cost paths is 8.

Usage Guidelines

Use the **maximum paths** command to change the router's default number of multiple equal-cost IS-IS paths for load balancing of outgoing traffic packets. The SmartEdge router load balances among these IS-IS paths if, in the routing table, they are the best paths among paths provided by all running routing protocols.

Use the **no** or **default** form of this command to restore the default setting.

Examples

The following example sets the maximum number of paths to 4:

```
[local]Redback(config-ctx)#router isis isis01  
[local]Redback(config-isis)#maximum paths 4
```

Related Commands

None

maximum redistribute

maximum redistribute *prefixes* [**retry-interval** *interval*]

no maximum redistribute

Purpose

Limits the maximum number of routes that can be redistributed into the specified Intermediate System-to-Intermediate System (IS-IS) instance.

Command Mode

IS-IS router configuration

Syntax Description

<i>prefixes</i>	Maximum number of prefixes that can be redistributed into the IS-IS routing instance. The range of values is 1 to 1,000,000.
retry-interval <i>interval</i>	Optional. Amount of time, in seconds, before IS-IS attempts to redistribute routes after the maximum prefix value is exceeded. The range of values is 120 to 7,200; the default value is 600.

Default

There is no maximum limit for the number of prefixes that can be redistributed. The retry interval is 600 seconds.

Usage Guidelines

Use the **maximum redistribute** command to limit the maximum number of routes that can be redistributed into the specified IS-IS instance.

If the maximum number of redistributed prefixes is reached, IS-IS stops redistributing external routes for the duration specified by the **retry-interval interval** construct.

Use the **no** form of this command to restore the default settings.

Examples

The following example redistributes up to 50000 prefixes into the `isis01` IS-IS instance. If this number is exceeded, routes are not redistributed again for 300 seconds (5 minutes):

```
[local]Redback(config-ctx)#router isis isis01  
[local]Redback(config-isis)#maximum redistribute 50000 retry-interval 300
```

Related Commands

lsp gen-interval
lsp refresh-interval

metric

metric *metric* [**level-1** | **level-2**]

no metric

Purpose

When entered in IS-IS interface configuration mode, configures the common Intermediate System-to-Intermediate System (IS-IS) metric for the interface.

When entered in IS-IS interface address family configuration mode, configures the IS-IS interface metric for a specific address family.

Command Mode

IS-IS interface configuration

Syntax Description

<i>metric</i>	Metric used for calculating the Shortest Path First (SPF). The range of values is 1 to 63 for narrow-style metrics, and 0 to 16,777,215 for wide-style metrics; the default value is 10 for an active IS-IS circuit and is 1 for a passive IS-IS interface.
level-1	Optional. Configures the metric for IS-IS level 1 routing independently.
level-2	Optional. Configures the metric for IS-IS level 2 routing independently.

Default

The default common metric is 10 for an active IS-IS circuit and is 1 for a passive IS-IS interface. When you enter this command without specifying either level 1 or level 2 routing, the same metric value is used for both levels.

The default address family-specific IS-IS interface metric is not configured.

Usage Guidelines

Use the **metric** command in IS-IS interface configuration mode to configure the common IS-IS metric for the interface.

Use the **metric** command in IS-IS interface address family configuration mode to configure the IS-IS interface metric for a specific address family.

Metric values are determined by circuit distance, load-sharing requirements, and other traffic engineering factors.

Use the **no** form of the **metric** command in IS-IS interface configuration mode to restore the IS-IS common metric for the interface to the default value.

Use the **no** form of the **metric** command in IS-IS interface address family configuration mode to remove the address family-specific IS-IS interface metric configuration. When the IS-IS interface metric specific to an address family is not configured, then the common IS-IS metric for the interface is used for that address family.

Note Address family IPv4 unicast always uses the common IS-IS interface metric. The **metric** command is not available for address family IPv4 unicast.

Examples

The following example assigns an IS-IS metric of 43 to the fa4/1 interface for level 2 routing:

```
[local]Redback(config-ctx)#router isis ip-backbone  
[local]Redback(config-isis)#interface fa4/1  
[local]Redback(config-isis-if)#metric 43 level-2
```

Related Commands

address-family
metric-style

metric-style

metric-style [**narrow** | **transition** | **wide**] [**level-1** | **level-2**]

no metric-style

Purpose

Allows the advertisement of short or wide metrics and migration of existing traditional Intermediate System-to-Intermediate-System (IS-IS) networks into the new scheme on a per-level basis.

Command Mode

IS-IS router configuration

Syntax Description

narrow	Optional. Allows advertisement of metrics with values in the range from 0 to 63. If enabled on a level, no device operating in wide mode can be present in the same area. All metrics from redistributed and calculated routing information is clipped to a maximum of 63.
transition	Optional. Allows advertisement of metrics with values in the range from 0 to 63. Higher metrics can be specified and redistributed, but are only used when the metric style is changed to wide mode. Devices with narrow or wide mode enabled can be present in the same area.
wide	Optional. Allows advertisement of metrics longer than 63. If enabled on a level, no device operating in narrow mode can be present in the same area.
level-1	Optional. Sets the metric style independently for level 1. If wide metric style is enabled, routes can be advertised from the level 2 area into the level 1 area, and level 1 devices can select the best level 2 device on a per-destination basis. If narrow mode is enabled, level 1 devices must forward traffic to the closest level 2 device.
level-2	Optional. Sets the metric style independently for level 2.

Default

The SmartEdge router uses the wide metric style for both IS-IS level 1 and level 2.

Usage Guidelines

Use the **metric-style** command to allow the advertisement of short or wide metrics and migration of existing traditional IS-IS networks into the new scheme on a per-level basis. Implementation of this command adheres to the IETF draft-ietf-isis-traffic-02.txt document, *IS-IS Extensions for Traffic Engineering*.

The wide-style metric can be enabled when traffic engineering capabilities or metrics longer than 63 are preferred. With the exception of devices in transition mode, all devices in the area must apply the same metric style; otherwise the IP topology becomes partitioned.

Use the **no** form of this command to restore the default behavior of using the wide metric style for both IS-IS levels 1 and 2.

Examples

The following example sets the metric style to **transition** for **level-1** routing:

```
[local]Redback(config-ctx)#router isis isis01  
[local]Redback(config-isis)#metric-style transition level-1
```

Related Commands

metric

net

net *net*

no net *net*

Purpose

Configures a network entity title (NET) for the Intermediate System-to-Intermediate System (IS-IS) routing process.

Command Mode

IS-IS router configuration

Syntax Description

<i>net</i>	Area address and system ID for the IS-IS routing process. This argument can be either an address in hexadecimal-dotted byte format or a name.

Default

A NET is mandatory for IS-IS operation. If this option is not configured, the IS-IS instance is disabled.

Usage Guidelines

Use the **net** command to configure a NET for the IS-IS routing process.

Network entity titles can be anywhere between 8 and 20 bytes in length, and are provided in a hexadecimal-dotted byte format, such as 47.0005.80ff.e200.02aa.0a00.0002.00. The last byte, which is the Network Service Access Point (NSAP) n-selector, must be zero. The 6 bytes before the last byte indicate the system ID. This ID must be the same for all NETs configured for the system, and must be unique within the IS-IS domain. The bytes before that indicate an area ID, which is a variable from 1 to 13 bytes. Multiple areas can be specified in scenarios of area merges and the necessity of renumbering. The protocol will not form a level 1 adjacency between two devices if they have no areas in common.

Use the **no** form of this command to remove a NET.

Examples

The following example assigns a NET of 47.0001.0002.0002.0002.00 to the ip-backbone IS-IS instance:

```
[local]Redback(config-ctx)#router isis ip-backbone
[local]Redback(config-isis)#net 47.0001.0002.0002.0002.00
```

Related Commands

router isis

optional-checksums

optional-checksums [level-1 | level-2]

no optional-checksums [level-1 | level-2]

Purpose

Enables optional Intermediate System-to-Intermediate System (IS-IS) checksums on the interface.

Command Mode

IS-IS interface configuration

Syntax Description

level-1 Optional. Enables checksums for IS-IS level 1 routing independently.

level-2 Optional. Enables checksums for IS-IS level 2 routing independently.

Default

The command is disabled.

Usage Guidelines

Use the **optional-checksums** command to enable optional IS-IS checksums on the interface.

Use the **no** form of this command to disable optional IS-IS checksums.

Examples

The following example enables optional checksums on the fa4/1 interface:

```
[local]Redback(config-ctx)#router isis ip-backbone  
[local]Redback(config-isis)#interface fa4/1  
[local]Redback(config-isis-if)#optional-checksums
```

Related Commands

None

passive-interface

passive-interface

no passive-interface

Purpose

Configures the Intermediate System-to-Intermediate System (IS-IS) instance to advertise the interface's IP address without actively running IS-IS on the interface.

Command Mode

IS-IS interface configuration

Syntax Description

This command has no keywords or arguments.

Default

Passive mode is disabled.

Usage Guidelines

Use the **passive-interface** command to configure the IS-IS instance to advertise the interface's IP addresses without actively running IS-IS on the interface.

When an IS-IS interface is configured in passive mode, IS-IS packets are sent and no adjacency is formed on the interface. IS-IS advertises the interface's IP address in its link-state protocol data units (LSPs).

The default metric value for a passive interface is 1. To change the metric value, use the **metric** command in IS-IS interface configuration mode.

Use the **no** form of this command to disable this option.

Examples

The following example configures the fa4/1 interface as a passive IS-IS interface:

```
[local]Redback(config-ctx)#router isis ip-backbone  
[local]Redback(config-isis)#interface fa4/1  
[local]Redback(config-isis-if)#passive-interface
```

Related Commands

metric

priority

priority *priority* [**level-1** | **level-2**]

no priority

Purpose

Configures the Intermediate System-to-Intermediate System (IS-IS) designated router priority setting for the specified LAN interface.

Command Mode

IS-IS interface configuration

Syntax Description

<i>priority</i>	Priority setting. The range of values is 0 to 127; the default value is 64. Higher numbers signify a higher priority.
level-1	Optional. Sets the priority for IS-IS level 1 routing independently.
level-2	Optional. Sets the priority for IS-IS level 2 routing independently.

Default

The priority setting is 64.

Usage Guidelines

Use the **priority** command to configure the IS-IS designated router priority setting for the specified LAN interface.

A priority value determines which router on a network is the first router chosen for sending and receiving traffic. The priority value is advertised in Hello packets. The router with the highest priority becomes the Designated Intermediate System (DIS).

In IS-IS, there is no backup designated router. If a router is set to priority 0, it has a smaller chance of becoming the DIS, but it may not be prevented from becoming the DIS. When a router with a higher priority becomes available on the network, it takes over as the current DIS. In the case of equal priorities, the highest medium access control (MAC) address breaks the tie.

Use the **no** form of this command to restore the default priority.

Examples

The following example sets the priority for the **fa4/1** interface to 80, making it more likely to become the DIS for IS-IS **level-1** routing:

```
[local]Redback(config-ctx)#router isis ip-backbone
[local]Redback(config-isis)#interface fa4/1
[local]Redback(config-isis-if)#priority 80 level-1
```

Related Commands

None

redistribute

```

redistribute { bgp asn | connected | isis instance-name | nat | ospf instance-id | rip instance-name |
  static [dvsr] | subscriber [address | static]} [level-1 | level-2] [metric metric]
  [metric-type {internal | external}] [route-map map-name]

no redistribute { bgp asn | connected | isis instance-name | nat | ospf instance-id | rip instance-name |
  static [dvsr] | subscriber [address | static]} [level-1 | level-2] [metric metric]
  [metric-type {internal | external}] [route-map map-name]

```

Purpose

Redistributes IP routes learned through external routing protocols into the Intermediate System-to-Intermediate System (IS-IS) routing instance.

Command Mode

IS-IS address family configuration

Syntax Description

bgp <i>asn</i>	Border Gateway Protocol (BGP) autonomous system number (ASN). Redistributes routes from BGP into the IS-IS routing instance. The range of values for the <i>asn</i> argument is 1 to 65,535.
connected	Redistributes routes from directly attached networks into the IS-IS routing instance.
isis <i>instance-name</i>	IS-IS instance name. Redistributes routes from the specified IS-IS routing instance into the current IS-IS routing instance.
nat	Redistributes network address translation (NAT) routes into the IS-IS routing instance.
ospf <i>instance-id</i>	Open Shortest Path First (OSPF) instance ID. Redistributes routes from the specified OSPF routing instance into the IS-IS routing instance. The range of values is 1 to 65,535.
rip <i>instance-name</i>	Routing Information Protocol (RIP) instance name. Redistributes routes from the specified RIP routing instance into the IS-IS routing instance.
static	Redistributes static routes into the IS-IS routing instance. Optional with the subscriber keyword; redistributes only static subscriber routes into the IS-IS routing domain.
dvsr	Optional. Redistributes dynamically verified static routing (DVSR) subtype of static routes into the IS-IS routing instance.
subscriber	Redistributes routes configured within subscriber records into the IS-IS routing instance.
address	Optional. Redistributes only subscriber address routes into the IS-IS routing instance.

level-1	Optional. Redistributes only level 1 routes into the IS-IS routing instance.
level-2	Optional. Redistributes only level 2 routes into the IS-IS routing instance independently.
metric <i>metric</i>	Optional. Metric assigned to the redistributed routes. The range of values is 0 to 16,777,215; the default metric is 0.
metric-type	Optional. Assigns a metric type to the redistributed routes; the default metric type is internal.
internal	Assigns an internal metric type to redistributed routes. When the system receives an LSP with an internal metric type, the total cost is the cost the route from itself to the redistributing system plus the advertised cost to reach the destination.
external	Assigns an external metric type to redistributed routes. When the system receives a link-state protocol data unit (LSP) with an external metric type, it considers only the advertised cost to reach the destination
route-map <i>map-name</i>	Optional. Route map name. Applies a previously configured route map that filters the routes that are redistributed into the IS-IS routing instance. If this option is not specified, all routes from the specified protocol are redistributed into the IS-IS routing instance.

Default

Routes learned by other protocols are not distributed into the IS-IS routing instance.

Usage Guidelines

Use the **redistribute** command to redistribute routes learned through external protocols into the IS-IS routing instance.

Note Currently, this command is only available for address family IPv4 unicast.

You must enter multiple **redistribute** commands to redistribute routes from several different kinds of routing protocols into the IS-IS routing instance.

Use the **no** form of this command to disable redistribution into the IS-IS routing instance.

Examples

The following example redistributes static IP routes into an IS-IS level-1 area with an advertised metric of 10. The internal metric type is used by default.

```
[local]Redback(config-ctx)#router isis ip-backbone
[local]Redback(config-isis)#address-family ipv4 unicast
[local]Redback(config-isis-af)#redistribute static level-1 metric 10
```

Related Commands

address-family

summary-address

router isis

router isis *instance-name*

no router isis *instance-name*

Purpose

Creates an Intermediate System-to-Intermediate System (IS-IS) instance and enters IS-IS router configuration mode.

Command Mode

context configuration

Syntax Description

instance-name IS-IS instance name.

Default

No instance of IS-IS is configured.

Usage Guidelines

Use the **router isis** command to create an IS-IS instance and to enter IS-IS router configuration mode. To enable the IS-IS routing process, you must assign a network entity title (NET) to the instance. Use the **net** command in IS-IS router configuration mode.

A context can have multiple IS-IS instances. No more than one instance of IS-IS can operate on a single interface. To enable IS-IS on an interface, use the **interface** command in IS-IS router configuration mode.

Use the **no** form of this command to delete the IS-IS instance.



Caution Risk of IS-IS configuration settings loss. The **no router isis** command removes the IS-IS instance and all related configuration settings, which is different from deleting the last NET. Deleting the last NET disables the IS-IS instance while preserving all configuration information. To reduce the risk, delete the last NET.

Examples

The following example configures the `ip-backbone` IS-IS instance and assigns it a NET of `47.001.002.002.00`:

```
[local]Redback(config-ctx)#router isis ip-backbone
[local]Redback(config-isis)#net 47.0001.0002.0002.0002.00
```

Related Commands

interface
net

set-overload-bit

set-overload-bit [**on-startup** *[interval]*] | **bgp-converge-delay** *[interval]* | **strict-bgp-tracking**
no set-overload-bit

Purpose

Sets the overload bit so that other devices do not use the SmartEdge router to forward traffic.

Command Mode

IS-IS router configuration

Syntax Description

on-startup	Optional. Sets the overload bit on startup, and continues until the timer expires.
<i>interval</i>	Optional. Timer interval in seconds. The range of values is 10 to 3,600 seconds; the default value is 210 seconds.
bgp-converge-delay	Optional. Sets the overload bit on startup, and continues until timer expires or the Border Gateway Protocol (BGP) converges. The overload bit is removed as soon as BGP converges.
strict-bgp-tracking	Optional. Sets the overload bit until BGP converges. If BGP is not converged or not running, the overload bit remains set. There is no time out for the overload bit as long as BGP is not converged.

Default

The overload bit is not set.

Usage Guidelines

Use the **set-overload-bit** command to set the overload bit so that other devices do not use the SmartEdge router to forward traffic. The other routers in the domain can still forward traffic to IP networks directly connected to this router.

The overload bit is designed by the IS-IS protocol to indicate a router overload condition, such as memory shortage; however, this overload bit can be manually set or dynamically set for other network conditions. For example, when a router resides in a web server location, it may only want to attract traffic destined to the web servers, and not attract general traffic headed to other routers. When BGP is running on the router, and if it is not fully converged, the router may not have all the routing information for transit traffic.

Use the **set-overload-bit** command without any option to indefinitely set the overload bit. This is suitable for the web server location example above.

Use the **on-startup** keyword if BGP is not configured on the router, or if BGP convergence is not an issue. When the router starts, IS-IS temporarily sets the overload bit to allow the router to reach full functionality with complete routing information on the router.

Use the **bgp-converge-delay** keyword if BGP is not fully converged, and you want to use the IS-IS overload bit feature to delay other routers from sending transit traffic through the router until BGP converges. If the BGP converge delay time expires, the overload bit is removed, even if BGP has not converged; therefore, you should adjust the BGP converge delay time so that it is appropriate to your network size and the amount information in the BGP routing table.

Use the **strict-bgp-tracking** keyword if BGP is not fully converged, and you want to use the overload bit feature to stop other routers from sending transit traffic through the router to until BGP converges. The overload bit is removed only when full BGP convergence is reached.

Use the **no** form of this command to remove the overload bit.

Examples

The following example enables ISIS to use the overload bit to delay transit traffic for 60 seconds:

```
[local]Redback(config-ctx)#router isis test  
[local]Redback(config-isis)#set -overload-bit bgp-converge-delay 60
```

Related Commands

maximum update-delay
stub-router

spf holddown

spf holddown *interval* [**level-1** | **level-2**]

no spf holddown

Purpose

Modifies the delay time between an event that triggers a Shortest Path First (SPF) calculation and the calculation itself.

Command Mode

IS-IS router configuration

Syntax Description

<i>interval</i>	Delay interval, in seconds, between the trigger event and the SPF computation. The range of values is 1 through 120; the default value is 5.
level-1	Optional. Sets the holddown for level 1 routes independently.
level-2	Optional. Sets the holddown for level 2 routes independently.

Default

The SPF holddown is five seconds. When you enter this command without specifying level 1 or level 2 routing, SPF holddown value is the same for both level 1 and level 2.

Usage Guidelines

Use the **spf holddown** command to modify the delay time between an event that triggers an SPF calculation and the calculation itself. The purpose of that delay is to capitalize on the fact that computation triggers, such as new link-state protocol data units (LSPs), tend to occur in bursts. Starting the computation after the first event would cause another computation to be scheduled immediately after that due to further events.

Because SPF calculations are performed when the topology changes, increasing this value offloads the processor, especially in large topologies, but slows down the convergence of the network.

Use the **no** form of this command to restore the default delay value.

Examples

The following example sets the delay between the event that triggers an SPF calculation and the calculation itself to 20 seconds for **level-1** routing:

```
[local]Redback(config-ctx)#router isis isis1
[local]Redback(config-isis)#spf holddown 20 level-1
```

Related Commands

spf interval

spf interval

spf interval *seconds* [**level-1** | **level-2**]

no spf interval

Purpose

Configures the minimum interval between Shortest Path First (SPF) calculations.

Command Mode

IS-IS router configuration

Syntax Description

<i>seconds</i>	Minimum amount of time, in seconds, between SPF calculations. The range of values is 1 to 120; the default value is 10.
level-1	Optional. Sets the interval for level 1 routes independently.
level-2	Optional. Sets the interval for level 2 routes independently.

Default

The SPF interval is 10 seconds. When you enter this command without specifying level 1 or level 2 routing, the same SPF interval is used for both levels.

Usage Guidelines

Use the **spf interval** command to configure the minimum interval between SPF calculations.

Because SPF calculations are performed when the topology changes, increasing this value offloads the processor, especially in large topologies, but slows down the convergence of the network.

Use the **no** form of this command to restore the default SPF interval.

Examples

The following example sets the minimum time between SPF calculations to 25 seconds:

```
[local]Redback(config-ctx)#router isis isis1  
[local]Redback(config-isis)#spf interval 25
```

Related Commands

None

summary-address

summary-address *ip-addr* {*netmask* | /*prefix-length*} [**level-1** | **level-2**]

no summary-address *ip-addr* {*netmask* | /*prefix-length*} [**level-1** | **level-2**]

Purpose

Provides IP route aggregation during the processes of route leaking and route redistribution.

Command Mode

IS-IS address family configuration

Syntax Description

<i>ip-addr</i>	IP address of the route.
<i>netmask</i>	Network mask in the form <i>A.B.C.D</i> .
<i>prefix-length</i>	Prefix length. The range of values is 0 to 32.
level-1	Optional. Sets IP route aggregation for level 1 routes independently.
level-2	Optional. Sets IP route aggregation for level 2 routes independently.

Default

No route aggregation is applied. When you enter this command without specifying the IS-IS level, a summary address is only applied to an IS-IS level 2 domain.

Usage Guidelines

Use the **summary-address** command to provide IP route aggregation during the processes of route leaking and route redistribution.

Note Currently, this command is only available for address family IPv4 unicast.

A summary address is active if one or multiple more-specific routes are found during route leaking, redistribution, or both. Otherwise, the summary address is nonactive, and all IP addresses are included in the local link-state protocol data units (LSPs). If the summary address is active, all more-specific addresses in the summary range are suppressed during the local LSP generation. The metric of the summary address is equal to the lowest metric of all more-specific routes. A black hole is installed for an active summary address.

Use the **no** form of this command to remove the route aggregation from the configuration.

Examples

The following example suppresses all more-specific level 2 routes that match the 10.0.0.0 255.0.0.0 constraint:

```
[local]Redback(config-ctx)#router isis isis1  
[local]Redback(config-isis)#address-family ipv4 unicast  
[local]Redback(config-isis)#summary-address 10.0.0.0 255.0.0.0
```

Related Commands

address-family
interarea-distribute
redistribute

traffic-engineering

traffic-engineering [*level-1* | *level-2* | *level-1-2*]

no traffic-engineering

Purpose

Enables Multiprotocol Label Switching (MPLS) traffic engineering within Intermediate System-to-Intermediate System (IS-IS) routing.

Command Mode

IS-IS router configuration

Syntax Description

<i>level-1</i>	Optional. Traffic engineering for IS-IS level 1 routing only.
<i>level-2</i>	Optional. Traffic engineering for IS-IS level 2 routing only.
<i>level-1-2</i>	Optional. Traffic engineering for IS-IS both routing levels.

Default

MPLS traffic engineering is disabled.

Usage Guidelines

Use the **traffic-engineering** command to enable MPLS traffic engineering within IS-IS routing. Enabling traffic engineering allows IS-IS link-state protocol data units (LSPs) to carry traffic engineering information on IS-IS interfaces. Traffic engineering information includes link IP addresses, link bandwidth and link administrative colors.

Traffic engineering can be enabled on either IS-IS level 1, level 2, or both level 1 and level 2 routing.

Note Resource Reservation Protocol (RSVP) must be configured on the interface for IS-IS traffic engineering information to be included in its LSP for the link.

Note An IS-IS metric style of wide or transition must be used for traffic engineering to take effect.

Note The global **router-id** command in context configuration mode must be configured for the IS-IS LSP to carry the specified IP address of the router ID interface.

Use the **show isis database extensive** command to see the traffic engineering information for the IS-IS link in the LSPs, and the **show isis interface detail** to see if the interface has traffic engineering information for the routing level.

Use the **no** form of this command to disable MPLS traffic engineering within IS-IS routing.

Examples

The following example displays that IS-IS traffic engineering is enabled for IS-IS level-2 routing:

```
[local]Redback(config-ctx)#router isis ip-backbone  
[local]Redback(config-isis)#traffic-engineering level-2
```

Related Commands

router-id
router rsvp
metric-style

IP Multicast Configuration

This chapter provides an overview of IP multicast, and describes the tasks and commands used to configure IP multicast features through the SmartEdge® OS.

For information about the tasks and commands used to monitor, troubleshoot, and administer IP multicast, see the “IP Multicast Operations” chapter in the *Routing Protocols Operations Guide* for the SmartEdge OS.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples
- Command Descriptions

Overview

There are three basic types of IP communication: unicast, broadcast, and multicast. Unicast communication occurs between a source host and a single, unique destination host; it is one-to-one communication. Unicast packet headers specify a single IP address of a destination host. Broadcast communication occurs between a source host and all other hosts on the network; it is one-to-all communication. Broadcast packet headers specify an IP broadcast address that includes all destination hosts on the subnet. Multicast communication, by contrast, falls somewhere between unicast and broadcast communication.

Multicast communication enables a source host to send IP packets to any number of hosts, anywhere within an IP network; it is one-to-any communication. That is, multicast communication is not limited to sending packets to a single destination host, or sending packets to every host on the network. Instead, multicast enables a source host to send IP packets to as many destination hosts as necessary, but no more than that. The advantages of multicast communication, unlike broadcast communication, which floods the network with unnecessary traffic, is that a source host can communicate with more than one destination host without sending traffic to every host on the network. This results in an economic use of bandwidth.

The main challenge for multicast communication is developing a method for determining which hosts will receive multicast traffic, and which hosts will not receive the traffic. Several different multicast protocols have been developed, each with its own unique approach to addressing the multicast challenge. The SmartEdge OS supports the following multicast protocols:

- Internet Group Management Protocol
- Protocol Independent Multicast
- Source-Specific Multicast
- Multicast Source Discovery Protocol
- Anycast RP
- Multicast VPNs
- Remote Multicast Replication

Internet Group Management Protocol

Internet Group Management Protocol (IGMP) is the method by which local hosts join a multicast group. A host that wants to join a multicast group should immediately transmit an unsolicited Membership Report for that group to the multicast-enabled router for that network. The router maintains a list of multicast group memberships for each attached network, and a timer for each membership. The designated router (DR), which is the multicast-enabled router with the highest IP address on the network, periodically sends a general query to learn which groups have members on an attached network, and a group-specific query to learn if a particular group has any members on an attached network.

The following sections describe additional IGMP-related features:

- IGMP Bandwidth Limitation
- IGMP Membership Tracking

IGMP Bandwidth Limitation

The IGMP bandwidth limitation feature is targeted at applications where many potential receivers share the same port. When too many receivers join at the same time, the aggregate bandwidth exceeds that of the physical port, resulting in unacceptable service. The loss of packets is more visible for video and audio types of applications, in the form of interruptions, than for unicast Transmission Control Protocol (TCP) applications, where the sender backs off and retransmits. With this feature, you can decide when to reject new IGMP joins, and you can set priorities among receivers.

IGMP Membership Tracking

The IGMP membership tracking feature allows explicit tracking of group membership for all multicast hosts in a multiaccess network. Because it allows the instant-leave feature to work on a multiaccess network, membership tracking enables much lower leave latency and faster channel surfing.

Membership tracking, which is enabled by default, works with IGMP Version 2 (IGMPv2) and IGMP Version 3 (IGMPv3). The following sections describe how membership tracking works within each IGMP version:

- Membership Tracking with IGMPv2

- Membership Tracking with IGMPv3

Membership Tracking with IGMPv2

When a host running IGMPv2 joins a group, it sends a membership report to the router. The router adds the host's IP address to the group membership list, which enables the router to track which hosts are members of a particular group on the same multiaccess network. When a host sends an IGMPv2 Leave message, it is removed from the group membership list.

Membership Tracking with IGMPv3

When a host running IGMPv3 joins a group from a source list, it sends a membership report for a group and source as the include source list. The router adds the host's IP address to the list of interested members for all the sources in the source list. When a host removes a source from its source list, the router removes the host from the group's source record, and if the host was the last interested host for that source, and the circuit is configured with instant-leave, the router performs an instant-leave operation for the source record.

Protocol Independent Multicast

Protocol Independent Multicast (PIM) is a multicast routing protocol that runs over an existing unicast infrastructure. As its name implies, PIM is IP routing protocol-independent; that is, regardless of the unicast routing protocol used to populate the unicast routing tables, PIM uses those tables to perform multicast forwarding tasks. PIM also relies on IGMP to provide and maintain all multicast group membership information.

There are two implementations of PIM:

- Protocol Independent Multicast-Dense Mode
- Protocol Independent Multicast-Sparse Mode

Protocol Independent Multicast-Dense Mode

Protocol Independent Multicast-Dense Mode (PIM-DM) uses source distribution, or shortest-path trees (SPTs), to distribute multicast traffic to receivers in the network. PIM-DM uses Hello messages to establish neighbor adjacencies, and builds an initial SPT based on the neighbor adjacencies. The initial form of the SPT is also referred to as a broadcast tree, because PIM routers use it to distribute multicast traffic in a broadcast-like manner; that is, multicast traffic is sent to all PIM-DM routers, regardless of whether they want to receive the traffic.

After the initial flood of multicast traffic in a PIM-DM network down a broadcast tree, the tree is trimmed back to a minimum spanning tree. PIM-DM routers send Prune messages to remove themselves from the SPT if they meet any of the following conditions:

- The PIM router is a leaf router and has no directly connected receivers.
- The PIM router is a non-leaf router on a point-to-point link and receives a Prune message from its neighbor.
- The PIM router is a non-leaf router on a LAN segment with no directly connected receivers, has received a Prune message from a neighbor on a LAN segment, and no other neighbor on the LAN segment overrides the Prune message.

Prune messages can be overridden by Join messages sent by downstream neighbors that want to continue, or begin, receiving multicast traffic on the specified SPT. Pruned branches are restored periodically to see if new multicast group members have joined since the branch was pruned.

The PIM-DM flooding and pruning mechanism is optimal only for densely populated groups.

Protocol Independent Multicast-Sparse Mode

Protocol Independent Multicast-Sparse Mode (PIM-SM) differs from PIM-DM in the following ways:

- Routers with directly attached multicast receivers, or downstream receivers, are required to join a sparse mode distribution tree by transmitting explicit join messages. If a router does not become part of the distribution tree, it does not receive multicast traffic.
- PIM-SM uses a rendezvous point (RP) to serve as a distribution point for multicast traffic from one or more related multicast sources.

PIM-SM sends multicast traffic only to locations on the network that explicitly request membership to a multicast group. The requests are called PIM Join messages, which are sent hop-by-hop towards the multicast source, creating an SPT. As the PIM Join message is sent up the tree, routers along the path establish the multicast forwarding state so that multicast traffic can be sent back down the path. Likewise, PIM Prune messages can be sent hop-by-hop towards the multicast source to remove locations from the multicast group.

On a PIM-SM network, SPTs are trees created by a collection of joins where the root of the tree is also the multicast source; however, the root of an SPT does not need to be the multicast source, but can be a location called the rendezvous point. SPTs with an RP as its source are called shared trees. With a shared tree, multiple multicast sources share the same tree structure by forwarding their multicast traffic to the RP where it is then distributed down the shared tree.

Any router on a network can be specified as the RP, or multiple routers can be specified as candidate RPs (C-RPs). In the case of C-RPs, an RP election process determines which router serves as the RP. The bootstrap router (BSR) eliminates the need to manually configure each router on the network with the RP information by distributing group-to-RP mapping information to all routers on the network. During the RP election process, all C-RPs send their candidacy advertisements to the BSR, and the BSR distributes the group-to-RP mappings.

For purposes of redundancy, multiple candidate BSRs (C-BSRs) can be specified. A BSR election process, based on the routers priority level, determines which C-BSR serves as the BSR.

Note The PIM-SM explicit join mechanism is optimal only for sparsely populated groups.

Source-Specific Multicast

The source-specific multicast (SSM) feature is an extension of multicast routing where traffic is forwarded to receivers from only those multicast sources to which the receivers have explicitly joined. For multicast groups configured to use SSM, only source-specific multicast distribution trees are created, and not shared trees.

The PIM-SSM routing protocol supports the implementation of SSM and is derived from PIM-SM. SSM is supported by IGMPv3.

The address range 232.0.0.0 through 232.255.255.255 is reserved for SSM applications and protocols. Existing IP multicast receivers cannot receive traffic when trying to use addresses in a defined SSM range, unless they are SSM enabled.

For more information on SSM routing, see the Internet Draft, *Source-Specific Multicast for IP*, draft-ietf-ssm-arch-00.txt.

Multicast Source Discovery Protocol

The Multicast Source Discovery Protocol (MSDP) is the method used to link interdomain RPs so that multicast messages can be forwarded to other domains that have active group membership. RPs in a PIM domain know about all active sources in its own domain, but not other domains; however, if an RP from one domain is peered with another RP in a different domain, it can send source active messages from one domain to the other.

Using MSDP provides the following benefits:

- A multicast distribution tree can be divided into different segments.
- Local members for each segment can join their local segments.
- Each segment depends only on its own RP. Each RP has information about multicast sources of each domain, so members in each segment can stay in their local segment.
- Each domain does not have to globally send its member information.

Anycast RP

In a basic PIM-SM network, a single RP is used by all multicast sources and receivers. Anycast RP is a mechanism that provides RP redundancy and load-sharing capabilities by allowing the use of multiple RPs within a single multicast domain. Assuming that the sources are evenly spaced around the network, an equal number of sources register with each RP. That is, the process of registering the sources are shared equally by all the RPs in the network.

All routers acting as RPs must be configured with a loopback interface using the same anycast RP address. All downstream routers use that anycast RP address as IP address for their local RP. To facilitate communication between RPs, each router acting as an RP must also be configured with its own unique IP address, which is used only to send and receive messages from the other RPs.

When a source registers with one RP, a message is sent to the other RPs informing them that there is an active source for a particular multicast group. The result is that each RP knows about the active sources in the area of the other RPs. If any of the RPs were to fail, IP routing would converge and one of the RPs would become the active RP in more than one area. New sources would register with the backup RP. Receivers would join toward the new RP and connectivity would be maintained.

Our implementation of anycast RP eliminates the dependency on MSDP by removing MSDP peering between the anycast RPs; however, to advertise internal sources to routers outside of the routing domain, MSDP may still be required.

Multicast VPNs

Standard Border Gateway Protocol/Multiprotocol Label Switching Virtual Private Networks (BGP/MPLS VPNs) do not provide a way for IP multicast traffic to travel from one VPN site to another. Implementing multicast domain trees (MDTs) provides a scalable solution to support IP multicast over BGP/MPLS VPNs. Currently, MDTs support only IPv4 multicast.

When a network uses many VPNs, where each VPN can have many multicast groups, and each multicast group can have many multicast transmitters, it is not scalable to have one or more distribution trees for each multicast group. A scalable IP multicast solution for MPLS/BGP VPNs requires that the amount of VPN-specific information maintained by the P routers must be proportional only to the number of VPNs that run over the backbone. The amount of VPN-specific information in the P routers is not sensitive to the number of multicast groups or to the number of multicast transmitters within the VPNs. However, there is a trade off to using this scalable solution; nodes that are not on a path to a multicast receiver may still receive multicast packets, and will have to discard them. That is, greater scalability reduces multicast route optimization.

A multicast-enabled VPN has a corresponding multicast domain. A provider edge (PE) router that attaches to a multicast-enabled VPN belongs to the corresponding multicast domain. For each multicast domain, there is a default MDT through the backbone, connecting all of the PE routers that belong to that multicast domain. A PE router may be in as many multicast domains as there are VPNs attached to it. However, each multicast domain has its own MDT. The MDTs are created by running PIM in the backbone, and in general an MDT also includes P routers on the paths between the PE routers. For MDTs to work properly, the following conditions must be met:

- PIM must be the multicast routing protocol used in the VPN.
- PIM must be the multicast routing protocol used in the backbone network.
- The backbone network must support IP multicast forwarding.

Default MDTs are constructed automatically as the PE routers in the domain come up. Construction of a default MDT does not depend on the existence of multicast traffic in the domain. That is, it exists before any multicast traffic is detected.

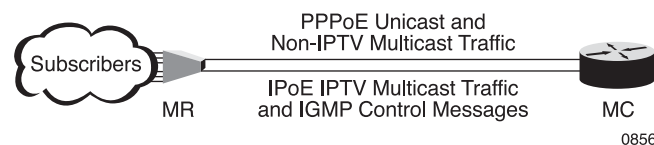
In a multicast-enabled VPN, each customer edge (CE) router has a PIM adjacency to a PE router, but CE routers at different sites do not have PIM adjacencies to each other. Multicast packets from within a VPN are received from a CE router by an ingress PE router. The ingress PE router encapsulates the multicast packets and forwards them across the default MDT to all PE routers connected to the specified VPN. If a PE router receiving the multicast packets is not on the path to any multicast receiver of that multicast group, it discards the multicast packet.

For the SmartEdge implementation of multicast VPNs, the default MDT group must be configured on an intercontext interface in a VPN-enabled context. This interface is similar to a loopback interface in that it is not bound to anything and does not need an IP address. It creates an intercontext circuit between the VPN-enabled context and the local context. PIM-SM must also be configured on this intercontext interface. The MDT encapsulation type must be configured on a loopback interface in the local context. The loopback interface is used to source multicast packets on the MDT.

Remote Multicast Replication

Remote multicast replication (RMR) is used to enable multicast services. It requires a multicast controller (MC), such as the SmartEdge router, to maintain complete subscriber awareness and subscriber control for all traffic types, and a multicast replicator (MR), such as a digital subscriber line access multiplexer (DSLAM), to perform IGMP snooping (examining the information in IGMP packets) and multicast traffic replication on the subscriber-side link. Figure 11-1 shows the network topology used for RMR.

Figure 11-1 Remote Multicast Replication Network Topology



The IP over Ethernet (IPoE) circuit is configured to carry the multicast traffic and IGMP control messages between the MC and the MR. The MC starts forwarding a multicast stream upon receiving the first IGMP join message, and stops forwarding the stream upon receiving the last IGMP leave message. The MR replicates the incoming multicast stream to all subscribers that need a copy of that stream, thus reducing the bandwidth usage on the IPoE circuit. The MR makes its multicast replication and forwarding decisions by snooping the IGMP join and leave messages from the subscriber.

The MR performs multicast replication, but it does not support any routing functions, user authentication, or billing functions. These functions are supported by the MC (SmartEdge router) via a Point-to-Point Protocol over Ethernet (PPPoE) circuit from the subscriber to the MC.

A single MC can support multiple MRs. When multiple MRs are used, the MC performs per-MR multicast replication, while each MR performs per-subscriber multicast replication.

To configure RMR on the SmartEdge router, an interface must be enabled to forward the multicast data and IGMP control messages, and an IGMP service profile must be enabled to forward multicast data for IGMP messages received on the PPPoE circuit on the IPoE interface.

Configuration Tasks

Note In this section, the command syntax in the task tables displays only the root command; for the complete command syntax, see the full description for the command in the “Command Descriptions” section.

To configure IP multicast, perform the tasks described in the following sections:

- Configuring IGMP
- Configuring an IGMP Service Profile
- Configuring PIM-DM
- Configuring PIM-SM
- Configuring MSDP
- Configuring an MSDP Peer

- Configuring Multicast for Subscribers
- Enabling PIM Graceful Restart
- Enabling SSM
- Enabling Multicast VPNs
- Enabling RMR

Configuring IGMP

To configure IGMP, perform the tasks described in Table 11-1. Enter all commands in interface configuration mode, unless otherwise noted.

Table 11-1 Configure IGMP

Task	Root Command	Notes
Configure a router to join a multicast group on the interface.	igmp join-group	
Configure IGMP membership on an interface.	igmp access-group	Only multicast groups permitted by the access control list (ACL) are accepted on the interface.
Configure the interval at which the router sends IGMP group-specific host query messages.	igmp last-member-query-interval	
Configure the interval at which the router sends IGMP host query messages.	igmp query-interval	
Configure the maximum response time specified in IGMP queries.	igmp query-max-response-time	
Configure the IGMP robustness variable.	igmp robust	The group membership interval, other query present interval, startup query count, and last member query count are all determined by the robustness variable.
Configure the interface to operate in either IGMP Version 1, Version 2, or Version 3 mode.	igmp version	
Configure the recommended bandwidth required by each of the specified groups.	igmp group-bandwidth	This is only a recommendation. Before configuring the recommended group bandwidth, you should know the rate at which senders send on each group. You can use inbound rate limiting to ensure that the groups' recommended bandwidth is not exceeded.
Configure the total maximum bandwidth allowed for multicast data traffic on a port or channel.	igmp maximum-bandwidth	If the addition of a new group would cause the bandwidth usage on this port to exceed the maximum bandwidth, and if a subscriber with a lower priority exists on this port, the lower priority group is dropped to reclaim the bandwidth; otherwise, the new group is dropped.
Ensure that all mtrace queries are received within the administratively scoped domain of the router.	igmp mtrace-prohibit	

Table 11-1 Configure IGMP (*continued*)

Task	Root Command	Notes
Configure an IGMP service profile.		For the complete list of tasks used to configure an IGMP service profile, see the "Configuring an IGMP Service Profile" section.
Enable the specified IGMP service profile on the interface.	igmp service-profile	

Configuring an IGMP Service Profile

To configure an IGMP service profile, perform the tasks described in Table 11-2. Enter all commands in IGMP service profile configuration mode, unless otherwise noted.

Table 11-2 Configure a Service Profile

Task	Root Command	Notes
Create a service profile, and access IGMP service profile configuration mode.	igmp service-profile	Enter this command in context configuration mode.
Enable Instant Leave on the interface.	instant-leave	Instant Leave allows IGMP to perform a 0-delay leave upon receiving an IGMPv2 leave message. If the router is an IGMP querier, it sends an IGMP last member query with a 100 ms last member query response time; however, the router does not wait for 100 ms before it prunes off the group. This allows channel surfing applications to function better.
Configure the maximum number of IGMP-joined groups allowed per interface.	max-groups	If the addition of a new group on a circuit causes the total number of joined groups to exceed the maximum number allowed, one of the following actions is taken: - If the drop-old keyword is specified for the service profile, the oldest IGMP group on the circuit is dropped and the new IGMP report accepted. - If the drop-old keyword is not specified for the service profile, the new IGMP membership report is dropped.
Enable the forwarding of multicast data for IGMP messages received on the PPPoE subscriber circuits on an out-of-band (separated from the PPPoE circuit) IPoE interface.	multicast destination	The IGMP service profile must be bound to a subscriber record through a configuration or a Remote Authentication Dial-In User Service (RADIUS) attribute. For the multicast destination command to work properly, the out-of-band IPoE interface on which the multicast data is to be forwarded must be multicast-enabled; use the multicast output command (in interface configuration mode) to enable the out-of-band IPoE interface to forward multicast data.
Configure the priority of the interface when the maximum bandwidth in the service profile has been exhausted.	priority	When the addition of a new group would cause the maximum bandwidth, as specified by the igmp maximum-bandwidth command, to be exceeded on the port, the router searches for subscribers joined on the same port with a lower priority. The router drops the lower priority subscribers and reclaims their bandwidth until it gets enough bandwidth to service the higher priority subscriber. If it cannot reclaim enough bandwidth the new group join will be dropped.

Table 11-2 Configure a Service Profile (*continued*)

Task	Root Command	Notes
Creates a static multicast route, (*,G) or (S,G), with a subscriber circuit as the outgoing interface (OIF).	static-group	PIM normally creates dynamic multicast routes; the static-group command allows you to create static multicast routes. An OIF is an outgoing circuit that receives traffic destined for a given multicast group. When you configure the static multicast route in IGMP service profile configuration mode, the OIF is a subscriber circuit. To configure all subscriber circuits on a multibind interface to receive multicast traffic for a specified multicast group, configure the static-group command in an IGMP service profile that is bound to a subscriber (default) profile.
Enable IGMP groups to be sticky.	sticky-groups	Groups defined by the ACL will never be dropped, unless an explicit leave for that group is received.

Configuring PIM-DM

To configure PIM-DM, perform the tasks described in Table 11-3. Enter the command in interface configuration mode.

Table 11-3 Configure PIM-DM

Task	Root Command	Notes
Enable PIM-DM on an interface.	pim dense-mode	

Configuring PIM-SM

To configure PIM-SM, perform the tasks described in Table 11-4. Enter all commands in interface configuration mode, unless otherwise noted.

Table 11-4 Configure PIM-SM

Task	Root Command	Notes
Enable PIM-SM on an interface.	pim sparse-mode	
Configure an administratively scoped boundary for multicast routing.	ip multicast boundary	An administratively scoped boundary prevents forwarding of multicast data packet destined for group addresses denied by the ACL.
Accept or reject an IP address as being a valid RP address for a specific multicast group.	pim accept-rp	Enter this command in context configuration mode. To determine if the RP should be accepted, the router checks the group-to-RP mapping cache for a matching entry for the group. If there is a matching entry, and the acl-name argument is specified, the router compares the RP address to the ACL to determine if the filter permits the RP address.
Enable anycast RP functionality on a PIM-SM router.	pim anycast-rp	Enter this command in context configuration mode.
Configure the router to neither send nor receive BSR messages.	pim bsr-border	This command should be configured on routers that connect to bordering PIM domains to create a PIM domain boundary that blocks the flow Protocol Independent Multicast Version 2 (PIMv2) BSR messages across the domain border.

Table 11-4 Configure PIM-SM (*continued*)

Task	Root Command	Notes
Configure a router to begin serving as a C-BSR, and participate in the BSR election process.	pim bsr-candidate	Enter this command in context configuration mode. If this router wins the BSR election, all candidate RPs will advertise their candidacy to this router. The BSR caches and advertises the RP sets via the PIM bootstrap messages to the entire PIM domain.
Specify the election priority value for a DR.	pim dr-priority	
Set the PIMv2 Hello interval.	pim hello-interval	
Filter PIM messages from neighbors.	pim neighbor-filter	
Set the protocol parameters to be compatible with PIM-SM specifications, or to be compatible with legacy implementations, such as traditional Cisco implementations.	pim operation-mode	
Configure a router with the RP address for all multicast group addresses permitted by an ACL.	pim rp-address	Enter this command in context configuration mode. The pim rp-address command is usually used on very simple PIM-SM networks where the RP address is manually configured on each router in the network. More complicated networks should use PIMv2's Bootstrap Router feature which allows routers on a network to dynamically learn the RP address. If an ACL is not specified, this RP address is used for the entire multicast address space.
Configure a C-RP on an interface for group address ranges permitted by an ACL.	pim rp-candidate	Enter this command in context configuration mode. If an ACL is not specified, this RP address is used for the entire multicast address space.
Enable a PIM-SM leaf router to continue using a shared tree, instead of switching to an SPT.	pim spt-threshold infinity	Enter this command in context configuration mode.
Create a static multicast route, (*,G) or (S,G), with the specified interface as the outgoing interface (OIF).	pim static group	Enter this command in context configuration mode. PIM normally creates dynamic multicast routes; the pim static group command allows you to create static multicast routes. An OIF is an outgoing circuit that receives traffic destined for a given multicast group. For this command, the OIF is a regular interface. For multibind interface OIFs, configure the static-group command in an IGMP service profile that is bound to a subscriber (default) profile.

Configuring MSDP

To configure MSDP, perform the tasks described in Table 11-5. Enter all commands in MSDP router configuration mode, unless otherwise noted.

Table 11-5 Configure MSDP

Task	Root Command	Notes
Enable MSDP within a context, and access MSDP router configuration mode.	router mspf	Enter this command in context configuration mode.
Configure a default peer from which to accept all MSDP SA messages.	default-peer	A default peer is needed in topologies where MSDP peers do not co-exist with BGP peers. In such a case the reverse path forwarding (RPF) check on SAs may fail, and no SAs will be accepted. In these cases you can configure the peer as a default peer, and bypass RPF checks. An MSDP peer must already be configured before it can be made a default peer.
Configure an MSDP peer to be a member of a mesh group.	mesh-group	The MSDP mesh group is a mechanism to reduce SA flooding. Peers in the same mesh group will not forward SA messages to each other. The originator will send the SAs to all its peers.
Configure an interface as the originating RP address.	originating-rp	The IP address of the interface is used as the RP address in all SAs originated by the router.
Configure an ACL to filter incoming SA messages learned from the local RP.	originating-rp sa-filter	
Configure an MSDP peer.	For the complete list of tasks used to configure an MSDP peer, see the "Configuring an MSDP Peer" section.	

Configuring an MSDP Peer

To configure an MSDP peer, perform the tasks described in Table 11-6. Enter all commands in MSDP peer configuration mode, unless otherwise noted.

Table 11-6 Configure an MSDP Peer

Task	Root Command	Notes
Create an MSDP peer and enter MSDP peer configuration mode for peer-specific configurations.	peer	Enter this command in MSDP router configuration mode. The no shutdown command is enabled by default after you configure an MSDP peer.
Associate a text description with an MSDP peer.	description	
Configure a peer's autonomous system (AS) number.	peer-as	
Configure an ACL to filter SA messages coming from another peer.	sa-filter	Use the following command syntax: sa-filter in acl-name
Configure an ACL to filter SA messages going to another peer.	sa-filter	Use the following command syntax: sa-filter out acl-name
Disable a configured MSDP peer.	shutdown	

Configuring Multicast for Subscribers

To configure multicast for subscribers, perform the tasks described in Table 11-7. Enter all commands in subscriber configuration mode.

Table 11-7 Configure Multicast for Subscribers

Task	Root Command	Notes
Enable an existing IGMP service profile on a single subscriber record, a named subscriber profile, or a default subscriber profile.	igmp service-profile	The service profile used is determined in the following order: - Subscriber profile - Default subscriber profile - Service profile configured on the subscriber's parent interface If a service profile is not defined in the subscriber record, it inherits the service profile from the default subscriber profile. If the default subscriber profile is not configured with an service profile, the service profile configured on the interface is used.
Configure the multicast receive permissions for a subscriber record or for the default subscriber record.	ip multicast receive	Permission attributes are applied in the following order: - Subscriber record - Default subscriber record - System defaults If a permission is not defined in the subscriber record, it inherits the value of the permission from the default subscriber record. If the permission is not defined in the default subscriber record, the system default values are used. For multicast routing to function on subscribers, you must use the pim sparse-mode command in interface configuration mode to enable PIM-SM on the interface.
Configure the multicast send permissions for a subscriber record or for the default subscriber record.	ip multicast send	If the permit keyword is used without the unsolicit keyword, the subscriber must join a group prior to sending unsolicited multicast data. If used together (permit unsolicit), a subscriber is allowed to send unsolicited multicast traffic. Permissions are examined in the following order: - Subscriber record - Default subscriber record - System defaults. If a permission is not defined in the subscriber record, it inherits the value of the permission from the default subscriber record. If the permission is undefined in the default subscriber record, the system default values are used. For multicast routing to function on subscribers, you must use the pim sparse-mode command in interface configuration mode to enable PIM-SM on the interface.

Enabling PIM Graceful Restart

PIM graceful restart allows the SmartEdge router and its neighbors to continue forwarding multicast packets without disrupting network traffic. Because neighboring routers assist, the SmartEdge router can quickly restart the PIM process without having to recalculate algorithms from scratch. To enable PIM graceful restart, perform the task described in Table 11-8. Enter the command in context configuration mode.

Table 11-8 Enable PIM Graceful Restart

Task	Root Command	Notes
Enable PIM graceful restart on the specified context.	pim graceful-restart	

Enabling SSM

To enable SSM, perform the task described in Table 11-9. Enter the command in context configuration mode.

Table 11-9 Enable SSM

Task	Root Command	Notes
Enable SSM routing on the specified context.	pim ssm	

Enabling Multicast VPNs

Multicast VPNs use MDTs on PE routers to support IP multicast over BGP/MPLS VPNs. To enable multicast VPNs, perform the task described in Table 11-10. Enter both commands in interface configuration mode.

Table 11-10 Enable Multicast VPNs

Task	Root Command	Notes
Specify the default MDT group.	mdt default-group	Configure this command on an intercontext interface in a VPN-enabled context. This interface is similar to a loopback interface in that it is not bound to anything and does not need an IP address. It creates an intercontext circuit between the VPN-enabled context and the local context. PIM-SM must also be configured on this intercontext interface.
Specify the multicast MDT encapsulation type.	mdt encapsulation	Configure this command on a loopback interface in the local context. The loopback interface is used to source multicast packets on the MDT.

Enabling RMR

Remote multicast replication (RMR) is used to enable IP multicast services. To enable RMR, perform the task described in Table 11-11.

Table 11-11 Enable Multicast VPNs

Task	Root Command	Notes
Enable an interface to forward multicast data, and to send and receive IGMP control messages.	multicast output	Enter this command in interface configuration mode. An IP over Ethernet (IPoE) circuit, on a Gigabit Ethernet port or an 802.1Q permanent virtual circuit (PVC) configured on it, must be configured on the interface to carry the multicast services. The MAC addresses received from IGMP control packets on the IPoE circuit are compared to the subscriber's MAC address received on the corresponding PPPoE circuit. By default, if the MAC addresses do not match, the IGMP control packet is dropped. Use the accept-unknown-mac keyword to accept IGMP control packets that have MAC addresses that do not match the subscriber's MAC address.
Enable the forwarding of multicast data for IGMP messages received on the PPPoE subscriber circuits on an out-of-band (separated from the PPPoE circuit) IPoE interface.	multicast destination	Enter this command in IGMP service profile configuration mode. The IGMP service profile must be bound to a subscriber record through a configuration or a Remote Authentication Dial-In User Service (RADIUS) attribute.

Configuration Examples

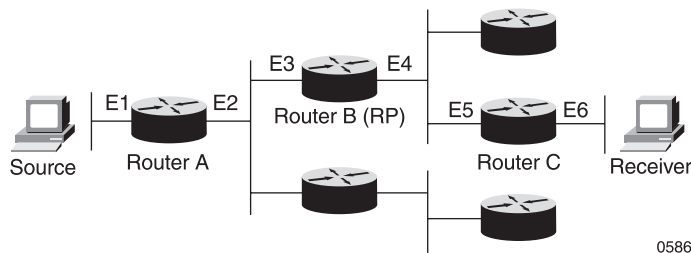
This section provides IP multicast configuration examples in the following sections:

- PIM-SM
- MSDP for Two PIM-SM Domains
- Multicast VPNs
- Remote Multicast Replication
- Anycast RP

PIM-SM

The following example demonstrates how three routers (Router A, Router B, and Router C) are configured to correctly operate on a PIM-SM local network. Figure 11-2 shows the simple PIM-SM network topology used for the configuration example.

Figure 11-2 Simple PIM-SM Network Topology



Router A is directly connected to the source, and Router C is directly connected to the receiver. Because Router A is the only router directly connected to the source, it serves as a PIM DR for the network. If multiple routers were connected to the source, the router with the highest IP address would be selected as the PIM DR.

The **pim sparse-mode** interface configuration mode command enables PIM-SM on the interface. The **pim rp-address** global configuration mode command enables all routers in the PIM-SM network to statically configure Router B as the rendezvous point (RP). An ACL can be specified with the **rp-addr** argument to permit multicast traffic for a particular group with this RP.

Enabling PIM-SM on an interface also enables IGMP on the same interface. For each local network, an IGMP querier is selected; for example, Router C is the IGMP querier for the local network connected to the receiver. If multiple routers were connected directly to the receiver, the router with the lowest IP address serves as the IGMP querier. The IGMP querier is responsible for sending IGMP host-query messages to all hosts on the local network.

Router A, which is directly connected to the source and the DR for its local network, sends PIM register messages on behalf of the source to the RP. Router C, on behalf of the receiver, sends PIM join and prune messages to the RP to advertise the group membership.

The configuration for Router A is as follows:

```
[local]RouterA#config
[local]RouterA(config)#context local
[local]RouterA(config-ctx)#interface E1
[local]RouterA(config-if)#ip address 10.2.1.1/24
[local]RouterA(config-if)#pim sparse-mode
[local]RouterA(config-if)#exit
[local]RouterA(config-ctx)#interface E2
[local]RouterA(config-if)#ip address 11.1.1.1/24
[local]RouterA(config-if)#pim sparse-mode
[local]RouterA(config-if)#exit
[local]RouterA(config-ctx)#ip access-list 1
[local]RouterA(config-access-list)#seq 10 permit 224.0.0.0 15.255.255.255
[local]RouterA(config-access-list)#exit
[local]RouterA(config-ctx)#pim rp-address 10.2.1.2 1
```

The configuration for RouterB (RP) is as follows:

```
[local]RouterB#config
[local]RouterB(config)#context local
[local]RouterB(config-ctx)#interface E3
[local]RouterB(config-if)#ip address 10.2.1.2/24
[local]RouterB(config-if)#pim sparse-mode
[local]RouterB(config-if)#exit
[local]RouterB(config-ctx)#interface E4
[local]RouterB(config-if)#ip address 10.4.1.1/24
[local]RouterB(config-if)#pim sparse-mode
[local]RouterB(config-if)#exit
[local]RouterB(config-ctx)#ip access-list 1
[local]RouterB(config-access-list)#seq 10 permit 224.0.0.0 15.255.255.255
[local]RouterB(config-access-list)#exit
[local]RouterB(config-ctx)#pim rp-address 10.2.1.2 1
```

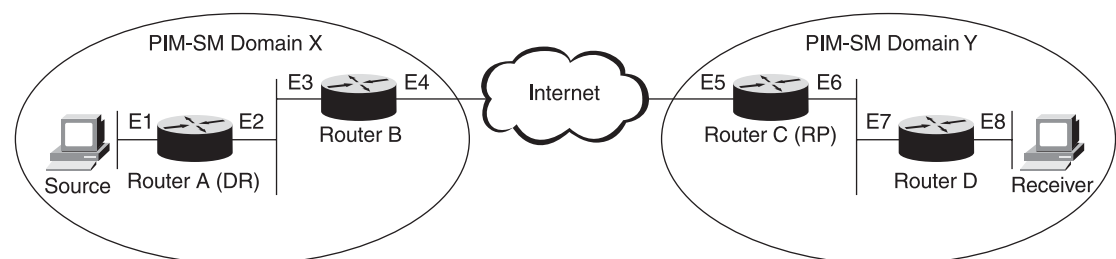
The configuration for RouterC (IGMP querier) is as follows:

```
[local]RouterC#config
[local]RouterC(config)#context local
[local]RouterC(config-ctx)#interface E5
[local]RouterC(config-if)#ip address 10.4.1.1/24
[local]RouterC(config-if)#pim sparse-mode
[local]RouterC(config-if)#exit
[local]RouterC(config-ctx)#interface E6
[local]RouterC(config-if)#ip address 44.1.1.1/24
[local]RouterC(config-if)#pim sparse-mode
[local]RouterC(config-if)#exit
[local]RouterC(config-ctx)#ip access-list 1
[local]RouterC(config-access-list)#seq 10 permit 224.0.0.0 15.255.255.255
[local]RouterC(config-access-list)#exit
[local]RouterC(config-ctx)#pim rp-address 10.2.1.2 1
```

MSDP for Two PIM-SM Domains

The following example demonstrates how to configure MSDP to link two PIM-SM domains, using MSDP, so that multicast messages can be forwarded from one domain to the other. Figure 11-3 shows the PIM-SM interdomain network topology used for the configuration example.

Figure 11-3 Interdomain PIM-SM Network Topology



0587

This example can be expanded to several PIM-SM domains. Each domain can use BGP for interdomain routing. MSDP is used for interdomain source discovery.

Each PIM-SM domain has one or more RPs that belong to the domain. MSDP allows RPs in different domains to share information about active sources. RPs know about the receivers in their local domain. Because RPs share information about the active sources in each domain, each RP can forward data accordingly if there is an active receiver in their local domain for a particular source.

For RPs to share information with each other, RPs are configured as MSDP peers. There can be multiple peers in between two RP MSDP peers. Each RP establishes an MSDP peering session with another RP in another domain.

To keep this configuration example simple, the following assumptions are made:

- The two domains, Domain X and Domain Y, are externally peered using MBGP, thus, Router B and Router C are external MBGP peers and MSDP peers.
- The two domains are different LAN segments.
- Static routing is being used instead of other Internet gateway protocols like Open Shortest Path First (OSPF), internal Border Gateway Protocol (iBGP), Intermediate System-to-Intermediate System (IS-IS), and so on.

The configuration for RouterA (DR) is as follows:

```
[local]RouterA#config
[local]RouterA(config)#context local
[local]RouterA(config-ctx)#interface lo1 loopback
[local]RouterA(config-if)#ip address 10.200.1.1/32
[local]RouterA(config-if)#pim sparse-mode
[local]RouterA(config-if)#exit
[local]RouterA(config-ctx)#interface E2
[local]RouterA(config-if)#ip address 102.1.1.1/24
[local]RouterA(config-if)#pim sparse-mode
[local]RouterA(config-if)#exit
[local]RouterA(config-ctx)#interface E4
[local]RouterA(config-if)#ip address 11.1.1.1/24
[local]RouterA(config-if)#pim sparse-mode
[local]RouterA(config-if)#exit
```

Static RP for Domain X configuration:

```
[local]RouterA(config-ctx)#pim rp-address 10.200.1.2
```

Static route configuration:

```
[local]RouterA(config-ctx)#ip route 10.200.1.2/32 102.1.1.2
```

The configuration for RouterB is as follows:

```
[local]RouterB#config
[local]RouterB(config)#context local
[local]RouterB(config-ctx)#interface lo1 loopback
[local]RouterB(config-if)#ip address 10.200.1.2/32
[local]RouterB(config-if)#pim sparse-mode
[local]RouterB(config-if)#exit
[local]RouterB(config-ctx)#interface E1
[local]RouterB(config-if)#ip address 102.1.1.2/24
[local]RouterB(config-if)#pim sparse-mode
[local]RouterB(config-if)#exit
[local]RouterB(config-ctx)#interface E2
```

```
[local]RouterB(config-if)#ip address 104.1.1.1/24
[local]RouterB(config-if)#pim sparse-mode
[local]RouterA(config-if)#exit
```

Static RP for Domain X configuration:

```
[local]RouterB(config-ctx)#ip pim rp-address 10.200.1.2
```

eBGP configuration:

```
[local]RouterB(config-ctx)#router bgp 100
[local]RouterB(config-bgp)#router-id 10.200.1.2
[local]RouterB(config-bgp)#address-family ipv4 multicast
[local]RouterB(config-addrfamily)#network 11.1.1.0/24
[local]RouterB(config-addrfamily)#exit
[local]RouterB(config-bgp)#peer-group eMBGP external
[local]RouterB(config-peergroup)#ebgp-multihop 5
[local]RouterB(config-peergroup)#update-source lo1
[local]RouterB(config-peergroup)#address-family ipv4 unicast
[local]RouterB(config-addrfamily)#exit
[local]RouterB(config-peergroup)#address-family ipv4 multicast
[local]RouterB(config-peergroup)#neighbor 10.200.1.3 external
[local]RouterB(config-neighbor)#remote-as 200
[local]RouterB(config-neighbor)#peer-group eMBGP
[local]RouterB(config-neighbor)#exit
[local]RouterB(config-peergroup)#exit
[local]RouterB(config-bgp)#exit
```

MSDP configuration—peering between two RPs:

```
[local]RouterB(config-ctx)#router msdp
[local]RouterB(config-msdp)#peer 10.200.1.3 local-tcp-source lo1
[local]RouterB(config-msdp-peer)#no shutdown
[local]RouterB(config-msdp-peer)#exit
[local]RouterB(config-msdp)#exit
```

Static route configuration:

```
[local]RouterB(config-ctx)#ip route 10.200.1.1/32 102.1.1.1
[local]RouterB(config-ctx)#ip route 10.200.1.3/32 104.1.1.2
[local]RouterB(config-ctx)#ip route 11.1.1.0/24 102.1.1.1
```

The configuration for RouterC (RP) is as follows:

```
[local]RouterC#config
[local]RouterC(config)#context local
[local]RouterC(config-ctx)#interface lo1 loopback
[local]RouterC(config-if)#ip address 10.200.1.3/32
[local]RouterC(config-if)#pim sparse-mode
[local]RouterC(config-if)#exit
[local]RouterC(config-ctx)#interface E2
[local]RouterC(config-if)#ip address 104.1.1.2/24
[local]RouterC(config-if)#pim sparse-mode
[local]RouterC(config-if)#exit
[local]RouterC(config-ctx)#interface E4
```

```
[local]RouterC(config-if)#ip address 105.1.1.1/24
[local]RouterC(config-if)#pim sparse-mode
[local]RouterC(config-if)#exit
```

eBGP configuration:

```
[local]RouterC(config-ctx)#router bgp 200
[local]RouterC(config-bgp)#router-id 10.200.1.3
[local]RouterC(config-bgp)#address-family ipv4 multicast
[local]RouterC(config-addrfamily)#network 44.1.1.0/24
[local]RouterC(config-addrfamily)#exit
[local]RouterC(config-bgp)#peer-group eMBGP external
[local]RouterC(config-peergroup)#ebgp-multihop 5
[local]RouterC(config-peergroup)#update-source lo1
[local]RouterC(config-peergroup)#address-family ipv4 multicast
[local]RouterC(config-addrfamily)#exit
[local]RouterC(config-peergroup)#neighbor 10.200.1.2 external
[local]RouterC(config-neighbor)#remote-as 100
[local]RouterC(config-neighbor)#peer-group eMBGP
[local]RouterC(config-neighbor)#exit
[local]RouterC(config-peergroup)#exit
[local]RouterC(config-bgp)#exit
```

Static RP for Domain Y configuration:

```
[local]RouterC(config-ctx)#ip pim rp-address 10.200.1.3
```

BGP configuration:

```
[local]RouterC(config-ctx)#router bgp 200
[local]RouterC(config-bgp)#router-id 10.200.1.3
[local]RouterC(config-bgp)#address-family ipv4 multicast
[local]RouterC(config-addrfamily)#network 44.1.1.0/24
[local]RouterC(config-addrfamily)#exit
[local]RouterC(config-bgp)#peer-group eMBGP external
[local]RouterC(config-peergroup)#ebgp-multihop 5
[local]RouterC(config-peergroup)#update-source lo1
[local]RouterC(config-peergroup)#address-family ipv4 unicast
[local]RouterC(config-addrfamily)#exit
[local]RouterC(config-peergroup)#address-family ipv4 multicast
[local]RouterC(config-addrfamily)#exit
[local]RouterC(config-peergroup)#neighbor 10.200.1.2 external
[local]RouterC(config-neighbor)#remote-as 100
[local]RouterC(config-neighbor)#peer-group eMBGP
[local]RouterC(config-neighbor)#exit
[local]RouterC(config-peergroup)#exit
[local]RouterC(config-bgp)#exit
```

Static route configuration:

```
[local]RouterC(config-ctx)#ip route 10.200.1.2/32 104.1.1.1
[local]RouterC(config-ctx)#ip route 10.200.1.4/32 105.1.1.2
[local]RouterC(config-ctx)#ip route 44.1.1.0/24 105.1.1.2
```


MSDP configuration—configure MSDP peering between two RPs:

```
[local]RouterC(config-ctx)#router msdp
[local]RouterC(config-msdp)#peer 10.200.1.2 local-tcp-source lo1
[local]RouterC(config-msdp-peer)#no shutdown
```

The configuration for RouterD is as follows:

```
[local]RouterD#config
[local]RouterD(config)#context local
[local]RouterD(config-ctx)#interface lo1 loopback
[local]RouterD(config-if)#ip address 10.200.1.4/32
[local]RouterD(config-if)#pim sparse-mode
[local]RouterD(config-if)#exit
[local]RouterD(config-ctx)#interface E1
[local]RouterD(config-if)#ip address 105.1.1.2/24
[local]RouterD(config-if)#pim sparse-mode
[local]RouterD(config-if)#exit
[local]RouterD(config-ctx)#interface E2
[local]RouterD(config-if)#ip address 44.1.1.1/24
[local]RouterD(config-if)#pim sparse-mode
```

Static RP for Domain Y configuration:

```
[local]RouterD(config-if)#ip pim rp-address 10.200.1.3
[local]RouterD(config-if)#exit
```

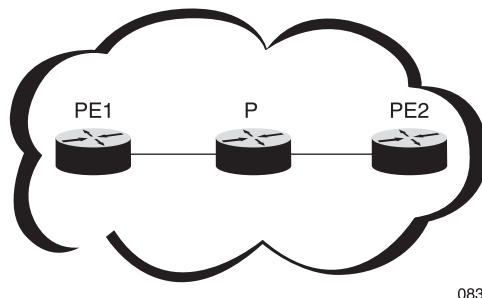
Static route configuration:

```
[local]RouterD(config-ctx)#ip route 10.200.1.3/32 105.1.1.1
```

Multicast VPNs

Multicast-enabled VPNs use MDTs to support IP multicast over BGP/MPLS VPNs. Figure 11-4 shows the multicast VPN network topology used for the configuration example.

Figure 11-4 Multicast VPN Network Topology



0837

Multicast-enabled VPNs are configured on both PE routers, PE1 and PE2. In the `local` context, the MDT encapsulation type is configured on loopback interface, `lo1`, which must be the same interface used for BGP peering. The loopback interface is used to source multicast packets on the MDT. An intercontext P2P interface is also configured in the local context, and is used to pass traffic between the VPN and the local context. (This interface does not need an IP address.)

A generic intercontext interface, `ic-local`, is configured in the VPN-enabled context, `VPN1`. This interface is similar to a loopback interface in that it is not bound to anything. It creates an intercontext circuit between the `VPN1` context and the `local` context. PIM-SM and the MDT default group are configured on this intercontext interface.

Note The IP address of the intercontext interface, `ic-local`, must be the same as that of the loopback interface in the local context used for BGP peering.

Because the MDT default group is configured in the `VPN1` context on each PE router, this information must be sent to the other PE router. When each PE router discovers that the other PE router is configured for MDTs, with the same MDT group, it sends a PIM join, with the remote PE router's loopback address as the multicast source, and the MDT group as the multicast group. This forms the MDT tree for forwarding traffic from CE router to the backbone.

The configuration for the PE1 router is as follows:

```
[local]PE1#config
[local]PE1(config)#service multiple-contexts
[local]PE1(config)#context local
[local]PE1(config-ctx)#no ip domain-lookup
[local]PE1(config-ctx)#interface ic-vpn1 intercontext p2p 1
[local]PE1(config-if)#pim sparse-mode passive
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#interface lo1 loopback
[local]PE1(config-if)#ip address 10.0.0.3/32
[local]PE1(config-if)#pim sparse-mode passive
[local]PE1(config-if)#mdt encapsulation gre
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#interface to_P
[local]PE1(config-if)#ip address 10.1.1.3/24
[local]PE1(config-if)#pim sparse-mode
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#router rip backbone
[local]PE1(config-rip)#redistribute connected
[local]PE1(config-rip)#interface to_P
[local]PE1(config-rip-if)#exit
[local]PE1(config-rip)#exit
[local]PE1(config-ctx)#router mpls
[local]PE1(config-mpls)#interface to_P
[local]PE1(config-mpls-if)#exit
[local]PE1(config-mpls)#exit
[local]PE1(config-ctx)#router ldp
[local]PE1(config-ldp)#interface lo1
[local]PE1(config-ldp)#interface to_P
[local]PE1(config-ldp)#exit
[local]PE1(config-ctx)#router bgp 100
[local]PE1(config-bgp)#neighbor 10.0.0.2 internal
```

```

[local]PE1(config-bgp-neighbor)#update-source lo1
[local]PE1(config-bgp-neighbor)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#exit
[local]PE1(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE1(config-bgp-af)#exit
[local]PE1(config-bgp-neighbor)#exit
[local]PE1(config-bgp)#exit
[local]PE1(config-ctx)#pim rp-address 10.1.1.2
[local]PE1(config-ctx)#exit
[local]PE1(config)#context VPN1 vpn-rd 10.0.0.3:1
[local]PE1(config-ctx)#interface ic-local intercontext p2p 1
[local]PE1(config-if)#ip address 10.0.0.3/24
[local]PE1(config-if)#pim sparse-mode
[local]PE1(config-if)#mdt default-group 239.1.1.1
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#interface to_CE1
[local]PE1(config-if)#ip address 11.1.1.2/24
[local]PE1(config-if)#pim sparse-mode
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#router bgp vpn
[local]PE1(config-bgp)#address-family ipv4 unicast
[local]PE1(config-bgp-af)#export route-target 100:1
[local]PE1(config-bgp-af)#import route-target 100:1
[local]PE1(config-bgp-af)#redistribute connected
[local]PE1(config-bgp-af)#exit
[local]PE1(config-bgp)#exit
[local]PE1(config-ctx)#pim rp-address 11.1.1.2
[local]PE1(config-ctx)#exit
[local]PE1(config)#card ether-12-port 4
[local]PE1(config)#port ethernet 4/8
[local]PE1(config-port)#no shutdown
[local]PE1(config-port)#bind interface to_P local
[local]PE1(config-port)#exit
[local]PE1(config)#port ethernet 4/11
[local]PE1(config-port)#no shutdown
[local]PE1(config-port)#bind interface to_CE1 VPN1
[local]PE1(config)#end

```

The configuration for the P router is as follows:

```

[local]P#config
[local]P(config)#context local
[local]P(config-ctx)#interface to_PE1
[local]P(config-if)#ip address 10.1.1.2/24
[local]P(config-if)#pim sparse-mode
[local]P(config-if)#exit
[local]P(config-ctx)#interface to_PE2
[local]P(config-if)#ip address 20.1.1.2/24
[local]P(config-if)#pim sparse-mode
[local]P(config-if)#exit
[local]P(config-ctx)#router rip backbone
[local]P(config-rip)#redistribute connected

```

```
[local]P(config-rip)#interface to_PE1
[local]P(config-rip-if)#exit
[local]P(config-rip)#interface to_PE2
[local]P(config-rip-if)#exit
[local]P(config-rip)#exit
[local]P(config-ctx)#router mpls
[local]P(config-mpls)#interface to_PE1
[local]P(config-mpls-if)#exit
[local]P(config-mpls)#interface to_PE2
[local]P(config-mpls-if)#exit
[local]P(config-mpls)#exit
[local]P(config-ctx)#router ldp
[local]P(config-ldp)#interface to_PE1
[local]P(config-ldp)#interface to_PE2
[local]P(config-ldp)#exit
[local]P(config-ctx)#pim rp-address 10.1.1.2
[local]P(config-ctx)#exit
[local]P(config)#card ether-12-port 13
[local]P(config)#port ethernet 13/6
[local]P(config-port)#no shutdown
[local]P(config-port)#bind interface to_PE1 local
[local]P(config-port)#exit
[local]P(config)#port ethernet 13/11
[local]P(config-port)#no shutdown
[local]P(config-port)#bind interface to_PE2 local
[local]P(config)#end
```

The configuration for the PE2 router is as follows:

```
[local]PE2#config
[local]PE2(config)#service multiple-contexts
[local]PE2(config)#context local
[local]PE2(config-ctx)#interface ic-vpn1 intercontext p2p 1
[local]PE2(config-if)#pim sparse-mode passive
[local]PE2(config-if)#exit
[local]PE2(config-ctx)#interface lo1 loopback
[local]PE2(config-if)#ip address 10.0.0.2/32
[local]PE2(config-if)#pim sparse-mode passive
[local]PE2(config-if)#mdt encapsulation gre
[local]PE2(config-if)#exit
[local]PE2(config-ctx)#interface to_P
[local]PE2(config-if)#ip address 20.1.1.3/24
[local]PE2(config-if)#pim sparse-mode
[local]PE2(config-if)#exit
[local]PE2(config-ctx)#router rip backbone
[local]PE2(config-rip)#redistribute connected
[local]PE2(config-rip)#interface to_P
[local]PE2(config-rip-if)#exit
[local]PE2(config-rip)#exit
[local]PE2(config-ctx)#router mpls
[local]PE2(config-mpls)#interface to_P
[local]PE2(config-mpls-if)#exit
```

```

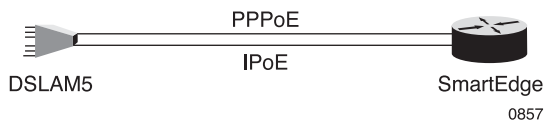
[local]PE2(config-mpls)#exit
[local]PE2(config-ctx)#router ldp
[local]PE2(config-ldp)#interface lo1
[local]PE2(config-ldp)#interface to_P
[local]PE2(config-ldp)#exit
[local]PE2(config-ctx)#router bgp 100
[local]PE2(config-bgp)#neighbor 10.0.0.3 internal
[local]PE2(config-bgp-neighbor)#update-source lo1
[local]PE2(config--bgp-neighbor)#address-family ipv4 unicast
[local]PE2(config--bgp-af)#exit
[local]PE2(config-bgp-neighbor)#address-family ipv4 vpn
[local]PE2(config-bgp-af)#exit
[local]PE2(config-bgp-neighbor)#exit
[local]PE2(config-bgp)#exit
[local]PE2(config-ctx)#pim rp-address 10.1.1.2
[local]PE2(config-ctx)#exit
[local]PE2(config)#context VPN1 vpn-rd 10.0.0.2:1
[local]PE2(config-ctx)#no ip domain-lookup
[local]PE2(config-ctx)#interface ic-local intercontext p2p 1
[local]PE2(config-if)#ip address 10.0.0.2/24
[local]PE2(config-if)#pim sparse-mode
[local]PE2(config-if)#mdt default-group 239.1.1.1
[local]PE2(config-if)#exit
[local]PE2(config-ctx)#interface to_CE2
[local]PE2(config-if)#ip address 21.1.1.2/24
[local]PE2(config-if)#pim sparse-mode
[local]PE2(config-if)#no logging console
[local]PE2(config-if)#exit
[local]PE2(config-ctx)#router bgp vpn
[local]PE2(config-bgp)#address-family ipv4 unicast
[local]PE2(config-bgp-af)#export route-target 100:1
[local]PE2(config-bgp-af)#import route-target 100:1
[local]PE2(config-bgp-af)#redistribute connected
[local]PE2(config-bgp-af)#exit
[local]PE2(config-bgp)#exit
[local]PE2(config-ctx)#pim rp-address 11.1.1.2
[local]PE2(config-ctx)#exit
[local]PE2(config)#card ether-12-port 1
[local]PE2(config)#port ethernet 1/3
[local]PE2(config-port)#no shutdown
[local]PE2(config-port)#bind interface to_CE2 VPN1
[local]PE2(config-port)#exit
[local]PE2(config)#port ethernet 1/12
[local]PE2(config-port)#no shutdown
[local]PE2(config-port)#bind interface to_P local
[local]PE2(config-port)#end

```

Remote Multicast Replication

RMR is used to enable IP multicast services. Figure 11-5 shows the RMR network topology used for the configuration example.

Figure 11-5 RMR Network Topology



The MC, a SmartEdge router, is connected to an MR, DSLAM5, with PPPoE and IPoE circuits. The PPPoE circuit is created on a 4-port gigabit Ethernet card on slot 14, and an IPoE circuit is created on the `ipoe_to_dslam5` interface, which is bound to a 12-port ethernet card on slot 4. The interface is enabled to forward multicast traffic, and to send and receive the IGMP control messages. The `foo` IGMP service profile is linked to the multicast-enabled `ipoe_to_dslam5` interface.

Subscribers are brought up from the PPPoE circuit. The multicast traffic and the IGMP control messages are forwarded on the IPoE circuit. DSLAM5 replicates the multicast stream for all interested subscribers.

The RMR configuration is as follows:

```
[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config-ctx)#interface ipoe_to_dslam5
[local]Redback(config-if)#ip address 11.1.1.1/24
[local]Redback(config-if)#igmp service-profile foo
[local]Redback(config-if)#multicast output accept-unknown-mac
[local]Redback(config-if)#pim sparse-mode passive
[local]Redback(config-if)#exit
[local]Redback(config)#interface pppoe_to_dslam5 multibind
[local]Redback(config-if)#ip address 192.1.1.1/16
[local]Redback(config-if)#ip pool 192.1.0.0/16
[local]Redback(config-if)#pim sparse-mode passive
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#igmp service-profile foo
[local]Redback(config-igmp-service-profile)#instant-leave
[local]Redback(config-igmp-service-profile)#static-group 224.1.1.1
[local]Redback(config-igmp-service-profile)#exit
[local]Redback(config-ctx)#igmp service-profile bar
[local]Redback(config-igmp-service-profile)#multicast destination ipoe_to_dslam5 local
[local]Redback(config-igmp-service-profile)#exit
[local]Redback(config-ctx)#subscriber name joe
[local]Redback(config-sub)#password test
[local]Redback(config-sub)#ip address pool
[local]Redback(config-sub)#ip igmp service-profile test
[local]Redback(config-sub)#exit
[local]Redback(config-ctx)#pim rp-address 21.1.1.1
[local]Redback(config-ctx)#pim static group 224.1.1.1 source 50.1.1.100 send-join
[local]Redback(config-ctx)#ip access-list 1
[local]Redback(config-access-list)#seq 10 deny ip host 224.1.1.1
[local]Redback(config-access-list)#exit
```

```

[local]Redback(config-ctx)#exit
[local]Redback(config)#card ether-12-port 4
[local]Redback(config)#port ethernet 4/2
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface ipoe_to_dslam5 local
[local]Redback(config-port)#exit
[local]Redback(config)#card gigaether-4-port 14
[local]Redback(config)#port ethernet 14/1
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#encapsulation pppoe
[local]Redback(config-port)#bind authentication chap pap context local maximum 8000
[local]Redback(config-port)#end

```

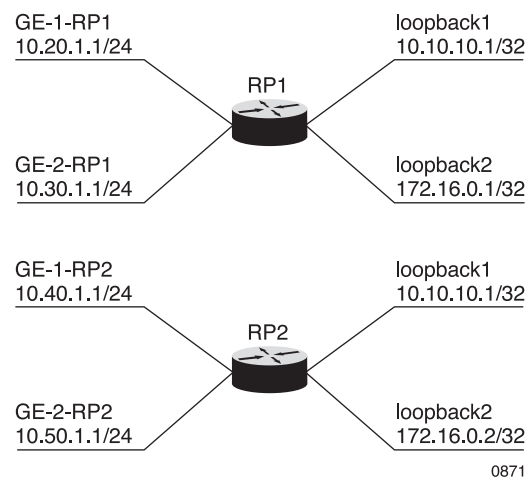
Anycast RP

Anycast RP is a mechanism that provides RP redundancy and load-sharing capabilities by allowing the use of multiple RPs within a single multicast domain. Assuming that the sources are evenly spaced around the network, an equal number of sources register with each RP. That is, the process of registering the sources are shared equally by all the RPs in the network.

All routers acting as RPs must be configured with a loopback interface using the same anycast RP address. All downstream routers use that anycast RP address as the IP address for their local RP. To facilitate communication between RPs, each router acting as an RP must also be configured with its own unique IP address, which is used only to send and receive messages from the other RPs.

Figure 11-6 shows the Anycast RP network topology used for the configuration example.

Figure 11-6 Anycast RP Network Topology



In this configuration example, two routers, RP1 and RP2, are configured for anycast RP. Both routers are configured with a loopback interface, loopback1, using the same IP address, which is used as the IP address for the anycast RP set. Both routers are also configured with a loopback interface, loopback2, using unique IP addresses. The loopback2 interface is used to facilitate communication between the two RPs. The other interfaces, GE - 1 - RP1, GE - 2 - RP1, GE - 1 - RP2, and GE - 2 - RP2, are physical interfaces that connect to the network, and are used to send and receive multicast packets.

The configuration for the RP1 router is as follows:

```
[local]RP1#config
[local]RP1(config)#context local
[local]RP1(config-ctx)#interface loopback1 loopback
[local]RP1(config-if)#description Anycast-RP-Looback
[local]RP1(config-if)#ip address 10.10.10.1/32
[local]RP1(config-if)#pim sparse-mode
[local]RP1(config-if)#exit
[local]RP1(config-ctx)#interface loopback2 loopback
[local]RP1(config-if)#description Unique-RP-Loopback
[local]RP1(config-if)#ip address 172.16.0.1/32
[local]RP1(config-if)#pim sparse-mode
[local]RP1(config-if)#exit
[local]RP1(config-ctx)#interface GE-1-RP1
[local]RP1(config-if)#ip address 10.20.1.1/24
[local]RP1(config-if)#pim sparse-mode
[local]RP1(config-if)#exit
[local]RP1(config-ctx)#interface GE-2-RP2
[local]RP1(config-if)#ip address 10.30.1.1/24
[local]RP1(config-if)#pim sparse-mode
[local]RP1(config-if)#exit
[local]RP1(config-ctx)#router ospf
[local]RP1(config-ospf)#area 0.0.0.0
[local]RP1(config-ospf-area)#interface GE-2-RP1
[local]RP1(config-ospf-if)#exit
[local]RP1(config-ospf-area)#interface GE-1-RP2
[local]RP1(config-ospf-if)#exit
[local]RP1(config-ospf-area)#interface loopback1
[local]RP1(config-ospf-if)#exit
[local]RP1(config-ospf-area)#interface loopback2
[local]RP1(config-ospf-if)#exit
[local]RP1(config-ospf-area)#exit
[local]RP1(config-ospf)#exit
[local]RP1(config-ctx)#pim anycast-rp 10.10.10.1 172.16.0.1
[local]RP1(config-ctx)#pim anycast-rp 10.10.10.1 172.16.0.2
[local]RP1(config-ctx)#pim rp-address 10.10.10.1
```

The configuration for the RP2 router is as follows:

```
[local]RP2#config
[local]RP2(config)#context local
[local]RP2(config-ctx)#interface loopback1 loopback
[local]RP2(config-if)#description Anycast-RP-Looback
[local]RP2(config-if)#ip address 10.10.10.1/32
[local]RP2(config-if)#pim sparse-mode
[local]RP2(config-if)#exit
[local]RP2(config-ctx)#interface loopback2 loopback
[local]RP2(config-if)#description Unique-RP-Loopback
[local]RP2(config-if)#ip address 172.16.0.2/32
[local]RP2(config-if)#pim sparse-mode
[local]RP2(config-if)#exit
[local]RP2(config-ctx)#interface GE-1-RP2
```



```
[local]RP2(config-if)#ip address 10.40.1.1/24
[local]RP2(config-if)#pim sparse-mode
[local]RP2(config-if)#exit
[local]RP2(config-ctx)#interface GE-2-RP2
[local]RP2(config-if)#ip address 10.50.1.1/24
[local]RP2(config-if)#pim sparse-mode
[local]RP2(config-if)#exit
[local]RP2(config-ctx)#router ospf
[local]RP2(config-ospf)#area 0.0.0.0
[local]RP2(config-ospf-area)#interface GE-2-RP2
[local]RP2(config-ospf-if)#exit
[local]RP2(config-ospf-area)#interface GE-1-RP2
[local]RP2(config-ospf-if)#exit
[local]RP2(config-ospf-area)#interface loopback1
[local]RP2(config-ospf-if)#exit
[local]RP2(config-ospf-area)#interface loopback2
[local]RP2(config-ospf-if)#exit
[local]RP2(config-ospf-area)#exit
[local]RP2(config-ospf)#exit
[local]RP2(config-ctx)#pim anycast-rp 10.10.10.1 172.16.0.1
[local]RP2(config-ctx)#pim anycast-rp 10.10.10.1 172.16.0.2
[local]RP2(config-ctx)#pim rp-address 10.10.10.1
```

Command Descriptions

This section describes the syntax and usage guidelines for the commands used to configure IP multicast features. The commands are presented in alphabetical order.

default-peer	originating-rp sa-filter
description	peer
igmp access-group	peer-as
igmp group-bandwidth	pim accept-rp
igmp join-group	pim anycast-rp
igmp last-member-query-interval	pim bsr-border
igmp maximum-bandwidth	pim bsr-candidate
igmp mtrace-prohibit	pim dense-mode
igmp query-interval	pim dr-priority
igmp query-max-response-time	pim graceful-restart
igmp robust	pim hello-interval
igmp service-profile	pim neighbor-filter
igmp version	pim operation-mode
instant-leave	pim rp-address
ip igmp service-profile	pim rp-candidate
ip multicast boundary	pim sparse-mode
ip multicast receive	pim spt-threshold infinity
ip multicast send	pim ssm
max-groups	pim static group
mdt default-group	priority
mdt encapsulation	router msdp
mesh-group	sa-filter
multicast destination	shutdown
multicast output	static-group
originating-rp	sticky-groups

default-peer

default-peer *peer-addr* [*pl-name*]

no default-peer *peer-addr* [*pl-name*]

Purpose

Configures a default peer from which to accept all Multicast Source Discovery Protocol (MSDP) source active (SA) messages.

Command Mode

MSDP router configuration

Syntax Description

<i>peer-addr</i>	Peer IP address to be set as the default peer.
<i>pl-name</i>	Optional. Name of the Border Gateway Protocol (BGP) prefix list which specifies that the peer will be a default peer only for the prefixes listed in the list. A BGP prefix list must be configured for this <i>pl-name</i> argument to have any effect.

Default

None

Usage Guidelines

Use the **default-peer** command to configure a default peer from which to accept all MSDP SA messages. A default peer is needed in topologies where MSDP peers do not coexist with BGP peers. In such a case, the reverse path forwarding (RPF) check on SA messages can fail, and no SA messages are accepted. In these cases, you can configure the peer as a default peer, and bypass RPF checks.

Note An MSDP peer must already be configured before it can be made a default peer.

The *peer-addr* argument must be the IP address of a previously configured peer.

Use the *pl-name* argument to allow only those SA entries whose RP is permitted in the prefix list; otherwise, all SA messages from the default peer are accepted.

Use the **no** form of this command to disable the default peer.

Examples

The following example configures the peer address, 192.168.3.8, as the default peer:

```
[local]Redback(config-ctx)#router msdp
[local]Redback(config-msdp)#default-peer 192.168.3.8
```

Related Commands

description
mesh-group
originating-rp
originating-rp sa-filter
peer
peer-as
router msdp
sa-filter
shutdown

igmp access-group

igmp access-group *acl-name*

no igmp access-group *acl-name*

Purpose

Configures Internet Group Management Protocol (IGMP) membership on an interface.

Command Mode

interface configuration

Syntax Description

<i>acl-name</i>	Name of the access control list (ACL) used to filter IGMP membership.

Default

None

Usage Guidelines

Use the **igmp access-group** command to configure IGMP membership on an interface.

Note Only multicast groups permitted by the ACL are accepted on the interface.

Use the **no** form of this command to remove the ACL filter, and allow all groups to have access on an interface.

Examples

The following example configures IGMP membership using the ACL, igmp_mem03:

```
[ local ]Redback(config-ctx)#interface enet01
[ local ]Redback(config-if)#igmp access-group igmp_mem03
```

Related Commands

- igmp join-group**
- igmp last-member-query-interval**
- igmp mtrace-prohibit**
- igmp query-interval**
- igmp query-max-response-time**
- igmp robust**
- igmp version**

igmp group-bandwidth

igmp group-bandwidth *rate* **group-list** *acl-name*

no igmp group-bandwidth *rate* **group-list** *acl-name*

Purpose

Configures the recommended bandwidth required by each of the specified groups.

Command Mode

context configuration

Syntax Description

<i>rate</i>	Recommended rate in Kbps for each group.
group-list <i>acl-name</i>	Access control list (ACL) name used to permit groups to the group bandwidth profile.

Default

None

Usage Guidelines

Use the **igmp group-bandwidth** command to configure the recommended bandwidth required by each of the specified groups. Before configuring the recommended group bandwidth, you should know the rate at which senders send on each group.

Note You can use inbound rate limiting to ensure that the groups' recommended bandwidth is not exceeded.

Use the **no** form of this command to delete a group bandwidth profile.

Examples

The following example configures a recommended bandwidth rate of 512 Kbps for each group permitted by the ACL, grp936:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#igmp group-bandwidth 512 group-list grp936
```

Related Commands

igmp maximum-bandwidth	max-groups
igmp service-profile	priority
igmp version	sticky-groups
instant-leave	

igmp join-group

igmp join-group *group-addr*

no igmp join-group *group-addr*

Purpose

Configures a router to join a multicast group.

Command Mode

interface configuration

Syntax Description

group-addr Multicast group IP address.

Default

None

Usage Guidelines

Use the **igmp join-group** command to configure a router to join a multicast group on the interface.

Use the **no** form of this command to remove a router from a multicast group.

Examples

The following example configures a router to join multicast group 224.1.1.1:

```
[local]Redback(config-ctx)#interface enet01  
[local]Redback(config-if)#igmp join-group 224.1.1.1
```



Caution Risk of reduced router performance. If local joins are configured, packets are punted from the Packet Processing ASIC (PPA) to the Cross-Connect Route Processor (XCRP) or XCRP Version 3 (XCRP3) Controller card. To reduce the risk, ensure that data is not sent at high rates for local joins.

Related Commands

igmp access-group
igmp last-member-query-interval
igmp mtrace-prohibit
igmp query-interval
igmp query-max-response-time
igmp robust
igmp version

igmp last-member-query-interval

igmp last-member-query-interval *interval*

no igmp last-member-query-interval

Purpose

Configures the interval at which the router sends Internet Group Management Protocol (IGMP) group-specific host query messages.

Command Mode

interface configuration

Syntax Description

<i>interval</i>	Interval, in milliseconds, at which IGMP group-specific host query messages are sent.

Default

The default last member query interval is 1,000 milliseconds (1 second).

Usage Guidelines

Use the **igmp last-member-query-interval** command to configure the interval at which the router sends IGMP group-specific host query messages.

Use the **no** form of this command to set the interval to the default value of 1,000 milliseconds.

Examples

The following example sets the last member query interval to 2500 milliseconds (2.5 seconds):

```
[local]Redback(config-ctx)#interface enet01  
[local]Redback(config-if)#igmp last-member-query-interval 2500
```

Related Commands

- igmp access-group**
- igmp join-group**
- igmp mtrace-prohibit**
- igmp query-interval**
- igmp query-max-response-time**
- igmp robust**
- igmp version**

igmp maximum-bandwidth

igmp maximum-bandwidth *rate* [**percent**]

no igmp maximum-bandwidth

Purpose

Configures the total maximum bandwidth allowed for multicast data traffic on a port or channel.

Command Mode

ATM configuration
 ATM DS-3 configuration
 AU-3 configuration
 DS-0 configuration
 DS-1 configuration
 DS-3 configuration
 E1 configuration
 E3 configuration
 port configuration
 STM-1 configuration

Syntax Description

<i>rate</i>	Maximum rate in Kbps when the percent keyword is not specified. When the percent keyword is specified, the <i>rate</i> value is taken as a percentage of the port bandwidth, and not a rate in Kbps.
percent	Optional. Specifies that the <i>rate</i> value is taken as a percentage of the port bandwidth, and not a rate in Kbps.

Default

None

Usage Guidelines

Use the **igmp maximum-bandwidth** command to configure the total maximum bandwidth allowed for multicast data traffic on a port or channel.

Note If the addition of a new group would cause the bandwidth usage on this port to exceed the maximum bandwidth, and if a subscriber with a lower priority exists on this port, the lower priority group is dropped to reclaim the bandwidth; otherwise, the new group is dropped.

Use the **no** command to remove maximum bandwidth restrictions a the port or channel.

Examples

The following example configures a maximum bandwidth of 300 Kbps for a Ethernet port in slot 7:

```
[local]Redback(config)#port ethernet 7/1  
[local]Redback(config-port)#igmp maximum-bandwidth 300
```

The following example configures a maximum bandwidth of 35 percent of an Ethernet port's maximum bandwidth:

```
[local]Redback(config)#port ethernet 7/1  
[local]Redback(config-port)#igmp maximum-bandwidth 35 percent
```

Related Commands

- igmp group-bandwidth**
- igmp service-profile**
- igmp version**
- instant-leave**
- max-groups**
- priority**
- sticky-groups**

igmp mtrace-prohibit

igmp mtrace-prohibit

Purpose

Ensures that all mtrace queries are received within the administratively scoped domain of the router.

Command Mode

context configuration

Syntax Description

This command has no keywords or arguments.

Default

None

Usage Guidelines

Use the **igmp mtrace-prohibit** command to ensure that all mtrace queries are received within the administratively scoped domain of the router.

Examples

The following example ensures that all mtrace queries are received within the administratively scoped domain of the router:

```
[local]Redback(config)#context
[local]Redback(config-ctx)#igmp mtrace-prohibit
[local]Redback(config-ctx)#
```

Related Commands

- igmp access-group**
- igmp join-group**
- igmp last-member-query-interval**
- igmp query-interval**
- igmp query-max-response-time**
- igmp robust**
- igmp version**

igmp query-max-response-time

igmp query-max-response-time *interval*

no igmp query-max-response-time

Purpose

Configures the maximum response time specified in Internet Group Management Protocol (IGMP) queries.

Command Mode

interface configuration

Syntax Description

<i>interval</i>	Interval, in seconds, specified in IGMP queries.

Default

The default IGMP query-max-response-time is 10 seconds.

Usage Guidelines

Use the **igmp query-max-response-time** command to configure the maximum response time specified in IGMP queries.

Use the **no** form of this command to set the interval to the default value of 10 seconds.

Examples

The following example sets the maximum response time to 30 seconds:

```
[local]Redback(config-ctx)#interface enet01
[local]Redback(config-if)#igmp query-max-response-time 30
```

Related Commands

igmp access-group
igmp join-group
igmp last-member-query-interval
igmp mtrace-prohibit
igmp query-interval
igmp robust
igmp version

igmp robust

igmp robust *robust-value*

no igmp robust

Purpose

Configures the Internet Group Management Protocol (IGMP) robustness variable.

Command Mode

interface configuration

Syntax Description

robust-value Robustness value. The range of values is 2 to 7; the default value is 2.

Default

The default robustness value is 2.

Usage Guidelines

Use the **igmp robust** command to configure the IGMP robustness value. The group membership interval, other querier present interval, startup query count, and last member query count are all determined by the robustness value.

Use the **no** form of this command to set the robustness to the default value of 2.

Examples

The following example configures the robustness variable to 4:

```
[local]Redback(config-ctx)#interface enet01  
[local]Redback(config-if)#igmp robust 4
```

Related Commands

igmp access-group
igmp join-group
igmp last-member-query-interval
igmp mtrace-prohibit
igmp query-interval
igmp query-max-response-time
igmp version

igmp service-profile

igmp service-profile *prof-name*

no igmp service-profile *prof-name*

Purpose

In context configuration mode, creates a service profile and enters IGMP service profile configuration mode.

In interface configuration mode, enables the specified service profile on the interface.

Command Mode

context configuration

interface configuration

Syntax Description

<i>prof-name</i>	In context configuration mode, name of the service profile to be created.
	In interface configuration mode, name of an existing service profile to enable on the interface.

Default

None

Usage Guidelines

Use the **igmp service-profile** command in context configuration mode to create a service profile and enters IGMP service profile configuration mode.

Use the **igmp service-profile** in interface configuration mode to enable the specified service profile on the interface.

Use the **no** form of this command in context configuration mode to delete the specified service profile.

Use the **no** form of this command in interface configuration mode to disable the specified service profile on the interface.

Examples

The following example creates a service profile, pro332, and enters IGMP service profile configuration mode:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#igmp service-profile pro332
[local]Redback(config-igmp-service-profile)#
```

The following example enables a service profile, pro332, on the interface, foo:

```
[local]Redback(config-ctx)#interface foo
[local]Redback(config-if)#igmp service-profile pro332
```


Related Commands

- `igmp group-bandwidth`
- `igmp maximum-bandwidth`
- `igmp version`
- `instant-leave`
- `max-groups`
- `priority`
- `static-group`
- `sticky-groups`

igmp version

igmp version {1 | 2 | 3}

no igmp version

Purpose

Configures the interface to operate in either Internet Group Management Protocol (IGMP) Version 1, Version 2, or Version 3 mode.

Command Mode

interface configuration

Syntax Description

1	Configures the interface to operate in IGMP Version 1 mode.
2	Configures the interface to operate in IGMP Version 2 mode.
3	Configures the interface to operate in IGMP Version 3 mode.

Default

The default mode is IGMP Version 2.

Usage Guidelines

Use the **igmp version** command to configure the interface to operate in either IGMP Version 1, Version 2, or Version 3 mode.

Use the **no** form of this command to configure the interface to the default value.

Examples

The following example configures the interface to operate in IGMP Version 2 mode:

```
[local]Redback(config-ctx)#interface enet01  
[local]Redback(config-if)#igmp version 2
```

Related Commands

igmp access-group
igmp join-group
igmp last-member-query-interval
igmp mtrace-prohibit
igmp query-interval
igmp query-max-response-time
igmp robust

instant-leave

instant-leave
no instant-leave

Purpose

Enables Instant Leave on the interface.

Command Mode

IGMP service profile configuration

Syntax Description

This command has no keywords or arguments.

Default

Instant Leave is disabled.

Usage Guidelines

Use the **instant-leave** command to enable Instant Leave on the interface.

Instant Leave allows Internet Group Management Protocol (IGMP) to perform a 0-delay leave upon receiving an IGMP Version 2 (IGMPv2) leave message. If the router is an IGMP querier, it sends an IGMP last member query with a 100 ms last member query response time; however, the router does not wait for 100 ms before it prunes off the group. This allows channel surfing applications to function better.

Use the **no** form of this command to disable Instant Leave on the interface.

Examples

The following example enables Instant Leave on the service profile, bar:

```
[local]Redback(config-ctx)#igmp service-profile bar  
[local]Redback(config-igmp-service-profile)#instant-leave
```

Related Commands

igmp group-bandwidth
igmp maximum-bandwidth
igmp service-profile
igmp version
max-groups
priority
static-group
sticky-groups

ip igmp service-profile

ip igmp service-profile *prof-name*

no ip igmp service-profile *prof-name*

Purpose

Enables an existing Internet Group Management Protocol (IGMP) service profile on a single subscriber record, a named subscriber profile, or a default subscriber profile.

Command Mode

subscriber configuration

Syntax Description

<i>prof-name</i>	Name of the IGMP service profile enabled on the subscriber profile.

Default

None

Usage Guidelines

Use the **ip igmp service-profile** command to enable a existing IGMP service profile on a single subscriber record, a named subscriber profile, or a default subscriber profile. The service profile used is determined in the following order:

- Subscriber profile
- Default subscriber profile
- Service profile configured on the subscriber's parent interface

If a service profile is not defined in the subscriber record, it inherits the service profile from the default subscriber profile. If the default subscriber profile is not configured with an service profile, the service profile configured on the interface is used.

Use the **no** form of this command to disable the service profile on the subscriber.

Examples

The following example enables the IGMP service profile, sp04, on the default subscriber profile:

```
[local]Redback(config-ctx)#subscriber default  
[local]Redback(config-sub)#ip igmp service-profile sp04
```

Related Commands

ip multicast receive
ip multicast send
pim sparse-mode

ip multicast boundary

ip multicast boundary *acl-name*

no ip multicast boundary *acl-name*

Purpose

Configures an administratively scoped boundary for multicast routing.

Command Mode

interface configuration

Syntax Description

acl-name

Name of the access control list (ACL) that controls the range of group addresses affected by the boundary.

Default

None

Usage Guidelines

Use the **ip multicast boundary** command to configure an administratively scoped boundary for multicast routing. This boundary prevents forwarding of multicast data packet destined for group addresses denied by the ACL.

Use the **no** form of this command to remove the multicast boundary from the interface.

Examples

The following example configures an administratively scoped boundary for multicast using ACL 20:

```
[local]Redback(config-ctx)#interface enet01  
[local]Redback(config-if)#ip multicast boundary 20
```

Related Commands

pim accept-rp
pim bsr-border
pim bsr-candidate
pim dr-priority
pim hello-interval
pim neighbor-filter
pim operation-mode
pim rp-address
pim rp-candidate
pim sparse-mode

ip multicast receive

ip multicast receive {permit | deny}

no ip multicast receive

Purpose

Configures the multicast receive permissions for a subscriber record, a named subscriber profile, or a default subscriber profile.

Command Mode

subscriber configuration

Syntax Description

permit	Allows the subscriber to receive multicast traffic.
deny	Denies the subscriber the ability to receive multicast traffic.

Default

The multicast receive permission is set to permit.

Usage Guidelines

Use the **ip multicast receive** command to configure the multicast receive permissions for a subscriber record, a named subscriber profile, or a default subscriber profile. Permission attributes are applied in the following order:

- Subscriber profile
- Default subscriber profile
- System defaults

If a permission is not defined in the subscriber, it inherits the value of the permission from the default subscriber profile. If the permission is not defined in the default subscriber profile, the system default values are used.

For multicast routing to function on subscribers, you must use the **pim sparse-mode** command in interface configuration mode to enable Protocol Independent Multicast Sparse-Mode (PIM-SM) on the interface.

Use the **no** form of this command to delete receive permissions for the profile to which the command is applied.

Examples

The following example sets receive permissions to permit for the default subscriber profile:

```
[local]Redback(config-ctx)#subscriber default  
[local]Redback(config-sub)#ip multicast receive permit
```

The following example sets receive permissions to deny for subscriber freddy:

```
[local]Redback(config-ctx)#subscriber name freddy  
[local]Redback(config-sub)#ip multicast receive deny
```

Related Commands

ip igmp service-profile
ip multicast send
pim sparse-mode

ip multicast send

ip multicast send {**permit** [**unsolicit**] | **deny**}

no ip multicast send

Purpose

Configures the multicast send permissions for a subscriber record, a named subscriber profile, or a default subscriber profile.

Command Mode

subscriber configuration

Syntax Description

permit	Allows the subscriber to send multicast traffic.
unsolicit	Optional. Used in conjunction with the permit keyword to indicate that the subscriber is allowed to send unsolicited multicast traffic.
deny	Denies the subscriber the ability to send multicast traffic.

Default

The multicast send permission is set to deny.

Usage Guidelines

Use the **ip multicast send** command to configure the multicast send permissions for a subscriber record, a named subscriber profile, or a default subscriber profile.

If the **permit** keyword is used without the **unsolicit** keyword, the subscriber must join a group prior to sending unsolicited multicast data. If used together (**permit unsolicit**), a subscriber is allowed to send unsolicited multicast traffic. Permissions are examined in the following order:

- Subscriber profile
- Default subscriber profile
- System defaults.

If a permission is not defined in the subscriber profile, it inherits the value of the permission from the default subscriber profile. If the permission is undefined in the default subscriber profile, the system default values are used.

For multicast routing to function on subscribers, you must use the **pim sparse-mode** command in interface configuration mode to enable Protocol Independent Multicast Sparse-Mode (PIM-SM) on the interface.

Use the **no** form of this command to delete all send permissions for the profile. Deleting the permissions in a subscriber profile causes the system to use the permissions from the default subscriber profile. If no such permissions exist in the default subscriber profile, the system default is used.

Examples

The following example configures the default subscriber profile with the permission to send multicast traffic; however, subscriber **mike** is denied sending multicast traffic:

```
[local]Redback(config-ctx)#subscriber default
[local]Redback(config-sub)#ip multicast send permit
[local]Redback(config-sub)#exit
[local]Redback(config-ctx)#subscriber name mike
[local]Redback(config-sub)#ip multicast send deny
```

The following example (using the **no** form) deletes send permissions in the default subscriber profile; however, the system default for multicast send is permit, so the subscriber **jane** can send and receive multicast traffic:

```
[local]Redback(config-ctx)#subscriber default
[local]Redback(config-sub)#no ip multicast send
[local]Redback(config-sub)#exit
[local]Redback(config-ctx)#subscriber name jane
[local]Redback(config-sub)#ip address 10.10.1.4
[local]Redback(config-sub)#exit
```

Related Commands

ip igmp service-profile
ip multicast receive
pim sparse-mode

max-groups

max-groups *count* [**drop-old**]

no max-groups

Purpose

Configures the maximum number of Internet Group Management Protocol (IGMP)-joined groups allowed per interface.

Command Mode

IGMP service profile configuration

Syntax Description

<i>count</i>	Maximum number of IGMP-joined groups. The range of values is 1 to 100,000.
drop-old	Optional. Drops the oldest IGMP group on the interface, and accepts the new IGMP report.

Default

Maximum number of IGMP-joined groups is not configured.

Usage Guidelines

Use the **max-groups** command to configure the maximum number of IGMP-joined groups allowed per interface.

If the addition of a new group on an interface causes the total number of joined groups to exceed the maximum number allowed, one of the following actions is taken:

- If the **drop-old** keyword is specified for the service profile, the oldest IGMP group on the interface is dropped and the new IGMP report accepted.
- If the **drop-old** keyword is not specified for the service profile, the new IGMP membership report is dropped.

Use the **no** form of this command to remove the maximum number of IGMP-joined groups restriction.

Examples

The following example configures a maximum of 5,000 IGMP-joined groups per interface:

```
[local]Redback(config-ctx)#igmp service-profile bar
[local]Redback(config-igmp-service-profile)#max-groups 5000
```

Related Commands

- `igmp group-bandwidth`
- `igmp maximum-bandwidth`
- `igmp service-profile`
- `igmp version`
- `instant-leave`
- `priority`
- `static-group`
- `sticky-groups`

mdt default-group

mdt default-group *ip-addr*

no mdt default-group *ip-addr*

Purpose

Specifies the default multicast domain tree (MDT) group.

Command Mode

interface configuration

Syntax Description

ip-addr IP address of the default MDT group in the form *A.B.C.D*.

Default

No default MDT group is specified.

Usage Guidelines

Use the **mdt default-group** command to specify the default MDT group.

You must configure the **mdt default-group** command on an intercontext interface in a VPN-enabled context. The intercontext interface creates an intercontext circuit between the VPN-enabled context and the local context.

Use the **no** form of this command to disable the default MDT group.

Examples

The following example specifies the default MDT group, 30.40.50.60, on an intercontext interface, to-local, in a VPN-enabled context, VPN1:

```
[local]Redback(config)#context VPN1 vpn-rd 101:202
[local]Redback(config-ctx)#interface to-local intercontext p2p 2
[local]Redback(config-if)#mdt default-group 30.40.50.60
```

Related Commands

mdt encapsulation

mdt encapsulation

mdt encapsulation {gre | ip}

no mdt encapsulation {gre | ip}

Purpose

Specifies the multicast domain tree (MDT) encapsulation type.

Command Mode

interface configuration

Syntax Description

gre	Uses the GRE encapsulation type.
ip	Uses the IP-in-IP encapsulation type.

Default

No MDT encapsulation type is specified.

Usage Guidelines

Use the **mdt encapsulation** command to specify the MDT encapsulation type.

You must configure this command on a loopback interface in the local context. The loopback interface is used to source multicast packets on the MDT.

Note The PIM-SM explicit join mechanism is optimal only for sparsely populated groups.

Use the **no** form of this command to remove the MDT encapsulation type.

Examples

The following example specifies the MDT encapsulation type, gre, for the loopback interface, to-vpn1:

```
[local]Redback(config)#context local  
[local]Redback(config-ctx)#interface to-vpn1 intercontext p2p 1  
[local]Redback(config-if)#mdt encapsulation gre
```

Related Commands

mdt default-group

mesh-group

mesh-group *group-name peer-addr*

no mesh-group *group-name peer-addr*

Purpose

Configures a Multicast Source Discovery Protocol (MSDP) peer to be a member of a mesh group.

Command Mode

MSDP router configuration

Syntax Description

<i>group-name</i>	Mesh group name.
<i>peer-addr</i>	IP address of the peer to be added to the mesh group.

Default

None

Usage Guidelines

Use the **mesh-group** command to configure an MSDP peer to be a member of a mesh group.

Use the **no** form of this command to remove an MSDP peer's membership from a mesh group.

Examples

The following example configures the MSDP peer with the IP address, 10.10.10.1, to be a member of the mesh group, foo:

```
[local]Redback(config-ctx)#router msdp  
[local]Redback(config-msdp)#mesh-group foo 10.10.10.1
```

Related Commands

default-peer

multicast destination

multicast destination [*if-name* *ctx-name* [**group-list** *acl-name*]]

no multicast destination

Purpose

Enables the forwarding of multicast data for Internet Group Management Protocol (IGMP) messages received on the Point-to-Point Protocol over Ethernet (PPPoE) subscriber circuits on an out-of-band (separated from the PPPoE circuit) IP over Ethernet (IPoE) interface.

Command Mode

IGMP service profile configuration

Syntax Description

<i>if-name</i>	Optional. Multicast-enabled interface name.
<i>ctx-name</i>	Optional. Context name in which the multicast-enabled interface resides.
group-list <i>acl-name</i>	Optional. Name of the access control list (ACL) used to filter IGMP control messages.

Default

Forwarding multicast data on an out-of-band IPoE interface is disabled.

Usage Guidelines

Use the **multicast destination** command to enable the forwarding of multicast data for IGMP messages received on the PPPoE subscriber circuits on an out-of-band IPoE interface.

The IGMP service profile must be bound to a subscriber record through a configuration or a Remote Authentication Dial-In User Service (RADIUS) attribute.

Note For the **multicast destination** command to work properly, the out-of-band IPoE interface on which the multicast data is to be forwarded must be multicast-enabled; use the **multicast output** command (in interface configuration mode) to enable the out-of-band IPoE interface to forward multicast data.

Use the **no** form of this command to disable the forwarding of multicast data for IGMP messages received on the PPPoE subscriber circuits on an out-of-band IPoE interface.

Examples

The following example enables the `to_dslam5` interface on the `local` context to forward multicast data, and configures the `foo` IGMP service profile to enable the forwarding of multicast data received on a PPPoE subscriber circuit on the `to_dslam5` interface:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#interface to_dslam5
[local]Redback(config-if)#multicast output
```

Command Descriptions

```
[local]Redback(config-if)#exit  
[local]Redback(config-ctx)#igmp service-profile foo  
[local]Redback(config-igmp-service-profile)#multicast destination to_dslam5
```

Related Commands

multicast output

multicast output

multicast output [accept-unknown-mac]

no multicast output [accept-unknown-mac]

Purpose

Enables an interface to forward multicast data, and to send and receive Internet Group Management Protocol (IGMP) control messages.

Command Mode

interface configuration

Syntax Description

accept-unknown-mac	Optional. Accepts IGMP control packets with unknown medium access control (MAC) addresses.

Default

No interface is enabled for multicast data.

Usage Guidelines

Use the **multicast output** command to enable an interface to forward multicast data, and to send and receive IGMP control messages.

An IP over Ethernet (IPoE) circuit, on a Gigabit Ethernet port or an 802.1Q permanent virtual circuit (PVC) configured on it, must be configured on the interface to carry the multicast services. The MAC addresses received from IGMP control packets on the IPoE circuit are compared to the subscriber's MAC address received on the corresponding Point-to-Point Protocol over Ethernet (PPPoE) circuit. By default, if the MAC addresses do not match, the IGMP control packet is dropped. Use the **accept-unknown-mac** keyword to accept IGMP control packets that have MAC addresses that do not match the subscriber's MAC address.

Note The **multicast output** command only enables an interface for multicast services; the **multicast destination** command (in IGMP service profile configuration mode) must also be configured to enable the forwarding of multicast data for IGMP messages received on the PPPoE subscriber circuits on the multicast-enabled interface. A single multicast-enabled interface carry multicast data for multiple IGMP service profiles with configured multicast destinations.

Use the **no** form of this command to disable an interface from forwarding multicast data, and from sending and receiving IGMP control messages.

Examples

The following example enables the `to_dslam5` interface on the `local` context to forward multicast data, and configures the `foo` IGMP service profile to enable the forwarding of multicast data received on a PPPoE subscriber circuit on the `to_dslam5` interface:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#interface to_dslam5
[local]Redback(config-if)#multicast output accept-unknown-mac
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#igmp service-profile foo
[local]Redback(config-igmp-service-profile)#multicast destination to_dslam5
```

Related Commands

multicast destination

originating-rp

originating-rp *if-name*

no originating-rp *if-name*

Purpose

Configures an interface as the originating rendezvous point (RP) address.

Command Mode

MSDP router configuration

Syntax Description

<i>if-name</i>	Name of the interface whose IP address is to be used as the originating RP address.

Default

None

Usage Guidelines

Use the **originating-rp** command to configure an interface as the originating RP address. The IP address of the interface is used as the RP address in all source active (SA) messages originated by the router.

Use the **no** form of this command to remove the interface's IP address for the originating RP address.

Examples

The following example configures the interface, ToLan04, to be used as the RP address:

```
[local]Redback(config-msdp)#originating-rp ToLan04
```

Related Commands

default-peer
description
mesh-group
originating-rp sa-filter
peer
peer-as
router msdp
sa-filter
shutdown

originating-rp sa-filter

originating-rp sa-filter *acl-name*

no originating-rp sa-filter *acl-name*

Purpose

Configures an access control list (ACL) to filter incoming source active (SA) messages learned from the local rendezvous point (RP).

Command Mode

MSDP router configuration

Syntax Description

acl-name Name of the ACL used to filter incoming SA messages.

Default

None

Usage Guidelines

Use the **originating-rp sa-filter** command to configure an ACL to filter incoming SA messages learned from the local RP.

Use the **no** form of this command to remove the ACL.

Examples

The following example configures ACL 320 to filter incoming SA messages:

```
[ local ]Redback(config-ctx)#router msdp
[ local ]Redback(config-msdp)#originating-rp sa-filter 320
```

Related Commands

default-peer
description
mesh-group
originating-rp
peer
peer-as
router msdp
sa-filter
shutdown

peer

peer *peer-addr* **local-tcp-source** *if-name*
no peer *peer-addr* **local-tcp-source** *if-name*

Purpose

Configures an Multicast Source Discovery Protocol (MSDP) peer and enters MSDP peer configuration mode.

Command Mode

MSDP router configuration

Syntax Description

<i>peer-addr</i>	IP address of the router that is to be the MSDP peer.
local-tcp-source <i>if-name</i>	Name of the interface whose address becomes the source IP address for Transmission Control Protocol (TCP) connection.

Default

None

Usage Guidelines

Use the **peer** command to configure an MSDP peer and enter MSDP peer configuration mode for peer-specific configurations.

Use the **no** form of this command to delete an MSDP peer.

Examples

The following example configures a router with an IP address of 192.168.1.1 to be an MSDP peer that uses the ToWan12 interface for the TCP connection:

```
[local]Redback(config-ctx)#router msdp  
[local]Redback(config-msdp)#peer 192.168.1.1 local-tcp-source ToWan12  
[local]Redback(config-msdp-peer)#
```

Related Commands

default-peer	peer-as
description	router msdp
mesh-group	sa-filter
originating-rp	shutdown
originating-rp sa-filter	

peer-as

peer-as {*asn* | *nn:nn*}

no peer-as {*asn* | *nn:nn*}

Purpose

Configures a peer's autonomous system number (ASN).

Command Mode

MSDP peer configuration

Syntax Description

<i>asn</i>	Autonomous system number, in integer format, of the autonomous system that includes the peer. The range of values is 1 to 65,535. The subrange 64,512 to 65,535 is reserved for private autonomous systems.
<i>nn:nn</i>	Optional. ASN, in 4-byte integer format, that includes the peer. With 4-byte integer format, the first <i>nn</i> indicates the two higher-order bytes, and the second <i>nn</i> denotes the two lower-order bytes.

Default

None

Usage Guidelines

Use the **peer-as** command to configure a peer's ASN.

Use the **no** form of this command to delete the source active (SA) number from the peer's configuration.

Examples

The following example configures a peer's SA number to 37:

```
[local]Redback(config-msdp)#peer 192.168.1.1 local-tcp-source ToWan12
[local]Redback(config-msdp-peer)#peer-as 37
```

Related Commands

default-peer	peer
description	router msdp
mesh-group	sa-filter
originating-rp	shutdown
originating-rp sa-filter	

pim accept-rp

pim accept-rp *rp-addr* [*acl-name*]

no pim accept-rp *rp-addr*

Purpose

Accepts an IP address as being a valid rendezvous point (RP) address for a specific Internet Group Management Protocol (IGMP) group.

Command Mode

context configuration

Syntax Description

<i>rp-addr</i>	IP address of the RP.
<i>acl-name</i>	Optional. Name of the access control list (ACL) used to filter RP addresses.

Default

None

Usage Guidelines

Use the **pim accept-rp** command to accept an IP address as being a valid RP address for a specific IGMP group.

To determine if the RP should be accepted, the router checks the Group-to-RP mapping cache for a matching entry for the group. If there is a matching entry, the RP is accepted.

Use the *acl-name* argument to compare the RP address to the specified ACL to determine if the filter permits the RP address.

Use the **no** form of this command to remove an accepted RP address.

Examples

The following example configures the router to accept or reject the RP address, 192.168.100.1, as a valid RP:

```
[local]Redback(config)#context isp1
[local]Redback(config-ctx)#pim accept-rp 192.168.100.1
```

Related Commands

ip multicast boundary	pim neighbor-filter
pim bsr-border	pim operation-mode
pim bsr-candidate	pim rp-address
pim dr-priority	pim rp-candidate
pim hello-interval	pim sparse-mode

pim anycast-rp

pim anycast-rp *anycast-addr rp-addr*

no pim anycast-rp *anycast-addr rp-addr*

Purpose

Configures anycast rendezvous point (RP) functionality on a Protocol Independent Multicast-Sparse Mode (PIM-SM) router.

Command Mode

context configuration

Syntax Description

<i>anycast-addr</i>	IP address of the anycast RP set. This is the IP address used by the multicast groups or sources to join or register.
<i>rp-addr</i>	IP address of the router configured with anycast RP. This is the IP address to where the Register messages are forwarded.

Default

Anycast RP is not configured on the router.

Usage Guidelines

Use the **pim anycast-rp** command to configure anycast RP functionality on a PIM-SM router.

Note This command must be configured for each router that belongs to the same anycast RP set in the domain.

Use the **no** form of this command to disable anycast RP functionality on a PIM-SM router.

Examples

The following example configures the IP address for the anycast RP to 10.10.10.20, and the IP address of the router to 192.168.20.34:

```
[local]Redback(config-ctx)#pim anycast-rp 10.10.10.20 192.168.20.34
```

Related Commands

pim sparse-mode

pim bsr-border

pim bsr-border

no pim bsr-border

Purpose

Configures the router to neither send nor receive bootstrap router (BSR) messages.

Command Mode

interface configuration

Syntax Description

This command has no keywords or arguments.

Default

None

Usage Guidelines

Use the **pim bsr-border** command to configure the router to neither send nor receive BSR messages.

Note This command should be configured on routers that connect to bordering Protocol Independent Multicast (PIM) domains to create a PIM domain boundary that blocks the flow of PIM Version 2 (PIMv2) BSR messages across the domain border.

Use the **no** form of this command to resume the flow of BSR messages to and from the router.

Examples

The following example configures the router to neither send nor receive BSR messages:

```
[local]Redback(config-ctx)#interface enet01  
[local]Redback(config-if)#pim bsr-border
```

Related Commands

ip multicast boundary
pim accept-rp
pim bsr-candidate
pim dr-priority
pim hello-interval
pim neighbor-filter
pim operation-mode
pim rp-address
pim rp-candidate
pim sparse-mode

pim bsr-candidate

pim bsr-candidate *if-name hash-mask-len priority*

no pim bsr-candidate *if-name hash-mask-len priority*

Purpose

Configures a router to begin serving as a candidate bootstrap router (C-BSR).

Command Mode

context configuration

Syntax Description

<i>if-name</i>	Unicast rendezvous point (RP) address corresponding to the IP address of the interface to be used by the BSR.
<i>hash-mask-len</i>	Value contained in BSR messages that will be used by all routers to hash (map) to an RP. It is recommended to use a value between 24 and 30.
<i>priority</i>	Value used to specify the BSR election priority among different candidate BSRs. A larger value wins over a smaller value.

Default

None

Usage Guidelines

Use the **pim bsr-candidate** command to configure a router to begin serving as a C-BSR, and participate in the BSR election process. If this router wins the BSR election, all candidate RPs advertise their candidacy to this router. The BSR caches and advertises the RP sets via the Protocol Independent Multicast (PIM) bootstrap messages to the entire PIM domain.

Use the **no** form of this command to decline the router's BSR candidacy.

Examples

The following example configures a router to begin serving as a C-BSR using the interface, `intfe1/1`, with a hash mask length of 27 and a priority of 12:

```
[local]Redback(config)#context isp01
[local]Redback(config-ctx)#pim bsr-candidate intfe1/1 27 12
```

Related Commands

ip multicast boundary	pim neighbor-filter
pim accept-rp	pim operation-mode
pim bsr-border	pim rp-address
pim dr-priority	pim rp-candidate
pim hello-interval	pim sparse-mode

pim dense-mode

pim dense-mode

no pim dense-mode

Purpose

Enables Protocol Independent Multicast-Dense Mode (PIM-DM).

Command Mode

interface configuration

Syntax Description

This command has no keywords or arguments.

Default

None

Usage Guidelines

Use the **pim dense-mode** command to enable PIM-DM on an interface.

Use the **no** form of this command to disable PIM-DM on an interface.

Examples

The following example enables PIM-DM on the interface, southpoint:

```
[local]Redback(config-ctx)#interface southpoint  
[local]Redback(config-if)#pim dense-mode
```

Related Commands

pim sparse-mode

pim dr-priority

pim dr-priority *priority*

no pim dr-priority

Purpose

Specifies the election priority value for a designated router (DR).

Command Mode

interface configuration

Syntax Description

<i>priority</i>	Value used in the DR election process. The router with the highest priority value is elected as the DR.

Default

The default priority value is 1.

Usage Guidelines

Use the **pim dr-priority** command to specify the election priority value for a DR.

Use the **no** form of this command to set the election priority to the default value of 1.

Examples

The following example sets the election priority value to 3:

```
[local]Redback(config-ctx)#interface enet1  
[local]Redback(config-if)#pim dr-priority 3
```

Related Commands

- ip multicast boundary**
- pim accept-rp**
- pim bsr-border**
- pim bsr-candidate**
- pim hello-interval**
- pim neighbor-filter**
- pim operation-mode**
- pim rp-address**
- pim rp-candidate**
- pim sparse-mode**

pim graceful-restart

pim graceful-restart

no pim graceful-restart

default pim graceful-restart

Purpose

Enables Protocol Independent Multicast (PIM) graceful restart on the specified context.

Command Mode

context configuration

Syntax Description

This command has no keywords or arguments.

Default

PIM graceful restart is enabled.

Usage Guidelines

Use the **pim graceful-restart** command to enable PIM graceful restart on the specified context. PIM graceful restart allows the SmartEdge router and its neighbors to continue forwarding multicast packets without disrupting network traffic. Because neighboring routers assist, the SmartEdge router can quickly restart the PIM process without having to recalculate algorithms from scratch.

A generation ID (GenID), used in Hello messages, is generated randomly when the PIM process initially starts, or restarts after a crash. PIM uses the GenID to establish neighbor relationships with other PIM routers in the network. All neighbors that support graceful restart acknowledge the new GenID by sending multicast updates to the restarting neighbor.

The SmartEdge router stores the GenID of every PIM neighbor, and when it detects a new GenID for a neighbor, it performs one of the following functions:

- If the neighbor restarts more than five times within its hello interval hold time, which is 105 seconds by default, PIM defers its neighbor recovery mechanism and generates the following INFO message:

```
Nbr restarted 6 times (> 5) within 105 secs, backoff nbr recovery
```
- If a reverse path forwarding (RPF) neighbor (which is an assert winner) restarts, PIM clears its RPF assert winner information and the RPF reverts back to the original RPF (pointed by unicast routing).
- If a candidate RP neighbor restarts, PIM sends a candidate RP advertisement to the bootstrap router (BSR).

If PIM graceful restart is enabled, the **show configuration pim verbose** command displays `pim graceful restart` in the configuration; however, if it is disabled, the **show configuration pim** command (non-verbose) displays `no pim graceful restart` in the configuration. For more information about the **show configuration pim** and **show configuration pim verbose** commands, see the “IP Multicast Operations” chapter in the *Routing Protocols Operations Guide* for the SmartEdge OS.

Use the **no** form of this command to disable PIM graceful restart.

Use the **default** form of this command to return to the default PIM graceful restart state, which is enabled.

Examples

The following example enables PIM graceful restart on the context, `foo`, where PIM graceful restart had been previously disabled:

```
[local]Redback(config)#context foo
[local]Redback(config-ctx)#pim graceful-restart
```

Related Commands

None

pim hello-interval

pim hello-interval *interval*

no pim hello-interval

Purpose

Sets the Protocol Independent Multicast Version 2 (PIMv2) Hello interval.

Command Mode

interface configuration

Syntax Description

interval Interval, in seconds, at which PIMv2 Hello messages are sent.

Default

The default PIM Hello interval is 30 seconds.

Usage Guidelines

Use the **pim hello-interval** command to set the PIMv2 Hello interval.

Use the **no** form of this command to set the Hello interval to the default value.

Examples

The following example sets the PIM Hello interval to 65 seconds:

```
[local]Redback(config-ctx)#interface enet1  
[local]Redback(config-if)#pim hello-interval 65
```

Related Commands

- ip multicast boundary**
- pim accept-rp**
- pim bsr-border**
- pim bsr-candidate**
- pim dr-priority**
- pim neighbor-filter**
- pim operation-mode**
- pim rp-address**
- pim rp-candidate**
- pim sparse-mode**

pim neighbor-filter

pim neighbor-filter *acl-name*

no pim neighbor-filter

Purpose

Filters Protocol Independent Multicast (PIM) messages from neighbors.

Command Mode

interface configuration

Syntax Description

acl-name

Name of the access control list (ACL) used to filter PIM messages from neighbors.

Default

None

Usage Guidelines

Use the **pim neighbor-filter** command to filter PIM messages from neighbors. PIM messages are accepted only if the neighbor's IP address is permitted by the ACL.

Use the **no** form of this command to accept all PIM messages from neighbors.

Examples

The following example filters PIM messages from neighbors using the Neighbors44 ACL:

```
[local]Redback(config-ctx)#interface enet1  
[local]Redback(config-if)#pim neighbor-filter Neighbors44
```

Related Commands

ip multicast boundary
pim accept-rp
pim bsr-border
pim bsr-candidate
pim dr-priority
pim hello-interval
pim operation-mode
pim rp-address
pim rp-candidate
pim sparse-mode

pim operation-mode

pim operation-mode {standard | legacy}

Purpose

Sets the protocol parameters to be compatible with Protocol Independent Multicast Sparse-Mode (PIM-SM) specifications, or to be compatible with legacy implementations.

Command Mode

interface configuration

Syntax Description

standard	Configures compatibility with PIM-SM specifications.
legacy	Configures compatibility with legacy implementations.

Default

The protocol parameters are compatible with legacy implementations.

Usage Guidelines

Use the **pim operation-mode** command to set the protocol parameters to be compatible with PIM-SM specifications, or to be compatible with legacy implementations, such as traditional Cisco implementations.

Examples

The following example sets the protocol parameters to be compatible with PIM-SM specifications:

```
[local]Redback(config-ctx)#interface enet1  
[local]Redback(config-if)#pim operation-mode standard
```

Related Commands

- ip multicast boundary**
- pim accept-rp**
- pim bsr-border**
- pim bsr-candidate**
- pim dr-priority**
- pim hello-interval**
- pim neighbor-filter**
- pim rp-address**
- pim rp-candidate**
- pim sparse-mode**

pim rp-address

pim rp-address *rp-addr* [*acl-name*]

no pim rp-address *rp-addr*

Purpose

Configures a router with the rendezvous point (RP) address.

Command Mode

context configuration

Syntax Description

<i>rp-addr</i>	IP address of the RP.
<i>acl-name</i>	Optional. Name of the access control list (ACL) used to filter multicast groups using the RP.

Default

None

Usage Guidelines

Use the **pim rp-address** command to configure a router with the RP address for all Internet Group Management Protocol (IGMP) group addresses permitted by an ACL. If an ACL is not specified, this RP address is used for the entire multicast address space.

The **pim rp-address** command is generally used on simple Protocol Independent Multicast sparse mode (PIM-SM) networks where the RP address is manually configured on each router in the network. More complicated networks should use the PIM Version 2 (PIMv2) bootstrap router (BSR) feature, which allows routers on a network to dynamically learn the RP address.

Use the **no** form of this command to remove the RP address from the router.

Examples

The following example configures a router with the RP address of 192.168.200.20:

```
[local]Redback(config)#context isp1
[local]Redback(config-ctx)#pim rp-address 192.168.200.20
```

Related Commands

ip multicast boundary	pim hello-interval
pim accept-rp	pim neighbor-filter
pim bsr-border	pim operation-mode
pim bsr-candidate	pim rp-candidate
pim dr-priority	pim sparse-mode

pim rp-candidate

pim rp-candidate *if-name* [**group-list** *acl-name*]

no pim rp-candidate *if-name*

Purpose

Configures a candidate rendezvous point (C-RP) on an interface.

Command Mode

context configuration

Syntax Description

<i>if-name</i>	Name of the interface to be used by the C-RP.
group-list <i>acl-name</i>	Optional. Name of the access control list (ACL) used to filter Internet Group Management Protocol (IGMP) group IP addresses.

Default

None

Usage Guidelines

Use the **pim rp-candidate** command to configure a C-RP on an interface for group address ranges permitted by an ACL. If an ACL is not specified, this RP address is used for the entire multicast address space.

Use the **no** form of this command to decline the C-RP's candidacy from the interface.

Examples

The following example configures a C-RP on the interface, Loopback22:

```
[ local ]Redback(config)#context isp1
[ local ]Redback(config-ctx)#pim rp-candidate loopback22
```

Related Commands

ip multicast boundary	pim hello-interval
pim accept-rp	pim neighbor-filter
pim bsr-border	pim operation-mode
pim bsr-candidate	pim rp-address
pim dr-priority	pim sparse-mode

pim sparse-mode

pim sparse-mode [passive]

no pim sparse-mode [passive]

Purpose

Enables Protocol Independent Multicast Sparse-Mode (PIM-SM).

Command Mode

interface configuration

Syntax Description

passive

Optional. Specifies that no PIM messages are exchanged out of the interface, but the interface, or circuits belonging to the interface, can be populated in a multicast forwarding entry by receiving an Internet Group Management Protocol (IGMP) report or a data packet.

Default

None

Usage Guidelines

Use the **pim sparse-mode** command to enable PIM-SM on an interface.

Use the **no** form of this command to disable PIM-SM on an interface.

Examples

The following example enables PIM-SM on the interface, Northpoint:

```
[local]Redback(config-ctx)#interface Northpoint  
[local]Redback(config-if)#pim sparse-mode
```

Related Commands

ip multicast boundary
pim accept-rp
pim bsr-border
pim bsr-candidate
pim dr-priority
pim hello-interval
pim neighbor-filter
pim operation-mode
pim rp-address
pim rp-candidate

pim spt-threshold infinity

pim spt-threshold infinity [**group-list** *acl*]

no pim spt-threshold infinity [**group-list** *acl*]

Purpose

Enables a Protocol Independent Multicast-Sparse Mode (PIM-SM) leaf router to continue using a shared tree, instead of switching to a shortest-path tree (SPT).

Command Mode

context configuration

Syntax Description

group-list *acl*

Optional. Groups permitted by the access control list (ACL) to stay on the shared tree. If the **group-list** *acl* construct is not used, or if the *acl* value is 0, the threshold applies to all groups.

Default

The SPT threshold is set to 0, and the switchover occurs immediately after the initial transmission has been established.

Usage Guidelines

Use the **pim spt-threshold infinity** command to enable a PIM-SM leaf router to continue using a shared tree, instead of switching to an SPT.

A multicast source initially sends traffic using the shared tree; however, after transmitting a certain number of bits (the SPT threshold), the PIM-SM router switches from using the shared tree to using the SPT. Using the **pim spt-threshold infinity** command sets the SPT threshold infinitely high, making it impossible for the switchover to occur.

Use the **no** form of this command to allow a PIM-SM leaf router to switch from a shared tree to an SPT.

Examples

The following example enables a PIM-SM leaf router to continue using a shared tree:

```
[local]Redback(config-ctx)#pim spt-threshold infinity
```

Related Commands

pim sparse-mode

pim ssm

```
pim ssm {default | range acl-name}
no pim ssm {default | range acl-name}
```

Purpose

Enables source-specific multicast (SSM) routing on the specified context.

Command Mode

context configuration

Syntax Description

default	Specifies a default SSM address range of 232.0.0.0/8.
range acl-name	Access control list (ACL) used to specify the SSM address range.

Default

The default SSM address range is 232.0.0.0/8.

Usage Guidelines

Use the **pim ssm** command to enable SSM routing on the specified context.

The SSM feature is an extension of multicast routing where traffic is forwarded to receivers from only those multicast sources to which the receivers have explicitly joined. For multicast groups configured to use SSM, only source-specific multicast distribution trees are created, and not shared trees.

Protocol Independent Multicast-SSM (PIM-SSM) is the routing protocol that supports the implementation of SSM and is derived from PIM sparse mode (PIM-SM). SSM is supported by IGMPv3.

The address range 232.0.0.0 to 232.255.255.255 is reserved for SSM applications and protocols. Existing IP multicast receivers cannot receive traffic when trying to use addresses in a defined SSM range, unless they are SSM enabled.

For more information on SSM routing, see the Internet Draft, *Source-Specific Multicast for IP*, draft-ietf-ssm-arch-00.txt.

Use the **no** form of this command to disable SSM routing on an interface.

Examples

The following example enables SSM routing on the `local` context using the `default` address range of 232.0.0.0/8:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#pim ssm default
```

Related Commands

None

pim static group

pim static group *group-addr* [**oif** *if-name* | **register** | **source** *ip-addr* [**oif** *if-name* | **register**]]

no pim static group *group-addr* [**oif** *if-name* | **register** | **source** *ip-addr* [**oif** *if-name* | **register**]]

Purpose

Creates a static multicast route, (*,G) or (S,G), with the specified interface as the outgoing interface (OIF).

Command Mode

context configuration

Syntax Description

<i>group-addr</i>	Multicast group IP address.
oif <i>if-name</i>	OIF name.
register	Enables the first-hop router to send register messages to the rendezvous point (RP).
source <i>ip-addr</i>	Multicast source IP address.

Default

No static multicast routes are created.

Usage Guidelines

Use the **pim static group** command to create a static multicast route, (*,G) or (S,G), with the specified interface as the OIF.

Note Protocol Independent Multicast (PIM) normally creates dynamic multicast routes; the **pim static group** command allows you to create static multicast routes.

An OIF is an outgoing circuit that receives traffic destined for a given multicast group. For this command, the OIF is a regular interface. For multibind interface OIFs, configure the **static-group** command in an Internet Group Management Protocol (IGMP) service profile that is bound to a subscriber (default) profile.

Use the **register** keyword to configure multicast static groups on the first-hop router, which is the router directly connected to the multicast source, so that this router can send register messages to the RP.

Use the **no** form of this command to delete the static multicast route.

Examples

The following example creates a static multicast route, 224.1.1.1, with fxp1 as its OIF:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#pim static group 224.1.1.1 oif fxp1
```

Related Commands

static-group

priority

priority *priority*

no priority

Purpose

Configures the priority of the interface when the maximum bandwidth in the service profile has been exhausted.

Command Mode

IGMP service profile configuration

Syntax Description

priority Priority setting for the interface. The range of values is 0 to 10.

Default

The interface has no priority setting.

Usage Guidelines

Use the **priority** command to configure the priority of the interface when the maximum bandwidth in the service profile has been exhausted.

When the addition of a new group would cause the maximum bandwidth, as specified by the **igmp maximum-bandwidth** command, to be exceeded on the port, the router searches for subscribers joined on the same port with a lower priority. The router drops the lower priority subscribers and reclaims their bandwidth until it gets enough bandwidth to service the higher priority subscriber. If it cannot reclaim enough bandwidth the new group join will be dropped.

Use the **no** form of this command to delete the priority setting for the interface.

Examples

The following example configures a priority of 8 for the interface:

```
[local]Redback(config-ctx)#igmp service-profile bar
[local]Redback(config-igmp-service-profile)#priority 8
```

Related Commands

igmp group-bandwidth	instant-leave
igmp maximum-bandwidth	max-groups
igmp service-profile	static-group
igmp version	sticky-groups

router msdp

router msdp

no router msdp

Purpose

Enables Multicast Source Discovery Protocol (MSDP) within a context and enters MSDP router configuration mode.

Command Mode

context configuration

Syntax Description

This command has no keywords or arguments.

Default

MSDP is disabled.

Usage Guidelines

Use the **router msdp** command to enable MSDP within a context and enter MSDP router configuration mode.

Use the **no** form of this command to disable MSDP within a context.

Examples

The following example enables MSDP and enters MSDP router configuration mode:

```
[local]Redback(config-ctx)#router msdp
[local]Redback(config-msdp)#
```

Related Commands

default-peer
description
mesh-group
originating-rp
originating-rp sa-filter
peer
peer-as
sa-filter
shutdown

sa-filter

sa-filter [**in** | **out**] *acl-name*

no sa-filter [**in** | **out**] *acl-name*

Purpose

Specifies an access control list (ACL) to filter source active (SA) messages coming in to, or going out of, the peer.

Command Mode

MSDP peer configuration

Syntax Description

in	Optional. Filters incoming SA messages only.
out	Optional. Filters outgoing SA messages only.
<i>acl-name</i>	Name of the ACL used to filter SA messages.

Default

None

Usage Guidelines

Use the **sa-filter** command to specify an ACL to filter SA messages coming in to, or going out of, the peer.

Use the **no** form of this command to remove the SA filter.

Examples

The following example filters incoming SA messages from a peer using the ACL, `peer-sa-filter-in-group`:

```
[local]Redback(config-ctx)#ip access-list peer-sa-filter-in-group
[local]Redback(config-access-list)#seq 10 deny ip any 224.137.0.0 0.0.255.255
[local]Redback(config-access-list)#seq 20 deny ip any 224.134.1.0 0.0.0.255
[local]Redback(config-access-list)#seq 30 deny ip any host 224.131.1.1
[local]Redback(config-access-list)#seq 40 permit any any

[local]Redback(config-ctx)#router msdp
[local]Redback(config-msdp)#peer 10.200.1.2 local-tcp-source lo1
[local]Redback(config-msdp-peer)#sa-filter in peer-sa-filter-in-group
```

The following example filters outgoing SA messages to a peer using the ACL, `peer-sa-filter-out-source-group`:

```
[local]Redback(config-ctx)#ip access-list peer-sa-filter-out-source-group
[local]Redback(config-access-list)#seq 10 deny ip 44.1.1.0 0.0.0.255 host 224.133.1.2
```

Command Descriptions

```
[local]Redback(config-access-list)#seq 20 deny ip 44.1.1.0 0.0.0.255 224.136.2.0
0.0.0.255
[local]Redback(config-access-list)#seq 30 permit ip any any

[local]Redback(config-ctx)#router msdp
[local]Redback(config-msdp)#peer 10.200.1.2 local-tcp-source lo1
[local]Redback(config-msdp-peer)#sa-filter out peer-sa-filter-out-source-group
```

Related Commands

- default-peer
- description
- mesh-group
- originating-rp
- originating-rp sa-filter
- peer
- peer-as
- router msdp
- shutdown

shutdown

shutdown
no shutdown

Purpose

Disables a configured Multicast Source Discovery Protocol (MSDP) peer.

Command Mode

MSDP peer configuration

Syntax Description

This command has no keywords or arguments.

Default

The peer is up when configured.

Usage Guidelines

Use the **shutdown** command to disable a configured MSDP peer.

Use the **no** form of this command to bring up a configured MSDP peer.

Examples

The following example disables an MSDP peer:

```
[local]Redback(config-ctx)#router msdp  
[local]Redback(config-msdp)#peer 10.200.1.2 local-tcp-source lo1  
[local]Redback(config-msdp-peer)#shutdown
```

Related Commands

default-peer
description
mesh-group
originating-rp
originating-rp sa-filter
peer
peer-as
router msdp
sa-filter

static-group

static-group *group-addr source-addr*

no static-group *group-addr source-addr*

Purpose

Creates a static multicast route, (*,G) or (S,G), with a subscriber circuit as the outgoing interface (OIF).

Command Mode

IGMP service profile configuration

Syntax Description

<i>group-addr</i>	Multicast group IP address.
<i>source-addr</i>	Multicast source IP address.

Default

None

Usage Guidelines

Use the **static-group** command to create a static multicast route, (*,G) or (S,G), with a subscriber circuit as the OIF.

Note Protocol Independent Multicast (PIM) normally creates dynamic multicast routes; the **static-group** command allows you to create static multicast routes.

An OIF is an outgoing circuit that receives traffic destined for a given multicast group. When the static multicast route is configured in IGMP service profile configuration mode, the OIF is a subscriber circuit.

To configure all subscriber circuits on a multibind interface to receive multicast traffic for a specified multicast group, configure the **static-group** command in an Internet Group Management Protocol (IGMP) service profile that is bound to a subscriber (default) profile.

Use the **no** form of this command to delete the static multicast route.

Examples

The following example creates a static multicast route, 10.10.10.1 20.20.20.0, for IGMP service profile, pro78, and then applies the service profile to the default subscriber profile:

```
[local]Redback(config-context)#igmp service-profile pro78
[local]Redback(config-igmp-service-profile)#static-group 10.10.10.1 20.20.20.2
[local]Redback(config-igmp-service-profile)#exit
[local]Redback(config-context)#subscriber default
[local]Redback(config-sub)#ip igmp service-profile pro78
```

Related Commands

- igmp service-profile**
- instant-leave**
- max-groups**
- pim static group**
- priority**
- sticky-groups**

sticky-groups

sticky-groups *acl-name*

no sticky-groups

Purpose

Enables Internet Group Management Protocol (IGMP) groups to be sticky.

Command Mode

IGMP service profile configuration

Syntax Description

acl-name Access control list (ACL) of groups to be sticky.

Default

Sticky groups are disabled.

Usage Guidelines

Use the **sticky-groups** command to enable IGMP groups to be sticky.

Groups defined by the ACL will never be dropped, unless an explicit leave for that group is received.

Use the **no** form of this command to disable sticky groups.

Examples

The following example enables IGMP groups, as specified by the ACL, foo3, to be sticky:

```
[local]Redback(config-ctx)#igmp service-profile bar  
[local]Redback(config-igmp-service-profile)#sticky-groups foo3
```

Related Commands

igmp group-bandwidth
igmp maximum-bandwidth
igmp service-profile
igmp version
instant-leave
max-groups
priority
static-group

Routing Policy Configuration

This chapter provides an overview of routing policies and describes the tasks and commands used to configure routing policy features through the SmartEdge® OS.

For information about the tasks and commands used to monitor, troubleshoot, and administer routing policy features, see the “Routing Policy Operations” chapter in the *Routing Protocols Operations Guide* for the SmartEdge OS.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples
- Command Descriptions

Overview

Routing policies allow you to enforce routing policy decisions onto incoming, outgoing, and redistributed routes. The tools to configure routing policies include Border Gateway Protocol (BGP) autonomous system (AS) path lists, BGP community lists, BGP extended community lists, IP prefix lists, IP Version 6 (IPv6) prefix lists, and route maps with match and set conditions.

Note When IP Version 6 (IPv6) addresses are not referenced or explicitly specified, the term, IP address, can refer generally to IP Version 4 (IPv4) addresses, IPv6 addresses, or IP addressing. In instances where IPv6 addresses are referenced or explicitly specified, the term, IP address, refers only to IPv4 addresses.

Configuration Tasks

Note In this section, the command syntax in the task tables displays only the root command; for the complete command syntax, see the full description for the command in the “Command Descriptions” section.

To configure routing policies, perform the tasks described in the following sections:

- Configuring AS Path Lists
- Configuring BGP Community Lists
- Configuring BGP Extended Community Lists
- Configuring IP Prefix Lists
- Configuring IPv6 Prefix Lists
- Configuring Route Maps
- Configuring BGP Attribute-Based Accounting
- Configuring BGP Destination-Based QoS

Configuring AS Path Lists

To configure BGP AS path lists, perform the tasks described in the following sections:

- Create an AS Path List
- Configure an AS Path List Permit or Deny Condition

Create an AS Path List

To create an AS path list, perform the tasks described in Table 12-1.

Table 12-1 Create an AS Path List

Task	Root Command	Notes
Create an AS path list and enter AS path list configuration mode.	as-path-list	Enter this command in context configuration mode.
Associate a description with the BGP AS path list.	description	Enter this command in AS path list configuration mode.
Configure the AS path list permit or deny condition.	For the complete list of tasks used to configure the AS path list permit or deny condition, see the “Configure an AS Path List Permit or Deny Condition” section.	

Configure an AS Path List Permit or Deny Condition

When you create several permit or deny conditions for a single list, the system can automatically sequence the entries for you, or you can manually assign a number for each entry. A BGP AS path attribute is compared with BGP AS path list entries in order of ascending sequence number to determine if routes associated with the AS path attribute are permitted or denied.

When you allow the system to automatically sequence entries for you, the system increments each statement by a count of 10. The first statement you enter is assigned the sequence number of 10, the second is assigned the number 20, and so on. This allows room to assign intermediate sequence numbers to statements that you might want to add later. You can also resequence numbers to existing entries in an AS path list.

To configure an AS path list permit or deny condition, perform the tasks described in Table 12-2. Enter all commands in AS path list configuration mode, unless otherwise noted.

Table 12-2 Configure an AS Path List Permit or Deny Condition

Task	Root Command	Notes
Permit or deny routes matching the specified criteria, and allow the system to automatically assign sequence numbers for the AS path list statement.	{permit deny}	Use the following command syntax: {permit deny} {reg-exp any}
Permit or deny routes matching the specified criteria, and manually assign a sequence number for the AS path list statement.	{permit deny}	Use the following command syntax: seq seq-num {permit deny} {reg-exp any}
Assign new sequence numbers to existing entries in a specified AS path list, so that entries are in increments of 10.	resequence as-path-list	Enter this command in context configuration mode. This command is useful when you have manually assigned sequence numbers and have no room to insert new entries in between existing entries.

Configuring BGP Community Lists

To configure BGP community lists, perform the tasks described in the following sections:

- Create a BGP Community List
- Configure a BGP Community List Permit or Deny Condition

Create a BGP Community List

To create a BGP community list, perform the tasks described in Table 12-3.

Table 12-3 Create a BGP Community List

Task	Root Command	Notes
Create a BGP community list and enter community list configuration mode.	community-list	Enter this command in context configuration mode. A reference to a community list that does not exist, or does not contain any configured entries, implicitly matches and permits all community lists.
Associate a description with the BGP community list.	description	Enter this command in community list configuration mode.
Configure the BGP community list permit or deny condition.	For the complete list of tasks used to configure the BGP community list permit or deny condition, see the "Configure a BGP Community List Permit or Deny Condition" section.	

Configure a BGP Community List Permit or Deny Condition

When you create several permit or deny conditions for a single BGP community list, the system can automatically sequence the entries for you, or you can manually assign a number for each entry. A BGP community attribute is compared with BGP community list entries in order of ascending sequence number to determine if they are permitted or denied.

When you allow the system to automatically sequence entries, the system increments each statement by a count of 10. The first statement you enter is assigned the sequence number of 10, the second is assigned the number 20, and so on. This allows room to assign intermediate sequence numbers to statements that you might want to add later. You can also resequence existing entries in a BGP community list.

To configure a BGP community list permit or deny condition, perform the tasks described in Table 12-4. Enter all commands in community list configuration mode, unless otherwise noted.

Table 12-4 Configure a BGP Community List Permit or Deny Condition

Task	Root Command	Notes
Permit or deny routes matching the specified criteria, and allow the system to automatically assign sequence numbers for the BGP community list statement.	{permit deny}	Use the following command syntax: {permit deny} {community-num local-as no-advertise no-export reg-exp reg-exp any}
Permit or deny routes matching the specified criteria, and manually assign a sequence number for the BGP community list statement.	{permit deny}	Use the following command syntax: seq seq-num {permit deny} {community-num local-as no-advertise no-export reg-exp reg-exp any}
Assign new sequence numbers to existing entries in a BGP community list, so that entries are in increments of 10.	resequence community-list	Enter this command in context configuration mode. This command is useful when you have manually assigned sequence numbers and have no room to insert new entries in between existing entries.

Configuring BGP Extended Community Lists

A BGP extended community is a group of destinations that share some common attributes. Extended community attributes are carried in BGP messages as attributes of the route. They identify the route as belonging to a specific collection of routes, all of which are treated the same with respect to routing policy. Each BGP extended community must be globally unique (contains either a public IP address or autonomous system number [ASN]).

BGP/Multiprotocol Label Switching Virtual Private Networks (BGP/MPLS VPNs) use BGP extended community attributes instead of conventional BGP community attributes.

To configure BGP extended community lists, perform the tasks described in the following sections:

- Create a BGP Extended Community List
- Configure a BGP Extended Community List Permit or Deny Condition

Create a BGP Extended Community List

To create a BGP extended community list, perform the tasks described in Table 12-5.

Table 12-5 Create a BGP Extended Community List

Task	Root Command	Notes
Create a BGP extended community list and enter community list configuration mode.	ext-community-list	Enter this command in context configuration mode. A reference to an extended community list that does not exist, or does not contain any configured entries, implicitly matches and permits all extended community lists.
Associate a description with the BGP extended community list.	description	Enter this command in extended community list configuration mode.
Configure the BGP extended community list permit or deny condition.	For the complete list of tasks used to configure the BGP extended community list permit or deny condition, see the “Configure a BGP Extended Community List Permit or Deny Condition” section.	

Configure a BGP Extended Community List Permit or Deny Condition

When you create several permit or deny conditions for a single BGP extended community list, the system can automatically sequence the entries for you, or you can manually assign a number for each entry. A BGP extended community attribute is compared with BGP extended community list entries in order of ascending sequence number to determine if they are permitted or denied.

When you allow the system to automatically sequence entries for you, the system increments each statement by a count of 10. The first statement you enter is assigned the sequence number of 10, the second is assigned the number 20, and so on. This allows room to assign intermediate sequence numbers to statements that you might want to add later. You can also resequence existing entries in a BGP extended community list.

To configure a BGP extended community list permit or deny condition, perform the tasks described in Table 12-6. Enter all commands in extended community list configuration mode, unless otherwise noted.

Table 12-6 Configure a BGP Extended Community List Permit or Deny Condition

Task	Root Command	Notes
Permit or deny routes matching the specified criteria, and allow the system to automatically assign sequence numbers for the BGP extended community list statement.	{permit deny}	Use the following command syntax: {permit deny} {ext-community-num reg-exp reg-exp any}
Permit or deny routes matching the specified criteria, and manually assign a sequence number for the BGP extended community list statement.	{permit deny}	Use the following command syntax: seq seq-num {permit deny} {ext-community-num reg-exp reg-exp any}
Assign new sequence numbers to existing entries in a BGP extended community list, so that entries are in increments of 10.	resequence ext-community-list	Enter this command in context configuration mode. This command is useful when you have manually assigned sequence numbers and have no room to insert new entries in between existing entries.

Configuring IP Prefix Lists

To configure IP prefix lists, perform the tasks described in the following sections:

- Create an IP Prefix List
- Configure an IP Prefix List Permit or Deny Condition

Create an IP Prefix List

To create an IP prefix list, perform the tasks described in Table 12-7.

Table 12-7 Create an IP Prefix List

Task	Root Command	Notes
Create an IP prefix list used to filter routes and enter IP prefix list configuration mode.	ip prefix-list	Enter this command in context configuration mode. A reference to an IP prefix list that does not exist, or does not contain any configured entries, implicitly matches and permits all IP prefixes.
Associate a description with the IP prefix list.	description	Enter this command in IP prefix list configuration mode.
Configure the IP prefix list permit or deny condition.	For the complete list of tasks used to configure the IP prefix list permit or deny condition, see the “Configure an IP Prefix List Permit or Deny Condition” section.	

Configure an IP Prefix List Permit or Deny Condition

When you create several permit or deny conditions for a single IP prefix list, the system can automatically sequence the entries for you, or you can manually assign a number for each entry.

When you allow the system to automatically sequence the entries for you, the system increments each statement by a count of 10. The first statement you enter is assigned the sequence number of 10, the second is assigned the number 20, and so on. This allows room to assign intermediate sequence numbers to statements that you might want to add later. You can also resequence existing entries in an IP prefix list.

To configure an IP prefix list permit or deny condition, perform the tasks described in Table 12-8. Enter all commands in IP prefix list configuration mode, unless otherwise noted.

Table 12-8 Configure an IP Prefix List Permit or Deny Condition

Task	Root Command	Notes
Permit or deny routes matching the specified criteria, and allow the system to automatically assign sequence numbers for the IP prefix list statement.	{permit deny}	Use the following command syntax: {permit deny} {ip-addr/prefix-length} [[eq eq-value ge ge-value le le-value]] any
Permit or deny routes matching the specified criteria, and manually assign a sequence number for the IP prefix list statement.	{permit deny}	Use the following command syntax: seq seq-num {permit deny} {ip-addr/prefix-length} [[eq eq-value ge ge-value le le-value]] any
Assign new sequence numbers to existing entries in an IP prefix list, so that entries are in increments of 10.	resequence ip prefix-list	Enter this command in context configuration mode. This command is useful when you have manually assigned sequence numbers and have no room to insert new entries in between existing entries.

Configuring IPv6 Prefix Lists

To configure IP Version 6 (IPv6) prefix lists, perform the tasks described in the following sections:

- Create an IPv6 Prefix List
- Configure an IPv6 Prefix List Permit or Deny Condition

Create an IPv6 Prefix List

To create an IPv6 prefix list, perform the tasks described in Table 12-9.

Table 12-9 Create an IPv6 Prefix List

Task	Root Command	Notes
Create an IPv6 prefix list used to filter routes and enter IPv6 prefix list configuration mode.	ipv6 prefix-list	Enter this command in context configuration mode. A reference to an IPv6 prefix list that does not exist, or does not contain any configured entries, implicitly matches and permits all IPv6 prefixes.
Associate a description with the IPv6 prefix list.	description	Enter this command in IPv6 prefix list configuration mode.
Configure the IPv6 prefix list permit or deny condition.	For the complete list of tasks used to configure the IPv6 prefix list permit or deny condition, see the “Configure an IPv6 Prefix List Permit or Deny Condition” section.	

Configure an IPv6 Prefix List Permit or Deny Condition

When you create several permit or deny conditions for a single IPv6 prefix list, the system can automatically sequence the entries for you, or you can manually assign a number for each entry.

When you allow the system to automatically sequence the entries for you, the system increments each statement by a count of 10. The first statement you enter is assigned the sequence number of 10, the second is assigned the number 20, and so on. This allows room to assign intermediate sequence numbers to statements that you might want to add later. You can also resequence existing entries in an IPv6 prefix list.

To configure an IPv6 prefix list permit or deny condition, perform the tasks described in Table 12-10. Enter all commands in IPv6 prefix list configuration mode, unless otherwise noted.

Table 12-10 Configure an IPv6 Prefix List Permit or Deny Condition

Task	Root Command	Notes
Permit or deny routes matching the specified criteria, and allow the system to automatically assign sequence numbers for the IPv6 prefix list statement.	{permit deny}	Use the following command syntax: {permit deny} {ip-addr/prefix-length} [{eq eq-value ge ge-value le le-value}] any
Permit or deny routes matching the specified criteria, and manually assign a sequence number for the IPv6 prefix list statement.	{permit deny}	Use the following command syntax: seq seq-num {permit deny} {ip-addr/prefix-length} [{eq eq-value ge ge-value le le-value}] any
Assign new sequence numbers to existing entries in an IPv6 prefix list, so that entries are in increments of 10.	resequence ipv6 prefix-list	Enter this command in context configuration mode. This command is useful when you have manually assigned sequence numbers and have no room to insert new entries in between existing entries.

Configuring Route Maps

When you configure route maps, you configure the route map name, and optionally, associate a description with the route map. You can also assign a sequence number to the route map, and permit or deny routes that use a specific sequence number.

After you create a route map, configure the match conditions that are looked at by the system when sending and receiving routes, and configure the set conditions that determine the action the system takes once a match for a route is found.

To configure route maps, perform the tasks described in the following sections:

- Create a Route Map
- Configure a Match Condition
- Configure a Set Condition

Create a Route Map

To create a route map, perform the tasks described in Table 12-11.

Table 12-11 Create a Route Map

Task	Root Command	Notes
Create a route map and implement a routing policy, and enter route map configuration mode.	route-map	Enter this command in context configuration mode. You can specify a sequence number for the route map entry, relative to other route map entries in the same route map. Route map entries are tested in order of ascending sequence number. That is, the route map entry with the lowest sequence number is examined first when routes are tested. A reference to a route map that does not exist, or does not contain any configured entries, implicitly matches and permits all routes.
Assign new sequence numbers to existing entries in a specified route map, so that entries are in increments of 10.	resequence route-map	Enter this command in route map configuration mode.
Configure the match condition.	For the complete list of tasks used to configure the match condition, see the “Configure a Match Condition” section.	
Configure the set condition.	For the complete list of tasks used to configure the set condition, see the “Configure a Set Condition” section.	

Configure a Match Condition

To configure a match condition, perform the tasks described in Table 12-12. Enter all commands in route map configuration mode, unless otherwise noted.

Table 12-12 Configure a Match Condition

Task	Root Command	Notes
Permit or deny routes with an associated BGP AS path attribute that matches the specified BGP AS path list.	match as-path-list	

Table 12-12 Configure a Match Condition (*continued*)

Task	Root Command	Notes
Permit or deny routes with an associated BGP community attribute that matches the specified community list.	match community-list	
Permit or deny routes with an associated BGP extended community attribute that matches the specified extended community list.	match ext-community-list	
Permit or deny routes that have a destination IP address permitted by a specified IP prefix list.	match ip address prefix-list	
Permit or deny routes with a next-hop IP address that is permitted by a specified IP prefix list.	match ip next-hop prefix-list	
Permit or deny routes that have a destination IPv6 address permitted by a specified IPv6 prefix list.	match ipv6 address prefix-list	
Permit or deny routes with a next-hop IPv6 address that is permitted by a specified IPv6 prefix list.	match ipv6 next-hop prefix-list	
Permit or deny routes with a specific metric value.	match metric	
Permit or deny routes that match a specific route type.	match route-type	
Permit or deny routes that match a specific route tag value.	match tag	

Configure a Set Condition

To configure a set condition, perform the tasks described in Table 12-13. Enter all commands in route map configuration mode, unless otherwise noted.

Table 12-13 Configure a Set Condition

Task	Root Command	Notes
Prepend an AS path to BGP routes that pass the route map conditions.	set as-path	The only global BGP metric available to influence the best path selection is the AS path length. Usually the local AS number is prepended multiple times, increasing the AS path length.
Set the BGP community attribute for routes that pass the route map conditions.	set community	A community is a group of destinations that share some common attributes. Each destination can belong to multiple communities. Up to eight communities can be specified. If the additive keyword is used, communities are added to the existing BGP community list. However, unlike AS path attributes, community attributes do not include duplicate entries.
Delete BGP communities matching the community list from the BGP community attribute for routes that pass the route map conditions.	set community-list	

Table 12-13 Configure a Set Condition (*continued*)

Task	Root Command	Notes
Set the BGP extended community attribute for routes that pass the route map conditions.	set ext-community	An extended community is a group of destinations that share some common attributes. Each destination can belong to multiple extended communities. Up to eight extended communities can be specified. If the additive keyword is used, extended communities are added to the existing BGP extended community list; however, unlike AS path attributes, extended community attributes do not include duplicate entries.
Set the BGP route dampening policy for routes that pass the route map conditions.	set dampening	
Set the next-hop IP address used to forward packets for routes that pass the route map conditions.	set ip next-hop	
Set the next-hop IPv6 address used to forward packets for routes that pass the route map conditions.	set ipv6 next-hop	
Set the MPLS label for routes that pass the route map conditions.	set label	
Set the advertisement scope for routes redistributed into Open Shortest Path First (OSPF) and Intermediate System-to-Intermediate System (IS-IS) routing domains for routes that pass the route map conditions.	set level	
Set the degree of preference for the BGP AS path for routes that pass the route map conditions.	set local-preference	
Set, increment, or decrement the metric value for routes passing the route map condition.	set metric	
Set the metric type for routes passing the route map condition.	set metric-type	
Set the origin of the BGP path for routes that pass the route map conditions.	set origin	
Set the route tag value for routes that pass the route map condition.	set tag	
Set the degree of preference for BGP routes that pass the route map conditions.	set weight	

Configuring BGP Attribute-Based Accounting

Traffic index counters are maintained on interfaces with traffic index accounting enabled. Traffic indexes are associated with BGP routes based on route-maps matching on BGP attributes. When IP packets are received on an interface with traffic index accounting enabled, and the route lookup for the packet's destination IP address corresponds to a BGP route with a traffic index assigned, the corresponding byte and packet counters are incremented.

To configure BGP attribute-based accounting, perform the tasks described in Table 12-14.

Table 12-14 Configure BGP Attribute-Based Accounting

Task	Root Command	Notes
Set the traffic index value for routes that pass the route map conditions.	set traffic-index	Enter this command in route map configuration mode.
Assign a traffic index to routes installed for a BGP address family.	table-map	Enter this command in BGP address family configuration mode. To determine the attribute modifications and filtering conditions of the applied route map, use the route-map command in context configuration mode. For more information about this command, see Chapter 8, “BGP Configuration.”
Enables BGP attribute-based accounting on an interface.	traffic-index accounting	Enter this command in interface configuration mode.

Configuring BGP Destination-Based QoS

BGP destination-based quality of service QoS provides multiple levels of service based on a customer’s IP destination. BGP routes can be assigned a Differentiated Services Code Point (DSCP) value based on the BGP traffic indexing and table map features associated with route maps. This feature is useful to treat traffic differently depending on which policy it matches.

If a packet’s destination matches a BGP route configured in a route map that contains a **set dscp** statement, and that route map is enabled via the **table-map** command in BGP address family configuration mode, and the ingress interface of the packet is enabled via the **mark dscp destination** command in interface configuration mode, the packet is marked according to the statement defined by the **set dscp** statement of the route map.

To configure BGP destination-based QoS, perform the tasks described in Table 12-15.

Table 12-15 Configure BGP Destination-Based QoS

Task	Root Command	Notes
Set the DSCP value for routes that pass route map conditions.	set dscp	Enter this command in route map configuration mode. BGP routes can be assigned a DSCP value based on the BGP table-map route-map. When a packet is received on an interface with mark dscp destination enabled, and the packet is routed using a route with an associated DSCP, the packet's DSCP is updated and the IP header checksum is re-calculated.
Assign the DSCP value to routes installed for a BGP address family.	table-map	Enter this command in BGP address family configuration mode. For more information about this command, see Chapter 8, “BGP Configuration.”
Set the DSCP byte, based on BGP attributes, such as community list and autonomous AS path, for incoming IP traffic on the specified interface.	mark dscp destination	Enter this command in interface configuration mode. BGP destination based QoS supports setting the DSCP byte for IP traffic based on BGP attributes including community list and AS path. This can be used by a service provider (SP) to provide multiple levels of service based on a customers IP destination.

Configuration Examples

This section provides the following configuration examples:

- Simple IP Prefix List
- Complex IP Prefix List
- Simple AS Path List
- Complex AS Path List
- Simple Community List
- Complex Community List
- Simple Route Map
- Complex Route Map
- BGP Attribute-Based Accounting
- BGP Destination-Based QoS

Simple IP Prefix List

The following example configures a simple IP prefix list that allows routes from networks 128.141.1.0/24, 129.142.2.0/24, and 130.143.3.0/24. The last prefix list entry (sequence 40) is optional, because denial is the default action for any prefix not explicitly specified.

```
[local]Redback(config-ctx)#ip prefix-list simple-prefix-list
[local]Redback(config-prefix-list)#seq 10 permit 128.141.1.0/24
[local]Redback(config-prefix-list)#seq 20 permit 129.142.2.0/24
[local]Redback(config-prefix-list)#seq 30 permit 130.143.3.0/24
[local]Redback(config-prefix-list)#seq 40 deny 0.0.0.0/0
```

The following example applies the IP prefix list, simple-prefix-list, to BGP neighbor, 192.100.100.1, as a BGP inbound route filter:

```
[local]Redback(config-ctx)#router bgp 100
[local]Redback(config-bgp)#neighbor 192.100.100.1 external
[local]Redback(config-neighbor)#address-family ipv4 unicast
[local]Redback(config-addrfamily)#prefix-list simple-prefix-list in
```

Complex IP Prefix List

This section contains an example of a more complex IP prefix list that allows routes from the following subnetworks:

- Any subnet in the class A network 10 with a prefix length greater than 16 and less than 20
- Any subnet in the class A network 11 with a prefix length exactly equal to 24
- Any subnet or host address in the class A network 12

The IP prefix list configuration is as follows:

```
[local]Redback(config-ctx)#ip prefix-list complex-prefix-list
[local]Redback(config-prefix-list)#seq 10 permit 10.0.0.0/8 ge 16 le 20
[local]Redback(config-prefix-list)#seq 20 permit 11.0.0.0/8 eq 24
[local]Redback(config-prefix-list)#seq 30 permit 12.0.0.0/8 le 32
[local]Redback(config-prefix-list)#seq 40 deny 0.0.0.0/0
```

The following example applies the complex-prefix-list IP prefix list to BGP neighbor, 192.100.101.5, as a BGP outbound route filter:

```
[local]Redback(config-ctx)#router bgp 100
[local]Redback(config-bgp)#neighbor 192.100.101.5 external
[local]Redback(config-neighbor)#address-family ipv4 unicast
[local]Redback(config-addrfamily)#prefix-list complex-prefix-list out
```

Simple AS Path List

The following example configures a simple AS path list that denies BGP path attributes starting with AS 100 or ending with AS 200, but allows everything else:

```
[local]Redback(config-ctx)#as-path-list simple-as-path
[local]Redback(config-as-path-list)#seq 10 deny ^100
[local]Redback(config-as-path-list)#seq 20 deny 200$
[local]Redback(config-as-path-list)#seq 30 permit any
```

The following example applies the AS path list, simple-as-path, to BGP neighbor, 192.100.105.10, as a BGP inbound route filter:

```
[local]Redback(config-ctx)#router bgp 100
[local]Redback(config-bgp)#neighbor 192.100.105.10 external
[local]Redback(config-neighbor)#address-family ipv4 unicast
[local]Redback(config-addrfamily)#as-path-list simple-as-path in
```

Complex AS Path List

The AS path list example in this section denies:

- Any AS path containing a private AS number (64500–65535)
- Any AS path with AS 100, AS 200, AS 300, or AS 400 anywhere in the sequence
- Any AS path ending in AS 500 or AS 600
- Any AS path starting with 666

The AS path list configuration is as follows:

```
[local]Redback(config-ctx)#as-path-list complex-as-path
[local]Redback(config-as-path-list)#seq 10 deny _(65[0-9][0-9][0-9]|64[5-9][0-9][0-9])_
[local]Redback(config-as-path-list)#seq 20 deny _(100|200|300|400)_
[local]Redback(config-as-path-list)#seq 30 deny (500|600)$
[local]Redback(config-as-path-list)#seq 40 deny $666
[local]Redback(config-as-path-list)#seq 50 permit any
```

The following example applies the `complex-as-path` AS path list to BGP neighbor, 192.100.106.20, as a BGP outbound route filter:

```
[local]Redback(config-ctx)#router bgp 100
[local]Redback(config-bgp)#neighbor 192.100.106.20 external
[local]Redback(config-neighbor)#address-family ipv4 unicast
[local]Redback(config-addrfamily)#as-path-list complex-as-path out
```

Simple Community List

This following example configures a simple community list that denies community lists containing 10:10, 20:20, or the well-known community no-export (65535:65281), but allows any others:

```
[local]Redback(config-ctx)#community-list simple-community-list
[local]Redback(config-community-list)#seq 10 deny 10:10
[local]Redback(config-community-list)#seq 20 deny 20:20
[local]Redback(config-community-list)#seq 30 deny no-export
[local]Redback(config-community-list)#seq 40 permit any
```

Complex Community List

This section contains an example of a complex community list that denies communities with:

- 400 as the first 16 bits (i.e., AS number) and anything for the second 16 bits of the community number
- 500 or 600 as the first 16 bits (i.e., AS number) and 1, 2, or 3 as the second 16 bits of the community number
- The community that maps to the 32-bit quantity 4 billion (4000000000)

The community list configuration is as follows:

```
[local]Redback(config-ctx)#community-list complex-community-list
[local]Redback(config-community-list)#seq 10 deny reg-exp _400:[0-9]._
[local]Redback(config-community-list)#seq 20 deny reg-exp _(500|600):(1|2|3)_
[local]Redback(config-community-list)#seq 30 deny 4000000000
[local]Redback(config-community-list)#seq 40 permit any
```

Simple Route Map

The following protocol redistribution example configures a simple route map that sets metrics based on network destination address:

```
[local]Redback(config-ctx)#ip prefix-list select-network-20
[local]Redback(config-prefix-list)#seq 10 permit 20.0.0.0/8
[local]Redback(config-prefix-list)#exit
[local]Redback(config-ctx)#ip prefix-list select-network-30
[local]Redback(config-prefix-list)#seq 10 permit 30.0.0.0/8
[local]Redback(config-prefix-list)#exit
[local]Redback(config-ctx)#route-map proto-redist permit 10
[local]Redback(config-route-map)#match ip address prefix-list select-network-20
[local]Redback(config-route-map)#set metric 100
[local]Redback(config-route-map)#exit
```

```
[local]Redback(config-ctx)#route-map proto-redis permit 20
[local]Redback(config-route-map)#match ip address prefix-list select-network-30
[local]Redback(config-route-map)#set metric 200
```

The following example applies the proto-redis route map to BGP neighbor, 192.100.105.100, as a BGP inbound route filter:

```
[local]Redback(config-ctx)#router bgp 100
[local]Redback(config-bgp)#neighbor 192.100.105.100 external
[local]Redback(config-neighbor)#address-family ipv4 unicast
[local]Redback(config-addrfamily)#route-map proto-redis in
```

Complex Route Map

This section contains an example of a complex route map that modifies communities based on AS path lists. For routes corresponding to paths containing private autonomous systems, it will set the community list attribute to the well-known community no-advertise. For routes corresponding to AS paths traversing AS 100, the communities 100:1, 100:2, and 100:3 are added to the BGP community list attribute. This route map and the corresponding communities can be used in conjunction with BGP.

The route map configuration is as follows:

```
[local]Redback(config-ctx)#as-path-list private-as
[local]Redback(config-as-path-list)#seq 10 permit _(65[0-9][0-9][0-9]|64[5-9][0-9][0-9])_
[local]Redback(config-as-path-list)#exit
[local]Redback(config-ctx)#as-path-list traverse-100
[local]Redback(config-as-path-list)#seq 10 permit _100_
[local]Redback(config-as-path-list)#exit
[local]Redback(config-ctx)#route-map modify-community permit 10
[local]Redback(config-route-map)#match as-path-list private-AS
[local]Redback(config-route-map)#set community no-advertise
[local]Redback(config-as-route-map)#exit
[local]Redback(config-ctx)#route-map modify-community permit 20
[local]Redback(config-route-map)#match as-path-list traverse-100
[local]Redback(config-route-map)#set community 100:1 100:2 100:3 additive
```

The following example applies the modify-community route map to BGP neighbor, 192.100.106.100, as a BGP outbound route filter:

```
[local]Redback(config-ctx)#router bgp 100
[local]Redback(config-bgp)#neighbor 192.100.106.100 external
[local]Redback(config-neighbor)#address-family ipv4 unicast
[local]Redback(config-addrfamily)#route-map modify-community out
```

BGP Attribute-Based Accounting

The following example configures BGP attribute-based accounting. Policies are configured to classify the routes which are to be used for BGP policy accounting, and traffic index values are set for routes that pass route map conditions. The bgp-accounting traffic index is assigned to routes installed for the BGP address family. BGP attribute-based accounting is enabled on the interface, joe-customer.

The BGP attribute-based accounting configuration is as follows:

1. Configure policies to classify the routes which are to be used for BGP attribute-based accounting.

```
[local]Redback(config-ctx)#community-list Customer04
[local]Redback(config-community-list)#seq 10 permit 200:20
[local]Redback(config-community-list)#exit
[local]Redback(config-ctx)#community-list SP-Network
[local]Redback(config-community-list)#seq 10 permit 200:30
[local]Redback(config-community-list)#exit
[local]Redback(config-ctx)#community-list SP-Services
[local]Redback(config-community-list)#seq 10 permit 200:10
[local]Redback(config-community-list)#exit
[local]Redback(config-ctx)#route-map bgp-accounting permit 10
[local]Redback(config-route-map)#match community-list SP-Services
[local]Redback(config-route-map)#set traffic-index 1
[local]Redback(config-route-map)#exit
[local]Redback(config-ctx)#route-map bgp-accounting permit 20
[local]Redback(config-route-map)#match community-list Customer04
[local]Redback(config-route-map)#set traffic-index 2
[local]Redback(config-route-map)#exit
[local]Redback(config-ctx)#route-map bgp-accounting permit 30
[local]Redback(config-route-map)#match community-list SP-Network
[local]Redback(config-route-map)#set traffic-index 3
```

2. Configure table-map to assign a traffic-index to routes installed for a particular BGP address family.

```
[local]Redback(config-ctx)#router bgp 1
[local]Redback(config-bgp)#address-family ipv4 unicast
[local]Redback(config-addrfamily)#table-map bgp-accounting
```

3. Enable traffic-index accounting on applicable interface.

```
[local]Redback(config-ctx)#interface joe-customer
[local]Redback(config-if)#ip address 10.200.1.1/30
[local]Redback(config-if)#traffic-index accounting
```

BGP Destination-Based QoS

BGP destination-based QoS supports setting the DSCP byte for IP traffic based on BGP attributes including community list and AS path. This can be used by a service provider (SP) to provide multiple levels of service based on a customers IP destination.

BGP routes can be assigned a DSCP value based on the BGP table-map route-map. When a packet is received on an interface with mark dscp destination enabled and the packet is routed using a route with associated DSCP, the packet's DSCP is updated and the IP header checksum is re-calculated.

The BGP destination-based QoS configuration is as follows:

1. Configure policies to classify the routes which are to be used for BGP attribute-based accounting.

```
[local]Redback(config-ctx)#community-list Bronze-service
[local]Redback(config-community-list)#seq 10 permit 200:10
[local]Redback(config-community-list)#exit
[local]Redback(config-ctx)#community-list Silver-service
[local]Redback(config-community-list)#seq 10 permit 200:20
[local]Redback(config-community-list)#exit
[local]Redback(config-ctx)#community-list Gold-service
[local]Redback(config-community-list)#seq 10 permit 200:30
[local]Redback(config-community-list)#exit
[local]Redback(config-ctx)#route-map destination-qos permit 10
[local]Redback(config-route-map)#match community-list Gold-service
[local]Redback(config-route-map)#set dscp ef
[local]Redback(config-route-map)#exit
[local]Redback(config-ctx)#route-map destination-qos permit 20
[local]Redback(config-route-map)#match community-list Silver-service
[local]Redback(config-route-map)#set dscp af11
[local]Redback(config-route-map)#exit
[local]Redback(config-ctx)#route-map destination-qos permit 20
[local]Redback(config-route-map)#match community-list Bronze-service
[local]Redback(config-route-map)#set dscp df
```

2. Configure table-map to assign a DSCP to routes installed for a particular BGP address family.

```
[local]Redback(config-ctx)#router bgp 1
[local]Redback(config-bgp)#address-family ipv4 unicast
[local]Redback(config-addrfamily)#table-map destination-qos
```

3. Enable mark dscp destination on applicable interface.

```
[local]Redback(config-ctx)#interface jane-customer
[local]Redback(config-if)#ip address 10.200.1.1/30
[local]Redback(config-if)#mark dscp destination
```

Command Descriptions

This section describes the syntax and usage guidelines for the commands used to configure routing policy features. The commands are presented in alphabetical order.

as-path-list	resequence ipv6 prefix-list
community-list	resequence route-map
description	route-map
ext-community-list	set as-path
ip prefix-list	set community
ipv6 prefix-list	set community-list
mark dscp destination	set dampening
match as-path-list	set dscp
match community-list	set ext-community
match ext-community-list	set ip next-hop
match ip address prefix-list	set ipv6 next-hop
match ip next-hop prefix-list	set label
match ipv6 address prefix-list	set level
match ipv6 next-hop prefix-list	set local-preference
match metric	set metric
match route-type	set metric-type
match tag	set origin
{permit deny}	set tag
resequence as-path-list	set traffic-index
resequence community-list	set weight
resequence ext-community-list	traffic-index accounting
resequence ip prefix-list	

as-path-list

as-path-list *apl-name*

no as-path-list *apl-name*

Purpose

Creates a Border Gateway Protocol (BGP) autonomous system (AS) path list and enters AS path list configuration mode.

Command Mode

context configuration

Syntax Description

apl-name Name of the AS path list.

Default

There are no preconfigured AS path lists.

Usage Guidelines

Use the **as-path-list** command to create a BGP AS path list and enter AS path list configuration mode where you can define conditions using the **permit** and **deny** commands.

You can specify an AS path list filter on both inbound and outbound BGP routes. Each filter is based on regular expressions. If the regular expression matches the representation of the AS path of the route as a set of AS numbers (ASNs), the **permit** or **deny** keyword applies. The AS path does not contain the local ASN. Apply the AS path list to a route map using the **match as-path-list** command. Apply the route map as appropriate.

A regular expression is a pattern that is matched against an input string. A regular expression contains the criteria shown in Table 12-16.

Table 12-16 Filter Expression Criteria

Criteria	Description
range	A sequence of characters contained within left and right square brackets; for example, [abcd].
atoms	One of the following single characters: . matches any single character. \$ matches the beginning of the input string. \ <i>character</i> matches the character. - matches a comma (,), left brace ({}), right brace ({}), the beginning of the input string, the end of the input string, or a space.
piece	One of the following symbols: * matches 0 or more sequence of the atom. + matches 1 or more sequences of the atom. ? matches the atom or the null string.
branch	Zero or more concatenated pieces.

The following examples display regular expressions:

```
_100_(via AS100)
^100$(origin AS100)
^100.* (coming from AS100)
```

Use the **no** form of this command to remove an AS path list.

Examples

The following examples create an AS path list, `aspath-1`, and enter AS path list configuration mode:

```
[local]Redback(config-ctx)#as-path-list aspath-1
[local]Redback(config-as-path-list)#
```

Related Commands

```
description
match as-path-list
{permit | deny}
```

community-list

community-list *cl-name*

no community-list *cl-name*

Purpose

Creates a Border Gateway Protocol (BGP) community list and enters community list configuration mode.

Command Mode

context configuration

Syntax Description

cl-name Name of the community list.

Default

There are no preconfigured community lists.

Usage Guidelines

Use the **community-list** command to create a BGP community list and enter community list configuration mode where you can define conditions using the **permit** and **deny** commands.

A community is an attribute shared among a group of prefixes; for example, 10.1.1.0/24, 20.1.1.0/24, and 30.1.1.0/24. A single prefix can be associated with multiple communities. You can specify multiple communities in a single community list entry using a regular expression. Like access control lists, community lists can have multiple entries that are examined in order of ascending sequence number.

To set the communities attribute and match clauses based on communities, use the **set community** and **match community-list** commands in route map configuration mode.

Note A reference to a community list that does not exist, or does not contain any configured entries, implicitly matches and permits all community lists.

Use the **no** form of this command to remove a community list.

Examples

The following example configures the community list, `permit_local`, and enters community list configuration mode:

```
[local]Redback(config-ctx)#community-list permit_local
[local]Redback(config-community-list)#
```

Related Commands

match community-list
{**permit** | **deny**}

set community

description

description *text*

no description

Purpose

Associates a description with the autonomous system (AS) path list, community list, extended community list, IP prefix list, or IPv6 prefix list.

Command Mode

AS path list configuration
 community list configuration
 extended community list configuration
 IP prefix list configuration
 IPv6 prefix list configuration

Syntax Description

<i>text</i>	Description of the AS path list, community list, extended community list, IP prefix list, or IPv6 prefix list.

Default

None

Usage Guidelines

Use the **description** command to associates a description with the AS path list, community list, extended community list, IP prefix list, or IPv6 prefix list. For more information, see the **as-path-list**, **community-list**, **ext-community-list**, **ip prefix-list**, and **ipv6 prefix-list** commands in context configuration mode.

Use the **no** form of this command to remove a description. Because there can be only one description for an AS path list, community list, extended community list, IP prefix list, or IPv6 prefix list, when you use the **no** form of this command, it is not necessary to include the *text* argument.

Examples

The following example configures a description for the community list, com-list1:

```
[local]Redback(config-ctx)#community-list com-list1
[local]Redback(config-community-list)#description filter for community1
```

Related Commands

as-path-list	ip prefix-list
community-list	ipv6 prefix-list
ext-community-list	

ext-community-list

ext-community-list *ecl-name*

no ext-community-list *ecl-name*

Purpose

Creates a Border Gateway Protocol (BGP) extended community list and enters community list configuration mode.

Command Mode

context configuration

Syntax Description

<i>ecl-name</i>	Name of the extended community list.

Default

There are no preconfigured extended community lists.

Usage Guidelines

Use the **ext-community-list** command to create a BGP extended community list and enter community list configuration mode where you can define conditions using the **permit** and **deny** commands.

The extended communities attribute consists of a set of extended communities. Each extended community is coded as an eight octet extended community number. An extended communities attribute is specified by configuring an extended communities list. You can specify multiple extended communities in a single extended community list entry. Like access control lists, extended community lists can have multiple entries that are examined in order of ascending sequence number.

All routes with the extended communities attribute belong to the communities listed in the attribute.

To set the extended communities attribute and match clauses based on extended communities, use the **set ext-community** and **match ext-community-list** commands in route map configuration mode.

Note A reference to an extended community list that does not exist, or does not contain any configured entries, implicitly matches and permits all extended community lists.

Use the **no** form of this command to remove an extended community list.

Examples

The following example configures the extended community list, `permit_local`, and enters community list configuration mode:

```
[local]Redback(config-ctx)#ext-community-list permit_local
[local]Redback(config-community-list)#
```

Related Commands

`match ext-community-list`
`{permit | deny}`
`set ext-community`

ip prefix-list

ip prefix-list *pl-name*
no ip prefix-list *pl-name*

Purpose

Creates an IP prefix list used to filter routes and enters IP prefix list configuration mode.

Command Mode

context configuration

Syntax Description

pl-name IP prefix list name.

Default

There are no preconfigured IP prefix lists.

Usage Guidelines

Use the **ip prefix-list** command to create an IP prefix list used to filter routes and to enter IP prefix list configuration mode where you can define conditions using the **permit** and **deny** commands.

Note A reference to an IP prefix list that does not exist, or does not contain any configured entries, implicitly matches and permits all IP prefixes.

Use the **no** form of this command to remove an IP prefix list.

Examples

The following example creates the IP prefix list, `list102`, and enters IP prefix list configuration mode:

```
[local]Redback(config-ctx)#ip prefix-list list102  
[local]Redback(config-prefix-list)#
```

Related Commands

description
match ip address prefix-list
match ip next-hop prefix-list
{permit | deny}

ipv6 prefix-list

ipv6 prefix-list *pl-name*

no ipv6 prefix-list *pl-name*

Purpose

Creates an IP Version 6 (IPv6) prefix list used to filter routes and enters IPv6 prefix list configuration mode.

Command Mode

context configuration

Syntax Description

pl-name IPv6 prefix list name.

Default

There are no preconfigured IPv6 prefix lists.

Usage Guidelines

Use the **ipv6 prefix-list** command to create an IPv6 prefix list used to filter routes and to enter IPv6 prefix list configuration mode where you can define conditions using the **permit** and **deny** commands.

Note A reference to an IPv6 prefix list that does not exist, or does not contain any configured entries, implicitly matches and permits all IPv6 prefixes.

Use the **no** form of this command to remove an IPv6 prefix list.

Examples

The following example creates the IPv6 prefix list, `list102`, and enters IPv6 prefix list configuration mode:

```
[local]Redback(config-ctx)#ipv6 prefix-list list102
[local]Redback(config-ipv6-prefix-list)#
```

Related Commands

description
match ip address prefix-list
match ip next-hop prefix-list
{permit | deny}

mark dscp destination

mark dscp destination

no mark dscp destination

Purpose

Sets the Differentiated Services Code Point (DSCP) byte, based on Border Gateway Protocol (BGP) attributes, such as community list and autonomous system (AS) path, for incoming IP traffic on the specified interface.

Command Mode

interface configuration

Syntax Description

This command has no keywords or arguments.

Default

Disabled

Usage Guidelines

Use the **mark dscp destination** command to set the DSCP byte, based on BGP attributes, such as community list and autonomous AS path, for incoming IP traffic on the specified interface.

BGP destination-based quality of service (QoS) provides multiple levels of service based on a customer's IP destination. BGP routes can be assigned a DSCP value based on the BGP traffic indexing and table map features associated with route maps. BGP routes can be assigned a traffic index. The byte and packet counters for the traffic index are incremented based on the route traversed by IP traffic received on the ingress interface.

When a packet is received on an interface with **mark dscp destination** enabled and the packet is routed using a route with associated DSCP, the packet's DCSP is updated and the IP header checksum is re-calculated.

When an ingress packet is routed using a BGP route, and a DSCP marking is associated with the route, the packet's DCSP is updated and the IP header checksum is recalculated. The packet is then placed in the appropriate priority queue on the egress circuit, depending on the new DSCP value and the QoS Policy configured for that circuit.



Caution Risk of overriding configurations. Because marking can be configured at different levels, the SmartEdge OS checks for and applies marking in a specific order. To reduce the risk, remember the following points:

- Circuit-based marking overrides class-based marking. Circuit-based marking is configured through the **conform** and **exceed** commands in QoS policy rate configuration mode. Class-based marking is configured through the **class** command in policy ACL configuration mode and the **mark** command in policy ACL class configuration mode.
- BGP destination-based marking, through route maps, overrides both circuit-based and class-based marking.

Use the **no** form of this command to disable the DSCP byte marking for incoming IP traffic for the specified interface.

Examples

The following example enables BGP-based marking on the appropriate ingress interface:

```
[local]Redback(config)#context local  
[local]Redback(config-ctx)#interface CustomerOne  
[local]Redback(config-if)#ip address 10.200.1.1/30  
[local]Redback(config-if)#mark dscp destination
```

Related Commands

route-map
set dscp
table-map

match as-path-list

match as-path-list *apl-name*

no match as-path-list *apl-name*

Purpose

Permits or denies routes that include the specified Border Gateway Protocol (BGP) autonomous system (AS) path list.

Command Mode

route map configuration

Syntax Description

apl-name AS path list name.

Default

There are no preconfigured route map match conditions.

Usage Guidelines

Use the **match as-path-list** command to permit or deny routes that include the specified BGP AS path list. A route map can have several entries. Any route that does not match at least one match clause corresponding to a route map is ignored; that is, the route is not advertised for outbound route maps and is not accepted for inbound route maps. To modify only some data, you must configure a second route map section with an explicit match condition specified.

Use the **no** form of this command to remove the match condition.

Examples

The following example permits routes that include AS path list 5:

```
[local]Redback(config-ctx)#route-map asp-regex permit 10  
[local]Redback(config-route-map)#match as-path-list 5
```

Related Commands

as-path-list
route-map
set as-path

match community-list

match community-list *cl-name* [**exact-match**]

no match community-list *cl-name*

Purpose

Permits or denies routes with an associated Border Gateway Protocol (BGP) community attribute that matches the specified community list.

Command Mode

route map configuration

Syntax Description

<i>cl-name</i>	Name of the community list.
exact-match	Optional. Defines communities in the community list that must match exactly.

Default

There are no preconfigured route map match conditions.

Usage Guidelines

Use the **match community-list** command to permit or deny routes with an associated BGP community attribute that matches the specified community list.

When the **exact-match** keyword is specified, the community list entries must match the BGP community attribute exactly. In other words, the community list must have the same number of entries as the BGP community attribute, and each community list entry, community number, or well-known community must be present in the BGP community attribute. In addition, the community list used for exact matching must not have any deny entries or any entries with a regular expression specification.

A route map can have several sequenced entries. Any route that does not satisfy all the match conditions associated with a route map entry is ignored and the next higher sequenced route map entry is examined.

See the **community-list** command in context configuration mode for more information.

Use the **no** form of this command to disable the match condition.

Examples

The following example permits any route that includes the attribute community list 1:

```
[local]Redback(config-ctx)#community-list 1
[local]Redback(config-community-list)#permit 11
[local]Redback(config-community-list)#exit
[local]Redback(config-ctx)#route-map map_A
[local]Redback(config-route-map)#match community-list 1
```

Related Commands

`community-list`
`route-map`
`set community`

match ext-community-list

match ext-community-list *ecl-name* [**exact-match**]

no match community-list *ecl-name*

Purpose

Permits or denies routes with an associated Border Gateway Protocol (BGP) extended community attribute that matches the specified extended community list.

Command Mode

route map configuration

Syntax Description

<i>ecl-name</i>	Name of the extended community list.
exact-match	Optional. Defines extended communities in the extended community list that must match exactly.

Default

There are no preconfigured route map match conditions.

Usage Guidelines

Use the **match ext-community-list** command to permit or deny routes with an associated BGP extended community attribute that matches the specified extended community list.

When the **exact-match** keyword is specified, the extended community list entries must match the BGP extended community attribute exactly. In other words, the extended community list must have the same number of entries as the BGP extended community attribute, and each extended community list entry, extended community number, or well-known extended community must be present in the BGP extended community attribute. In addition, the extended community list used for exact matching must not have any deny entries or any entries with a regular expression specification.

A route map can have several sequenced entries. Any route that does not satisfy all the match conditions associated with a route map entry is ignored and the next higher sequenced route map entry is examined.

See the **ext-community-list** command in context configuration mode for more information.

Use the **no** form of this command to disable the match condition.

Examples

The following example permits any route that includes the extended community list 1 attribute:

```
[local]Redback(config-ctx)#ext-community-list 1
[local]Redback(config-community-list)#permit 11
[local]Redback(config-community-list)#exit
[local]Redback(config-ctx)#route-map map_A
[local]Redback(config-route-map)#match ext-community-list 1
```


Related Commands

- `ext-community-list`
- `route-map`
- `send ext-community`
- `set ext-community`

match ip address prefix-list

match ip address prefix-list *pl-name*

no match ip address prefix-list *pl-name*

Purpose

Permits or denies routes with a destination IP address permitted by the specified IP prefix list.

Command Mode

route map configuration

Syntax Description

pl-name Name of the IP prefix list used to match route destinations.

Default

There are no preconfigured route map match conditions.

Usage Guidelines

Use the **match ip address prefix-list** command to permit or deny routes with a destination IP address permitted by the specified IP prefix list. To create an IP prefix list, use the **ip prefix-list** command in context configuration mode.

Use the **no** form of this command to disable IP address matching.

Examples

The following example permits routes that have destination IP addresses specified in an IP prefix list, prefix8:

```
[local]Redback(config-ctx)#route-map rmap_B  
[local]Redback(config-route-map)#match ip address prefix-list prefix8
```

Related Commands

ip prefix-list
route-map

match ip next-hop prefix-list

match ip next-hop prefix-list *pl-name*

no match ip next-hop prefix-list *pl-name*

Purpose

Permits or denies routes with a next-hop IP address that is permitted by the specified IP prefix list.

Command Mode

route map configuration

Syntax Description

pl-name Name of the IP prefix list used to match the next-hop IP address.

Default

There are no preconfigured route map match conditions.

Usage Guidelines

Use the **match ip next-hop prefix-list** command to permit or deny routes with a next-hop IP address permitted by the specified IP prefix list. To create an IP prefix list, use the **ip prefix-list** command in context configuration mode.

Use the **no** form of this command to disable next-hop IP address matching.

Examples

The following example permits routes that have a next-hop IP address permitted by either prefix list, prefix11 or prefix98:

```
[local]Redback(config-ctx)#route-map rmap_C  
[local]Redback(config-route-map)#match ip next-hop prefix-list prefix11 prefix98
```

Related Commands

ip prefix-list
route-map
set ip next-hop

match ipv6 address prefix-list

match ipv6 address prefix-list *ipv6-pl-name*

no match ipv6 address prefix-list *ipv6-pl-name*

Purpose

Permits or denies routes with a destination IP Version 6 (IPv6) address permitted by the specified IPv6 prefix list.

Command Mode

route map configuration

Syntax Description

ipv6-pl-name Name of the IPv6 prefix list used to match route destinations.

Default

There are no preconfigured route map match conditions.

Usage Guidelines

Use the **match ipv6 address prefix-list** command to permit or deny routes with a destination IPv6 address permitted by the specified IPv6 prefix list. To create an IPv6 prefix list, use the **ipv6 prefix-list** command in context configuration mode.

Use the **no** form of this command to disable IPv6 address matching.

Examples

The following example permits routes that have destination IPv6 addresses specified in an IPv6 prefix list, prefix8:

```
[local]Redback(config-ctx)#route-map rmap_B  
[local]Redback(config-route-map)#match ipv6 address prefix-list prefix8
```

Related Commands

ipv6 prefix-list
route-map

match ipv6 next-hop prefix-list

match ip next-hop prefix-list *ipv6-pl-name*

no match ip next-hop prefix-list *ipv6-pl-name*

Purpose

Permits or denies routes with a next-hop IP Version 6 (IPv6) address that is permitted by the specified IPv6 prefix list.

Command Mode

route map configuration

Syntax Description

ipv6-pl-name Name of the IPv6 prefix list used to match the next-hop IPv6 address.

Default

There are no preconfigured route map match conditions.

Usage Guidelines

Use the **match ipv6 next-hop prefix-list** command to permit or deny routes with a next-hop IPv6 address permitted by the specified IPv6 prefix list. To create an IPv6 prefix list, use the **ipv6 prefix-list** command in context configuration mode.

Use the **no** form of this command to disable next-hop IPv6 address matching.

Examples

The following example permits routes that have a next-hop IPv6 address permitted by either IPv6 prefix list, ipv6pl4 or ipv6pl72:

```
[local]Redback(config-ctx)#route-map rmap_C  
[local]Redback(config-route-map)#match ipv6 next-hop prefix-list ipv6pl4 ipv6pl72
```

Related Commands

ipv6 prefix-list
route-map
set ip next-hop

match metric

match metric *metric*

no match metric *metric*

Purpose

Permits or denies routes with a specified metric value.

Command Mode

route map configuration

Syntax Description

metric Route metric value. The range of values is 0 to 4,294,967,295.

Default

There are no preconfigured route map match conditions.

Usage Guidelines

Use the **match metric** command to permit or deny routes with a specified metric value.

Use the **no** form of this command to disable the match condition.

Examples

The following example permits routes with a metric value of 5:

```
[ local ]Redback(config-ctx)#route-map rmap_D  
[ local ]Redback(config-route-map)#match metric 5
```

Related Commands

route-map

set metric

match route-type

```
match route-type {internal | external [type-1 | type-2] | level-1 | level-2 | nssa-external  
[type-1 | type-2] | dvsr}  
no match route-type
```

Purpose

Permits or denies routes that match a specified route type.

Command Mode

route map configuration

Syntax Description

internal	Matches internal Open Shortest Path First (OSPF) intra-area and interarea routes.
external	Specifies Border Gateway Protocol (BGP) and OSPF external routes.
type-1	Optional. Matches OSPF Type 1 external routes when used with the external keyword. Matches OSPF NSSA Type 1 external routes when used with the nssa-external keyword.
type-2	Optional. Matches OSPF Type 2 external routes when use with the external keyword. Matches OSPF not-so-stubby-area (NSSA) Type 2 external routes when used with the nssa-external keyword.
level-1	Matches Intermediate System-to-Intermediate System (IS-IS) Level 1 routes.
level-2	Matches IS-IS Level 2 routes.
nssa-external	Matches OSPF NSSA external routes.
dvsr	Matches dynamically verified static routing (DVSr) subtype of static route.

Default

There are no preconfigured route map match conditions.

Usage Guidelines

Use the **match route-type** command to permit or deny routes that match a specified route type.

Use the **no** form of this command to disable route type matching.

Examples

The following example permits or denies internal OSPF routes:

```
[local]Redback(config-ctx)#route-map map_E  
[local]Redback(config-route-map)#match route-type internal
```

Related Commands

route-map

match tag

match tag *tag*

no match tag

Purpose

Permits or denies routes that match a specified route tag value.

Command Mode

route map configuration

Syntax Description

tag Unsigned integer. The range of values is 0 to 4,294,967,295.

Default

There are no preconfigured route map match conditions.

Usage Guidelines

Use the **match tag** command to permit or deny routes that match a specified route tag value.

Use the **no** form of this command to disable route tag matching.

Examples

The following example permits routes using a route tag value of 5:

```
[local]Redback(config-ctx)#route-map map_F  
[local]Redback(config-route-map)#match tag 5
```

Related Commands

route-map

{permit | deny}

```
{permit | deny} {reg-exp | any} | {community-num | ext-community-num | local-as | no-advertise |  
no-export | any | reg-exp reg-exp} | { {ip-addr/prefix-length | ipv6-addr/prefix-length}  
[ {eq eq-value | ge ge-value | [le le-value] } ] | any }  
  
seq seq-num {permit | deny} {reg-exp | any} | {community-num | ext-community-num | local-as |  
no-advertise | no-export | any | reg-exp reg-exp} | {ip-addr/prefix-length [ {eq eq-value |  
ge ge-value | [le le-value] } ] | any }  
  
no seq seq-num
```

Purpose

Permits or denies routes matching the specified criteria.

Command Mode

AS path list configuration
community list configuration
extended community list configuration
IP prefix list configuration
IPv6 prefix list configuration

Syntax Description

AS path list configuration mode:

<i>reg-exp</i>	AS path regular expression.
any	Wildcard that matches on any AS path list number.

Community list and extended community list configuration mode:

<i>community-num</i>	Community number, which can be specified only when configuring a community list. It can be expressed in either of the following formats: • <i>asn:nn</i>, where <i>asn</i> is the autonomous system number (ASN) and <i>nn</i> is a 16-bit integer. The range of <i>nn</i> values is 0 to 65,535. • An unsigned decimal value. The range of values is 1 to 4,294,967,040. You can specify a single community number or multiple community numbers separated by a space. (All numbers must match a community in the route being tested in order for the statement to match.)

<i>ext-community-num</i>	Extended community number, which can be specified only when configuring an extended community list. It can be expressed in either of the following formats: <i>tt:asn:nnnn</i>, where <i>tt</i> is the extended community type, <i>asn</i> is the ASN, and <i>nnnn</i> is a 32-bit integer. The extended community type identifies either a target or origin community. The target community identifies the destination to which the route is going, and the origin community identifies source from where the route originated. The <i>tt</i> argument is a placeholder for either the ro (route origin) keyword, or the rt (route target) keyword. <i>tt:ip-addr:nn</i>, where <i>tt</i> is the extended community type, <i>ip-addr</i> is the IP address in the form <i>A.B.C.D</i>, and <i>nn</i> is a 16-bit integer. You can specify a single extended community number or multiple extended community numbers separated by a space. (All numbers must match an extended community in the route being tested in order for the statement to match.)
local-as	Propagates this route to peers in other subautonomous systems within the confederation. Does not advertise this route to an external Border Gateway Protocol (eBGP) peer.
no-advertise	Does not advertise this route to any peer (internal or external).
no-export	Does not advertise this route out of the confederation, or out of the local AS, if this peer is not part of a confederation.
reg-exp <i>reg-exp</i>	Regular expression used to match the ASCII representation of the route's community attribute. The ASCII representation of the community attributes includes all the communities in <i>aa:nn</i> format. Each entry must be separated by a space.
any	Wildcard that matches on any community number.
IP prefix list configuration mode:	
<i>ip-addr</i>	IP address in the form <i>A.B.C.D</i> .
<i>prefix-length</i>	Prefix length. The range of values is 0 to 32.
eq <i>eq-value</i>	Optional. Equal to value. The <i>eq-value</i> argument specifies a value to which a route's prefix length must match; the eq keyword indicates that the route's prefix length must exactly match the <i>eq-value</i> . The range of values for the <i>eq-value</i> argument is 1 to 32.
ge <i>ge-value</i>	Optional. Greater than or equal to value. The <i>ge-value</i> argument specifies a value to which a route's prefix length must match; the ge keyword indicates that the route's prefix length must be greater than or equal to the <i>ge-value</i> to match. The range of values for the <i>ge-value</i> argument is 1 to 32.

le *le-value* Optional. Less than or equal to value. The *le-value* argument specifies a value to which a route's prefix length must match; the **le** keyword indicates that the route's prefix length must be less than or equal to the *le-value* to match. The range of values for the *le-value* argument is 1 to 32.

any Wildcard that matches on any prefix.

IPv6 prefix list configuration mode:

ipv6-addr IP Version 6 (IPv6) address in the form *A:B:C:D:E:F:G:H*.

prefix-length Prefix length. The range of values is 0 to 128.

eq *eq-value* Optional. Equal to value. The *eq-value* argument specifies a value to which a route's prefix length must match; the **eq** keyword indicates that the route's prefix length must exactly match the *eq-value*. The range of values for the *eq-value* argument is 1 to 128.

ge *ge-value* Optional. Greater than or equal to value. The *ge-value* argument specifies a value to which a route's prefix length must match; the **ge** keyword indicates that the route's prefix length must be greater than or equal to the *ge-value* to match. The range of values for the *ge-value* argument is 1 to 128.

le *le-value* Optional. Less than or equal to value. The *le-value* argument specifies a value to which a route's prefix length must match; the **le** keyword indicates that the route's prefix length must be less than or equal to the *le-value* to match. The range of values for the *le-value* argument is 1 to 128.

any Wildcard that matches on any prefix.

Note A high prefix length value specifies a small subnet, and a low prefix length value specifies a large subnet. Using the **ge** keyword permits or denies routes with higher prefix length values (smaller subnets), and the **le** keyword permits or denies routes with lower prefix length values (larger subnets).

Default

None

Usage Guidelines

Use the {**permit** | **deny**} command to permit or deny any routes matching the specified criteria.

Use the **seq** *seq-num* form of this command to specify the sequence number of the statement you are creating. If you do not use the **seq** *seq-num* construct, the system automatically assigns sequence numbers in increments of 10. The range of values is 1 to 4,294,967,295.

Use the **no seq** *seq-num* form of this command to delete a specific sequence number from the AS path list, community list, extended community list, IP prefix list, or IPv6 prefix list.

Examples

The following example ensures that the BGP neighbor at IP address 10.1.1.1 is not sent advertisements about any path to or from the adjacent autonomous system 3:

```
[local]Redback(config-ctx)#as-path-list aspath-1
[local]Redback(config-as-path-list)#seq 5 deny _3_
[local]Redback(config-ctx)#as-path-list 10 seq 10 permit .*
[local]Redback(config-ctx)#route-map drop-asp-3 permit 10
[local]Redback(config-route-map)#match as-path-list 10
.
.
.
[local]Redback(config-ctx)#router bgp 65015
[local]Redback(config-group)#neighbor 10.1.1.1
[local]Redback(config-peer)#route-map drop-asp-3 out
```

The following example configures community list permit_local to propagate routes to peers within the local autonomous system (local-AS):

```
[local]Redback(config-ctx)#community-list permit_local
[local]Redback(config-community-list)#seq 10
[local]Redback(config-community-list)#permit local-AS
```

Related Commands

- as-path-list
- community-list
- ext-community-list
- ip prefix-list
- ipv6 prefix-list
- route-map

resequence as-path-list

resequence as-path-list *apl-name*

Purpose

Assigns new sequence numbers to existing entries in the specified autonomous system (AS) path list so that entries are in increments of 10.

Command Mode

context configuration

Syntax Description

<i>apl-name</i>	Name of the AS path list to be resequenced.

Default

Sequence numbers are assigned by the system in increments of 10.

Usage Guidelines

Use the **resequence as-path-list** command to assign new sequence numbers to existing entries in the specified AS path list so that entries are in increments of 10.

This command is useful when you have manually assigned sequence numbers and have no room to insert new entries in between existing entries. You can manually assign sequence numbers using the **seq seq-num** construct in the **as-path-list** command in context configuration mode.

Note Two resequence commands, **resequence ip access-list** and **resequence policy access-list**, are not included in this guide. For more information on these commands, see the “ACL Configuration” chapter in the *IP Services and Security Configuration Guide* for the SmartEdge OS.

Examples

The following example resequences entries in the AS path list, `filter1`, by increments of 10:

```
[local]Redback(config-ctx)#resequence as-path-list filter1
```

Related Commands

- as-path-list**
- resequence community-list**
- resequence ext-community-list**
- resequence ip prefix-list**
- resequence ipv6 prefix-list**
- resequence route-map**

resequence community-list

resequence community-list *cl-name*

Purpose

Assigns new sequence numbers to existing entries in the specified community list so that entries are in increments of 10.

Command Mode

context configuration

Syntax Description

<i>cl-name</i>	Name of the community list to be resequenced.

Default

Sequence numbers are assigned by the system in increments of 10.

Usage Guidelines

Use the **resequence community-list** command to assign new sequence numbers to existing entries in the specified community list so that entries are in increments of 10.

This command is useful when you have manually assigned sequence numbers and have no room to insert new entries in between existing entries. You can manually assign sequence numbers using the **seq seq-num** construct in the **community-list** command in context configuration mode.

Note Two resequence commands, **resequence ip access-list** and **resequence policy access-list**, are not included in this guide. For more information on these commands, see the “ACL Configuration” chapter in the *IP Services and Security Configuration Guide* for the SmartEdge OS.

Examples

The following example resequences entries in the community list, **cl012**, by increments of 10:

```
[local]Redback(config-ctx)#resequence community-list cl012
```

Related Commands

- community-list**
- resequence as-path-list**
- resequence ext-community-list**
- resequence ip prefix-list**
- resequence ipv6 prefix-list**
- resequence route-map**

resequence ext-community-list

resequence ext-community-list *ecl-name*

Purpose

Assigns new sequence numbers to existing entries in the specified extended community list so that entries are in increments of 10.

Command Mode

context configuration

Syntax Description

ecl-name Name of the extended community list to be resequenced.

Default

Sequence numbers are assigned by the system in increments of 10.

Usage Guidelines

Use the **resequence ext-community-list** command to assign new sequence numbers to existing entries in the specified extended community list so that entries are in increments of 10.

This command is useful when you have manually assigned sequence numbers and have no room to insert new entries in between existing entries. You can manually assign sequence numbers using the **seq seq-num** construct in the **ext-community-list** command in context configuration mode.

Note Two resequence commands, **resequence ip access-list** and **resequence policy access-list**, are not included in this guide. For more information on these commands, see the “ACL Configuration” chapter in the *IP Services and Security Configuration Guide* for the SmartEdge OS.

Examples

The following example resequences entries in the extended community list, `ec105`, by increments of 10:

```
[local]Redback(config-ctx)#resequence ext-community-list ec105
```

Related Commands

community-list
resequence as-path-list
resequence community-list
resequence ip prefix-list
resequence ipv6 prefix-list
resequence route-map

resequence ip prefix-list

resequence ip prefix-list *pl-name*

Purpose

Assigns new sequence numbers to existing entries in the specified IP prefix list so that entries are in increments of 10.

Command Mode

context configuration

Syntax Description

<i>pl-name</i>	Name of the IP prefix list to be resequenced.

Default

Sequence numbers are assigned by the system in increments of 10.

Usage Guidelines

Use the **resequence ip prefix-list** command to assign new sequence numbers to existing entries in the specified IP prefix list so that entries are in increments of 10.

This command is useful when you have manually assigned sequence numbers and have no room to insert new entries in between existing entries. You can manually assign sequence numbers using the **seq seq-num** construct in the **ip prefix-list** command in context configuration mode.

Note Two resequence commands, **resequence ip access-list** and **resequence policy access-list**, are not included in this guide. For more information on these commands, see the “ACL Configuration” chapter in the *IP Services and Security Configuration Guide* for the SmartEdge OS.

Examples

The following example resequences entries in the prefix list, pl226, by increments of 10:

```
[local]Redback(config-ctx)#resequence ip prefix-list pl226
```

Related Commands

- ip prefix-list**
- resequence as-path-list**
- resequence community-list**
- resequence ext-community-list**
- resequence ipv6 prefix-list**
- resequence route-map**

resequence ipv6 prefix-list

resequence ipv6 prefix-list *ipv6-pl-name*

Purpose

Assigns new sequence numbers to existing entries in the specified IP Version 6 (IPv6) prefix list so that entries are in increments of 10.

Command Mode

context configuration

Syntax Description

<i>ipv6-pl-name</i>	Name of the IPv6 prefix list to be resequenced.

Default

Sequence numbers are assigned by the system in increments of 10.

Usage Guidelines

Use the **resequence ipv6 prefix-list** command to assign new sequence numbers to existing entries in the specified IPv6 prefix list so that entries are in increments of 10.

This command is useful when you have manually assigned sequence numbers and have no room to insert new entries in between existing entries. You can manually assign sequence numbers using the **seq seq-num** construct in the **ipv6 prefix-list** command in context configuration mode.

Note Two resequence commands, **resequence ip access-list** and **resequence policy access-list**, are not included in this guide. For more information on these commands, see the “ACL Configuration” chapter in the *IP Services and Security Configuration Guide* for the SmartEdge OS.

Examples

The following example resequences entries in the prefix list, `ipv6p65`, by increments of 10:

```
[local]Redback(config-ctx)#resequence ipv6 prefix-list ipv6p65
```

Related Commands

- ipv6 prefix-list**
- resequence as-path-list**
- resequence community-list**
- resequence ext-community-list**
- resequence ip prefix-list**
- resequence route-map**

resequence route-map

resequence route-map *map-name*

Purpose

Assigns new sequence numbers to existing entries in the specified route map so that entries are in increments of 10.

Command Mode

context configuration

Syntax Description

<i>map-name</i>	Name of the route map to be resequenced.

Default

Sequence numbers are assigned by the system in increments of 10.

Usage Guidelines

Use the **resequence route-map** command to assign new sequence numbers to existing entries in the specified route map so that entries are in increments of 10.

This command is useful when you have manually assigned sequence numbers and have no room to insert new entries in between existing entries. You can manually assign sequence numbers using the **seq seq-num** construct in the **route-map** command in context configuration mode.

Note Two resequence commands, **resequence ip access-list** and **resequence policy access-list**, are not included in this guide. For more information on these commands, see the “ACL Configuration” chapter in the *IP Services and Security Configuration Guide* for the SmartEdge OS.

Examples

The following example resequences entries in the route map, rm045, by increments of 10:

```
[local]Redback(config-ctx)#resequence route-map rm045
```

Related Commands

resequence as-path-list
resequence community-list
resequence ext-community-list
resequence ip prefix-list
resequence ipv6 prefix-list
route-map

route-map

route-map *map-name* [*seq-num*] [**deny** *seq-num* | **permit** *seq-num*] | [**description** *text*]

no route-map *map-name* [*seq-num*] [**deny** *seq-num* | **permit** *seq-num*] | [**description**]

Purpose

Creates a route map for policy routing and enters route map configuration mode.

Command Mode

context configuration

Syntax Description

<i>map-name</i>	Descriptive name for the route map.
<i>seq-num</i>	Optional. Sequence number for the route map entry, relative to other route map entries in the same route map. Route map entries are tested in order of ascending sequence number; that is, the route map entry with the lowest sequence number is examined first when Border Gateway Protocol (BGP) routes are tested. The range of values is 1 to 4,294,967,295; the default value is 10 greater than the largest sequence number of any route map entry in the route map.
deny <i>seq-num</i>	Optional. Sequence number for the route map entry. The range of values is 1 to 4,294,967,295. Routes using the specified sequence number are denied.
permit <i>seq-num</i>	Optional. Sequence number for the route map entry. The range of values is 1 to 4,294,967,295. Routes using the specified sequence number are permitted.
description <i>text</i>	Optional. Description of the route map. No <i>text</i> argument is specified when the description keyword is used with the no form of this command.

Default

The action is permit. If not specified, the sequence number is 10 greater than the largest sequence number for a route map entry with the same *map-name* argument.

Usage Guidelines

Use the **route-map** command to create a route map for policy routing and enter route map configuration mode. Use this command in conjunction with the **match** commands in route map configuration mode to specify the conditions under which a route is accepted or rejected by the routing application that is using the route map. If the route entry indicates permit, the **set** commands can be used to modify the accepted routes attributes.

Route map entries are tested in ascending order. For a route to match a particular route map entry, all match conditions must be satisfied. A route map entry with no match conditions can be used to unconditionally change a route's attributes by applying set actions.

Note A reference to a route map that does not exist, or does not contain any configured entries, implicitly matches and permits all routes.

Use the **no** form of this command to delete a specific route map entry or to delete the entire route map. Because there can be only one description for a route map, when you use the **no** form of this command to delete the route map description, it is not necessary to include the *text* argument.

Examples

The following example redistributes static routes with destination addresses that match the IP access list `acc03` into the BGP routing process. The `set` command is used to modify the metric of selected routes.

```
[local]Redback(config-ctx)#ip prefix-list acc03
[local]Redback(config-prefix-list)#permit 81.1.0.0/16 le 32
[local]Redback(config-prefix-list)#permit 77.0.0.0/8 le 32
[local]Redback(config-prefix-list)#exit
[local]Redback(config-ctx)#route-map rmap1 permit 10
[local]Redback(config-route-map)#match ip address prefix-list acc03
[local]Redback(config-route-map)#set metric 10
[local]Redback(config-route-map)#exit
[local]Redback(config-ctx)#router bgp 65012
[local]Redback(config-bgp)#address-family ipv4 unicast
[local]Redback(config-addrfamily)#redistribute static route-map rmap1
```

Related Commands

- match as-path-list
- match community-list
- match ip address prefix-list
- match ip next-hop prefix-list
- match metric
- match route-type
- match tag
- redistribute
- route-map
- set as-path
- set community
- set dscp
- set ip next-hop
- set local-preference
- set metric
- set origin
- set tag
- set traffic-index
- set weight

set as-path

set as-path {prepend {asn... | nn:nn...} | tag}

no set as-path

Purpose

Prepends an autonomous system (AS) path to Border Gateway Protocol (BGP) routes that pass the route map conditions.

Command Mode

route map configuration

Syntax Description

prepend	Increases the AS path by adding AS numbers (ASNs) to the AS path.
<i>asn</i>	ASN in integer format. The range of values is 1 to 65,535. The subrange 64,512 to 65,535 is reserved for private autonomous systems. You can specify up to 16 ASNs. Each ASN must be separated by a space.
<i>nn:nn</i>	ASN in unsigned 4-byte <i>nn:nn</i> format, where the first <i>nn</i> represents the first 2 bytes of the ASN, and the second <i>nn</i> represents the second 2 bytes of the ASN. The range of values is 1 to 4,294,967,295. You can specify up to 16 ASNs. Each ASN must be separated by a space.
tag	Sets the AS path to the value of the route tag.

Default

There are no preconfigured route map set actions. The AS path attribute for selected BGP routes is not modified.

Usage Guidelines

Use the **set as-path** command to prepend an AS path to BGP routes that pass the route map conditions. 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 peer can influence the best path selection. Usually the local AS number is prepended multiple times, increasing the AS path length.

Use the **no** form of this command to disable the configured set action.

Examples

The following example prepends 11 to all the routes advertised to 10.1.1.1:

```
[local]Redback(config-ctx)#router bgp 11
[local]Redback(config-group)#neighbor 10.1.1.1
[local]Redback(config-peer)#route-map set-as-path out
```

```
.  
.   
.   
[local]Redback(config-ctx)#route-map set-as-path  
[local]Redback(config-route-map)#match as-path 1  
[local]Redback(config-route-map)#set as-path prepend 11 11
```

Related Commands

as-path-list
match as-path-list
route-map

set community

set community {*community-num* [**no-export**] [**local-as**] [**no-advertise**] [**additive**] | **none**}
no set community

Purpose

Sets the Border Gateway Protocol (BGP) community attribute for routes that pass the route map conditions.

Command Mode

route map configuration

Syntax Description

<i>community-num</i>	32-bit value expressed as either an unsigned decimal or in <i>nn:nn</i> format, where the first <i>nn</i> is the autonomous system number (ASN) and the second <i>nn</i> is a 2-byte number defined by the autonomous system. The range of unsigned decimal values is 1 to 4,294,967,295. The range of values for <i>aa</i> is 1 to 65,535. The range of values for either <i>nn</i> argument is 1 to 65,535. You can specify up to eight community numbers. Each entry must be separated by a space.
no-export	Optional. Does not advertise this route out of the local AS confederation, or out of the local AS, if it is not part of a confederation.
local-as	Optional. Propagates this route only to peers in the local autonomous system. Does not send this route to external peers even if they are in the same confederation.
no-advertise	Optional. Does not advertise this route to any peer (internal or external).
additive	Optional. Adds the community to the existing communities.
none	Removes the community attribute from the prefixes that pass the route map conditions.

Default

There are no preconfigured route map set actions. The community attribute for selected BGP routes is not modified.

Usage Guidelines

Use the **set community** command to set the BGP community attribute for routes that pass the route map conditions. A community is a group of destinations that share some common attributes. Each destination can belong to multiple communities.

Use the **no** form of this command to disable the configured set action.

Examples

The following example ensures that routes that pass the autonomous system (AS) path 1 conditions have the community set to 9. Routes that pass the autonomous system path list 2 conditions have the community set to no-export (these routes are not advertised out of the local AS confederation, or out of the local AS, if it is not part of a confederation):

```
[local]Redback(config-ctx)#route-map set_community 10 permit
[local]Redback(config-route-map)#match as-path 1
[local]Redback(config-route-map)#set community 9
.
.
.
[local]Redback(config-ctx)#route-map set_community 20 permit
[local]Redback(config-route-map)#match as-path 2
[local]Redback(config-route-map)#set community no-export
```

Related Commands

community-list
match community-list
route-map

set community-list

set community-list *ecl-name* **delete**

no set community-list

Purpose

Deletes Border Gateway Protocol (BGP) communities matching the community list from the BGP community attribute for routes that pass the route map conditions.

Command Mode

route map configuration

Syntax Description

<i>ecl-name</i>	Name of the community list.
delete	Deletes communities that match the specified community list from the BGP community attribute.

Default

There are no preconfigured route map set actions. The community list for selected BGP routes is not modified.

Usage Guidelines

Use the **set community-list** command to delete BGP communities matching the community list from the BGP community attribute for routes that pass the route map conditions.

Use the **no** form of this command to disable BGP community deletion.

Examples

The following example deletes communities in the community list, comm06:

```
[local]Redback(config-ctx)#route-map map04  
[local]Redback(config-route-map)#match as-path-list aspath02  
[local]Redback(config-route-map)#set community-list comm06 delete
```

Related Commands

community-list
match community-list
route-map

set dampening

set dampening *half-life reuse-threshold suppress-threshold max-suppress*

no set dampening

Purpose

Sets the Border Gateway Protocol (BGP) dampening policy for routes that pass the route map conditions.

Command Mode

route map configuration

Syntax Description

<i>half-life</i>	Amount of time (in minutes) before a penalty is decreased by half. After a route is assigned a penalty, that penalty is decreased by half after each half-life period elapses. The range of values is 1 to 45 minutes.
<i>reuse-threshold</i>	Route is no longer suppressed when a route penalty level falls below this setting. The range of values is 1 to 20,000.
<i>suppress-threshold</i>	Route is suppressed when a route penalty level exceeds this setting. The range of values is 1 to 20,000.
<i>max-suppress</i>	Maximum amount of time (in minutes) a route can be suppressed. The range of values is 1 to 255.

Default

There are no preconfigured route map set actions. No route advertisement dampening is performed for selected routes.

Usage Guidelines

Use the **set dampening** command to set the BGP dampening policy for routes that pass the route map conditions.

Use the **no** form of this command to disable the configured set action.

Examples

The following example sets the half life to 20 minutes, the reuse threshold to 800, the suppress threshold to 2500, and the maximum suppress time to 80 minutes:

```
[local]Redback(config-ctx)#route-map rmap_Q permit 10  
[local]Redback(config-route-map)#match ip address prefix-list list1  
[local]Redback(config-route-map)#set dampening 20 800 2500 80
```

Related Commands

route-map

set dscp

set dscp *dscp-value*

no set dscp

Purpose

Sets the Differentiated Services Code Point (DSCP) value for routes that pass the route map conditions.

Command Mode

route map configuration

Syntax Description

<i>dscp-value</i>	DSCP value. The range of values is 0 to 63. A keyword value as defined in Table 30-4 can also be specified.

Default

There are no preconfigured route map set actions. The DSCP value for selected routes are not modified.

Usage Guidelines

Use the **set dscp** command to set the DSCP value for routes that pass route-map conditions.

Border Gateway Protocol (BGP) destination-based quality of service (QoS) supports setting the DSCP byte for IP traffic based on BGP attributes including community list and AS path. This can be used by a service provider (SP) to provide multiple levels of service based on a customers IP destination. BGP routes can be assigned a DSCP value based on the BGP table map, route map. When a packet is received on an interface with **mark dscp destination** enabled, and the packet is routed using a route with an associated DSCP, the packet's DSCP is updated and the IP header checksum is re-calculated.

Use the **no** form of this command to disable the configured set action.

Examples

The following example sets the dscp value to 5 for routes passing IP access control list 23 conditions:

```
[local]Redback(config-ctx)#route-map map12 permit 10
[local]Redback(config-route-map)#match ip access-list 23
[local]Redback(config-route-map)#set dscp 5
```

Related Commands

mark dscp destination
route-map

set ext-community

set ext-community {*ext-community-num* [**additive**] | **none**}

no set ext-community

Purpose

Sets the Border Gateway Protocol (BGP) extended community attribute for routes that pass the route map conditions.

Command Mode

route map configuration

Syntax Description

<i>ext-community-num</i>	Extended community number, which can be specified only when configuring an extended community list. It can be expressed in either of the following formats: • <i>tt:asn:nnnn</i>, where <i>tt</i> is the extended community type, <i>asn</i> is the autonomous system number (ASN), and <i>nnnn</i> is a 32-bit integer. The extended community type identifies either a target or origin community. The target community identifies the destination to which the route is going, and the origin community identifies source from where the route originated. The <i>tt</i> argument is a placeholder for either the ro (route origin) keyword, or the rt (route target) keyword. • <i>tt:ip-addr:nn</i>, where <i>tt</i> is the extended community type, <i>ip-addr</i> is the IP address in the form <i>A.B.C.D</i>, and <i>nn</i> is a 16-bit integer.
additive	Optional. Adds the specified extended community numbers to the extended community. You can specify up to eight extended community numbers. Each entry must be separated by a space.
none	Removes the extended community attribute from the routes that pass the route map conditions.

Default

There are no preconfigured route map set actions. The extended community attribute for selected BGP routes is not modified.

Usage Guidelines

Use the **set ext-community** command to set the BGP extended community attribute for routes that pass the route map conditions.

An extended community is a group of destinations that share some common attributes. Each destination can belong to multiple extended communities. Up to eight extended communities can be specified. If the **additive** keyword is used, extended communities are added to the existing BGP extended community list; however, unlike AS path attributes, extended community attributes do not include duplicate entries.

Use the **no** form of this command to disable the configured set action.

Examples

The following example ensures that routes that pass the autonomous system (AS) path list 1 conditions have their extended community attribute set to `rt:10.10.10.1:15`.

```
[local]Redback(config-ctx)#route-map set_ext_community 10 permit  
[local]Redback(config-route-map)#match as-path 1  
[local]Redback(config-route-map)#set ext-community rt:10.10.10.1:15
```

The following example ensures that routes that pass the AS path list 2 conditions have their extended community attribute removed:

```
[local]Redback(config-ctx)#route-map set_ext_community 20 permit  
[local]Redback(config-route-map)#match as-path 2  
[local]Redback(config-route-map)#set ext-community none
```

Related Commands

ext-community-list
match ext-community-list
route-map

set ip next-hop

set ip next-hop {*ip-addr* | **peer-address**}

no set ip next-hop

Purpose

Determines the next-hop IP address used to forward packets for routes that pass the route map conditions.

Command Mode

route map configuration

Syntax Description

<i>ip-addr</i>	Next-hop IP address in the form <i>A.B.C.D</i> .
peer-address	Sets the next-hop IP address to a Border Gateway Protocol (BGP) peer address. For an inbound route map, the system uses the IP address of the BGP neighbor's peer. For an outbound route map, the system uses the IP address of the local BGP peer.

Default

There are no preconfigured route map set actions. The next hops of selected routes are not modified.

Usage Guidelines

Use the **set ip next-hop** command to determine the next-hop IP address used to forward packets for routes that pass the route map conditions. If the **peer-address** keyword is applied to an inbound route map, the next hop of received matching routes is set to the IP address of the BGP neighbor's peer, overriding any third-party next hops. If the **peer-address** keyword is applied to an outbound route map, the next hop of the advertised matching routes is set to the IP address of the local BGP speaker, thus disabling the next-hop calculation.

Use the **no** form of this command to disable the configured set action.

Examples

The following example sets the next hop for routes passing IP access list 1 to the BGP neighbor's peer IP address:

```
[local]Redback(config-ctx)#route-map rmap_Q permit 10
[local]Redback(config-route-map)#match ip access-list 1
[local]Redback(config-route-map)#set ip next-hop peer-address
```

Related Commands

match ip next-hop prefix-list
route-map

set ipv6 next-hop

set ipv6 next-hop {*ipv6-addr* | **peer-address**}

no set ipv6 next-hop

Purpose

Determines the next-hop IP Version 6 (IPv6) address used to forward packets for routes that pass the route map conditions.

Command Mode

route map configuration

Syntax Description

<i>ipv6-addr</i>	Next-hop IPv6 address in the form <i>A:B:C:D:E:F:G</i> .
peer-address	Sets the next-hop IPv6 address to a Border Gateway Protocol (BGP) peer address. For an inbound route map, the system uses the IPv6 address of the BGP neighbor's peer. For an outbound route map, the system uses the IPv6 address of the local BGP peer.

Default

There are no preconfigured route map set actions. The next hops of selected routes are not modified.

Usage Guidelines

Use the **set ipv6 next-hop** command to determine the next-hop IPv6 address used to forward packets for routes that pass the route map conditions. If you apply the **peer-address** keyword to an inbound route map, the next hop of received matching routes is set to the IPv6 address of the BGP neighbor's peer, overriding any third-party next hops. If you apply the **peer-address** keyword to an outbound route map, the next hop of the advertised matching routes is set to the IPv6 address of the local BGP speaker, thus disabling the next-hop calculation.

Use the **no** form of this command to disable the configured set action.

Examples

The following example sets the next hop for routes passing IPv6 access list 1 to the BGP neighbor's peer IPv6 address:

```
[local]Redback(config-ctx)#route-map rmap_Q permit 10
[local]Redback(config-route-map)#match ip access-list 1
[local]Redback(config-route-map)#set ipv6 next-hop peer-address
```

Related Commands

match ipv6 next-hop prefix-list
route-map

set label

set label

no set label

Purpose

Sets the Multiprotocol Label Switching (MPLS) label for routes that pass the route map conditions.

Command mode

route map configuration

Syntax Description

This command has no arguments or keywords.

Default

There are no predefined route map set actions. The label for the route is unmodified.

Usage Guidelines

Use the **set label** command to set the MPLS label for routes that pass the route map conditions.

Use the **no** form of this command to remove the MPLS label setting.

Examples

The following example sets the MPLS label for routes that pass the conditions specified by the route map, foo:

```
[local]Redback(config-ctx)#route-map foo  
[local]Redback(config-route-map)#set label  
[local]Redback(config-route-map)#
```

Related Commands

route-map

set level

set level {**level-1** | **level-1-2** | **level-2** | **nssa-areas** | **transit-areas**}

no set level

Purpose

For routes that pass the route map conditions, sets the advertisement scope for routes redistributed into Open Shortest Path First (OSPF) and Intermediate System-to-Intermediate System (IS-IS) routing domains.

Command Mode

route map configuration

Syntax Description

level-1	Redistributes routes into IS-IS level 1 areas. Routes are not advertised in IS-IS level 2 areas.
level-1-2	Redistributes routes into IS-IS level 1 and level 2 areas.
level-2	Redistributes routes into IS-IS level 2 areas. Routes are not advertised in IS-IS level 1 areas.
nssa-areas	Redistributes routes into OSPF not-so-stubby-areas (NSSAs). Routes are not advertised in OSPF transit areas.
transit-areas	Redistributes routes into OSPF transit areas. Routes are not advertised in OSPF NSSAs.

Default

There are no preconfigured route map set actions. For OSPF, routes are advertised into both regular and transit areas. For IS-IS, routes are advertised into both level 1 and level 2 areas.

Usage Guidelines

Use the **set level** command to set the advertisement scope for routes redistributed into OSPF and IS-IS routing domains.

Use this command in conjunction with the **route-map** command in context configuration mode, with the **redistribute** command in OSPF router configuration mode, and with the **redistribute** command in IS-IS configuration mode.

When a redistributed route is advertised into an OSPF transit area, it is advertised as a type 5 link-state advertisement (LSA). When a redistributed route is advertised into an OSPF NSSA, it is advertised as a type 7 LSA. When the **nssa-area** keyword is specified for a router that is part of an NSSA, but is not an area border router (ABR), the corresponding routes are advertised as type 7 LSAs without the P (propagate) bit set. The propagate bit is described in RFC 1587, *The OSPF NSSA Option*.

Use the **no** form of this command to return the system to its default behavior.

Examples

The following example limits the redistribution of static routes into OSPF transit areas:

```
[local]Redback(config-ctx)#route-map no-nssa-areas permit 10
[local]Redback(config-route-map)#set level transit-areas
[local]Redback(config-route-map)#exit
[local]Redback(config-ctx)#router ospf 1
[local]Redback(config-ospf)#redistribute static route-map no-nssa-areas
```

Related Commands

redistribute—BGP router configuration mode
redistribute—OSPF router configuration mode
redistribute—RIP router configuration mode
route-map

set local-preference

set local-preference *local-pref*

no set local-preference

Purpose

Sets the degree of preference for the Border Gateway Protocol (BGP) autonomous system (AS) path for routes that pass the route map conditions.

Command Mode

route map configuration

Syntax Description

local-pref Integer. The range of values is 0 to 4,294,967,295; the default value is 100.

Default

There are no preconfigured route map set actions. The preference value is for BGP routes is 100.

Usage Guidelines

Use the **set local-preference** command to set the degree of preference for the BGP AS path for routes that pass the route map conditions. The preference is sent only to routers in the local autonomous system. A route with a high value is preferred over a route with a lower value.

Use the **no** form of this command to disable the configured set action.

Examples

The following example sets the local preference for all routes included in route access list 1 to 50:

```
[local]Redback(config-ctx)#route-map rmap_P  
[local]Redback(config-route-map)#match route-access-list 1  
[local]Redback(config-route-map)#set local-preference 50
```

Related Commands

route-map

set metric

set metric [+ | -] *metric*

no set metric

Purpose

Sets, increments, or decrements the metric value for the destination routing protocol for routes that pass the route map conditions.

Command Mode

route map configuration

Syntax Description

+	Optional. Adds the specified metric value.
-	Optional. Subtracts the specified metric value.
<i>metric</i>	Metric value. The range of values is 0 to 4,294,967,295.

Default

There are no preconfigured route map set actions. The metric for selected routes is not modified. The metric value is determined by the application and routing protocol.

Usage Guidelines

Use the **set metric** command to set, increment, or decrement the metric value for the destination routing protocol for routes that pass the route map conditions.

Use the **no** form of this command to disable the configured metric value.

Examples

The following example sets the metric value for the routing protocol to 50:

```
[ local ]Redback(config-ctx)#route-map rmap_M
[ local ]Redback(config-route-map)#set metric 50
```

The following example adds 11 to the metric value for the routing protocol:

```
[ local ]Redback(config-ctx)#route-map add_metric permit 20
[ local ]Redback(config-route-map)#set metric +11
```

Related Commands

match metric
redistribute

route-map
set metric-type

set metric-type

set metric-type { **external** | **internal** | **type-1** | **type-2** }

no set metric-type

Purpose

Sets the metric type for the destination routing protocol for routes that pass the route map conditions.

Command Mode

route map configuration

Syntax Description

external	Specifies the Intermediate System-to-Intermediate System (IS-IS) external metric.
internal	Specifies the Internal Gateway Protocol (IGP) as the Multi-Exit Discriminator (MED) for Border Gateway Protocol (BGP).
type-1	Specifies the Open Shortest Path First (OSPF) external Type 1 metric.
type-2	Specifies OSPF external Type 2 metric.

Default

There are no preconfigured route map set actions. The metric type for selected routes is not modified. For routes redistributed into OSPF, the default metric is Type 2.

Usage Guidelines

Use the **set metric-type** command to set the metric type for the destination routing protocol for routes that pass the route map conditions.

Use the **no** form of this command to disable the configured set action.

Examples

The following example sets the metric type to **external**:

```
[local]Redback(config-ctx)#route-map rmap_M  
[local]Redback(config-route-map)#set metric-type external
```

Related Commands

match metric
redistribute
route-map
set metric

set origin

set origin {egp | igp | incomplete}

no set origin

Purpose

Sets the origin of the Border Gateway Protocol (BGP) path for routes that pass the route map conditions.

Command Mode

route map configuration

Syntax Description

egp	Indicates that the path information originated from another autonomous system (AS).
igp	Sets the origin to the local Interior Gateway Protocol (IGP).
incomplete	Indicates that the origin is unknown.

Default

There are no preconfigured route map set actions. The origin for selected BGP routes is not modified. The origin is determined by the route type.

Usage Guidelines

Use the **set origin** command to set the BGP origin path for routes that pass the route map conditions.

Use the **no** form of this command to disable the configured set action.

Examples

The following example sets the origin of routes that pass the route map conditions to IGP:

```
[local]Redback(config-ctx)#route-map rmap_H  
[local]Redback(config-route-map)#match route-access-list 10  
[local]Redback(config-route-map)#set origin igp
```

Related Commands

route-map

set tag

set tag *tag*

no set tag

Purpose

Sets the route tag value for routes that pass the route map conditions.

Command Mode

route map configuration

Syntax Description

<i>tag</i>	Route tag value. An unsigned 32-bit integer, the range of values is 1 to 4,294,967,295; the default value is 0.

Default

There are no preconfigured route map set actions. The route tag for selected routes is not modified.

Usage Guidelines

Use the **set tag** command to set the route tag value for routes that pass the route map conditions.

Use the **no** form of this command to remove the route tag setting.

Examples

The following example sets the route tag to 8 for routes that pass the route map conditions:

```
[local]Redback(config-ctx)#route-map map_F  
[local]Redback(config-route-map)#set tag 8
```

Related Commands

route-map

set traffic-index

set traffic-index *value*

no set traffic-index

Purpose

Sets the traffic index value for routes that pass the route map conditions.

Command Mode

route map configuration

Syntax Description

value Traffic index number. The range of values is 1 to 8.

Default

There are no preconfigured route map set actions. The traffic-index for selected routes is not modified.

Usage Guidelines

Use the **set traffic-index** command to set the traffic index value for routes that pass the route map conditions.

Per index counters for interfaces with Border Gateway Protocol (BGP) attribute-based accounting enabled are maintained for BGP routes assigned a traffic index. The byte and packet counters for a traffic index are incremented based on the route traversed by IP traffic received on the ingress interface. For more information, see the **traffic-index accounting** command in this chapter, and the **table-map** command in Chapter 12, “BGP Configuration.”

Use the **no** form of this command to remove the traffic index setting.

Examples

The following example sets the traffic index to 3 for routes that pass the route map conditions:

```
[local]Redback(config-ctx)#route-map bgp-accounting permit 10
[local]Redback(config-route-map)#set traffic-index 3
```

Related Commands

table-map
traffic-index accounting

set weight

set weight *weight*

no set weight

Purpose

Sets the degree of preference for Border Gateway Protocol (BGP) routes that pass the route map conditions.

Command Mode

route map configuration

Syntax Description

weight Weight value of a specified BGP route. The range of values is 0 to 65,535.

Default

There are no preconfigured route map set actions. The weight for selected BGP routes is not modified.

Usage Guidelines

Use the **set weight** command to set the degree of preference for BGP routes that pass the route map conditions. A route with a high value is preferred over a route with a lower value.

Use the **no** form of this command to disable the configured set action.

Examples

The following example sets the BGP weight to 50 for routes that are permitted by route access list 10:

```
[local]Redback(config-ctx)#route-map rmap_6  
[local]Redback(config-route-map)#match route-access-list 10  
[local]Redback(config-route-map)#set weight 50
```

Related Commands

route-map

traffic-index accounting

traffic-index accounting

no traffic-index accounting

Purpose

Enables Border Gateway Protocol (BGP) attribute-based accounting on an interface.

Command Mode

interface configuration

Syntax Description

This command has no keywords or arguments.

Default

BGP attribute-based accounting is disabled.

Usage Guidelines

Use the **traffic-index accounting** command to enable BGP attribute-based accounting on an interface.

Per index counters for interfaces with BGP attribute-based accounting enabled are maintained for BGP routes assigned a traffic index. The byte and packet counters for a traffic index are incremented based on the route traversed by IP traffic received on the ingress interface. For more information, see the **set traffic-index** and **table-map** commands.

Use the **no** form of this command to disable BGP attribute-based accounting on an interface.

Examples

The following example enables BGP policy accounting on the interface, *value-added*:

```
[local]Redback(config)#interface value-added  
[local]Redback(config-if)#ip address 10.200.1.1/30  
[local]Redback(config-if)#traffic-index accounting
```

Related Commands

set traffic-index
table-map

MPLS Routing

This part describes the SmartEdge[®] OS tasks and commands used to configure Multiprotocol Label Switching (MPLS), Layer 2 Virtual Private Networks (L2VPNs), Label Distribution Protocol (LDP), and Virtual Private LAN Services (VPLS); it consists of the following chapters:

- Chapter 13, “MPLS Configuration”
- Chapter 14, “L2VPN Configuration”
- Chapter 15, “LDP Configuration”
- Chapter 16, “VPLS Configuration”

MPLS Configuration

This chapter provides an overview of Multiprotocol Label Switching (MPLS), and describes the tasks and commands used to configure MPLS features through the SmartEdge[®] OS.

For information about the tasks and commands used to monitor, troubleshoot, and administer MPLS, see the “MPLS Operations” chapter in the *Routing Protocols Operations Guide* for the SmartEdge OS.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples
- Command Descriptions

Overview

The following sections provide an overview of MPLS and MPLS-related features supported by the SmartEdge router:

- MPLS Architecture
- MPLS QoS
- MPLS TTL
- Next-Hop Fast Reroute

MPLS Architecture

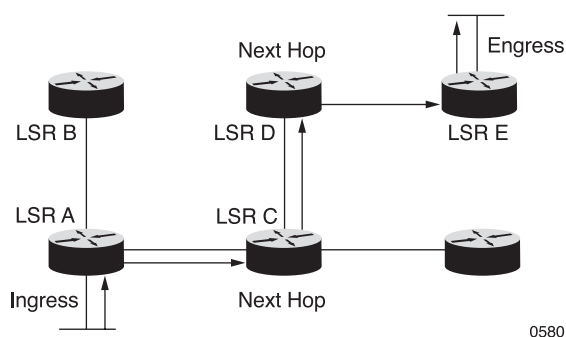
The SmartEdge OS supports Multiprotocol Label Switching (MPLS), which is a method for efficiently forwarding packets through a network. MPLS operates across an interface in an MPLS-enabled context.

In a conventional network, routers forward packets through the network, from one router to the next, with each router making an independent forwarding decision by analyzing the packet header. This conventional approach to forwarding packets has become insufficient to support current networking demands.

With MPLS, the complete analysis of the packet header is performed only once, when it enters an MPLS-enabled network. At each incoming (ingress) point of the network, packets are assigned a label by an edge label-switched router (LSR). Packets are forwarded along a label-switched path (LSP) where each LSR makes forwarding decisions based on the label information. At each hop, the LSR swaps the existing label for a new label that tells the next hop how to forward the packet. At the outgoing (egress) point, an edge LSR removes the label, and forwards the packet to its destination. MPLS uses the Resource Reservation Protocol (RSVP) to communicate labels and their meaning among LSRs.

An LSP is a specific traffic path through an MPLS-enabled network, and can be signaled or static. RSVP LSPs are dynamic. You specify the ingress LSR and the egress LSR, but the next hops through the network are determined using Label Distribution Protocol (LDP), which assign labels in LSRs based on information from existing routing protocols. However, you can also use the **source-path** command (in RSVP LSP configuration mode) to assign an explicit route (a list of specific hops through a network) to an RSVP LSP. RSVP LSPs can usually change according to changes in network conditions, but an RSVP LSP with an assigned source path fails if changing network conditions make it topologically impossible. With static LSPs, you manually specify the ingress LSR, all next-hop LSRs, and the egress LSR. It cannot change with changes in network conditions. Figure 13-1 shows a static LSP through a simple MPLS-enabled network. A packet enters the network at the ingress LSR A, is forwarded to the next-hop LSRs C and D, and exits the network through the egress LSR E.

Figure 13-1 Static LSP in a Simple MPLS-Enabled Network



To enable MPLS forwarding, you must enable an interface for MPLS by creating an MPLS instance, and adding an interface to it. To enable RSVP signaling, you must enable one or more interfaces for RSVP by creating an RSVP instance and adding an interface to it. For static LSPs, there is no need to have RSVP enabled; however, for RSVP LSPs, both RSVP and MPLS must be enabled. If MPLS is not properly enabled in the correct interface, you may have RSVP LSPs that are up, but MPLS forwarding does not yet work.

MPLS QoS

The SmartEdge quality of service (QoS) feature uses the Differentiated Services Code Point (DSCP) value to classify and mark ingress IP packets. At each transit node, the DSCP value is used to select the per-hop behavior (PHB) that determines the scheduling treatment and, in some cases, drop probability for each packet.

QoS DSCP can also be used over MPLS networks by copying the three most significant DSCP bits into the EXP field of MPLS labels at label imposition time.

The default SmartEdge MPLS QoS behavior adheres to the following rules:

- If there are two labels (tunnel and VPN labels) then the DSCP bits are copied into the EXP field of both labels. If penultimate hop popping is enabled, the tunnel label is taken off at the penultimate hop. The egress router will then use the VPN label EXP bits for egress queueing decisions. If there is no VPN label, then the egress router uses the DSCP value.
- If access control list (ACL)-based QoS or policing is used to change the DSCP at the ingress router, then bits 0–2 of this new value must be copied into the EXP field.
- The DSCP value is never changed after the ingress router, even if the EXP value in the tunnel or VPN label is changed.

The SmartEdge OS provides commands that allow you to change the default MPLS QoS behavior to accommodate situations, such as VPN configurations, where you may want to change the way the DSCP bits are handled.

For information about configuring MPLS QoS, see the “QoS Circuit Configuration” chapter in the *IP Services and Security Operations Guide* for the SmartEdge OS.

MPLS TTL

The time-to-live (TTL) field in the IP packet header indicates how many hops a packet can travel before being dropped. The TTL value is decremented by one at each hop, until it reaches zero, and the packet is dropped; however, there needs to be a mechanism to ensure that the TTL field is decremented whenever a packet is labeled and forwarded through an MPLS LSP.

The default SmartEdge behavior ensures that the TTL value is properly decremented by performing the following operations:

- At the ingress LSR, the IP TTL field is propagated to the MPLS TTL field located in the label header.
- The MPLS TTL field is decremented at each hop in the LSP.
- At the egress LSR, the MPLS TTL field replaces the IP TTL field, and the label is popped.

The SmartEdge OS provides commands that allow you to change the default MPLS TTL behavior to accommodate situations, such as VPN configurations, where you may want to change the way the TTL field is handled.

Next-Hop Fast Reroute

Next-hop fast reroute (NFRF) is a feature that allows you to quickly reroute IP and MPLS traffic in the event of a link failure or a node failure. This is done by creating a bypass RSVP LSP for link protection or node protection.

A bypass LSP is no different from any other RSVP LSP, except that it does not carry traffic under normal conditions. When a link or node failure is detected, traffic is quickly rerouted onto a bypass RSVP to circumvent the failure. Traffic enters the headend router of a bypass RSVP, which is called the point of local repair (PLR), and exits the tail end router of the bypass RSVP LSP, which is called the merge point (MP). Any type of traffic intended to use the next hop can be switched onto the bypass LSP.

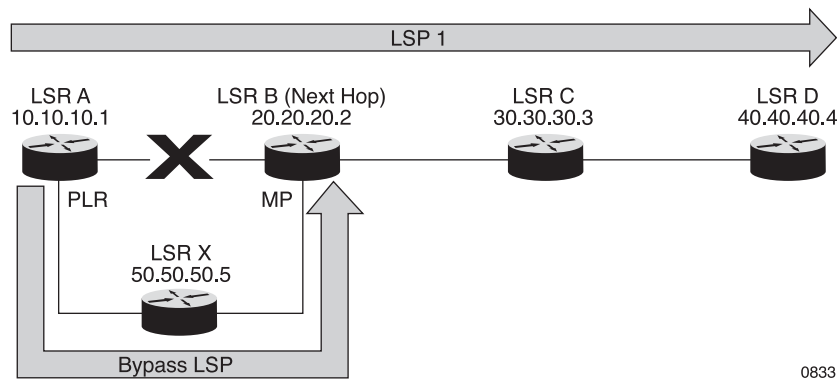
The following sections provide information on the two different types of NFRF:

- NFRF for Link Protection
- NFRF for Node Protection

NFRR for Link Protection

A bypass RSVP LSP for link protection reroutes traffic when a link failure is detected between an LSR and the next-hop LSR. Figure 13-2 shows an example where a bypass RSVP LSP has been created to protect against a link failure. The bypass RSVP LSP is created on LSR A, which is also the PLR, and when the IP address 20.20.20.2 is unreachable across LSP 1, the bypass RSVP LSP provides a path to reroute traffic to LSR B, which is also the MP. Traffic then continues across LSP 1 to LSR C and LSR D.

Figure 13-2 Next-Hop Fast Reroute for Link Protection

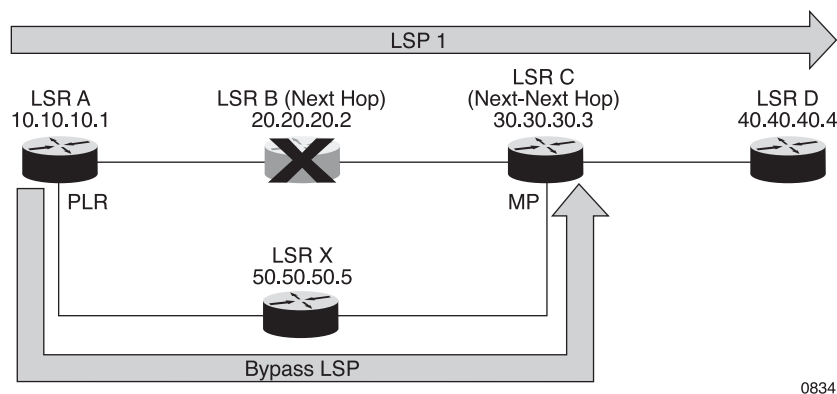


Note When creating a bypass RSVP LSP for link protection you, must specify only the LSR to protect against.

NFRR for Node Protection

A bypass RSVP LSP for node protection reroutes traffic when a next-hop LSR failure is detected. Figure 13-3 shows an example where a bypass RSVP LSP has been created to protect against a node failure. The bypass RSVP LSP is created on LSR A, which is also the PLR, and when LSR B failure is detected, the bypass RSVP LSP provides a path to reroute traffic to LSR C, which is LSR A's next-next hop and the MP. Traffic then continues across LSP 1 to LSR D.

Figure 13-3 Next-Hop Fast Reroute for Node Protection



Note When creating a bypass RSVP LSP for node protection, you must specify the LSR to protect against and the next-next-hop LSR.

Configuration Tasks

Note In this section, the command syntax in the task tables displays only the root command; for the complete command syntax, see the full description for the command in the “Command Descriptions” section.

To configure MPLS and MPLS-related features, perform the tasks described in the following sections:

- Configuring MPLS
- Configuring MPLS Static
- Configuring RSVP

Configuring MPLS

To configure MPLS, perform the tasks described in the following sections:

- Create an MPLS Routing Instance
- Configure the MPLS TTL

Create an MPLS Routing Instance

To create an MPLS routing instance, perform the tasks described in Table 13-1.

Table 13-1 Create an MPLS Routing Instance

Task	Root Command	Notes
Create an MPLS routing instance, and to access MPLS router configuration mode.	router mpls	Enter this command in context configuration mode.
Enable MPLS on an interface.	interface	Enter this command in MPLS router configuration mode.
Configure MPLS TTL.	For the complete list of tasks used to configure MPLS TTL, see the “Configure the MPLS TTL” section.	

Configure the MPLS TTL

To configure the MPLS TTL, perform the tasks described in Table 13-2. Enter all commands in MPLS router configuration mode.

Table 13-2 Configure the MPLS TTL

Task	Root Command	Notes
Enable transit routers to decrement the MPLS TTL by 1 at each hop.	decrement ttl	The default behavior of the SmartEdge router is to decrement the MPLS TTL by 1 at each hop, so the decrement ttl command is used to return the router to its default behavior after it has been changed by the no form of this command.
Enable the propagation of the IP TTL to the MPLS tunnel label TTL at the ingress router.	propagate ttl ip-to-mpls	The default behavior of the SmartEdge router is to propagate of the MPLS tunnel label TTL to the IP TTL at the egress router, so the propagate ttl ip-to-mpls command is used to return the router to its default behavior after it has been changed by the no form of this command.

Table 13-2 Configure the MPLS TTL (*continued*)

Task	Root Command	Notes
Enable the propagation of the MPLS tunnel label TTL to the IP TTL at the egress router.	propagate ttl mpls-to-ip	The default behavior of the SmartEdge router is to propagate of the MPLS tunnel label TTL to the IP TTL at the egress router, so the propagate ttl mpls-to-ip command is used to return the router to its default behavior after it has been changed by the no form of this command.

Configuring MPLS Static

To configure MPLS static, perform the tasks described in the following sections:

- Create an MPLS Static Routing Instance
- Configure an MPLS Static interface
- Configure an MPLS Static LSP

Create an MPLS Static Routing Instance

To create an MPLS static routing instance, perform the task described in Table 13-3.

Table 13-3 Configure an MPLS Static Routing Instance

Task	Root Command	Notes
Create an MPLS static routing instance, and to enter MPLS static router configuration mode.	router mpls-static	Enter this command in context configuration mode.

Configure an MPLS Static interface

To configure an MPLS static interface, perform the tasks described in Table 13-4. Enter all commands in MPLS static interface configuration mode, unless otherwise noted.

Table 13-4 Configure a Static Interface

Task	Root Command	Notes
Enable MPLS static on an interface, and access MPLS interface configuration mode.	interface	Enter this command in MPLS static router configuration mode.
Configure a static MPLS label-action mapping for an intermediate LSR.	label-action	Use the following command syntax: label-action in-label-num [php egress-addr pop swap out-label-num next-hop-addr] Use the swap keyword to replace the incoming label with the outgoing label.
Configure a static MPLS label-action mapping for an egress LSR.	label-action	Use the following command syntax: label-action in-label-num pop Use the pop keyword to remove the top label in the label stack.

Configure an MPLS Static LSP

To configure an MPLS static LSP, perform the tasks described in Table 13-5. Enter all commands in MPLS static LSP configuration mode, unless otherwise noted.

Table 13-5 Configure an MPLS Static LSP

Task	Root Command	Notes
Create a static LSP and enter MPLS static LSP configuration mode.	lsp	Enter this command in MPLS static router configuration mode.
Associate a description with a static LSP.	description	
Configure a next-hop entry for a static LSP.	next-hop	
Specify the IP address of the egress LSR in a static LSP.	egress	An egress LSR is the last router in the chain of routers that constitute an LSP; see Figure 13-1.
Configure the outgoing label number for a static LSP.	out-label	

Configuring RSVP

To configure RSVP, perform the tasks described in the following sections:

- Create an RSVP Routing Instance
- Configure an RSVP LSP
- Configure a Bypass RSVP LSP
- Configure an Explicit Route
- Configure an RSVP Interface
- Configure the RSVP Reservation State Lifetime
- Configure RSVP Graceful Restart

Create an RSVP Routing Instance

To create an RSVP routing instance, perform the tasks described in Table 13-6. Enter all commands in RSVP router configuration mode, unless otherwise noted.

Table 13-6 Create an RSVP Routing Instance

Task	Root Command	Notes
Create an RSVP routing instance within a context and enter RSVP router configuration mode.	router rsvp	Enter this command in context configuration mode.
Enable an egress router to advertise an explicit null label (value 0), in place of an implicit null label (value 3), to the penultimate hop router.	explicit-null	By default, RSVP advertises an implicit null label for directly connected prefixes. An implicit null label causes the upstream router to perform penultimate hop popping (PHP), and the implicit null label is not transmitted on the egress router. In some cases, such as QoS enforcement, PHP may not be desirable. In those cases, using the explicit-null command causes the egress router to advertise an explicit null label in place of an implicit null label for directly connected prefixes, which forces the upstream router to transmit packets with an explicit null label on the last hop.

Table 13-6 Create an RSVP Routing Instance (*continued*)

Task	Root Command	Notes
Enable RSVP LSPs to serve as Interior Gateway Protocol (IGP) shortcuts to nodes in a network.	igp-shortcut	When RSVP LSPs are enabled to serve as IGP shortcuts, link-state protocols, such as Intermediate System-to-Intermediate System (IS-IS) and Open Shortest Path First (OSPF), include the RSVP LSPs in their Shortest Path First (SPF) calculation when determining the shortest-path tree to all nodes in a network. This command (in RSVP router configuration mode) enables all RSVP LSPs for the specified RSVP routing instance to serve as IGP shortcuts. To enable only a specific RSVP LSP to serve as an IGP shortcut, enter this command in RSVP LSP configuration mode.
Enable the generation of RSVP-INFO messages when any RSVP LSP changes state.	log-lsp-up-down	The generation of RSVP-INFO messages cannot be disabled using the no terminal monitor command. Use the no log-lsp-up-down command to disable the generation of RSVP-INFO messages.
Configure the RSVP record route object (RRO) IP prefix type.	rro-prefix-type	Enter this command in RSVP router configuration mode. You can change the IP prefix inside an RRO to be either the router ID or the interface IP address. This is used for node protection and interarea node protection. During NFRR, the PLR LSR needs to match the bypass RSVP LSP egress IP address with the IP prefix inside the RRO of the next-next-hop node.

Configure an RSVP LSP

To configure an RSVP LSP, perform the tasks described in Table 13-7. Enter all commands in RSVP LSP configuration mode, unless otherwise noted.

Note Depending on the command syntax you use for the **lsp** command in RSVP router configuration mode, you can create a standard or backup RSVP LSP.

Table 13-7 Configure an RSVP LSP

Task	Root Command	Notes
Create a standard RSVP LSP and enter RSVP LSP configuration mode.	lsp	Enter this command in RSVP router configuration mode. Use the following command syntax: lsp lsp-name
Create a backup RSVP LSP and enter RSVP LSP configuration mode.	lsp	Enter this command in RSVP router configuration mode. Use the following command syntax: lsp lsp-name backup-for
Specify the bandwidth consumed by an RSVP LSP.	bandwidth	
Associate a description with an RSVP LSP.	description	
Enable an RSVP LSP to serve as an IGP shortcut to nodes in a network.	igp-shortcut	When a RSVP LSP is enabled to serve as an IGP shortcut, link-state protocols, such as IS-IS and OSPF, include the RSVP LSP in their Shortest Path First (SPF) calculation when determining the shortest-path tree to all nodes in a network. This command (in RSVP LSP configuration mode) enables the specified RSVP LSP to serve as an IGP shortcut. To enable all RSVP LSPs for an RSVP routing instance to serve as IGP shortcuts, enter this command in RSVP router configuration mode.

Table 13-7 Configure an RSVP LSP (*continued*)

Task	Root Command	Notes
Specify the IP address of the ingress LSR in an RSVP LSP.	ingress	An ingress LSR is the first router in the chain of routers that constitute an LSP; see Figure 13-1. An ingress IP address does not have to be specified for an RSVP LSP. If it is not specified, the IP address of the interface used to reach the egress IP address is used. If the interface changes, the ingress IP address will also change; however, if an ingress IP address is specified, then the specified address is always used.
Specify the IP address of the egress LSR in an RSVP LSP.	egress	An egress LSR is the last router in the chain of routers that constitute an LSP; see Figure 13-1.
Permit an LSP to be protected by a bypass RSVP LSP.	local-protection	When configured, the LSP advertises to the ingress and transit nodes that a bypass RSVP LSP can be used to provide MPLS fast reroute protection. This configuration affects both ingress LSR and the transit LSRs of the LSP operation.
Assign a configured explicit route to an LSP.	source-path	Before you can assign a source path to an LSP, you must configure an explicit route to use as the source path. Use the explicit-route command in MPLS router configuration mode to indicate a list of specific hops through a network that you want for your LSP, and then use the source-path command to assign that explicit route to your LSP.
Configure an RSVP LSP to actively record the routes through which it forwards packets.	record-route	You can use the recorded route information for troubleshooting, and to prevent routing loops.
Configure the setup priority value for an RSVP LSP.	setup-priority	
Enable or disable an RSVP LSP.	shutdown	Use the no form of this command to enable an existing RSVP LSP.

Configure a Bypass RSVP LSP

To configure a bypass RSVP LSP, perform the tasks described in Table 13-8. Enter all commands in RSVP LSP configuration mode, unless otherwise noted.

Note Depending on the command syntax you use for the **lsp** command in RSVP router configuration mode, you can create a bypass RSVP for one of the following protection schemes:

- Link protection
- Node protection

Table 13-8 Configure a Bypass RSVP LSP

Task	Root Command	Notes
Create a bypass RSVP LSP for link protection and enter RSVP LSP configuration mode.	lsp	Enter this command in RSVP router configuration mode. Use the following command syntax: lsp <i>lsp-name</i> bypass <i>ip-addr</i>
Create a bypass RSVP LSP for node protection and enter RSVP LSP configuration mode.	lsp	Enter this command in RSVP router configuration mode. Use the following command syntax: lsp <i>lsp-name</i> bypass <i>ip-addr</i> node-protect-lsp-egress <i>ip-addr</i>
Specify the bandwidth consumed by a bypass RSVP LSP.	bandwidth	
Associate a description with a bypass RSVP LSP.	description	

Table 13-8 Configure a Bypass RSVP LSP (*continued*)

Task	Root Command	Notes
Configure a bypass RSVP LSP to match the next-next-hop interface IP address.	fast-reroute	If the next-next-hop node does not use the router ID in the RSVP RRO, the PLR LSR can optionally configure the bypass LSP to match a known next-next-hop interface IP address. This is also useful in the case of interarea node protection.
Enable an RSVP LSP to serve as an IGP shortcut to nodes in a network.	igp-shortcut	When a RSVP LSP is enabled to serve as an IGP shortcut, link-state protocols, such as IS-IS and OSPF, include the RSVP LSP in their Shortest Path First (SPF) calculation when determining the shortest-path tree to all nodes in a network. This command (in RSVP LSP configuration mode) enables the specified RSVP LSP to serve as an IGP shortcut. To enable all RSVP LSPs for an RSVP routing instance to serve as IGP shortcuts, enter this command in RSVP router configuration mode.
Specify the IP address of the ingress LSR in a bypass RSVP LSP.	ingress	An ingress LSR is the first router in the chain of routers that constitute an LSP; see Figure 13-1. An ingress IP address does not have to be specified for an RSVP LSP. If it is not specified, the IP address of the interface used to reach the egress IP address is used. If the interface changes, the ingress IP address will also change; however, if an ingress IP address is specified, then the specified address is always used.
Specify the IP address of the egress LSR in a bypass RSVP LSP.	egress	An egress LSR is the last router in the chain of routers that constitute an LSP; see Figure 13-1.
Assign a configured explicit route to an LSP.	source-path	Before you can assign a source path to an LSP, you must configure an explicit route to use as the source path. Use the explicit-route command in MPLS router configuration mode to indicate a list of specific hops through a network that you want for your LSP, and then use the source-path command to assign that explicit route to your LSP.
Configure a bypass RSVP LSP to actively record the routes through which it forwards packets.	record-route	You can use the recorded route information for troubleshooting, and to prevent routing loops.
Configure the setup priority value for a bypass RSVP LSP.	setup-priority	
Enable or disable a bypass RSVP LSP.	shutdown	Use the no form of this command to enable an existing RSVP LSP.

Configure an Explicit Route

When an LSP is configured to use an explicit route, it uses the path determined by that explicit route. If the path defined by the explicit route is not topologically possible, either because the network is partitioned, or insufficient resources are available, the LSP fails. No alternate paths can be used. If the LSP succeeds, it continues to use the explicit route.

To configure an explicit route, perform the tasks described in Table 13-9.

Table 13-9 Configure an Explicit Route

Task	Root Command	Notes
Create an explicit route and access RSVP explicit route configuration mode.	explicit-route	Enter this command in RSVP router configuration mode.
Configure a next-hop entry for an RSVP explicit route.	next-hop	Enter this command in RSVP explicit route configuration mode.

Configure an RSVP Interface

To configure an RSVP interface, perform the tasks described in Table 13-10.

Table 13-10 Configure an RSVP Interface

Task	Root Command	Notes
Enable RSVP on an interface, and access RSVP interface configuration mode.	interface	Enter this command in RSVP router configuration mode.
Enable authentication for an RSVP interface.	authentication	Enter this command in RSVP interface configuration mode. Key chains allow you to control authentication for SmartEdge OS routing protocols. Neighboring routers using RSVP to exchange reservation and path messages must utilize an accepted key ID and key string. If multiple key IDs have been configured, the one with the most recent send time exceeding the current time is used. All key IDs that have not expired and that have a receive time exceeding the current time are accepted. Routes within the same area are not required to use the same authentication key ID. However, if two routers directly exchange updates, they must have the same authentication key ID.
Configure the RSVP reservation state lifetime.		For the complete list of tasks used to configure the RSVP reservation state lifetime, see the “Configure the RSVP Reservation State Lifetime” section.
Configure RSVP graceful restart.		For the complete list of tasks used to configure RSVP graceful restart, see the “Configure RSVP Graceful Restart” section.

Configure the RSVP Reservation State Lifetime

When RSVP is enabled, refresh messages are frequently generated and sent so that reservation states in neighboring nodes do not expire. The lifetime of a reservation state is determined by using two interrelated timing parameters: the keep-multiplier and the refresh-interval. Use the following formula to determine the lifetime of a reservation state:

$$\text{Lifetime} = (\text{keep-multiplier} + 0.5) * 1.5 * \text{refresh-interval}$$

To configure an RSVP reservation state lifetime, perform the tasks described in Table 13-11. Enter all commands in RSVP interface configuration mode, unless otherwise noted.

Table 13-11 Configure the RSVP Reservation State Lifetime

Task	Root Command	Notes
Configure the frequency of generating refresh messages.	refresh-interval	Before you can specify the lifetime of a reservation state using the refresh-interval command, you must ensure that the keep-multiplier timing parameter has also been specified.
Configure the RSVP keep-multiplier timing parameter.	keep-multiplier	Before you can specify the lifetime of a reservation state using the keep-multiplier command, you must ensure that the refresh-interval timing parameter has also been specified.

Configure RSVP Graceful Restart

To configure RSVP graceful restart, perform the tasks described in Table 13-12. Enter all commands in RSVP interface configuration mode, unless otherwise noted.

Table 13-12 Configure the RSVP Graceful Restart

Task	Root Command	Notes
Enable graceful restart for RSVP instance.	graceful-restart	Enter this command in RSVP router configuration mode. RSVP graceful restart relies on RSVP Hello messages to determine if a neighbor is down, and if it should initiate graceful restart procedures. Use the hello interval and hello keep-multiplier commands in RSVP interface configuration mode to enable and configure RSVP Hello messages.
Configure the interval at which RSVP Hello messages are sent out from the specified interface.	hello interval	
Configure the number of lost RSVP Hello messages that can be missed by a neighbor before it declares that the peer adjacency is down.	hello keep-multiplier	

Configuration Examples

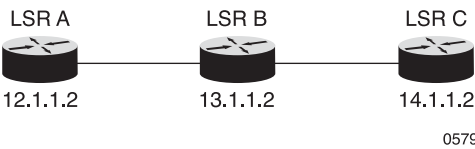
This section provides MPLS configuration examples in the following sections:

- MPLS Static LSP Tunnel
- RSVP LSP Tunnel

MPLS Static LSP Tunnel

The following example illustrates three routers configured to create a MPLS static LSP tunnel between LSR_A and LSR_C, using LSR_B as a next hop. Figure 13-4 shows the network topology for the configuration.

Figure 13-4 MPLS Static LSP Tunnel Network Topology



The configuration for LSR_A is as follows:

```
[local]LSR_A#config
[local]LSR_A(config)#context local
[local]LSR_A(config-ctx)#router mpls-static
[local]LSR_A(config-mpls-static)#lsp new
[local]LSR_A(config-mpls-static-lsp)#next-hop 13.1.1.2
[local]LSR_A(config-mpls-static-lsp)#out-label 30
```

```
[local]LSR_A(config-mpls-static-lsp)#egress 14.1.1.2
[local]LSR_A(config-mpls-static-lsp)#end
```

The configuration for LSR_B is as follows:

```
[local]LSR_B#config
[local]LSR_B(config)#context local
[local]LSR_B(config-ctx)#router mpls-static
[local]LSR_B(config-mpls-static)#interface foo
[local]LSR_B(config-mpls-static-if)#label-action 30 swap 37 14.1.1.2
[local]LSR_B(config-mpls-static-if)#end
```

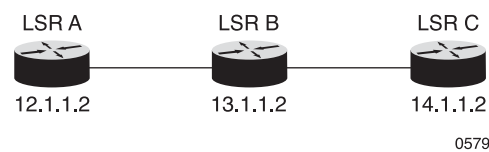
The configuration for LSR_C is as follows:

```
[local]LSR_C#config
[local]LSR_C(config)#context local
[local]LSR_C(config-ctx)#router mpls-static
[local]LSR_C(config-mpls-static)#interface foo
[local]LSR_C(config-mpls-static-if)#label-action 37 pop
[local]LSR_C(config-mpls-static-if)#end
```

RSVP LSP Tunnel

The following example illustrates three routers configured to create an RSVP LSP tunnel between LSR A and LSR C, using LSR B as a next hop. Figure 13-5 shows the network topology for the configuration.

Figure 13-5 RSVP LSP Tunnel Topology



The configuration for LSR_A is as follows:

```
[local]LSR_A#config
[local]LSR_A(config)#context local
[local]LSR_A(config-ctx)#router rsvp
[local]LSR_A(config-rsvp)#interface foo
[local]LSR_A(config-rsvp-if)#exit
[local]LSR_A(config-rsvp)#explicit-route two
[local]LSR_A(config-rsvp-explicit-route)#next-hop 13.1.1.2
[local]LSR_A(config-rsvp-explicit-route)#next-hop 14.1.1.2
[local]LSR_A(config-rsvp-explicit-route)#exit
[local]LSR_A(config-rsvp)#lsp new test
[local]LSR_A(config-rsvp-lsp)#ingress 12.1.1.2
[local]LSR_A(config-rsvp-lsp)#egress 14.1.1.2
[local]LSR_A(config-rsvp-lsp)#source-path two
[local]LSR_A(config-rsvp-lsp)#end
```

The configuration for LSR_B is as follows:

```
[ local]LSR_B#config
[ local]LSR_B(config)#context local
[ local]LSR_B(config-ctx)#router rsvp
[ local]LSR_B(config-rsvp)#interface foo
[ local]LSR_B(config-rsvp-if)#end
```

The configuration for LSR_C is as follows:

```
[ local]LSR_C#config
[ local]LSR_C(config)#context local
[ local]LSR_C(config-ctx)#router rsvp
[ local]LSR_C(config-rsvp)#interface foo
[ local]LSR_C(config-rsvp-if)#end
```

Command Descriptions

This section describes the syntax and usage guidelines for the commands used to configure MPLS features. The commands are presented in alphabetical order.

authentication	local-protection
bandwidth	log-lsp-up-down
decrement ttl	lsp
description	next-hop
egress	out-label
explicit-null	propagate ttl ip-to-mpls
explicit-route	propagate ttl mpls-to-ip
fast-reroute	record-route
graceful-restart	refresh-interval
hello interval	router mpls
hello keep-multiplier	router mpls-static
igp-shortcut	router rsvp
ingress	rro-prefix-type
interface	setup-priority
keep-multiplier	shutdown
label-action	source-path

authentication

authentication *key-chain*

no authentication

Purpose

Enables authentication for a Resource Reservation Protocol (RSVP) interface.

Command Mode

RSVP interface configuration

Syntax Description

key-chain Name of the key chain used for authentication.

Default

Authentication is not enabled.

Usage Guidelines

Use the **authentication** command to enable authentication for an RSVP interface.

Key chains allow you to control authentication for SmartEdge OS routing protocols. Neighboring routers using RSVP to exchange reservation and path messages must utilize an accepted key ID and key string. If multiple key IDs have been configured, the one with the most recent send time exceeded the current time is used. All key IDs that have not expired and that have a receive time exceeding the current time are accepted.

Routes within the same area are not required to use the same authentication key ID. However, if two routers directly exchange updates, they must have the same authentication key ID.

Use the **no** form of this command to disable authentication.

Examples

The following example configures authentication for the RSVP interface, 192.169.1.2:

```
[local]Redback(config-ctx)#router rsvp  
[local]Redback(config-rsvp)#interface 192.169.1.2  
[local]Redback(config-rsvp-if)#authentication auth01
```

Related Commands

keep-multiplier
refresh-interval

bandwidth

bandwidth *value*

Purpose

Specifies the bandwidth consumed by a Resource Reservation Protocol (RSVP) label-switched path (LSP).

Command Mode

RSVP LSP configuration

Syntax Description

<i>value</i>	Bandwidth value specified in bytes per second. The bandwidth value is signalled to the other RSVP peers. Valid values are 1 to 1,000,000,000.

Default

No bandwidth restriction is applied to an LSP.

Usage Guidelines

Use the **bandwidth** command to specify the bandwidth consumed by an RSVP LSP.

Examples

The following example specifies a bandwidth for the RSVP LSP, lsp04, of 1500000 bytes per second:

```
[local]Redback(config-ctx)#router rsvp
[local]Redback(config-rsvp)#lsp lsp04
[local]Redback(config-rsvp-lsp)#bandwidth 1500000
```

Related Commands

- description**
- egress**
- igp-shortcut**
- ingress**
- local-protection**
- lsp**
- record-route**
- setup-priority**
- shutdown**
- source-path**

decrement ttl

decrement ttl

no decrement ttl

Purpose

Enables transit routers to decrement the Multiprotocol Label Switching (MPLS) time-to-live (TTL) by 1 at each hop.

Command Mode

MPLS router configuration

Syntax Description

This command has no keywords or arguments.

Default

Transit routers are enabled to decrement the MPLS TTL by 1 at each hop.

Usage Guidelines

Use the **decrement ttl** command to enable transit routers to decrement the MPLS TTL by 1 at each hop.

Use the **no** form of this command to disable transit routers from decrementing the MPLS TTL by 1 at each hop.

Note The default behavior of the SmartEdge router is to decrement the MPLS TTL by 1 at each hop, so the **decrement ttl** command is used to return the router to its default behavior after it has been changed by the **no** form of this command.

Examples

The following example enables transit routers to decrement the MPLS TTL by 1 at each hop:

```
[local]Redback(config-ctx)#router mpls 234  
[local]Redback(config-mpls)#decrement ttl
```

Related Commands

propagate ttl ip-to-mpls
propagate ttl mpls-to-ip

description

description *text*

no description

Purpose

Associates a description with a static label-switched path (LSP) or a Resource Reservation Protocol (RSVP) LSP.

Command Mode

MPLS static LSP configuration
RSVP LSP configuration

Syntax Description

text Description of the LSP (maximum of 80 characters).

Default

None

Usage Guidelines

Use the **description** command to associate a description with a static LSP or an RSVP LSP. This command does not affect the LSP; it is used only as a note in the configuration.

Use the **no** form of this command to remove a description from the configuration. Because there can be only one description for an LSP, when you use the **no** form of this command, it is not necessary to include the *text* argument.

Examples

The following example provides the description, `Shortcut to Net 41A`, for the MPLS static LSP, `To41A`:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#router mpls-static
[local]Redback(config-mpls-static)#lsp To41A
[local]Redback(config-mpls-static-lsp)#description Shortcut to Net 41A
[local]Redback(config-mpls-static-lsp)#
```

Related Commands

bandwidth	next-hop
egress	out-label
igp-shortcut	record-route
ingress	setup-priority
local-protection	shutdown
lsp	source-path

egress

egress *egress-addr*

Purpose

Specifies the IP address of the egress label-switched router (LSR) in a label-switched path (LSP).

Command Mode

RSVP LSP configuration
MPLS static LSP configuration

Syntax Description

<i>egress-addr</i>	IP address of the egress LSR.

Default

None

Usage Guidelines

Use the **egress** command to specify the IP address of the egress LSR in an LSP.

An egress LSR is the last LSR in the chain of LSRs that constitute an LSP. It forwards packets out of a network. The IP address of the egress LSR must be specified in both signaled and static LSPs.

Examples

The following example configures the egress IP address to 192.168.1.2 for the static LSP, lsp01:

```
[local]Redback(config-ctx)#router mpls-static  
[local]Redback(config-mpls-static)#lsp lsp01  
[local]Redback(config-mpls-static-lsp)#egress 192.168.1.2
```

Related Commands

bandwidth
description
igp-shortcut
ingress
local-protection
lsp
record-route
setup-priority
shutdown
source-path

explicit-null

explicit-null

no explicit-null

Purpose

Enables an egress router to advertise an explicit null label (value 0), in place of an implicit null label (value 3), to the penultimate hop router.

Command Mode

RSVP router configuration

Syntax Description

This command has no keywords or arguments.

Default

The implicit null label (value 3) is advertised.

Usage Guidelines

Use the **explicit-null** command to enable an egress router to advertise an explicit null label (value 0), in place of an implicit null label (value 3), to the penultimate hop router.

By default, Resource Reservation Protocol (RSVP) advertises an implicit null label for directly connected prefixes. An implicit null label causes the upstream router to perform penultimate hop popping (PHP), and the implicit null label is not transmitted on the egress router. In some cases, such as quality of service (QoS) enforcement, PHP may not be desirable. In those cases, using the **explicit-null** command causes the egress router to advertise an explicit null label in place of an implicit null label for directly connected prefixes, which forces the upstream router to transmit packets with an explicit null label on the last hop.

Use the **no** form of this command to use the implicit null label.

Examples

The following example enables the explicit null value:

```
[local]Redback(config-ctx)#router rsvp  
[local]Redback(config-rsvp)#explicit-null
```

Related Commands

explicit-null—LDP router configuration mode

igp-shortcut

log-lsp-up-down

router rsvp

rro-prefix-type

explicit-route

explicit-route *er-name*

no explicit-route *er-name*

Purpose

Creates an explicit route and enters RSVP explicit route configuration mode.

Command Mode

RSVP router configuration

Syntax Description

<i>er-name</i>	Name of the explicit route; an alphanumeric string.

Default

None

Usage Guidelines

Use the **explicit-route** command to create an explicit route and to enter RSVP explicit route configuration mode.

When an LSP is configured to use an explicit route, it uses the path determined by the specified explicit route. If the path defined by the explicit route is not topologically possible, either because the network is partitioned, or because of insufficient resources, the label-switched path (LSP) fails. No alternate paths can be used. If the LSP does not fail, it continues to use the explicit route.

Use the **no** form of this command to delete an explicit route.

Examples

The following example creates an Resource Reservation Protocol (RSVP) explicit route, `ex-route02`, which consists of two next hops:

```
[local]Redback(config-ctx)#router rsvp
[local]Redback(config-rsvp)#explicit-route ex-route02
[local]Redback(config-rsvp-explicit-route)#next-hop 13.1.1.2
[local]Redback(config-rsvp-explicit-route)#next-hop 14.1.1.2
```

Related Commands

lsp

next-hop—RSVP explicit route configuration mode

router rsvp

fast-reroute

fast-reroute *nnhop-intf-address ip-addr*

no fast-reroute *nnhop-intf-address ip-addr*

Purpose

Configures a bypass Resource Reservation Protocol (RSVP) label-switched path (LSP) in node protection to match the next-next-hop interface IP address.

Command Mode

RSVP LSP configuration

Syntax Description

nnhop-intf-address *ip-addr* Next-next-hop node interface IP address.

Default

None

Usage Guidelines

Use the **fast-reroute** command to configure a bypass RSVP LSP for node protection to match the next-next-hop interface IP address. If the next-next-hop node does not use the router ID in the RSVP record route object (RRO), the point of local repair (PLR) node can optionally configure the bypass LSP to match a known next-next-hop interface IP address. This is also useful in the case of interarea MPLS fast reroute for node-protection.

Note The **fast-reroute** command is available only if the bypass RSVP LSP is configured for node protection.

Examples

The following example configures the RSVP LSP, *to-r1-edge*, to match the next-next-hop interface IP address, *10.2.2.2*:

```
[local]Redback(config-ctx)#router rsvp  
[local]Redback(config-rsvp)#lsp to-r1-edge bypass 10.1.1.1 node-protect-lsp-egress  
192.168.1.1  
[local]Redback(config-rsvp-lsp)#fast-reroute nnhop-intf-address 10.2.2.2
```

Related Commands

local-protection
lsp
rro-prefix-type

graceful-restart

graceful-restart
no graceful-restart

Purpose

Enables graceful restart for the Resource Reservation Protocol (RSVP) instance.

Command Mode

RSVP router configuration

Syntax Description

This command has no keywords or arguments.

Default

Graceful restart is disabled.

Usage Guidelines

Use the **graceful-restart** command to enable an RSVP instance to attempt to restart gracefully after a planned or unplanned restart (crash). This implies that the forwarding state is maintained while RSVP reestablishes its neighbor adjacencies and rediscovers LSP soft state. It also implies that the RSVP instance advertises its intent to restart gracefully to its neighbors.

RSVP graceful restart relies on RSVP Hello messages to determine if a neighbor is down, and if it should initiate graceful restart procedures. Use the **hello interval** and **hello keep-multiplier** commands in RSVP interface configuration mode to enable and configure RSVP Hello messages.

Use the **no** form of this command to disable RSVP graceful restart.

Examples

The following example enables an RSVP instance to restart gracefully:

```
[local]Redback(config-ctx)#router rsvp  
[local]Redback(config-rsvp)#graceful-restart
```

Related Commands

hello interval
hello keep-multiplier
router rsvp

hello interval

hello interval *interval*

no hello interval

default hello interval

Purpose

Configures the interval at which Resource Reservation Protocol (RSVP) Hello messages are sent out from the specified interface.

Command Mode

RSVP interface configuration

Syntax Description

<i>interval</i>	Amount of time, in seconds, between consecutive RSVP Hello messages. The range of values is 1 to 60.

Default

The default RSVP Hello interval value is 1.

Usage Guidelines

Use the **hello interval** command to configure the interval at which RSVP Hello messages are sent out from the specified interface.

RSVP Hello messages allow the router to detect the loss of RSVP peer adjacencies, such as when when a neighboring router restarts or the link fails. At regular intervals, RSVP Hello messages containing a HELLO REQUEST object are sent to all adjacent RSVP neighbors. Neighbors receiving the Hello message generate and send an RSVP Hello message containing a HELLO ACK object, which acknowledges that it received the original RSVP Hello message. If a router stops receiving the RSVP Hello message acknowledgements, then it declares that the peer adjacency is down.

Use the **hello keep-multiplier** command to configure the number of lost (unacknowledged) RSVP Hello messages that can be missed by a neighbor before it declares that the peer adjacency is down.

Use the **no** form of this command to disable the sending of RSVP Hello messages.

Use the **default** form of this command to return to the default RSVP Hello interval value of 1.

Examples

The following example configures the `test12` interface to send RSVP Hello messages at intervals of 10 seconds:

```
[local]Redback(config-ctx)#router rsvp
[local]Redback(config-rsvp)#interface test12
[local]Redback(config-rsvp-if)#hello interval 10
```

Related Commands

graceful-restart
hello keep-multiplier
interface—RSVP interface configuration mode

hello keep-multiplier

hello keep-multiplier *multiplier*

default hello keep-multiplier

Purpose

Configures the number of lost (unacknowledged) Resource Reservation Protocol (RSVP) Hello messages that can be missed by a neighbor before it declares that the peer adjacency is down.

Command Mode

RSVP interface configuration

Syntax Description

<i>multiplier</i>	Number of RSVP Hello messages a neighbor can miss. The range of values is 3 to 255.

Default

The default keep multiplier value is 3.

Usage Guidelines

Use the **hello keep-multiplier** command to configure the number of lost (unacknowledged) RSVP Hello messages that can be missed by a neighbor before it declares that the peer adjacency is down.

The advertised holdtime in RSVP Hello packets is the value of the *multiplier* argument multiplied by the value of the *seconds* argument set through the **hello interval** command in RSVP interface configuration mode.

Use the **default** form of this command to return to the default RSVP Hello keep multiplier value of 3.

Examples

The following example specifies that 15 RSVP Hello messages can be missed (unacknowledged) by a neighbor before it declares the RSVP peer adjacency down:

```
[local]Redback(config-ctx)#router rsvp  
[local]Redback(config-rsvp)#interface rsvp05  
[local]Redback(config-rsvp-if)#keep-multiplier 15
```

Related Commands

graceful-restart
hello interval
interface—RSVP interface configuration mode

igp-shortcut

igp-shortcut

no igp-shortcut

Purpose

Enables Resource Reservation Protocol (RSVP) label-switched paths (LSPs) to serve as Interior Gateway Protocol (IGP) shortcuts to nodes in a network.

Command Mode

RSVP router configuration

RSVP LSP configuration

Syntax Description

This command has no keywords or arguments.

Default

IGP shortcuts are disabled.

Usage Guidelines

Use the **igp-shortcut** command to enable RSVP LSPs to serve as IGP shortcuts to nodes in a network. When RSVP LSPs are enabled to serve as IGP shortcuts, link-state protocols, such as Intermediate System-to-Intermediate System (IS-IS) and Open Shortest Path First (OSPF), include the RSVP LSPs in their Shortest Path First (SPF) calculation when determining the shortest-path tree to all nodes in a network.

When entered in RSVP router configuration mode, this command enables all RSVP LSPs for the specified RSVP routing instance to serve as IGP shortcuts. When entered in RSVP LSP configuration mode, only the specified RSVP LSP is enabled to serve as an IGP shortcut.

For more information about IGP shortcuts, see RFC 3906, *Calculating Interior Gateway Protocol (IGP) Routes Over Traffic Engineering Tunnels*.

Use the **no** form of this command to disable RSVP LSPs from serving as IGP shortcuts.

Examples

The following example enables the RSVP LSP, `lspfoo`, to serve as an IGP shortcut:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#router rsvp
[local]Redback(config-rsvp)#lsp lspfoo
[local]Redback(config-rsvp-lsp)#igp-shortcut
```

Related Commands

bandwidth
description
egress
explicit-null
ingress
local-protection
log-lsp-up-down
lsp
record-route
router rsvp
rro-prefix-type
setup-priority
shutdown
source-path

ingress

ingress *ingress-addr*

Purpose

Specifies the IP address of the ingress label-switched router (LSR) in a Resource Reservation Protocol (RSVP) label-switched path (LSP).

Command Mode

RSVP LSP configuration

Syntax Description

ingress-addr IP address of the ingress LSR.

Default

None

Usage Guidelines

Use the **ingress** command to specify the IP address of the ingress LSR in an RSVP LSP. The ingress LSR is an edge LSR that forwards packets into a network, and is the first router in the chain of routers that constitute an LSP.

Note An ingress IP address does not have to be specified for an RSVP LSP. If it is not specified, the IP address of the interface used to reach the egress IP address is used. If the interface changes, the ingress IP address will also change; however, if an ingress IP address is specified, then the specified address is always used.

Examples

The following example configures the ingress IP address to 192.168.1.5 for the RSVP LSP, lsp01:

```
[local]Redback(config-ctx)#router rsvp
[local]Redback(config-rsvp)#lsp lsp01
[local]Redback(config-rsvp-lsp)#ingress 192.168.1.5
```

Related Commands

bandwidth	lsp
description	record-route
egress	setup-priority
igp-shortcut	shutdown
local-protection	source-path

interface

interface *if-name*

no interface *if-name*

Purpose

When entered in MPLS router configuration, enables Multiprotocol Label Switching (MPLS) routing on an interface.

When entered in MPLS static router configuration, enables static MPLS routing on an interface, and enters MPLS static interface configuration mode.

When entered in RSVP router configuration mode, enables Resource Reservation Protocol (RSVP) routing on an interface, and enters RSVP interface configuration mode.

Command Mode

MPLS router configuration

MPLS static router configuration

RSVP router configuration

Syntax Description

<i>if-name</i>	Name of the interface; an alphanumeric string.

Default

None

Usage Guidelines

Use the **interface** command in MPLS router configuration to enable MPLS routing on an interface.

Use the **interface** command in MPLS static router configuration to enable static MPLS routing on an interface, and enter MPLS static interface configuration mode.

Use the **interface** command in RSVP router configuration mode to enable RSVP routing on an interface, and enter RSVP interface configuration mode.

Note If an RSVP interface is not created, RSVP packets cannot be received, and the label-switched path (LSP) setup will fail.

Use the **no** form of this command to delete an interface.

Examples

The following example enables MPLS routing on the `mpls22` interface:

```
[local]Redback(config-ctx)#router mpls
[local]Redback(config-mpls)#interface mpls22
[local]Redback(config-mpls-if)#
```

The following example enables static MPLS routing on the `statmpls` interface and enters MPLS static interface configuration mode:

```
[local]Redback(config-ctx)#router mpls-static  
[local]Redback(config-mpls)#interface statmpls  
[local]Redback(config-mpls-static-if)#
```

The following example enables RSVP routing on the `rsvp05` interface and enters RSVP interface configuration mode:

```
[local]Redback(config-ctx)#router rsvp  
[local]Redback(config-rsvp)#interface rsvp05  
[local]Redback(config-rsvp-if)#
```

Related Commands

authentication—RSVP interface configuration mode
decrement ttl
egress
explicit-null
explicit-route
graceful-restart
hello interval
hello keep-multiplier
keep-multiplier
label-action
log-lsp-up-down
lsp
refresh-interval

keep-multiplier

keep-multiplier *multiplier*

Purpose

Configures the Resource Reservation Protocol (RSVP) keep-multiplier timing parameter.

Command Mode

RSVP interface configuration

Syntax Description

<i>multiplier</i>	Multiplier used for calculating the lifetime of a reservation state. The range of values is 1 to 255.

Default

The default keep-multiplier value is 3.

Usage Guidelines

Use the **keep-multiplier** command to configure the RSVP keep-multiplier timing parameter.

When RSVP is enabled, refresh messages are sent periodically so that reservation states in neighboring nodes do not expire. The lifetime of a reservation state is determined by using two interrelated timing parameters: the keep-multiplier and the refresh-interval. Use the following formula to determine the lifetime of a reservation state:

$$\text{Lifetime} = (\text{keep-multiplier} + 0.5) * 1.5 * \text{refresh-interval}$$

Examples

The following example configures the keep-multiplier timing parameter to 15:

```
[local]Redback(config-ctx)#router rsvp
[local]Redback(config-rsvp)#interface rsvp05
[local]Redback(config-rsvp-if)#keep-multiplier 15
```

Related Commands

authentication
refresh-interval

label-action

label-action *in-label-num* [**php** *egress-addr* | **pop** | **swap** *out-label-num next-hop-addr*]

no label-action *in-label-num* [**php** *egress-addr* | **pop** | **swap** *out-label-num next-hop-addr*]

Purpose

Configures a static Multiprotocol Label Switching (MPLS) label-action mapping.

Command Mode

MPLS static interface configuration

Syntax Description

<i>in-label-num</i>	Number of the incoming label. The range of values is 16 to 1,024.
php	Optional. Penultimate Hop Pop pops (removes) the label before forwarding the IP-only packet from the egress label-switched router (LSR). The egress LSR then forwards the packet based on its destination address.
<i>egress-addr</i>	Optional. IP address of the egress LSR.
pop	Optional. Pops (removes) the top label in the stack and forwards the remaining payload as either a labeled packet, or an unlabeled IP packet.
swap	Optional. Replaces the incoming label with the outgoing label, and forwards to the IP address of the next hop.
<i>out-label-num</i>	Optional. Number of the outgoing label. The range of values is 16 to 1,024.
<i>next-hop-addr</i>	Optional. IP address of the next hop.

Default

None

Usage Guidelines

Use the **label-action** command to configure a static MPLS label-action mapping for the MPLS static interface.

Label actions change the label information for labeled packets as they are forwarded through an LSR. For instance, a label can be removed from a stack of labels, a label can be swapped for another label, or the label can be completely removed from the packet.

Use the **no** form of this command to delete a static MPLS label-action mapping.

Examples

The following example swaps the MPLS label 16 for label 24 and forwards the labeled packet to the next hop 10.10.10.2:

```
[local]Redback(config-ctx)#router mpls-static  
[local]Redback(config-mpls-static)#interface isp6  
[local]Redback(config-mpls-static-if)#label-action 16 swap 24 10.10.10.2
```

Related Commands

- egress**
- interface**
- lsp**
- next-hop**
- out-label**
- router mpls-static**

local-protection

local-protection

no local-protection

Purpose

Permits a label-switched path (LSP) to be protected by a bypass Resource Reservation Protocol (RSVP) LSP.

Command Mode

RSVP LSP configuration

Syntax Description

This command has no keywords or arguments.

Default

Local protection is permitted.

Usage Guidelines

Use the **local-protection** command to permit an LSP to be protected by a bypass RSVP LSP. When configured, the LSP advertises to the ingress and transit nodes that a bypass RSVP LSP can be used to provide Multiprotocol Label Switching (MPLS) fast reroute protection. This configuration will affect both ingress node and the transit nodes of the LSP operation.

Use the **no** form of this command to deny an LSP from being protected by a bypass RSVP LSP. Local protection can be denied for operational or resource issues.

Examples

The following example configures an RSVP LSP, `to-r2-core`, to deny MPLS fast reroute protection:

```
[local]Redback(config-ctx)#router rsvp  
[local]Redback(config-rsvp)#lsp to-r2-core  
[local]Redback(config-rsvp-lsp)#no local-protection
```

Related Commands

bandwidth
description
egress
igp-shortcut
ingress

lsp
record-route
setup-priority
shutdown
source-path

log-lsp-up-down

log-lsp-up-down

no log-lsp-up-down

Purpose

Enables the logging of RSVP-INFO messages when any Resource Reservation Protocol (RSVP) label-switched path (LSP) changes state.

Command Mode

RSVP router configuration

Syntax Description

This command has no keywords or arguments.

Default

RSVP-INFO messages are not logged.

Usage Guidelines

Use the **log-lsp-up-down** command to enable the logging of RSVP-INFO messages when any RSVP LSP changes state. The state can change from Up to Down, or from Down to Up.

Note The generation of RSVP-INFO messages cannot be disabled using the **no terminal monitor** command.

Use the **no** form of this command to disable the logging of RSVP-INFO messages.

Examples

The following example enables the logging of RSVP-INFO messages when any RSVP LSP changes state:

```
[ local ]Redback(config-ctx)#router rsvp  
[ local ]Redback(config-rsvp)#log-lsp-up-down
```

Related Commands

explicit-null
igp-shortcut
router rsvp
rro-prefix-type

lsp

```
lsp lsp-name [backup-for lsp-name | bypass ip-addr [node-protect-lsp-egress ip-addr]]  
no lsp lsp-name [backup-for lsp-name | bypass ip-addr [node-protect-lsp-egress ip-addr]]
```

Purpose

When entered in MPLS static router configuration mode, creates a static label-switched path (LSP), and enters MPLS static LSP configuration mode.

When entered in RSVP router configuration mode, creates an RSVP LSP, and enters RSVP LSP configuration mode.

Command Mode

MPLS static router configuration
RSVP router configuration

Syntax Description

<i>lsp-name</i>	Name of the LSP.
backup-for <i>lsp-name</i>	Optional. Primary RSVP LSP name. Creates an LSP to back up a primary RSVP LSP. This option is only available when configuring an RSVP LSP in RSVP LSP configuration mode.
bypass <i>ip-addr</i>	Optional. Bypass LSP for next-hop fast reroute (NFRR) link protection. The <i>ip-addr</i> argument is the IP address of the directly connected next-hop node being protected. This option is only available when configuring a signaled LSP in RSVP LSP configuration mode.
node-protect-lsp-egress <i>ip-addr</i>	Optional. Bypass LSP for NFRR node protection. The <i>ip-addr</i> argument specifies the egress IP address of the bypass LSP. This option is only available when configuring a signaled LSP in RSVP LSP configuration mode, and when the LSP is being configured as a bypass LSP.

Default

None

Usage Guidelines

Use the **lsp** command in MPLS static router configuration mode to create a static LSP, and enter MPLS static LSP configuration mode.

Use the **lsp** command in RSVP router configuration mode to create an RSVP LSP, and enter RSVP LSP configuration mode.

Use the **backup-for** *lsp-name* construct to create a backup RSVP LSP for a primary RSVP LSP. A backup RSVP LSP remains in hot standby, which means that it is always consuming resources and available for passing traffic. If RSVP signals that the primary RSVP LSP as gone down, the backup RSVP LSP immediately begins passing traffic.

Use the **bypass** *ip-addr* construct to configure the RSVP LSP as a bypass LSP for NFRR link protection. A bypass LSP is no different from any other RSVP LSP, except that it does not carry traffic under normal conditions. It is configured to reach the next-hop router in the event of a link failure. Any type of traffic intended to use the next hop can be switched onto the bypass LSP.

Use the **node-protect-lsp-egress** *ip-addr* construct to use the bypass LSP for NFRR node protection. In the event of a link failure or a next-hop node failure, traffic is switched to the bypass LSP. If a bypass LSP is configured without enabling node protection, then the bypass LSP is used only for link protection.

Use the **no** form of this command to delete an LSP.

Examples

The following example configures the static LSP, `sl10`, to use the next-hop label-switched router (LSR), `192.168.1.24`, the egress LSR, `192.168.100.2`, and to set the outgoing label value to 3:

```
[local]Redback(config-ctx)#router mpls-static
[local]Redback(config-mpls-static)#lsp sl10
[local]Redback(config-mpls-static-lsp)#next-hop 192.168.1.24
[local]Redback(config-mpls-static-lsp)#egress 192.168.100.2
[local]Redback(config-mpls-static-lsp)#out-label 3
```

The following example configures the RSVP LSP, `12`, to use the ingress LSR, `13.1.1.1`, the egress LSR, `14.1.1.1`, and the explicit route `two` as its source path:

```
[local]Redback(config-ctx)#router rsvp
[local]Redback(config-rsvp)#lsp 12
[local]Redback(config-rsvp-lsp)#ingress 13.1.1.1
[local]Redback(config-rsvp-lsp)#egress 14.1.1.2
[local]Redback(config-rsvp-lsp)#source-path two
```

The following example configures the RSVP LSP, `to-r2-core`, as a bypass LSP for link protection:

```
[local]Redback(config-ctx)#router rsvp
[local]Redback(config-rsvp)#lsp to-r2-core bypass 10.1.1.1
[local]Redback(config-rsvp-lsp)#egress 192.168.1.1
```

Related Commands

bandwidth	local-protection
description	next-hop
egress	out-label
explicit-route	record-route
fast-reroute	rro-prefix-type
igp-shortcut	setup-priority
ingress	shutdown
label-action	source-path

next-hop

next-hop *next-hop-addr*
no next-hop *next-hop-addr*

Purpose

Configures a next-hop entry for a Resource Reservation Protocol (RSVP) explicit route, or for a static label-switched path (LSP).

Command Mode

MPLS static LSP configuration
RSVP explicit route configuration

Syntax Description

next-hop-addr IP address of the next-hop label-switched router (LSR).

Default

None

Usage Guidelines

Use the **next-hop** command to configure a next-hop entry for an RSVP explicit route, or for a static LSP.

Use the **no** form of this command to remove a next-hop entry from an RSVP explicit route. You cannot remove a next-hop entry from a static LSP.

Examples

The following example configures two next-hop entries for an RSVP explicit route:

```
[local]Redback(config-ctx)#router rsvp  
[local]Redback(config-rsvp)#explicit-route ex-route02  
[local]Redback(config-rsvp-explicit-route)#next-hop 13.1.1.2  
[local]Redback(config-rsvp-explicit-route)#next-hop 14.1.1.2
```

The following example configures two next-hop entries for a static LSP:

```
[local]Redback(config-ctx)#router mpls-static  
[local]Redback(config-mpls-static)#lsp 24  
[local]Redback(config-mpls-static-lsp)#next-hop 20.20.20.10  
[local]Redback(config-mpls-static-lsp)#next-hop 30.20.20.16
```

Related Commands

description	out-label
egress	router mpls-static
explicit-route	router rsvp
lsp	

out-label

out-label *out-label-num*

Purpose

Configures the outgoing label number for a static label-switched path (LSP).

Command Mode

MPLS static LSP configuration

Syntax Description

<i>out-label-num</i>	Number of the outgoing label. The range of values is 16 to 1,024.

Default

None

Usage Guidelines

Use the **out-label** command to configure the outgoing label number for a static LSP.

Examples

The following example configures the outgoing label for the LSP, test14, to the value of 20:

```
[ local]Redback(config-ctx)#router mpls-static  
[ local]Redback(config-mpls-static)#lsp test14  
[ local]Redback(config-mpls-static-lsp)#out-label 20
```

Related Commands

description
egress
lsp
next-hop

propagate ttl ip-to-mpls

propagate ttl ip-to-mpls

no propagate ttl ip-to-mpls

Purpose

Enables the propagation of the IP time-to-live (TTL) to the Multiprotocol Label Switching (MPLS) tunnel label TTL at the ingress router.

Command Mode

MPLS router configuration

Syntax Description

This command has no keywords or arguments.

Default

The IP TTL is propagated to the MPLS tunnel label TTL at the ingress router.

Usage Guidelines

Use the **propagate ttl ip-to-mpls** command to enable the propagation of the IP TTL to the MPLS tunnel label TTL at the ingress router.

Use the **no** form of this command to disable the propagation of the IP TTL to the MPLS tunnel label TTL at the ingress router.

Note The default behavior of the SmartEdge router is to propagate the IP TTL to the MPLS tunnel label TTL at the ingress router; therefore, the **propagate ttl ip-to-mpls** command is only used to return the router to its default behavior after it has been changed using the **no** form of this command.

Examples

The following example enables the propagation of the IP TTL to the MPLS tunnel label TTL:

```
[local]Redback(config-ctx)#router mpls 234
[local]Redback(config-mpls)#propagate ttl ip-to-mpls
[local]Redback(config-mpls)#
```

Related Commands

decrement ttl

propagate ttl mpls-to-ip

propagate ttl mpls-to-ip

propagate ttl mpls-to-ip

no propagate ttl mpls-to-ip

Purpose

Enables the propagation of the Multiprotocol Label Switching (MPLS) tunnel label time-to-live (TTL) to the IP TTL at the egress router.

Command Mode

MPLS router configuration

Syntax Description

This command has no keywords or arguments.

Default

The MPLS TTL tunnel label is propagated to the IP TTL at the egress router.

Usage Guidelines

Use the **propagate ttl mpls-to-ip** command to enable the propagation of the MPLS tunnel label TTL to the IP TTL at the egress router.

Use the **no** form of this command to disable the propagation of the MPLS tunnel label TTL to the IP TTL at the egress router.

Note The default behavior of the SmartEdge router is to propagate of the MPLS tunnel label TTL to the IP TTL at the egress router, so the **propagate ttl mpls-to-ip** command is only used to return the router to its default behavior after it has been changed using the **no** form of this command.

Examples

The following example enables the propagation of the MPLS tunnel label TTL to the IP TTL at the egress router:

```
[local]Redback(config-ctx)#router mpls 234
[local]Redback(config-mpls)#propagate ttl mpls-to-ip
[local]Redback(config-mpls)#
```

Related Commands

decrement ttl

propagate ttl ip-to-mpls

record-route

record-route

no record-route

Purpose

Configures a Resource Reservation Protocol (RSVP) label-switched path (LSP) to actively record the routes through which the LSP forwards packets.

Command Mode

RSVP LSP configuration

Syntax Description

This command has no keywords or arguments.

Default

Route information is recorded.

Usage Guidelines

Use the **record-route** command to configure an RSVP LSP to actively record the routes through which the LSP forwards packets.

Use the **show rsvp lsp** command to display the detailed output containing information about the recorded route, which you can use for troubleshooting purposes, and to prevent routing loops.

Use the **no** form of this command to disable route recording for the RSVP LSP.

Examples

The following example configures the LSP, `test07`, to actively record the routes through which it forwards packets:

```
[local]Redback(config-ctx)#router rsvp  
[local]Redback(config-rsvp)#lsp test07  
[local]Redback(config-rsvp-lsp)#record-route
```

Related Commands

bandwidth
description
egress
igp-shortcut
ingress

local-protection
lsp
setup-priority
shutdown
source-path

refresh-interval

refresh-interval *interval*

Purpose

Configures the frequency of generating refresh messages.

Command Mode

RSVP interface configuration

Syntax Description

<i>interval</i>	Frequency, in seconds, at which refresh messages are generated. The range of values is 1 to 65,535.

Default

Refresh messages are generated every 30 seconds.

Usage Guidelines

Use the **refresh-interval** command to configure the frequency of generating refresh messages.

When RSVP is enabled, refresh messages are sent periodically so that reservation states in neighboring nodes do not expire. The lifetime of a reservation state is determined by using two interrelated timing parameters: the keep-multiplier and the refresh-interval. Use the following formula to determine the lifetime of a reservation state:

$$\text{Lifetime} = (\text{keep-multiplier} + 0.5) * 1.5 * \text{refresh-interval}$$

Examples

The following example sets the refresh-interval timing parameter to 45 seconds:

```
[local]Redback(config-ctx)#router rsvp
[local]Redback(config-rsvp)#interface rsvp05
[local]Redback(config-rsvp-if)#refresh-interval 45
```

Related Commands

authentication—RSVP interface configuration mode
keep-multiplier

router mpls

router mpls

no router mpls

Purpose

Enables Multiprotocol Label Switching (MPLS) routing within a context and enters MPLS router configuration mode.

Command Mode

context configuration

Syntax Description

This command has no keywords or arguments.

Default

MPLS routing is disabled.

Usage Guidelines

Use the **router mpls** command to enable MPLS routing within a context and enter MPLS router configuration mode.

Use the **no** form of this command to disable MPLS routing.

Examples

The following example enables MPLS routing and enters MPLS router configuration mode:

```
[local]Redback(config)#context isp33  
[local]Redback(config-ctx)#router mpls  
[local]Redback(config-mpls)#
```

Related Commands

decrement ttl
egress
igp-shortcut
interface
propagate ttl ip-to-mpls
propagate ttl mpls-to-ip
router mpls-static
router rsvp

router mpls-static

router mpls-static

no router mpls-static

Purpose

Enables static Multiprotocol Label Switching (MPLS) routing within a context and enters MPLS static router configuration mode.

Command Mode

context configuration

Syntax Description

This command has no keywords or arguments.

Default

Static MPLS routing is disabled.

Usage Guidelines

Use the **router mpls-static** command to enable static MPLS routing within a context and enter MPLS static router configuration mode.

Use the **no** form of this command to disable static MPLS routing.

Examples

The following example enables static MPLS routing and enters MPLS static router configuration mode:

```
[local]Redback(config)#context isp33  
[local]Redback(config-ctx)#router mpls-static  
[local]Redback(config-mpls-static)#
```

Related Commands

interface

lsp

router mpls

router rsvp

router rsvp

router rsvp

no router rsvp

Purpose

Enables Resource Reservation Protocol (RSVP) routing within a context and enters RSVP router configuration mode.

Command Mode

context configuration

Syntax Description

This command has no keywords or arguments.

Default

RSVP is disabled.

Usage Guidelines

Use the **router rsvp** command to enable RSVP routing within a context and enter RSVP router configuration mode.

Use the **no** form of this command to disable RSVP routing within a context.

Examples

The following example enables RSVP routing and enters RSVP router configuration mode:

```
[local]Redback(config)#context isp35  
[local]Redback(config-ctx)#router rsvp  
[local]Redback(config-rsvp)#
```

Related Commands

authentication
explicit-null
igp-shortcut
interface
keep-multiplier
label-action
log-lsp-up-down
lsp
refresh-interval
router mpls
rro-prefix-type

rro-prefix-type

rro-prefix-type {**router-id** | **interface**}

no rro-prefix-type {**router-id** | **interface**}

Purpose

Configures the Resource Reservation Protocol (RSVP) record route object (RRO) IP prefix type.

Command Mode

RSVP router configuration

Syntax Description

router-id	Uses the router ID as the IP prefix when sending an RRO.
interface	Uses the outbound interface IP address when sending an RRO.

Default

The router ID is used as the IP prefix type when sending an RRO.

Usage Guidelines

Use the **rro-prefix-type** command to configure the RSVP RRO IP prefix type. You can change the IP prefix inside an RRO to be either the router ID or the interface IP address. This can be used for Multiprotocol Label Switching (MPLS) fast reroute for node protection and interarea node protection. During MPLS fast reroute, the point of local repair (PLR) router needs to match the bypass label-switched path (LSP) egress address with the IP prefix inside the RRO of the next-next-hop node.

Examples

The following example configures the RSVP RRO to use the outbound interface IP address when sending an RRO:

```
[local]Redback(config-ctx)#router rsvp
[local]Redback(config-rsvp)#rro-prefix-type interface
```

Related Commands

explicit-null
igp-shortcut
log-lsp-up-down
router rsvp

setup-priority

setup-priority *value*

Purpose

Configures the setup priority value for a Resource Reservation Protocol (RSVP) label-switched path (LSP).

Command Mode

RSVP LSP configuration

Syntax Description

value Setup priority value. Valid values are 0 to 7.

Default

None

Usage Guidelines

Use the **setup-priority** command to configure the setup priority value for an RSVP LSP.

Examples

The following example configures the setup priority value for the RSVP LSP, lsp04, to 5:

```
[local]Redback(config-ctx)#router rsvp  
[local]Redback(config-rsvp)#lsp lsp04  
[local]Redback(config-rsvp-lsp)#setup-priority 5
```

Related Commands

bandwidth
description
egress
igp-shortcut
ingress
local-protection
lsp
record-route
shutdown
source-path

shutdown

shutdown

no shutdown

Purpose

Disables a Resource Reservation Protocol (RSVP) label-switched path (LSP).

Command Mode

RSVP LSP configuration

Syntax Description

This command has no keywords or arguments.

Default

The RSVP LSP is enabled when configured.

Usage Guidelines

Use the **shutdown** command to disable an RSVP LSP.

Use the **no** form of this command to enable an existing RSVP LSP that has been disabled.

Examples

The following example disables the RSVP LSP, test03:

```
[local]Redback(config-ctx)#router rsvp
[local]Redback(config-rsvp)#lsp test03
[local]Redback(config-rsvp-lsp)#shutdown
```

Related Commands

bandwidth
description
egress
igp-shortcut
ingress
local-protection
lsp
record-route
setup-priority
source-path

source-path

source-path *er-name*

no source-path *er-name*

Purpose

Assigns a configured explicit route to a label-switched path (LSP).

Command Mode

RSVP LSP configuration

Syntax Description

<i>er-name</i>	Name of the explicit route to be used by the LSP.

Default

None

Usage Guidelines

Use the **source-path** command to assign a configured explicit route to an LSP.

Before you can assign a source path to an LSP, you must configure an explicit route to use as the source path. Use the **explicit-route** command in RSVP router configuration mode to indicate a list of specific hops through a network, and then use the **source-path** command to assign that explicit route to your LSP.

Use the **no** form of this command to remove an explicit route from an LSP.

Examples

The following example assigns the explicit route ER03 as the source path for the LSP, Prod23:

```
[local]Redback(config-ctx)#router rsvp
[local]Redback(config-rsvp)#lsp Prod23
[local]Redback(config-rsvp-lsp)#source-path ER03
```

Related Commands

bandwidth	local-protection
description	lsp
egress	record-route
igp-shortcut	setup-priority
ingress	shutdown

L2VPN Configuration

This chapter provides an overview of Layer 2 Virtual Private Networks (L2VPNs) and describes the tasks and commands used to configure L2VPN features through the SmartEdge® OS.

For information about the tasks and commands used to monitor, troubleshoot, and administer L2VPNs, see the “L2VPN Operations” chapter in the *Routing Protocols Operations Guide* for the SmartEdge OS.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples
- Command Descriptions

Overview

The following sections provide an overview of L2VPN:

- L2VPN Implementation
- Supported Encapsulation Types
- Supported Encapsulation Interconnectivity
- QoS Policies for L2VPN Circuits
- L2VPN over GRE

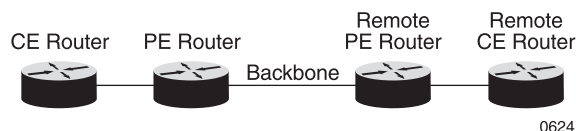
L2VPN Implementation

Customer edge (CE) routers send Layer 2 (L2) traffic to provider edge (PE) routers over L2 circuits configured between the PE and the CE routers. An L2 circuit can be either an Ethernet port, an 802.1Q virtual LAN (VLAN), a Frame Relay permanent virtual circuit (PVC), or an Asynchronous Transfer Mode (ATM) PVC.

An L2VPN is configured on PE routers. The purpose of an L2VPN configuration is to cross-connect a local L2 circuit with a corresponding remote L2 circuit through an label-switched path (LSP) tunnel that crosses the network backbone.

Figure 14-1 displays the network topology for an L2VPN configuration. The cross-connection between the local L2 circuit and the remote L2 circuit can be configured statically, or Label Distribution Protocol (LDP) can be used to discover the cross-connection between the local and remote L2 circuits.

Figure 14-1 L2VPN Network Topology



There are two stages for configuring L2VPN circuits. First, L2 circuits must be enabled for L2VPN operation. Then, the L2 circuits must be cross-connected.

An L2VPN is enabled on a context, in context configuration mode. (Currently, an L2VPN can only be enabled on the local context.) An L2 circuit is linked to an L2VPN by mapping to the associated L2VPN-enabled context.

Configuration has to be symmetric. That is, both PE routers (local and remote) must be configured using the same inner label or virtual circuit identifier, and must also use the address of the remote PE as the peer address.

Supported Encapsulation Types

The SmartEdge router L2VPN implementation supports the following encapsulation types:

- Frame Relay Martini Encapsulation
- Ethernet VLAN
- Ethernet
- ATM AAL5

Frame Relay Martini Encapsulation

Frame Relay Martini encapsulation is supported according to the Internet Draft, *draft-martini-l2circuit-trans-mpls-10.txt*. The Frame Relay virtual circuit (VC) type is always set to 0x0001. LDP sets the C-bit when establishing the VCs. When sending VC traffic to the core, a control word is attached to the packets, and Frame Relay data-link connection identifier (DLCI) information is stripped from the packets. The egress PE router strips the control word from the packets, and rebuilds the Frame Relay DLCI header before sending the traffic to the CE router.

The following considerations apply when configuring Frame Relay encapsulation:

- The VC type should be the same on both PE routers.
- The VC ID should be the same for both PE routers.
- The two CE routers can have different DLCIs, because the Frame Relay DLCI information is stripped at ingress and rebuilt at egress.

Ethernet VLAN

Ethernet VLAN is supported in the raw mode with the Ethernet VLAN facility. With raw mode, no control word is sent with the traffic, and no C-bit is set. In raw mode, the whole VLAN header is sent to the remote PE router. On the egress side, the VLAN ID/tag is stripped and rebuilt according to the local VLAN tag.

The following considerations apply when configuring VLAN VCs:

- The VC type should be the same on both sides.
- The VC ID should be same for both sides for a VC.
- The two CE routers can have the same or different VLAN tags/permanent virtual circuits (PVCs) for the VC.

Note The SmartEdge OS supports Ethernet VLAN tag stacking to support Extreme switches' virtual metropolitan area network (VMAN) type of configuration. This configuration requires support for VLAN/VMAN tag 9100 in addition to the standard VLAN tag 8100. This support does not require any special L2VPN configuration on the SmartEdge OS side. A sample configuration for this L2VPN environment is provided at the end of this section.

Ethernet

Ethernet implementation is the same as the Ethernet VLAN. Only raw mode is supported for Ethernet encapsulation.

ATM AAL5

With our ATM implementation, the entire incoming protocol data unit (PDU) is transported to, and then rebuilt on, the other side.

The following considerations apply when configuring ATM VCs:

- The VC type should be the same on both sides of the VC.
- The VC ID should be the same on both sides.
- The ATM PVCs should be the same on both sides.

Supported Encapsulation Interconnectivity

The SmartEdge router L2VPN implementation supports the following encapsulation types for interconnectivity between two end-to-end cross-connections:

- ATM RFC 1483 bridged to dot1q
- ATM RFC 1483 bridged to Ethernet

Note This feature is supported only if both end PE routers are SmartEdge 800 routers.

QoS Policies for L2VPN Circuits

Quality of service (QoS) policies that are valid for L2VPN type of circuits can be applied to L2VPN VCs. The following QoS policies can be applied to L2VPN circuits:

- Rate limiting policing policies on ingress L2VPN circuits.
- Metering type of shaping policies on egress L2VPN circuits.

Note The other QoS policies are denied on L2VPN port-level configuration.

In addition to supporting rate limiting policing policies, and metering type of shaping policies, L2VPN implementation also supports the following:

- L2VPN cross-connections with a Multiprotocol Label Switching (MPLS) experimental (EXP) bit configuration to forward traffic on certain backbone queues.
- dot1q profile configurations on L2VPN circuits to propagate dot1p bits to MPLS EXP bits, and MPLS EXP bits to dot1q bits.

For information about QoS policies, see the “QoS Rate- and Class-Limiting Configuration” and “QoS Circuit Configuration” chapters in the *IP Services and Security Configuration Guide* for the SmartEdge OS.

L2VPN over GRE

Encapsulating packets via Generic Routing Encapsulation (GRE) from an ingress PE router to an egress PE router is called soft GRE tunneling. Soft GRE tunnels are not Interior Gateway Protocol (IGP) visible links, and routing adjacencies are not supported across these tunnels. As a result, soft GRE tunnels have little in common with traditional (hard) GRE tunnels. The tunnel exists only in the sense of GRE encapsulation and decapsulation.

Only the ingress PE router and the egress PE router need to support the soft GRE functionality, and the PE routers can span over multiple autonomous systems.

Using soft GRE tunnels to transport L2VPN-encapsulated packets is called L2VPN over GRE, and can be used instead of an MPLS tunnel in the backbone. L2VPN over GRE does not require preconfiguration of the remote GRE endpoint. The GRE tunnel endpoint is the remote PE’s address to which the L2VPN packets are being transported.

For more information about soft GRE, see Chapter 9, “BGP/MPLS VPN Configuration.”

Configuration Tasks

Note In this section, the command syntax in the task tables displays only the root command; for the complete command syntax, see the full description for the command in the “Command Descriptions” section.

To configure an L2VPN, perform the tasks described in the following sections:

- Enabling an L2 Circuit for L2VPN Operation

- Configuring an LDP L2VPN Cross-Connection
- Configuring a Static L2VPN Cross-Connection
- Enabling Soft GRE Tunneling

Enabling an L2 Circuit for L2VPN Operation

To enabling an L2 circuit for L2VPN operation, perform the tasks described in Table 14-1.

Table 14-1 Enable an L2 Circuit for L2VPN Operation

Task	Root Command	Notes
Enable an ATM PVC for L2VPN operation.	l2vpn ctx-name	Enter this command in ATM PVC configuration mode.
Enable an 802.1Q PVC for L2VPN operation.	l2vpn ctx-name	Enter this command in dot1q PVC configuration mode.
Enable a Frame Relay PVC for L2VPN operation.	l2vpn ctx-name	Enter this command in Frame Relay PVC configuration mode.
Enable an Ethernet port for L2VPN operation.	l2vpn ctx-name	Enter this command in port configuration mode.

Configuring an LDP L2VPN Cross-Connection

To configure an LDP L2VPN cross-connection, perform the tasks described in Table 14-2.

Table 14-2 Configure an LDP L2VPN Cross-Connection

Task	Root Command	Notes
Enter L2VPN configuration mode.	l2vpn	Enter this command in context configuration mode. You cannot enter L2VPN configuration mode in a non-local context. L2VPN configuration is allowed only in the local context.
Access L2VPN LDP configuration mode.	l2vpn-cct-bindings ldp	Enter this command in L2VPN configuration mode.
Create an LDP L2VPN cross-connection.	xc vc-id	Enter this command in L2VPN LDP configuration mode. When creating a cross-connection to a remote circuit that uses an encapsulation type that is different than the encapsulation type of the local circuit, use the remote-encap keyword to specify the encapsulation type used at the remote end of the cross-connection. The SmartEdge router supports the following encapsulation interconnectivity: • ATM RFC 1483 bridged to dot1q • ATM RFC 1483 bridged to Ethernet For ATM OC cards, you must specify a default channel number of 1 in the xc vc-id command; for example, if the card is an ATM-OC3c/STM-1c, then you must specify a default channel number of 1. ATM PVC cross-connections support PDU mode, and not cell mode.

Configuring a Static L2VPN Cross-Connection

To configure a static L2VPN cross-connection, perform the tasks described in Table 14-3.

Table 14-3 Configure a Static L2VPN Cross-Connection

Task	Root Command	Notes
Enter L2VPN configuration mode.	l2vpn	Enter this command in context configuration mode. You cannot enter L2VPN configuration mode in a non-local context. L2VPN configuration is allowed only in the local context.
Access L2VPN static configuration mode.	l2vpn-cct-bindings static	Enter this command in L2VPN configuration mode.
Create a static L2VPN cross-connection.	xc vpn-label	Enter this command in L2VPN static configuration mode. For ATM OC cards, you must specify default channel number of 1 in the xc vpn-label command; for example, if the card is an ATM-OC3c/STM-1c, then you must specify a default channel number of 1.

Enabling Soft GRE Tunneling

To enable soft GRE tunneling, perform the tasks described in Table 14-4.

Table 14-4 Enable Soft GRE Tunneling

Task	Root Command	Notes
Enable soft GRE tunneling on the specified context.	ip soft-gre	Enter this command in context configuration mode. Using soft GRE tunnels to transport L2VPN-encapsulated packets is called L2VPN over GRE, and can be used instead of an MPLS tunnel in the backbone. L2VPN over GRE does not require preconfiguration of the remote GRE endpoint. The GRE tunnel endpoint is the remote PE's address to which the L2VPN packets are being transported.

Configuration Examples

This section provides L2VPN configuration examples in the following sections:

- Static L2VPN
- LDP L2VPN
- CE Router with RFC 1483 Bridged Encapsulation for ATM AAL5
- L2VPN for Extreme Networks Equipment Interoperability
- QoS Rate Limiting Policy on Ingress L2VPN Circuits
- QoS Metering Policies on Egress L2VPN Circuits
- EXP-Bit for L2VPN VCs
- dot1q Bit Propagation on L2VPN Cross-Connections
- ATM RFC 1483 Bridged to dot1q Interconnection
- ATM RFC 1483 Bridged to Ethernet Interconnection
- L2VPN over GRE

Static L2VPN

The following example configures a typical static L2VPN on a local PE router and a remote PE router. For this example, the L2VPN cross-connects 802.1Q PVCs.

The static L2VPN configuration for the local PE_Router is as follows:

```
[local]PE_Router(config)#context local
[local]PE_Router(config-ctx)#interface foo
[local]PE_Router(config-if)#ip address 100.1.1.1/32
[local]PE_Router(config-if)#exit
[local]PE_Router(config-ctx)#l2vpn
[local]PE_Router(config-l2vpn)#l2vpn static
[local]PE_Router(config-l2vpn-static)#xc 1/1 vlan-id 300 vpn-label 5000 peer 200.2.2.2
[local]PE_Router(config-l2vpn-static)#exit
[local]PE_Router(config-l2vpn)#exit
[local]PE_Router(config)#port ethernet 1/1
[local]PE_Router(config-port)#no shutdown
[local]PE_Router(config-port)#encapsulation dot1q
[local]PE_Router(config-port)#dot1q pvc 300
[local]PE_Router(config-dot1q-pvc)#l2vpn
```

The static L2VPN configuration for the remote PE_Router is as follows:

```
[local]PE_Router(config)#context local
[local]PE_Router(config-ctx)#interface foo
[local]PE_Router(config-if)#ip address 200.2.2.2/32
[local]PE_Router(config-if)#exit
[local]PE_Router(config-ctx)#l2vpn
[local]PE_Router(config-l2vpn)#l2vpn static
[local]PE_Router(config-l2vpn-static)#xc 4/1 vlan-id 300 vpn-label 5000 peer 100.1.1.1
[local]PE_Router(config-l2vpn-static)#exit
[local]PE_Router(config-l2vpn)#exit
[local]PE_Router(config)#port ethernet 4/1
[local]PE_Router(config-port)#no shutdown
[local]PE_Router(config-port)#encapsulation dot1q
[local]PE_Router(config-port)#dot1q pvc 300
[local]PE_Router(config-dot1q-pvc)#l2vpn
```

LDP L2VPN

The LDP L2VPN configuration examples assume that the following conditions are true:

- MPLS core backbone configuration is up and running.

For more information on configuring MPLS, see Chapter 13, “MPLS Configuration.”

- LDP targeted discovery has been enabled between PE peers.

For more information on configuring LDP targeted discovery, see the “Targeted LDP” section in Chapter 15, “LDP Configuration.”

The following LDP L2VPN examples configure LDP L2VPN on a local PE router and a remote PE router using the following encapsulation types:

- LDP L2VPN with Frame Relay Martini Encapsulation
- LDP L2VPN with Ethernet VLAN Encapsulation
- LDP L2VPN with Ethernet Encapsulation
- LDP L2VPN with ATM DS-3 Encapsulation
- LDP L2VPN with ATM OC Encapsulation

Note L2VPNs can also be configured using different encapsulation types at each end. The SmartEdge router supports the following encapsulation interconnectivity:

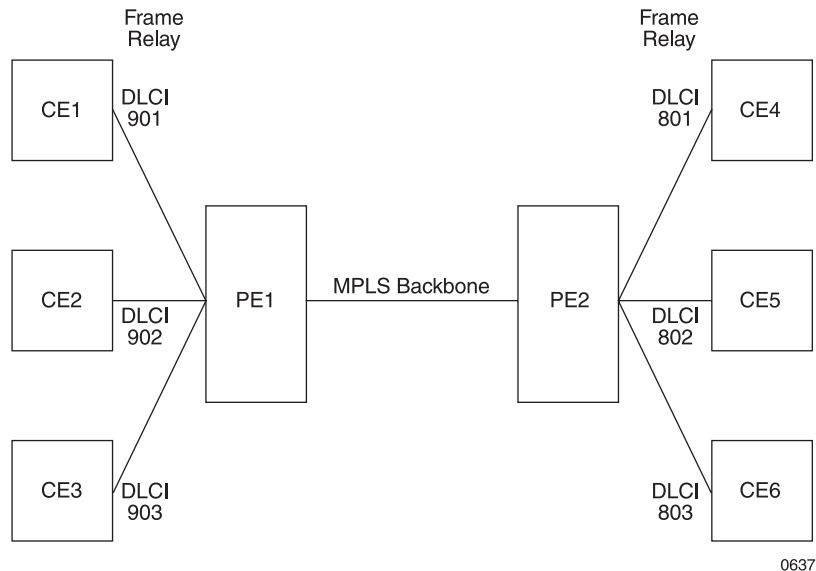
- ATM RFC 1483 bridged to dot1q
- ATM RFC 1483 bridged to Ethernet

LDP L2VPN with Frame Relay Martini Encapsulation

The following example demonstrates how two PE routers (PE1 and PE2) are configured to correctly operate LDP L2VPN using Frame Relay Martini encapsulation.

Figure 14-2 displays the network topology for this example.

Figure 14-2 LDP L2VPN with Frame Relay Martini Encapsulation Network Topology



Note Though the Frame Relay PVCs are different on the two sides, the VC IDs are still the same.

The configuration for the PE1 router is as follows:

```
[local]PE1(config)#context local
[local]PE1(config-ctx)#no ip domain-lookup
[local]PE1(config-ctx)#interface loop1 loopback
```

```

[local]PE1(config-if)#ip address 11.200.1.2/32
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#l2vpn
[local]PE1(config-l2vpn)#l2vpn ldp
[local]PE1(config-l2vpn-ldp)#xc 12/4 dlci 901 vc-id 901 peer 11.200.1.1
[local]PE1(config-l2vpn-ldp)#xc 12/4 dlci 902 vc-id 902 peer 11.200.1.1
[local]PE1(config-l2vpn-ldp)#xc 12/4 dlci 903 vc-id 903 peer 11.200.1.1
[local]PE1(config-l2vpn-ldp)#exit
[local]PE1(config-l2vpn)#exit
[local]PE1(config-ctx)#exit
[local]PE1(config)#port pos 12/4
[local]PE1(config-port)#no shutdown
[local]PE1(config-port)#encapsulation frame-relay
[local]PE1(config-port)#frame-relay pvc 901
[local]PE1(config-port)#l2vpn local
[local]PE1(config-port)#frame-relay pvc 902
[local]PE1(config-port)#l2vpn local
[local]PE1(config-port)#frame-relay pvc 903
[local]PE1(config-port)#l2vpn local
[local]PE1(config-port)#end

```

The configuration for the PE2 router is as follows:

```

[local]PE2(config)#context local
[local]PE2(config-ctx)#no ip domain-lookup
[local]PE2(config-ctx)#interface loop1 loopback
[local]PE2(config-if)#ip address 11.200.1.1/32
[local]PE2(config-if)#exit
[local]PE2(config-ctx)#router ldp
[local]PE2(config-ldp)#neighbor 11.200.1.2 targeted
[local]PE2(config-ldp)#exit
[local]PE2(config-ctx)#l2vpn
[local]PE2(config-l2vpn)#l2vpn ldp
[local]PE2(config-l2vpn-ldp)#xc 12/3 dlci 801 vc-id 901 peer 11.200.1.2
[local]PE2(config-l2vpn-ldp)#xc 12/3 dlci 802 vc-id 902 peer 11.200.1.2
[local]PE2(config-l2vpn-ldp)#xc 12/3 dlci 803 vc-id 903 peer 11.200.1.2
[local]PE2(config-l2vpn-ldp)#exit
[local]PE2(config-l2vpn)#exit
[local]PE2(config-ctx)#exit
[local]PE2(config)#port pos 12/3
[local]PE2(config-port)#no shutdown
[local]PE2(config-port)#encapsulation frame-relay
[local]PE2(config-port)#frame-relay pvc 801
[local]PE2(config-port)#l2vpn local
[local]PE2(config-port)#frame-relay pvc 802
[local]PE2(config-port)#l2vpn local
[local]PE2(config-port)#frame-relay pvc 803
[local]PE2(config-port)#l2vpn local
[local]PE2(config-port)#end

```

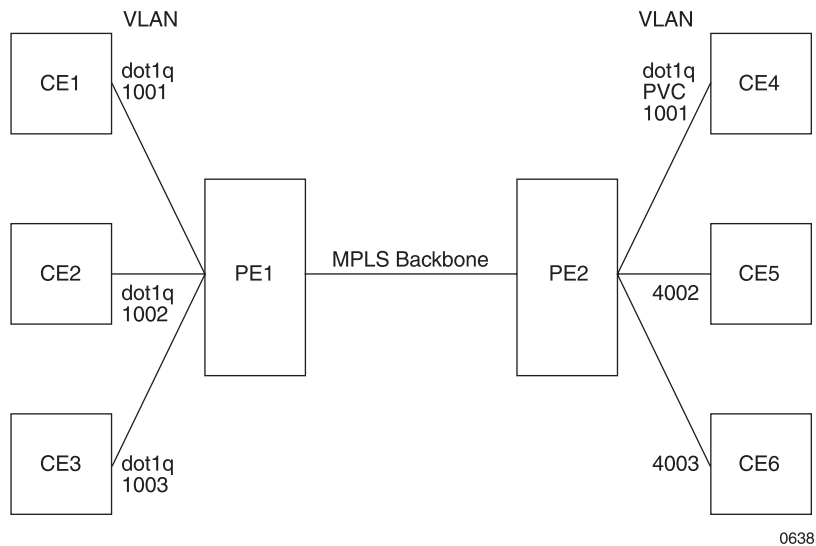
LDP L2VPN with Ethernet VLAN Encapsulation

The following example demonstrates how two PE routers (PE1 and PE2) are configured to correctly operate LDP L2VPN using Ethernet VLAN encapsulation.

Note The two CE ends are using either the same or different dot1q PVCs in this example.

Figure 14-3 displays the network topology for this example.

Figure 14-3 LDP L2VPN with Ethernet VLAN Encapsulation Network Topology



The configuration for the PE1 router is as follows:

```

[local]PE1(config)#context local
[local]PE1(config-ctx)#interface loop1 loopback
[local]PE1(config-if)#ip address 11.200.1.2/32
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#router ldp
[local]PE1(config-ldp)#neighbor 11.200.1.1 targeted
[local]PE1(config-ldp)#exit
[local]PE1(config-ctx)#l2vpn
[local]PE1(config-l2vpn)#l2vpn ldp
[local]PE1(config-l2vpn-ldp)#xc 10/2 vlan-id 1001 vc-id 1001 peer 11.200.1.1
[local]PE1(config-l2vpn-ldp)#xc 10/2 vlan-id 1002 vc-id 1002 peer 11.200.1.1
[local]PE1(config-l2vpn-ldp)#xc 10/2 vlan-id 1003 vc-id 1003 peer 11.200.1.1
[local]PE1(config-l2vpn-ldp)#exit
[local]PE1(config-l2vpn)#exit
[local]PE1(config-ctx)#exit
[local]PE1(config)#card gigaether-4-port 10
[local]PE1(config)#port ethernet 10/2
[local]PE1(config-port)#no shutdown
[local]PE1(config-port)#encapsulation dot1q
[local]PE1(config-port)#dot1q pvc 1001
[local]PE1(config-port)#l2vpn local
[local]PE1(config-port)#dot1q pvc 1002
  
```

```
[local]PE1(config-port)#l2vpn local
[local]PE1(config-port)#dot1q pvc 1003
[local]PE1(config-port)#l2vpn local
[local]PE1(config-port)#end
```

The configuration for the PE2 router is as follows:

```
[local]PE2(config)#context local
[local]PE2(config-ctx)#no ip domain-lookup
[local]PE2(config-ctx)#interface loop1 loopback
[local]PE2(config-if)#ip address 11.200.1.1/32
[local]PE2(config-if)#exit
[local]PE2(config-ctx)#router ldp
[local]PE2(config-ldp)#neighbor 11.200.1.2 targeted
[local]PE2(config-ldp)#exit
[local]PE2(config-ctx)#l2vpn
[local]PE2(config-l2vpn)#l2vpn ldp
[local]PE2(config-l2vpn-ldp)#xc 10/3 vlan-id 1001 vc-id 1001 peer 11.200.1.2
[local]PE2(config-l2vpn-ldp)#xc 10/3 vlan-id 4002 vc-id 1002 peer 11.200.1.2
[local]PE2(config-l2vpn-ldp)#xc 10/3 vlan-id 4003 vc-id 1003 peer 11.200.1.2
[local]PE2(config-l2vpn-ldp)#exit
[local]PE2(config-l2vpn)#exit
[local]PE2(config-ctx)#exit
[local]PE2(config)#port ethernet 10/3
[local]PE2(config-port)#no shutdown
[local]PE2(config-port)#encapsulation dot1q
[local]PE2(config-port)#dot1q pvc 1001
[local]PE2(config-port)#l2vpn local
[local]PE2(config-port)#dot1q pvc 4002
[local]PE2(config-port)#l2vpn local
[local]PE2(config-port)#dot1q pvc 4003
[local]PE2(config-port)#l2vpn local
[local]PE2(config-port)#end
```

LDP L2VPN with Ethernet Encapsulation

The following example demonstrates how two PE routers (PE1 and PE2) are configured to correctly operate LDP L2VPN using Ethernet encapsulation.

The configuration for the PE2 router is as follows:

```
[local]PE2(config)#context local
[local]PE2(config-ctx)#no ip domain-lookup
[local]PE2(config-ctx)#interface loop1 loopback
[local]PE2(config-if)#ip address 11.200.1.2/32
[local]PE2(config-if)#exit
[local]PE2(config-ctx)#exit
[local]PE2(config)#l2vpn
[local]PE2(config)#l2vpn ldp
[local]PE2(config-l2vpn-ldp)#xc 10/2 vc-id 1001 peer 11.200.1.1
[local]PE2(config-l2vpn-ldp)#xc 10/4 vc-id 1002 peer 11.200.1.1
[local]PE2(config-l2vpn-ldp)#exit
[local]PE2(config-l2vpn)#exit
```

```
[local]PE2(config-ctx)#exit
[local]PE2(config)#card gigaether-4-port 10
[local]PE2(config)#port ethernet 10/2
[local]PE2(config-port)#no shutdown
[local]PE2(config-port)#l2vpn local
[local]PE2(config-port)#exit
[local]PE2(config)#port ethernet 10/4
[local]PE2(config-port)#no shutdown
[local]PE2(config-port)#l2vpn local
[local]PE2(config-port)#end
```

LDP L2VPN with ATM DS-3 Encapsulation

The following example demonstrates how two PE routers (PE1 and PE2) are configured to correctly operate LDP L2VPN using ATM DS-3 encapsulation.

The configuration for the PE1 router is as follows:

```
[local]PE1(config-ctx)#l2vpn
[local]PE1(config-l2vpn)#l2vpn ldp
[local]PE1(config-l2vpn-ldp)#xc 4/1 vpi-vci 104 104 vc-id 104 peer 11.200.1.2
[local]PE1(config-l2vpn-ldp)#xc 4/1 vpi-vci 105 105 vc-id 105 peer 11.200.1.2
[local]PE1(config-l2vpn-ldp)#xc 4/2 vpi-vci 106 106 vc-id 106 peer 11.200.1.2
[local]PE1(config-l2vpn-ldp)#xc 4/2 vpi-vci 107 107 vc-id 107 peer 11.200.1.2
[local]PE1(config-l2vpn-ldp)#exit
[local]PE1(config-l2vpn)#exit
[local]PE1(config-ctx)#exit
[local]PE1(config)#atm profile l2vpn-atm-ds3
[local]PE1(config-atmpro)#counters l2
[local]PE1(config-atmpro)#shaping ubr
[local]PE1(config-atmpro)#exit
[local]PE1(config)#card atm-ds3-12-port 4
[local]PE1(config)#port atm 4/1
[local]PE1(config-atm)#no shutdown
[local]PE1(config-atm)#atm pvc 104 104 profile l2vpn-atm-ds3 encap bridge1483
[local]PE1(config-atmpvc)#l2vpn local
[local]PE1(config-atm)#atm pvc 105 105 profile l2vpn-atm-ds3 encap bridge1483
[local]PE1(config-atmpvc)#l2vpn local
[local]PE1(config-atmpvc)#exit
[local]PE1(config)#port atm 4/2
[local]PE1(config-atm)#no shutdown
[local]PE1(config-atm)#atm pvc 106 106 profile l2vpn-atm-ds3 encap bridge1483
[local]PE1(config-atmpvc)#l2vpn local
[local]PE1(config-atm)#atm pvc 107 107 profile l2vpn-atm-ds3 encap bridge1483
[local]PE1(config-atmpvc)#l2vpn local
[local]PE1(config-atmpvc)#end
```

The configuration for the PE2 router is as follows:

```
[local]PE2(config-ctx)#l2vpn
[local]PE2(config-l2vpn)#l2vpn ldp
[local]PE2(config-l2vpn-ldp)#xc 4/1 vpi-vci 104 104 vc-id 104 peer 11.200.1.1
[local]PE2(config-l2vpn-ldp)#xc 4/1 vpi-vci 105 105 vc-id 105 peer 11.200.1.1
```

```
[local]PE2(config-l2vpn-ldp)#xc 4/2 vpi-vci 106 106 vc-id 106 peer 11.200.1.1
[local]PE2(config-l2vpn-ldp)#xc 4/2 vpi-vci 107 107 vc-id 107 peer 11.200.1.1
[local]PE2(config-l2vpn-ldp)#exit
[local]PE2(config-l2vpn)#exit
[local]PE2(config-ctx)#exit
[local]PE2(config)#atm profile l2vpn-atm-ds3
[local]PE2(config-atmpro)#counters l2
[local]PE2(config-atmpro)#shaping ubr
[local]PE2(config-atmpro)#exit
[local]PE2(config)#port atm 4/1
[local]PE2(config-atm)#no shutdown
[local]PE2(config-atm)#atm pvc 104 104 profile l2vpn-atm-ds3 encap bridge1483
[local]PE2(config-atmpvc)#l2vpn local
[local]PE2(config-atm)#atm pvc 105 105 profile l2vpn-atm-ds3 encap bridge1483
[local]PE2(config-atmpvc)#l2vpn local
[local]PE2(config-atmpvc)#exit
[local]PE2(config)#port atm 4/2
[local]PE2(config-atm)#no shutdown
[local]PE2(config-atm)#atm pvc 106 106 profile l2vpn-atm-ds3 encap bridge1483
[local]PE2(config-atmpvc)#l2vpn local
[local]PE2(config-atm)#atm pvc 107 107 profile l2vpn-atm-ds3 encap bridge1483
[local]PE2(config-atmpvc)#l2vpn local
[local]PE2(config-atmpvc)#end
```

LDP L2VPN with ATM OC Encapsulation

The following example demonstrates how two PE routers (PE1 and PE2) are configured to correctly operate LDP L2VPN using ATM OC encapsulation.

The configuration for the PE1 router is as follows:

```
[local]PE1(config)#context local
[local]PE1(config-ctx)#no ip domain-lookup
[local]PE1(config-ctx)#l2vpn
[local]PE1(config-l2vpn)#l2vpn ldp
[local]PE1(config-l2vpn-ldp)#xc 5/1:1 vpi-vci 101 101 vc-id 101 peer 11.200.1.2
[local]PE1(config-l2vpn-ldp)#exit
[local]PE1(config-l2vpn)#exit
[local]PE1(config-ctx)#exit
[local]PE1(config)#atm profile l2vpn-atm
[local]PE1(config-atmpro)#counters l2
[local]PE1(config-atmpro)#shaping ubr
[local]PE1(config-atmpro)#exit
[local]PE1(config)#port atm 5/1
[local]PE1(config-atm)#no shutdown
[local]PE1(config-atm)#atm pvc 101 101 profile l2vpn-atm encap bridge1483
[local]PE1(config-atmpvc)#l2vpn local
[local]PE1(config-atmpvc)#end
```

The configuration for the PE2 router is as follows:

```
[local]PE2(config)#context local
[local]PE2(config-ctx)#l2vpn
[local]PE2(config-l2vpn)#l2vpn ldp
[local]PE2(config-l2vpn-ldp)#xc 5/1:1 vpi-vci 101 101 vc-id 101 peer 11.200.1.1
[local]PE2(config-l2vpn-ldp)#exit
[local]PE2(config-l2vpn)#exit
[local]PE2(config-ctx)#exit
[local]PE2(config)#atm profile l2vpn-atm
[local]PE2(config-atmpro)#counters l2
[local]PE2(config-atmpro)#shaping ubr
[local]PE2(config)#port atm 5/1
[local]PE2(config-atm)#no shutdown
[local]PE2(config-atm)#atm pvc 101 101 profile l2vpn-atm encap bridge1483
[local]PE2(config-atmpvc)#l2vpn local
[local]PE2(config-atmpvc)#end
```

CE Router with RFC 1483 Bridged Encapsulation for ATM AAL5

The following example configures a CE router with RFC 1483 bridged encapsulation for ATM AAL5:

```
[local]CE(config)#context CE1-atm-ds3-104
[local]CE(config-ctx)#no ip domain-lookup
[local]CE(config-ctx)#interface ce1-atm-ds3-104
[local]CE(config-if)#ip address 104.1.1.1/24
[local]CE(config-if)#exit
[local]CE(config-ctx)#exit
[local]CE(config)#atm profile l2vpn-atm-ds3
[local]CE(config-atmpro)#counters l2
[local]CE(config-atmpro)#shaping ubr
[local]CE(config-atmpro)#exit
[local]CE(config)#port atm 4/7
[local]CE(config-atm)#no shutdown
[local]CE(config-atm)#atm pvc 104 104 profile l2vpn-atm-ds3 encap bridge1483
[local]CE(config-atmpvc)#bind interface ce1-atm-ds3-104 CE1-atm-ds3-104
[local]CE(config-atmpvc)#end
```

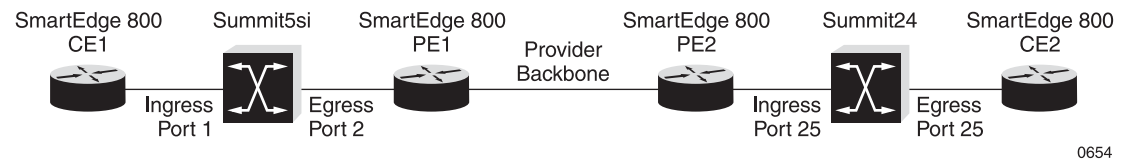
L2VPN for Extreme Networks Equipment Interoperability

This setup is used for testing interoperability with an Extreme Networks' switch VMAN-type packets. The SmartEdge 800 L2VPN does not require a specific configuration for this example. Extreme switches use 9100 as the Ethertype for these configurations.

For this example, SmartEdge 800 routers are used as PE routers. The CE1 router is connected to the PE1 router through an Extreme Summit5si switch. The ingress port for the tunnel is 1 and egress port on the VMAN tunnel on the Extreme Summit5i is 2. The PE2 router is connected to the CE2 router through an Extreme Summit24 switch. The PE2 router's port is Gigabit Ethernet, but the CE2 router's port is Ethernet; they are connected together over a VLAN/VMAN configuration.

Figure 14-4 displays the network topology for this configuration example.

Figure 14-4 Network Topology for Extreme Networks Equipment Interoperability



This setup uses the same VLAN ID on both ends, but should also work properly with different VLAN IDs.

Note This example does not show the MPLS Layer 3 backbone configuration. See Chapter 13, “MPLS Configuration,” for MPLS backbone configuration examples.

The L2VPN configuration for the Extreme Summit5si switch is as follows:

```
configure dot1q ethertype 9100
create vlan "l2vpn-CE1"
# Config information for VLAN l2vpn-CE1.
config vlan "l2vpn-CE1" tag 1000      # VLAN-ID=0x3e8 Global Tag 72
config vlan "l2vpn-CE1" protocol "ANY"
config vlan "l2vpn-CE1" qosprofile "QP1"
# No IP address is configured for VLAN l2vpn-CE1.
configure vlan "l2vpn-CE1" add port 1 untagged
config vlan "l2vpn-CE1" add port 2 tagged
configure jumbo-frame size 1530
disable red port 1
disable dlcs port 1
configure port 1 auto on
enable jumbo-frame ports 1
enable edp port 1
disable red port 2
disable dlcs port 2
configure port 2 auto on
enable jumbo-frame ports 2
```

The L2VPN configuration for the CE1 router is as follows:

```
[local]CE1#config
[local]CE1(config)#context CE1-extreme-1000
[local]CE1(config-ctx)#no ip domain-lookup
[local]CE1(config-ctx)#interface ce1-extreme-1000
[local]CE1(config-if)#ip address 1.1.1.1/24
[local]CE1(config-if)#exit
[local]CE1(config-ctx)#exit
[local]CE1(config)#port ethernet 10/2
[local]CE1(config-port)#no shutdown
[local]CE1(config-port)#encapsulation dot1q
[local]CE1(config-port)#dot1q pvc 1000
[local]CE1(config-port)#bind interface ce1-extreme-1000 CE1-extreme-1000
[local]CE1(config-port)#end
```

The L2VPN configuration for the PE1 router is as follows:

```
[local]PE1#config
[local]PE1(config)#context local
[local]PE1(config-ctx)#interface loop1 loopback
[local]PE1(config-if)#ip address 11.200.1.2/32
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#router ldp
[local]PE1(config-ldp)#neighbor 11.200.1.1 targeted
[local]PE1(config-ldp)#exit
[local]PE1(config-ctx)#l2vpn
[local]PE1(config-l2vpn)#l2vpn-cct-bindings ldp
[local]PE1(config-l2vpn-ldp)#xc 10/1 vlan-id 1000 vc-id 1000 peer 11.200.1.1
[local]PE1(config-l2vpn-ldp)#exit
[local]PE1(config-l2vpn)#exit
[local]PE1(config-ctx)#exit
[local]PE1(config)#card gigaether-4-port 10
[local]PE1(config)#port ethernet 10/1
[local]PE1(config-port)#description to-Extereme-port2
[local]PE1(config-port)#no shutdown
[local]PE1(config-port)#encapsulation dot1q
[local]PE1(config-port)#dot1q pvc 1000
[local]PE1(config-port)#l2vpn local
[local]PE1(config-port)#end
```

The L2VPN configuration for the PE2 router is as follows:

```
[local]PE2(config)#context local
[local]PE2(config-ctx)#interface loop1 loopback
[local]PE2(config-if)#ip address 11.200.1.1/32
[local]PE2(config-if)#exit
[local]PE2(config-ctx)#router ldp
[local]PE2(config-ldp)#neighbor 11.200.1.2 targeted
[local]PE2(config-ldp)#exit
[local]PE2(config-ctx)#l2vpn
[local]PE2(config-l2vpn)#l2vpn-cct-bindings ldp
[local]PE2(config-l2vpn-ldp)#xc 10/2 vlan-id 1000 vc-id 1000 peer 11.200.1.2
[local]PE2(config-l2vpn-ldp)#exit
[local]PE2(config-l2vpn)#exit
[local]PE2(config-ctx)#exit
[local]PE2(config)#port ethernet 10/2
[local]PE2(config-port)#no shutdown
[local]PE2(config)#encapsulation dot1q
[local]PE2(config-port)#dot1q pvc 1000
[local]PE2(config-port)#l2vpn local
[local]PE2(config-port)#end
```

The L2VPN configuration for the CE2 router is as follows:

```
[local]CE2(config)#context CE2-FE-dot1q-1000
[local]CE2(config-ctx)#no ip domain-lookup
[local]CE2(config-ctx)#interface ce2-fe-dot1q-1000
[local]CE2(config-if)#ip address 1.1.1.2/24
```

```

[local]CE2(config-if)#exit
[local]CE2(config-ctx)#exit
[local]CE2(config)#card ether-12-port 14
[local]CE2(config)#port ethernet 14/1
[local]CE2(config-port)#no shutdown
[local]CE2(config-port)#encapsulation dot1q
[local]CE2(config-port)#dot1q pvc 1000
[local]CE2(config-port)#bind interface ce2-fe-dot1q-1000 CE2-FE-dot1q-1000
[local]CE2(config-port)#end

```

The L2VPN configuration on the Extreme Summit 24 switch is as follows:

```

configure dot1q ethertype 9100
enable jumbo
# Config information for VLAN l2vpn-CE2.
config vlan "l2vpn-CE2" tag 1000      # VLAN-ID=0x3e8 Global Tag 256
config vlan "l2vpn-CE2" protocol "ANY"
config vlan "l2vpn-CE2" qosprofile "QP1"
# No IP address is configured for VLAN l2vpn-CE2.
configure vlan "l2vpn-CE2" add port 2 untagged
config vlan "l2vpn-CE2" add port 25 tagged

```

QoS Rate Limiting Policy on Ingress L2VPN Circuits

The following example configures the QoS rate limiting policy, l2vpn, for an ingress L2VPN circuit with Ethernet VLAN encapsulation. Incoming packets that exceed the 40000 kbps rate are dropped by default.

```

[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config-ctx)#l2vpn
[local]Redback(config-l2vpn)#l2vpn-cct-bindings ldp
[local]Redback(config-l2vpn-ldp)#xc 10/2 vlan-id 1001 vc-id 1001 peer 11.200.1.2
[local]Redback(config-l2vpn-ldp)#exit
[local]Redback(config-l2vpn)#exit
[local]Redback(config-ctx)#exit
[local]Redback(config)#qos policy l2vpn policing
[local]Redback(config-qos-pol-rl)#rate 40000 burst 20000
[local]Redback(config-qos-pol-rate)#exit
[local]Redback(config-qos-pol-rl)#exit
[local]Redback(config)#port ethernet 10/2
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#encapsulation dot1q
[local]Redback(config-port)#dot1q pvc 1001
[local]Redback(config-dot1q-pvc)#l2vpn local
[local]Redback(config-dot1q-pvc)#qos policy l2vpn in

```

QoS Metering Policies on Egress L2VPN Circuits

The following example configures the QoS metering policy, `l2vpn-shaping`, on the egress side of an L2VPN cross-connection. Outgoing packets that exceed the 10000 rate are dropped.

```
[local]Redback#config
[local]Redback(config)#port ethernet 9/2
[local]Redback(config-port)#dot1q pvc 1
[local]Redback(config-pvc)#qos policy metering l2vpn-shaping
[local]Redback(config-pvc)#exit
[local]Redback(config-port)#exit
[local]Redback(config)#qos policy l2vpn-shaping metering
[local]Redback(config-qos-pol-rl)#rate 10000 burst 2000
[local]Redback(config-qos-pol-rl)#end
```

EXP-Bit for L2VPN VCs

EXP bits can be set for L2VPN virtual circuits (VCs) to be applied to the outgoing backbone queues. The EXP bit is set for the Layer 2 label and is then copied to the appropriate Layer 3 label. This sets the corresponding outgoing backbone queue. For information on QoS queues, see the “QoS Circuit Configuration” chapter in the *IP Services and Security Configuration Guide* for the SmartEdge OS.

The following configuration example sets the EXP bits for L2VPN circuits.

Note This example is a relevant partial configuration; for a complete Layer 3 configuration, see Chapter 13, “MPLS Configuration.”

```
[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config-ctx)#no ip domain-lookup
[local]Redback(config-ctx)#interface loop1 loopback
[local]Redback(config-if)#ip address 11.200.1.1/32
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface to-P
[local]Redback(config-if)#ip address 101.1.1.4/24
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#router mpls
[local]Redback(config-mpls)#interface loop1
[local]Redback(config-mpls-if)#exit
[local]Redback(config-mpls)#interface to-P
[local]Redback(config-mpls-if)#exit
[local]Redback(config-mpls)#exit
[local]Redback(config-ctx)#router rsvp
[local]Redback(config-rsvp-explicit-route)#explicit-route to-MPLS2-via-P
[local]Redback(config-rsvp-explicit-route)#next-hop 101.1.1.5
[local]Redback(config-rsvp-explicit-route)#next-hop 4.1.1.5
[local]Redback(config-rsvp-explicit-route)#exit
[local]Redback(config-rsvp)#lsp S4_P_S2
[local]Redback(config-rsvp-lsp)#ingress 11.200.1.1
[local]Redback(config-rsvp-lsp)#egress 11.200.1.2
[local]Redback(config-rsvp-lsp)#source-path to-MPLS2-via-P
[local]Redback(config-rsvp-lsp)#exit
```

```

[local]Redback(config-rsvp)#interface loop1
[local]Redback(config-rsvp-if)#exit
[local]Redback(config-rsvp)#interface to-P
[local]Redback(config-rsvp-if)#exit
[local]Redback(config-rsvp)#exit
[local]Redback(config-ctx)#router ldp
[local]Redback(config-ldp)#neighbor 11.200.1.2 targeted
[local]Redback(config-ldp)#exit
[local]Redback(config-ctx)#l2vpn
[local]Redback(config-l2vpn)#l2vpn-cct-bindings ldp
[local]Redback(config-l2vpn-ldp)#xc 10/2 vlan-id 4001 vc-id 4001 peer 11.200.1.2 exp-bits 7
[local]Redback(config-l2vpn-ldp)#xc 10/2 vlan-id 4002 vc-id 4002 peer 11.200.1.2 exp-bits 6
[local]Redback(config-l2vpn-ldp)#xc 10/2 vlan-id 4003 vc-id 4003 peer 11.200.1.2 exp-bits 5
[local]Redback(config-l2vpn-ldp)#exit
[local]Redback(config-l2vpn)#exit
[local]Redback(config-ctx)#exit
[local]Redback(config)#qos queue-map default
[local]Redback(config-queue-map)#num-queues 2
[local]Redback(config-qos-queue-map-num-queues)#queue 0 priority 0
[local]Redback(config-qos-queue-map-num-queues)#queue 1 priority 1 2 3 4 5 6 7
[local]Redback(config-qos-queue-map-num-queues)#exit
[local]Redback(config-queue-map)#num-queues 4
[local]Redback(config-qos-queue-map-num-queues)#queue 0 priority 0
[local]Redback(config-qos-queue-map-num-queues)#queue 1 priority 1 2
[local]Redback(config-qos-queue-map-num-queues)#queue 2 priority 3 4 5 6
[local]Redback(config-qos-queue-map-num-queues)#queue 3 priority 7
[local]Redback(config-qos-queue-map-num-queues)#exit
[local]Redback(config-queue-map)#num-queues 8
[local]Redback(config-qos-queue-map-num-queues)#queue 0 priority 0
[local]Redback(config-qos-queue-map-num-queues)#queue 1 priority 1
[local]Redback(config-qos-queue-map-num-queues)#queue 2 priority 2
[local]Redback(config-qos-queue-map-num-queues)#queue 3 priority 3
[local]Redback(config-qos-queue-map-num-queues)#queue 4 priority 4
[local]Redback(config-qos-queue-map-num-queues)#queue 5 priority 5
[local]Redback(config-qos-queue-map-num-queues)#queue 6 priority 6
[local]Redback(config-qos-queue-map-num-queues)#queue 7 priority 7
[local]Redback(config-qos-queue-map-num-queues)#exit
[local]Redback(config-queue-map)#exit
[local]Redback(config)#qos policy pq2 pq
[local]Redback(config)#port ethernet 10/2
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#encapsulation dot1q
[local]Redback(config-port)#dot1q pvc 4001
[local]Redback(config-dot1q-pvc)#l2vpn local
[local]Redback(config-dot1q-pvc)#exit
[local]Redback(config-port)#dot1q pvc 4002
[local]Redback(config-dot1q-pvc)#l2vpn local
[local]Redback(config-dot1q-pvc)#exit
[local]Redback(config-port)#dot1q pvc 4003
[local]Redback(config-dot1q-pvc)#l2vpn local
[local]Redback(config-dot1q-pvc)#exit
[local]Redback(config-port)#exit

```

```
[local]Redback(config)#port ethernet 10/3
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#bind interface to-P local
[local]Redback(config-port)#qos policy queuing pq2
```

dot1q Bit Propagation on L2VPN Cross-Connections

L2VPN circuits support propagating dot1p bits to EXP bits on ingress routers, and EXP bits to dot1q bits on egress router. When a dot1q profile is applied to an ingress L2VPN circuit, the dot1q bits are propagated to QoS bits, and then MPLS propagates the QoS bits to the EXP bits, for both L2 and L3 labels. When the dot1p profile is applied to an egress L2VPN circuit, MPLS propagates the EXP bits to the QoS bits, and then the the QoS bits are propagated to the dot1q bits.

The following example propagates dot1p bits to EXP bits by applying the dot1q-qos dot1q profile to an ingress L2VPN circuit:

```
[local]Redback#config
[local]Redback(config)#dot1q profile dot1q-qos
[local]Redback(config-dot1q-profile)#propagate qos from ethernet
[local]Redback(config-dot1q-profile)#commit
[local]Redback(config-dot1q-profile)#exit
[local]Redback(config)#context local
[local]Redback(config-ctx)#router mpls
[local]Redback(config-mpls)#propagate qos to-mpls
[local]Redback(config-mpls)#commit
[local]Redback(config-mpls)#exit
[local]Redback(config)#port ethernet 9/2
[local]Redback(config-port)#dot1q pvc 1001 profile dpt1q-qos
[local]Redback(config-dot1q-pvc)#l2vpn local
[local]Redback(config-dot1q-pvc)#end
```

The following example propagates EXP bits to dot1p bits by applying the qos-dot1q dot1q profile to an egress L2VPN circuit:

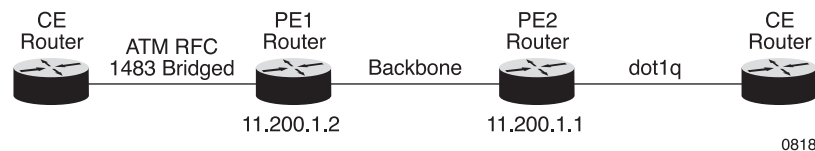
```
[local]Redback#config
[local]Redback(config)#dot1q profile qos-dot1q
[local]Redback(config-dot1q-profile)#propagate qos to ethernet
[local]Redback(config-dot1q-profile)#commit
[local]Redback(config-dot1q-profile)#exit
[local]Redback(config)#context local
[local]Redback(config-ctx)#router mpls
[local]Redback(config-mpls)#propagate qos from-mpls
[local]Redback(config-mpls)#commit
[local]Redback(config-mpls)#exit
[local]Redback(config)#port ethernet 9/2
[local]Redback(config-port)#dot1q pvc 1001 profile qos-dot1q
[local]Redback(config-dot1q-pvc)#l2vpn local
[local]Redback(config-dot1q-pvc)#end
```

ATM RFC 1483 Bridged to dot1q Interconnection

The SmartEdge OS supports L2VPN cross-connectivity when one end of the cross-connection is an ATM RFC 1483 bridged circuit, and the other end is a dot1q circuit. The following example configures an interconnection between ATM RFC 1483 bridged and dot1q on two sides of an L2VPN cross-connection.

Figure 14-5 displays the network topology for this configuration example.

Figure 14-5 ATM RFC 1483 Bridged to dot1q Network Topology



The L2VPN configuration for the PE1 router is as follows:

```

[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config-ctx)#l2vpn
[local]Redback(config-l2vpn)#l2vpn-cct-bindings ldp
[local]Redback(config-l2vpn-ldp)#xc 10/1:1 vpi-vci 104 104 vc-id 104 peer 11.200.1.1
remote-encap dot1q
[local]Redback(config-l2vpn-ldp)#xc 10/1:1 vpi-vci 105 105 vc-id 105 peer 11.200.1.1
remote-encap dot1q
[local]Redback(config-l2vpn-ldp)#xc 10/1:1 vpi-vci 106 106 vc-id 106 peer 11.200.1.1
remote-encap dot1q
[local]Redback(config-l2vpn-ldp)#exit
[local]Redback(config-l2vpn)#exit
[local]Redback(config-ctx)#exit
[local]Redback(config)#port atm 10/1
[local]Redback(config-atm)#no shutdown
[local]Redback(config-atm)#atm pvc 104 104 profile l2vpn-atm encap bridge1483
[local]Redback(config-atmpvc)#l2vpn local
[local]Redback(config-atmpvc)#exit
[local]Redback(config-atm)#atm pvc 105 105 profile l2vpn-atm encap bridge1483
[local]Redback(config-atmpvc)#l2vpn local
[local]Redback(config-atmpvc)#exit
[local]Redback(config-atm)#atm pvc 106 106 profile l2vpn-atm encap bridge1483
[local]Redback(config-atmpvc)#l2vpn local
[local]Redback(config-atmpvc)#end
  
```

The L2VPN configuration for the PE2 router is as follows:

```

[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config-ctx)#l2vpn
[local]Redback(config-l2vpn)#l2vpn-cct-bindings ldp
[local]Redback(config-l2vpn-ldp)#xc 5/1 vlan-id 1001 vc-id 104 peer 11.200.1.2
remote-encap bridge1483
[local]Redback(config-l2vpn-ldp)#xc 5/1 vlan-id 1002 vc-id 105 peer 11.200.1.2
remote-encap bridge1483
  
```

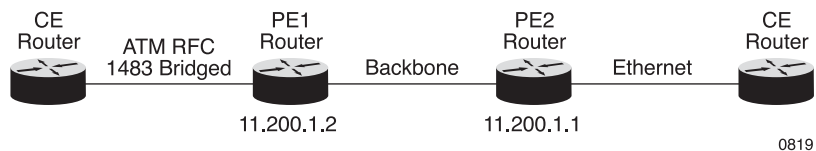
```
[local]Redback(config-l2vpn-ldp)#xc 5/1 vlan-id 1003 vc-id 106 peer 11.200.1.2
remote-encap bridge1483
[local]Redback(config-l2vpn-ldp)#exit
[local]Redback(config-l2vpn)#exit
[local]Redback(config-ctx)#exit
[local]Redback(config)#port eth 5/1
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#encapsulation dot1q
[local]Redback(config-port)#dot1q pvc 1001
[local]Redback(config-dot1q-pvc)#l2vpn local
[local]Redback(config-dot1q-pvc)#exit
[local]Redback(config-port)#dot1q pvc 1002
[local]Redback(config-dot1q-pvc)#l2vpn local
[local]Redback(config-dot1q-pvc)#exit
[local]Redback(config-port)#dot1q pvc 1003
[local]Redback(config-dot1q-pvc)#l2vpn local
[local]Redback(config-dot1q-pvc)#end
```

ATM RFC 1483 Bridged to Ethernet Interconnection

The SmartEdge OS supports L2VPN cross-connectivity when one end of the cross-connection is an ATM RFC 1483 bridged circuit, and the other end is an Ethernet circuit. The following example configures an interconnection between ATM RFC 1483 bridged and Ethernet on two sides of an L2VPN cross-connection.

Figure 14-6 Displays the network topology for this configuration example.

Figure 14-6 ATM RFC 1483 Bridged to Ethernet Network Topology



The L2VPN configuration for the PE1 router is as follows:

```
[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config-ctx)#l2vpn
[local]Redback(config-l2vpn)#l2vpn-cct-bindings ldp
[local]Redback(config-l2vpn-ldp)#xc 13/1:1 vpi-vci 104 104 vc-id 1001 peer 11.200.1.1
remote-encap ethernet
[local]Redback(config-l2vpn-ldp)#exit
[local]Redback(config-l2vpn)#exit
[local]Redback(config-ctx)#exit
[local]Redback(config)#port atm 13/1
[local]Redback(config-atm)#no shutdown
[local]Redback(config-atm)#atm pvc 104 104 profile l2vpn-atm encapsulation bridge1483
[local]Redback(config-atmpvc)#l2vpn local
[local]Redback(config-atmpvc)#end
```


The L2VPN configuration for the PE2 router is as follows:

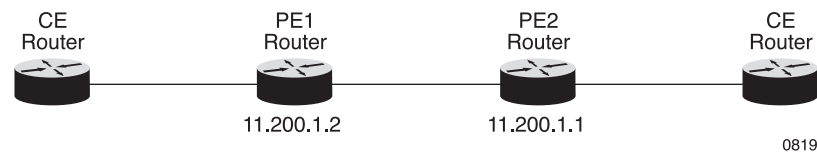
```
[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config-ctx)#l2vpn
[local]Redback(config-l2vpn)#l2vpn-cct-bindings ldp
[local]Redback(config-l2vpn-ldp)#xc 10/3 vc-id 1001 peer 11.200.1.2 remote-encap
bridge1483
[local]Redback(config-l2vpn-ldp)#exit
[local]Redback(config-l2vpn)#exit
[local]Redback(config-ctx)#exit
[local]Redback(config)#port ethernet 10/3
[local]Redback(config-port)#no shutdown
[local]Redback(config-port)#l2vpn local
[local]Redback(config-port)#end
```

L2VPN over GRE

The SmartEdge OS supports L2VPN over GRE, which is a method of transporting L2VPN-encapsulated packets using soft GRE tunnels. For L2VPN over GRE to work properly, the ingress and egress PE routers must both be configured to support soft GRE functionality. The following example enables soft GRE tunneling.

Figure 14-7 Displays the network topology for this configuration example.

Figure 14-7 L2VPN over GRE Network Topology



The L2VPN over GRE configuration for the PE1 router is as follows:

```
[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config-ctx)#ip soft-gre source 11.200.1.2
[local]Redback(config-ctx)#end
```

The L2VPN over GRE configuration for the PE2 router is as follows:

```
[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config-ctx)#ip soft-gre source 11.200.1.1
[local]Redback(config-ctx)#end
```

Command Descriptions

This section describes the syntax and usage guidelines for the commands used to configure L2VPN features. The commands are presented in alphabetical order.

ip soft-gre	l2vpn ctx-name
l2vpn	xc vc-id
l2vpn-cct-bindings ldp	xc vpn-label
l2vpn-cct-bindings static	

ip soft-gre

ip soft-gre [source *src-addr*]

no ip soft-gre [source *src-addr*]

Purpose

Enables soft Generic Routing Encapsulation (GRE) tunneling on the specified context.

Command Mode

context configuration

Syntax Description

source <i>src-addr</i>	Optional. Source address for the soft GRE tunnel. The IP address is in the form <i>A.B.C.D</i> .

Default

Soft GRE tunneling is disabled.

Usage Guidelines

Use the **ip soft-gre** command to enable soft GRE tunneling on the specified context.

Encapsulating packets with GRE from an ingress provider edge (PE) router to an egress PE router is called soft GRE tunneling. Soft GRE tunnels are not Interior Gateway Protocol (IGP)-visible links, and routing adjacencies are not supported across these tunnels. As a result, soft GRE tunnels have little in common with traditional (hard) GRE tunnels. The tunnel exists only in the sense of GRE encapsulation and decapsulation.

Only the ingress PE router and the egress PE router need to support the soft GRE functionality, and the PE routers can span over multiple autonomous systems.

Using soft GRE tunnels to transport Layer 2 Virtual Private Network (L2VPN)-encapsulated packets is called L2VPN over GRE, and can be used instead of a Multiprotocol Label Switching (MPLS) tunnel in the backbone. L2VPN over GRE does not require pre-configuration of the remote GRE endpoint. The GRE tunnel endpoint is the remote PE's address to which the L2VPN packets are being transported.

Note The **ip soft-gre** command is also documented in Chapter 9, "BGP/MPLS VPN Configuration," where it is used to enable BGP/MPLS VPN over GRE.

Use the **no** form of this command to disable soft GRE tunneling on the specified context.

Examples

The following example enables soft GRE tunneling in the `local` context:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#ip soft-gre
```

Related Commands

None

l2vpn

l2vpn
no l2vpn

Purpose

Enters L2VPN configuration mode.

Command Mode

context configuration

Syntax Description

This command has no keywords or arguments.

Default

None

Usage Guidelines

Use the **l2vpn** command to enter L2VPN configuration mode.

Note You cannot enter L2VPN configuration mode in a non-local context. L2VPN configuration mode is allowed only in the local context.

Use the **no** form of this command to delete all configured Layer 2 Virtual Private Network (L2VPN) cross-connections.

Examples

The following example changes the command mode from `context` configuration to L2VPN configuration:

```
[local]Redback(config)#context local  
[local]Redback(config-ctx)#l2vpn  
[local]Redback(config-l2vpn)#
```

Related Commands

l2vpn-cct-bindings ldp
l2vpn-cct-bindings static
l2vpn ctx-name
xc vc-id
xc vpn-label

l2vpn-cct-bindings ldp

l2vpn-cct-bindings ldp

no l2vpn-cct-bindings ldp

Purpose

Enters L2VPN LDP configuration mode.

Command Mode

L2VPN configuration

Syntax Description

This command has no keywords or arguments.

Default

None

Usage Guidelines

Use the **l2vpn-cct-bindings ldp** command to enter L2VPN LDP configuration mode.

Use the **no** form of this command to delete all Label Distribution Protocol (LDP) Layer 2 Virtual Private Network (L2VPN) cross-connections.

Examples

The following example changes the command mode from L2VPN configuration to L2VPN LDP configuration:

```
[local]Redback(config-ctx)#l2vpn  
[local]Redback(config-l2vpn)#l2vpn-cct-bindings ldp  
[local]Redback(config-l2vpn-ldp)#
```

Related Commands

l2vpn

l2vpn-cct-bindings static

l2vpn ctx-name

xc vc-id

xc vpn-label

l2vpn-cct-bindings static

l2vpn-cct-bindings static
no l2vpn-cct-bindings static

Purpose

Enters L2VPN static configuration mode.

Command Mode

L2VPN configuration

Syntax Description

This command has no keywords or arguments.

Default

None

Usage Guidelines

Use the **l2vpn-cct-bindings static** command to enter L2VPN static configuration mode.

Use the **no** form of this command to delete all static Layer 2 Virtual Private Network (L2VPN) cross-connections.

Examples

The following example changes the command mode from L2VPN configuration to L2VPN **static** configuration:

```
[local]Redback(config-ctx)#l2vpn  
[local]Redback(config-l2vpn)#l2vpn-cct-bindings static  
[local]Redback(config-l2vpn-static)#
```

Related Commands

l2vpn
l2vpn-cct-bindings ldp
l2vpn ctx-name
xc vc-id
xc vpn-label

l2vpn ctx-name

l2vpn *ctx-name*

no l2vpn *ctx-name*

Purpose

Enables a Layer 2 (L2) circuit for Layer 2 Virtual Private Network (L2VPN) operation.

Command Mode

ATM PVC configuration
dot1q PVC configuration
Frame Relay PVC configuration
port configuration

Syntax Description

<i>ctx-name</i>	Name of the context in which the L2VPN is created.

Default

L2 circuits are not enabled for L2VPN operation.

Usage Guidelines

Use the **l2vpn ctx-name** command in any L2 circuit configuration mode to enable an L2 circuit for L2VPN operation.

Use the **no** form of this command to disable L2 circuits for L2VPN operation.

Examples

The following example enables an Asynchronous Transfer Mode (ATM) permanent virtual circuit (PVC) for L2VPN operation:

```
[local]Redback(config)#port atm 6/1
[local]Redback(config-atm)#atm pvc 1 101 profile ubr encapsulation bridge1483
[local]Redback(config-atmpvc)#l2vpn
[local]Redback(config-atmpvc)#
```

The following example enables a dot1q PVC for L2VPN operation:

```
[local]Redback(config)#port ethernet 3/0
[local]Redback(config-port)#encapsulation dot1q
[local]Redback(config-port)#dot1q pvc 20
[local]Redback(config-dot1q-pvc)#l2vpn
[local]Redback(config-dot1q-pvc)#
```


The following example enables a Frame Relay PVC for L2VPN operation:

```
[local]Redback(config)#port pos 3/1  
[local]Redback(config-port)#frame-relay pvc 16  
[local]Redback(config-frpvc)#l2vpn  
[local]Redback(config-frpvc)#
```

The following example enables an Ethernet port for L2VPN operation:

```
[local]Redback(config)#port ethernet 3/0  
[local]Redback(config-port)#l2vpn local  
[local]Redback(config-port)#
```

Related Commands

- l2vpn**
- l2vpn-cct-bindings ldp**
- l2vpn-cct-bindings static**
- xc vc-id**
- xc vpn-label**

xc vc-id

```
xc slot/port[:chan-num] [:sub-chan-num] [circuit-id] vc-id vc-id peer peer-addr [remote encaps type]
[exp-bits bits-num]

no xc slot/port[:chan-num] [:sub-chan-num] [circuit-id] vc-id vc-id peer peer-addr [remote encaps
type] [exp-bits bits-num]
```

Purpose

Creates a Label Distribution Protocol (LDP) Layer 2 Virtual Private Network (L2VPN) cross-connection.

Command Mode

L2VPN LDP configuration

Syntax Description

<i>slot</i>	Chassis slot number with the port for which a cross-connection is to be specified.
<i>port</i>	Card port number of the port for which a cross-connection is to be specified.
<i>chan-num</i>	Optional. Channel number on the port for which a cross-connection is to be specified. The range of values is 0 to 32,767. For Asynchronous Transfer Mode (ATM) OC cards, a default channel number of 1 must be specified.
<i>sub-chan-num</i>	Optional. Subchannel number on the port for which a cross-connection is to be specified. The range of values is 0 to 255.
<i>circuit-id</i>	Optional. Layer 2 (L2) circuit ID. Depending on the type of circuit being cross-connected, the L2 circuit ID takes one of the following constructs: • vpi-vci <i>vpi vci</i>—ATM permanent virtual circuit (PVC). Specifies the virtual path identifier (VPI) and virtual channel identifier (VCI). The range of values for the <i>vpi</i> and <i>vci</i> arguments are 0 to 255, and 1 to 65,535 respectively. • vlan-id <i>vlan-id</i>—Virtual LAN (VLAN) tag value for an 802.1Q PVC. The <i>vlan-id</i> argument is one of the following constructs: • <i>tunl-vlan-id;pvc-vlan-id</i>—VLAN tag value for the tunnel followed by the VLAN tag value for the PVC within the tunnel. • <i>pvc-vlan-id</i>—VLAN tag value of a PVC that is not within an 802.1Q tunnel. The range of values for either VLAN tag value is 1 to 4,095. • dlci <i>dlci</i>—Data-link connection identifier (DLCI) for the Frame Relay PVC. The range of values is 16 to 991. • For Ethernet ports with no 802.1Q PVCs, no circuit descriptor is specified.

<i>vc-id</i>	Virtual circuit (VC) identifier associated with the LDP L2VPN cross-connection. The range of the <i>vc-id</i> argument values is 0 to 4,294,967,295.
peer <i>peer-addr</i>	IP address of the remote peer provider edge (PE) router.
remote-encap <i>type</i>	Optional. Specifies that a different encapsulation type is used at the remote end of the cross-connection. The <i>type</i> argument specifies one of the following encapsulation types: • 1qtunnel—Specifies the 802.1Q tunnel encapsulation type. • bridged1483—Specifies the RFC 1483 bridged encapsulation type. • dot1q—Specifies the 802.1Q Ethernet encapsulation type.
exp-bits <i>bits-num</i>	Optional. EXP bits to be used for transport. The range of the <i>bits-num</i> argument values is 0 to 7.

Default

None

Usage Guidelines

Use the **xc vc-id** command to create an LDP L2VPN cross-connection.

When creating a cross-connection to a remote circuit that uses an encapsulation type that is different than the encapsulation type of the local circuit, use the **remote-encap** keyword to specify the encapsulation type used at the remote end of the cross-connection.

Note The SmartEdge router supports the following encapsulation interconnectivity:

- ATM RFC 1483 bridged to dot1q
- ATM RFC 1483 bridged to Ethernet

For ATM OC cards, you must specify a default channel number of 1 in the **xc vc-id** command; for example, if the card is an ATM-OC3c/STM-1c, then you must specify a default channel number of 1.

Note ATM PVC cross-connections support PDU mode, and not cell mode.

Use the **no** form of this command to delete all LDP L2VPN cross-connections.

Examples

The following example creates a LDP L2VPN cross-connection between an ATM PVC and the remote peer PE router, 101.1.1.1:

```
[local]Redback(config-ctx)#l2vpn
[local]Redback(config-l2vpn)#l2vpn-cct-bindings ldp
[local]Redback(config-l2vpn-ldp)#xc 12/1 vpi-vci 200 1256 vc-id 2 peer 101.1.1.1
```

The following example creates a LDP L2VPN cross-connection between an 802.1Q PVC and the remote peer PE router, 101.1.1.1:

```
[local]Redback(config-ctx)#l2vpn
[local]Redback(config-l2vpn)#l2vpn-cct-bindings ldp
[local]Redback(config-l2vpn-ldp)#xc 12/1 vlan-id 200 vc-id 2 peer 101.1.1.1
```

The following example creates a LDP L2VPN cross-connection between an Frame Relay PVC and the remote peer PE router, 101.1.1.2:

```
[local]Redback(config-ctx)#l2vpn
[local]Redback(config-l2vpn)#l2vpn-cct-bindings ldp
[local]Redback(config-l2vpn-ldp)#xc 12/1 dlci 101 vc-id 2 peer 101.1.1.2
```

The following example creates a LDP L2VPN cross-connection between an Ethernet port and the remote peer PE router, 101.1.1.3:

```
[local]Redback(config-ctx)#l2vpn
[local]Redback(config-l2vpn)#l2vpn-cct-bindings ldp
[local]Redback(config-l2vpn-ldp)#xc 12/1 vc-id 2 peer 101.1.1.3
```

The following example creates a LDP L2VPN cross-connection between an Ethernet port and a remote circuit that uses 802.1Q PVC encapsulation:

```
[local]Redback(config-ctx)#l2vpn
[local]Redback(config-l2vpn)#l2vpn-cct-bindings ldp
[local]Redback(config-l2vpn-ldp)#xc 12/1 vc-id 2 peer 101.1.1.3 remote-encap dot1q
```

Related Commands

- l2vpn**
- l2vpn-cct-bindings ldp**
- l2vpn-cct-bindings static**
- l2vpn ctx-name**
- xc vpn-label**

xc vpn-label

xc *slot/port[:channel]* *circuit-id* **vpn-label** *label* **peer** *peer-addr*

no **xc** *slot/port[:channel]* *circuit-id* **vpn-label** *label* **peer** *peer-addr*

Purpose

Creates a static Layer 2 Virtual Private Network (L2VPN) cross-connection.

Command Mode

L2VPN static configuration

Syntax Description

<i>slot</i>	Chassis slot number with the port for which a cross-connection is to be specified.
<i>port</i>	Card port number of the port for which a cross-connection is to be specified.
<i>channel</i>	Optional. Channel number on the port for which a cross-connection is to be specified. For Asynchronous Transfer Mode (ATM) OC cards, a default channel number of 1 must be specified.
<i>circuit-id</i>	Layer 2 (L2) circuit ID. Depending on the type of circuit being cross-connected, the L2 circuit ID takes one of the following constructs: - For ATM permanent virtual circuits (PVCs), use the vpi-vci <i>vpi vci</i> construct, which denotes the virtual path identifier (VPI) and virtual channel identifier (VCI) for the ATM. The range of values for the VPI and VCI are 0 to 255, and 1 to 65,535 respectively. - For 802.1Q PVCs, use the vlan-id <i>vlan-id</i> construct, which denotes the VLAN tag value for the 802.1Q PVC. The range of values is 1 to 4,095. - For Frame Relay PVCs, use the dlci <i>dlci</i> construct, which denotes the data-link connection identifier (DLCI) for the Frame Relay PVC. The range of values is 16 to 991. - For Ethernet ports, no circuit descriptor is specified.
<i>label</i>	Inner label associated with the static L2VPN cross-connection. The range of the <i>label</i> argument values is 4,096 to 65,535.
peer <i>peer-addr</i>	IP address of the remote peer provider edge (PE) router.

Default

None

Usage Guidelines

Use the **xc vpn-label** command to create a static L2VPN cross-connection.

For ATM OC cards, you must specify default channel number of 1 in the **xc vpn-label** command; for example, if the card is an ATM-OC3c/STM-1c, then you must specify a default channel number of 1.

Note ATM PVC cross-connections support PDU mode, and not cell mode.

Use the **no** form of this command to delete all static L2VPN cross-connections.

Examples

The following example creates a static L2VPN cross-connection between an ATM PVC and the remote peer PE router, 192.168.1.1:

```
[local]Redback(config-ctx)#l2vpn
[local]Redback(config-l2vpn)#l2vpn-cct-bindings static
[local]Redback(config-l2vpn-static)#xc 12/1 vpi-vci 10 12 vpn-label 5000 peer 101.1.1.1
```

The following example creates a static L2VPN cross-connection between an 802.1Q PVC and the remote peer PE router, 192.168.1.1:

```
[local]Redback(config-ctx)#l2vpn
[local]Redback(config-l2vpn)#l2vpn-cct-bindings static
[local]Redback(config-l2vpn-static)#xc 12/1 vlan-id 200 vpn-label 5000 peer 101.1.1.1
```

The following example creates a static L2VPN cross-connection between an Frame Relay PVC and the remote peer PE router, 101.1.1.2:

```
[local]Redback(config-ctx)#l2vpn
[local]Redback(config-l2vpn)#l2vpn-cct-bindings static
[local]Redback(config-l2vpn-static)#xc 12/1 dlci 101 vpn-label 5000 peer 101.1.1.2
```

The following example creates a static L2VPN cross-connection between an Ethernet port and the remote peer PE router, 101.1.1.3:

```
[local]Redback(config-ctx)#l2vpn
[local]Redback(config-l2vpn)#l2vpn-cct-bindings static
[local]Redback(config-l2vpn-static)#xc 12/1 vpn-label 5000 peer 101.1.1.3
```

Related Commands

- l2vpn**
- l2vpn-cct-bindings ldp**
- l2vpn-cct-bindings static**
- l2vpn ctx-name**
- xc vc-id**

LDP Configuration

This chapter provides an overview of the Label Distribution Protocol (LDP) and describes the tasks and commands used to configure LDP features through the SmartEdge® OS.

For information about the tasks and commands used to monitor, troubleshoot, and administer LDP, see the “LDP Operations” chapter in the *Routing Protocols Operations Guide* for the SmartEdge OS.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples
- Command Descriptions

Overview

The following sections provide an overview of LDP concepts:

- LDP Implementation
- LDP Neighbor Discovery
- LDP Hello Messages

LDP Implementation

Our implementation of LDP supports RFC 3036, *LDP Specification*. LDP enables dynamic label allocation and distribution in a Multiprotocol Label Switching (MPLS) network. A label-switched router (LSR) enabled with LDP can establish label-switched paths (LSPs) to other LSRs in the network. LDP creates label bindings by assigning labels to connected routers and by advertising the bindings to neighbors. LDP also assigns labels to label bindings learned from neighbors, and readvertises the binding to other neighbors. When an LSR advertises a label binding for a route, the LSR is advertising the availability of an LSP to the destination of that route. LDP can learn several LSPs from different neighbors for the same route. In this case, LDP activates only the path selected by the underlying Interior Gateway Protocol (IGP). For this reason, LDP must work together with an IGP, such as the Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF) protocol.

To discover LDP peers, an LSR periodically transmits LDP Hello messages. After two LDP peers discover each other in this manner, LDP establishes a Transmission Control Protocol (TCP) connection between them. When the TCP connection is complete, an LDP session is established. In Redback's implementation, the LDP router ID is used as the transport address.

During the LDP session, LSRs send LDP label mapping and withdrawal messages. LSRs allocate labels to directly connected interfaces and learn about labels from neighbors. If a directly connected interface is shut down, an LSR withdraws the label and stops advertising it to neighbors. If a neighbor stops advertising a label to an LSR, the label is withdrawn from that LSR's Label Forwarding Information Base (LFIB). Teardown of LDP adjacencies or sessions results if Hello or keepalive messages are not received within the timer interval.

LDP Neighbor Discovery

There are two types of LDP neighbor discovery mechanisms: basic LDP discovery and extended LDP discovery. Basic LDP discovery is used to discover immediate neighbors; extended LDP discovery is used to discover neighbors that can be multiple hops away.

LDP Hello Messages

There are two types of LDP Hello messages: link Hello messages and targeted Hello messages. Link Hello messages are multicast on an interface to immediate neighbors. Link Hello messages are used in basic LDP discovery. Targeted Hello messages are unicast directly to remote neighbors. Targeted Hello messages are used in extended LDP discovery. Two LDP speaking LSRs can form LDP adjacencies after discovering each other. LDP adjacencies discovered by link Hello are link Hello adjacencies. LDP adjacencies discovered by targeted Hello are targeted Hello adjacencies.

Configuration Tasks

Note In this section, the command syntax in the task tables displays only the root command; for the complete command syntax, see the full description for the command in the "Command Descriptions" section.

For the context in which you configure LDP, you must also:

- Configure an MPLS routing instance.
- Enable MPLS on the interface on which you plan to enable LDP.

To ensure that the LDP router ID is always reachable, we recommend that you also configure a loopback interface that is advertised by the IGP, such as OSPF or IS-IS, routing instance.

Note To configure an IGP routing instance and interface, such as IS-IS or OSPF, see either Chapter 6, "OSPF Configuration," or Chapter 10, "IS-IS Configuration." To configure MPLS, see Chapter 13, "MPLS Configuration."

To configure LDP, perform the tasks described in the following sections:

- Configuring an LDP Routing Instance
- Configuring the Hello Adjacency Holdtime (Optional)
- Configuring the Hello Message Interval

Configuring an LDP Routing Instance

To configure an LDP routing instance, perform the tasks described in Table 15-1. Enter all commands in LDP router configuration mode, unless otherwise noted.

Table 15-1 Configure an LDP Routing Instance

Task	Root Command	Notes
Enable an LDP routing instance for a context, and to access LDP router configuration mode, use the following command in context configuration mode:	router ldp	Enter this command in context configuration mode. For LDP to work properly, LDP must work together with an Interior Gateway Protocol (IGP), such as OSPF, IS-IS, RIP, or static routing. Enable LDP in the same context in which the underlying IGP is configured. For LDP to be able to establish sessions, the LDP transport address of an LDP instance must be reachable. It is recommended that you configure a loopback interface whose address is advertised by the underlying IGP.
Enables the creation of LDP LSP pseudo-circuits.	create-lsp-circuit	Before packet statistics for LDP LSPs can be collected, LDP LSP pseudo-circuits must first be created.
Enable an egress router to advertise an explicit null label (value 0), in place of an implicit null label (value 3), to the penultimate hop router.	explicit-null	By default, LDP advertises an implicit null label for directly connected prefixes. An implicit null label causes the upstream router to perform penultimate hop popping (PHP), and the implicit null label is not transmitted on the last hop. In some cases, such as QoS enforcement, you may not want PHP. In those cases, you can use the explicit-null command to cause a router to advertise an explicit null label in place of an implicit null label for directly connected prefixes, which forces the upstream router to transmit packets with an explicit null label on the last hop. When an explicit-null command is specified for a particular neighbor, an if a context level explicit-null command has been configured, then the context level explicit-null command does not apply to the neighbor.
Enable or disable the graceful restart capability.	graceful-restart	Use the no form of this command to disable graceful restart. When graceful restart is enabled, the LSR restarts its LDP component while preserving its MPLS forwarding component across restart. After an LSR restarts its control plane, it starts an internal MPLS forwarding state holding timer, and continues to forward traffic using the preserved MPLS forwarding state entries. Before the MPLS forwarding state hold timer expires, the LSR creates local label bindings by following the normal LDP procedure. When the hold timer expires, the preserved forwarding entries are deleted, and normal operation resumes.
Enable LDP on an interface so that it can be used to exchange Hello messages with neighbors and to establish an LSP.	interface	You must also enable MPLS on the interface for the LSP to be established properly. You may also need to enable an IGP, such as IS-IS or OSPF, on the interface.
Apply an IP prefix list to filter LDP label advertisements.	label-binding	A typical filtering application is to apply a prefix list that restricts LDP to advertise labels for only loopback interface IP addresses. Limiting LDP label advertisements to loopback interfaces provides fast and reliable transportation of label binding information, and streamlines the efforts to build LSPs.

Table 15-1 Configure an LDP Routing Instance (*continued*)

Task	Root Command	Notes
Assign an encrypted MD5 password to an LDP neighbor.	neighbor password	For an LDP session to be established, the MD5 password must be the same on both the router and its neighbor.
Configure a remote LDP neighbor and enable extended LDP discovery of the specified neighbor.	neighbor targeted	LDP targeted neighbor discovery is required for L2VPN support if the PE routers are not directly connected. Using the targeted discovery mechanism, the PE routers establish an LDP session using an extended discovery mechanism where they do not have to be directly connected (as is required in hop-by-hop neighbors). LDP is used to distribute L2VPN labels to the remote router. LDP is used for distributing the VC labels across the path from the egress PE router to the ingress PE router. The VC label bindings are distributed using LDP downstream unsolicited mode. The PE routers establish an LDP session using an extended discovery mechanism where they do not have to be directly connected (as required in hop-by-hop neighbors). A new FEC type element is used for targeted discovery. A single VC forwarding equivalence class (FEC) element must be advertised per VC label. For distributing L2VPN labels, targeted LDP implementation supports the following features: • LDP downstream Unsolicited Mode • LDP request operation implemented in LDP • VC labels allocated from per platform label space
Configure the interface to be used as the LDP router ID.	router-id	Because the router ID is used as the transport IP address for establishing a TCP connection, changing the router ID causes an active LDP session to be torn down, and then re-established. Take care not to change the router ID when an LDP session is active. By default, the SmartEdge router determines the LDP router ID in the following sequence: • If a fixed LDP router ID configured through the router-id command in LDP configuration mode, it is used. • If an LDP router ID is not configured, and a system router ID is configured through the router-id command in context configuration mode, the system router ID is used. • If neither router ID is configured, the configured loopback interface with the highest IP address is used as the LDP router ID. • If a loopback interface is not configured, the operational interface with the highest IP address is used as the LDP router ID.
Enable LDP LSPs to inherit the Intermediate System-to-Intermediate System (IS-IS) routing metric for Border Gateway Protocol (BGP) to use when selecting a path.	track-igp-metric	
Configure the transport address advertised in LDP Hello messages.	transport address	Transport addresses are advertised in LDP Hello messages and are exchanged among LDP neighbors. LDP uses the local transport address as the source, and the received transport address as the destination when trying to establish a TCP connection to a neighbor. Therefore, transport addresses must be reachable. LDP also uses transport addresses to determine which of the two LSRs should perform active open. If a transport address is not explicitly configured, the LSR router ID is used as the transport address. In this case, the router ID must be reachable; however, if a transport address is explicitly configured, then the specified value is used. In this case, the router ID is not required to be reachable.

Table 15-1 Configure an LDP Routing Instance (*continued*)

Task	Root Command	Notes
Configure the Hello adjacency holdtime (optional).		For the complete list of tasks used to configure the Hello adjacency holdtime, see the "Configuring the Hello Adjacency Holdtime (Optional)" section.
Configure the Hello message interval.		For the complete list of tasks used to configure the Hello message interval, see the "Configuring the Hello Message Interval" section.

Configuring the Hello Adjacency Holdtime (Optional)

To configure the Hello adjacency holdtime, perform the tasks described in Table 15-2. Enter all commands in LDP router configuration mode.

Table 15-2 Configure the Hello Adjacency Holdtime

Task	Root Command	Notes
Configure the time for which an LDP link Hello adjacency is maintained in the absence of link Hello messages from the LDP neighbor.	hello holdtime	LDP neighbors periodically exchange Hello messages to maintain their adjacencies. The Hello holdtime determines the time after which, if LDP messages from the LDP neighbor are not received, the LDP hello adjacency is deleted. When the last LDP adjacency to a LDP neighbor is deleted, the LDP session to that LDP neighbor is torn down. For LDP neighbors to negotiate a Hello holdtime, each LDP neighbor includes a proposed Hello holdtime in their transmitted Hello message. The negotiated Hello holdtime used between the two neighbors is the lesser of the two proposed values. The locally configured link Hello holdtime as specified in hello holdtime command is included in the link Hello messages sent to immediate LDP neighbors. The negotiated holdtime used to timeout a link Hello adjacency is the lesser of the time value specified in the hello holdtime command and the hello holdtime received in link hello messages from the LDP neighbor of the adjacency. The default link Hello adjacency holdtime is 15 seconds.
Configure the time for which LDP targeted Hello adjacency is maintained in the absence of targeted Hello messages from an LDP neighbor.	targeted-hello holdtime	If LDP targeted Hello messages from an LDP neighbor are not received after the specified Hello holdtime, the LDP adjacency is deleted. If this is the last adjacency between the local LDP instance and an LDP neighbor, the LDP session to that LDP neighbor is torn down. The locally configured targeted Hello holdtime as specified by the targeted-hello holdtime command is included in the targeted Hello messages sent to remote LDP neighbors. The negotiated holdtime used to timeout a targeted Hello adjacency is the minimum of the time value specified by the targeted-hello holdtime command and the Hello holdtime received in targeted Hello messages from the LDP neighbor of the adjacency.

Configuring the Hello Message Interval

To configure the Hello message interval, perform the tasks described in Table 15-3. Enter all commands in LDP router configuration mode.

Table 15-3 Configure the Hello Message Interval

Task	Root Command	Notes
Configure the interval between consecutive LDP link Hello messages used in basic LDP discovery.	hello interval	If the Hello interval is explicitly configured, then the specified value is used to control the link Hello interval regardless of the link Hello holdtime; however, if the Hello interval is not explicitly configured, the Hello interval used is the negotiated LDP link Hello holdtime divided by three. The negotiated LDP link Hello holdtime is the lesser of the received LDP link Hello holdtime and the locally configured LDP link Hello holdtime.
Configure the interval between consecutive LDP targeted Hello messages used in extended LDP discovery.	targeted-hello interval	If the targeted Hello interval is explicitly configured, then the specified value is used to control targeted Hello interval regardless of the targeted Hello holdtime; however, if the targeted Hello interval is not explicitly configured, the targeted Hello interval used is the negotiated LDP targeted Hello holdtime divided by three. The negotiated LDP targeted Hello holdtime is the lesser of the received LDP targeted Hello holdtime and the locally configured LDP targeted Hello holdtime.

Configuration Examples

This section provides LDP configuration examples in the following sections:

- Basic LDP
- Targeted LDP

Basic LDP

The following example configures an IS-IS backbone network between two SmartEdge routers. Each router has an IS-IS, MPLS, and LDP routing instance and a single interface (the backbone between the two routers) enabled for IS-IS, MPLS, and LDP. Each router has an IS-IS loopback interface that is used as the LDP router ID. A filter restricts LDP to advertise labels for only loopback interface IP addresses.

The configuration for Router_A is as follows:

```
[local]Router_A(config)#context local
[local]Router_A(config-ctx)#router isis isis-backbone
[local]Router_A(config-isis)#net 49.2222.0010.0100.1001.00
[local]Router_A(config-isis)#exit
[local]Router_A(config-ctx)#ip prefix-list loop-only
[local]Router_A(config-prefix-list)#permit 0.0.0.0/0 eq 32
[local]Router_A(config-prefix-list)#exit
[local]Router_A(config-ctx)#interface backbone1
[local]Router_A(config-if)#ip address 10.1.1.1/24
[local]Router_A(config-if)#isis router isis-backbone
[local]Router_A(config-if)#exit
[local]Router_A(config-ctx)#interface loop1 loopback
```

```

[local]Router_A(config-if)#ip address 1.1.1.1/32
[local]Router_A(config-if)#isis router isis-backbone
[local]Router_A(config-if)#isis passive-interface
[local]Router_A(config-if)#exit
[local]Router_A(config-ctx)#router mpls 1
[local]Router_A(config-mpls)#interface backbone1
[local]Router_A(config-mpls-interface)#exit
[local]Router_A(config-mpls)#exit
[local]Router_A(config-ctx)#exit
[local]Router_A(config)#port pos 6/1
[local]Router_A(config-port)#bind interface backbone1 local
[local]Router_A(config-port)#no shutdown
[local]Router_A(config-port)#exit
[local]Router_A(config)#context local
[local]Router_A(config-ctx)#router ldp
[local]Router_A(config-ldp)#interface backbone1
[local]Router_A(config-ldp)#label-binding prefix-list loop-only out

```

The configuration for Router_B is as follows:

```

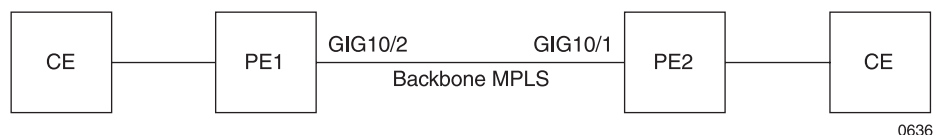
[local]Router_B(config)#context local
[local]Router_B(config-ctx)#router isis isis-backbone
[local]Router_B(config-isis)#net 49.2222.0010.0100.1001.00
[local]Router_B(config-isis)#exit
[local]Router_B(config-ctx)#ip prefix-list loop-only
[local]Router_B(config-prefix-list)#permit 0.0.0.0/0 eq 32
[local]Router_B(config-prefix-list)#exit
[local]Router_B(config-ctx)#interface backbone1
[local]Router_B(config-if)#ip address 10.2.2.2/24
[local]Router_B(config-if)#isis router isis-backbone
[local]Router_B(config-if)#exit
[local]Router_B(config-ctx)#interface loop1 loopback
[local]Router_B(config-if)#ip address 1.1.1.1/32
[local]Router_B(config-if)#isis router isis-backbone
[local]Router_B(config-if)#exit
[local]Router_B(config-ctx)#router mpls 1
[local]Router_B(config-mpls)#interface backbone1
[local]Router_B(config-mpls-interface)#exit
[local]Router_B(config-mpls)#exit
[local]Router_B(config-ctx)#exit
[local]Router_B(config)#port pos 6/1
[local]Router_B(config-port)#bind interface backbone1 local
[local]Router_B(config-port)#no shutdown
[local]Router_B(config-port)#exit
[local]Router_B(config)#context local
[local]Router_B(config-ctx)#router ldp
[local]Router_B(config-ldp)#interface backbone1
[local]Router_B(config-ldp)#label-binding prefix-list loop-only out

```

Targeted LDP

The following example configures two PE routers (PE1 and PE2) for targeted LDP discovery. The two routers are connected over an IGP in an MPLS network, so their router IDs are known to each other via IGP. Figure 15-1 shows the network topology for this example.

Figure 15-1 Targeted LDP Network Topology



The LDP router ID address is also used as the LDP transport address for establishing the LDP targeted neighbor. The **router-id** command is used in LDP router configuration mode to configure the LDP router ID on the router. If the **router-id** command is removed from the configuration example, the LDP router ID is picked up as follows:

- If one or more loopback addresses are present, the highest loopback address is used as the neighbor, and the router ID address is used as transport address.
- If no loopback addresses are present, the highest interface address is used as the LDP router ID.

The configuration for the PE1 router is as follows:

```
[local]PE1(config)#context local
[local]PE1(config-ctx)#interface loop1 loopback
[local]PE1(config-if)#ip address 11.200.1.1/32
[local]PE1(config-if)#exit
[local]PE1(config-ctx)#router ldp
[local]PE1(config-ldp)#router-id 11.200.1.1
[local]PE1(config-ldp)#neighbor 11.200.1.2 targeted
[local]PE1(config-ldp)#end
```

The configuration for the PE2 router is as follows:

```
[local]PE2(config)#context local
[local]PE2(config-ctx)#interface loop1 loopback
[local]PE2(config-if)#ip address 11.200.1.2
[local]PE2(config-if)#exit
[local]PE2(config-ctx)#router ldp
[local]PE2(config-ldp)#router-id 11.200.1.2
[local]PE2(config-ldp)#neighbor 11.200.1.1 targeted
[local]PE2(config-ldp)#end
```

Command Descriptions

This section describes the syntax and usage guidelines for the commands used to configure LDP features. The commands are presented in alphabetical order.

create-lsp-circuit	neighbor targeted
explicit-null	router-id
graceful-restart	router ldp
hello holdtime	targeted-hello holdtime
hello interval	targeted-hello interval
interface	track-igp-metric
label-binding	transport address
neighbor password	

create-lsp-circuit

create-lsp-circuit

no create-lsp-circuit

Purpose

Enables the creation of pseudo-circuits for Label Distribution Protocol (LDP) label-switched paths (LSPs).

Command Mode

LDP router configuration

Syntax Description

This command has no keywords or arguments.

Default

Pseudo-circuits are not created for LDP LSPs.

Usage Guidelines

Use the **create-lsp-circuit** command to enable the creation of pseudo-circuits for LDP LSPs. Before packet statistics for LDP LSPs can be collected, pseudo-circuits for the LDP LSPs must first be created.

Note Resource Reservation Protocol (RSVP) LSP circuit creation is always enabled.

Use the **no** form of this command to disable the creation of pseudo-circuits for LDP LSPs.

Examples

The following example enables the creation of pseudo-circuits for LDP LSPs:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#router ldp
[local]Redback(config-ldp)#create-lsp-circuit
[local]Redback(config-ldp)#
```

Related Commands

router ldp

explicit-null

[**neighbor** *ip-addr*] **explicit-null** [**prefix-list** *pl-name*]

no [**neighbor** *ip-addr*] **explicit-null** [**prefix-list** *pl-name*]

Purpose

Enables an egress router to advertise an explicit null label (value 0), in place of an implicit null label (value 3), to the penultimate hop router.

Command Mode

LDP router configuration

Syntax Description

neighbor <i>ip-addr</i>	Optional. Neighbor IP address. Enables the advertisement of explicit null labels to the neighbor specified by the <i>ip-addr</i> argument. When a neighbor is not specified, explicit null advertisement is enabled for all neighbors in the context.
prefix-list <i>pl-name</i>	Optional. Prefix list name. Applies the filters in the specified prefix list to label advertisements and enables advertisement of explicit null labels only for directly connected prefixes that are permitted by the prefix list. When the prefix list is not specified, explicit null label advertisement is enabled for all directly connected prefixes.

Default

The implicit null label (value 3) is advertised.

Usage Guidelines

Use the **explicit-null** command to enable an egress router to advertise an explicit null label (value 0), in place of an implicit null label (value 3), to the penultimate hop router.

By default, Label Distribution Protocol (LDP) advertises an implicit null label for directly connected prefixes. An implicit null label causes the upstream router to perform penultimate hop popping (PHP), and the implicit null label is not transmitted on the egress router. In some cases, such as quality of service (QoS) enforcement, PHP may not be desirable. In those cases, using the **explicit-null** command causes the egress router to advertise an explicit null label in place of an implicit null label for directly connected prefixes, which forces the upstream router to transmit packets with an explicit null label on the last hop.

If a neighbor IP address is specified, then the **explicit-null** command is neighbor-specific, and applies only to the LDP neighbor whose transport address matches the IP address specified in the command. If a neighbor address is not specified, then the **explicit-null** command is non neighbor-specific, and applies to all LDP neighbors in the context.

When both a neighbor-specific **explicit-null** command and a non neighbor-specific **explicit-null** command exist, only the neighbor-specific command applies to the neighbor whose transport address matches the IP address given in the neighbor-specific explicit-null command.

Use the **no** form of this command to disable explicit null label advertisement.

Examples

The following example enables advertising explicit-null label to neighbor 10.1.1.1 for directly connected prefixes that match the prefix-list, net01:

```
[local]Redback(config-ctx)#ip prefix-list net01 permit 155.0.0.0/8 ge 8
[local]Redback(config-ctx)#router ldp
[local]Redback(config-ldp)#neighbor 10.1.1.1 explicit-null prefix-list net01
```

Related Commands

- explicit-null**—RSVP router configuration mode
- hello holdtime**
- interface**—LDP router configuration mode
- label-binding**
- router-id**—LDP router configuration mode
- router ldp**

graceful-restart

graceful-restart

no graceful-restart

Purpose

Enables a label-switched router (LSR) to restart its Label Distribution Protocol (LDP) component while preserving its Multiprotocol Label Switching (MPLS) forwarding component across restart.

Command Mode

LDP router configuration

Syntax Description

This command has no keywords or arguments.

Default

Graceful restart is enabled.

Usage Guidelines

Use the **graceful-restart** command to enable an LSR to restart its LDP component while preserving its MPLS forwarding component across restart.

After an LSR restarts its control plane, it starts an internal MPLS forwarding state holding timer, and continues to forward traffic using the preserved MPLS forwarding state entries. Before the MPLS forwarding state hold timer expires, the LSR creates local label bindings by following the normal LDP procedure. When the hold timer expires, the preserved forwarding entries are deleted, and normal operation resumes.

Use the **no** form of this command to disable the graceful restart capability.

Examples

The following example disables an LSR from restarting its LDP component while preserving its MPLS forwarding component across restart:

```
[local]Redback(config-ldp)#no graceful-restart
```

Related Commands

router ldp

hello holdtime

hello holdtime *seconds*

default hello holdtime

Purpose

Changes the time for which a Label Distribution Protocol (LDP) link Hello adjacency is maintained in the absence of link Hello messages from the LDP neighbor.

Command Mode

LDP router configuration

Syntax Description

<i>seconds</i>	Number of seconds after which, if LDP link hello messages from the LDP neighbor is not received, the LDP adjacency is deleted. The range of values is 15 to 3,600.

Default

The default LDP link hello holdtime is 15 seconds.

Usage Guidelines

Use the **hello holdtime** command to change the time for which an LDP link Hello adjacency is maintained in the absence of link Hello messages from the LDP neighbor.

LDP neighbors periodically exchange Hello messages to maintain their adjacencies. The Hello holdtime determines the time after which, if LDP messages from the LDP neighbor are not received, the LDP hello adjacency is deleted. When the last LDP adjacency to a LDP neighbor is deleted, the LDP session to that LDP neighbor is torn down.

For LDP neighbors to negotiate a Hello holdtime, each LDP neighbor includes a proposed Hello holdtime in their transmitted Hello message. The negotiated Hello holdtime used between the two neighbors is the lesser of the two proposed values.

The locally configured link Hello holdtime as specified in **hello holdtime** command is included in the link Hello messages sent to immediate LDP neighbors. The negotiated holdtime used to timeout a link Hello adjacency is the lesser of the time value specified in “hello holdtime” command and the hello holdtime received in link hello messages from the LDP neighbor of the adjacency.

Use the **default** form of this command to return to the default value of 15 seconds.

Examples

The following example configures the LDP hold time to be 45 seconds:

```
[local]Redback(config-ctx)#router ldp
[local]Redback(config-ldp)#hello holdtime 45
```

Related Commands

explicit-null
hello interval
interface—LDP router configuration mode
label-binding
router-id—LDP router configuration mode
router ldp
targeted-hello holdtime
targeted-hello interval

hello interval

hello interval *seconds*

default hello interval

Purpose

Configures the interval between consecutive Label Distribution Protocol (LDP) link Hello messages used in basic LDP discovery.

Command Mode

LDP router configuration

Syntax Description

<i>seconds</i>	Number of seconds between consecutive LDP link Hello messages. The range of values is 5 to 1,200.

Default

The default LDP link Hello interval is five seconds.

Usage Guidelines

Use the **hello interval** command to configure the interval between consecutive LDP link Hello messages used in basic LDP discovery.

If the Hello interval is explicitly configured, then the specified value is used to control the link Hello interval regardless of the link Hello holdtime; however, if the Hello interval is not explicitly configured, the Hello interval used is the negotiated LDP link Hello holdtime divided by three. The negotiated LDP link Hello holdtime is the lesser of the received LDP link Hello holdtime and the locally configured LDP link Hello holdtime.

Use the **hello holdtime** command in LDP router configuration mode to change the locally configured LDP link Hello holdtime.

Use the **targeted-hello interval** command in LDP router configuration mode to change the locally configured LDP targeted hello interval.

Use the **default** form of this command to return to the default value of five seconds.

Examples

The following example configures an LDP link Hello interval of 10 seconds:

```
[local]Redback(config-ctx)#router ldp
[local]Redback(config-ldp)#hello interval 10
```

Related Commands

hello holdtime

interface—LDP router configuration mode

router-id—LDP router configuration mode

router ldp

targeted-hello holdtime

targeted-hello interval

interface

interface *if-name*

no interface *if-name*

Purpose

Enables the Label Distribution Protocol (LDP) on an interface so that the interface can be used to exchange Hello messages with neighbors and to establish a label-switched path (LSP).

Command Mode

LDP router configuration

Syntax Description

<i>if-name</i>	Name of the interface; an alphanumeric string.

Default

Disabled

Usage Guidelines

Use the **interface** command to enable LDP on an interface so that the interface can be used to exchange Hello messages with neighbors and to establish an LSP.

Note You must also enable Multiprotocol Label Switching (MPLS) on the interface for the LSP to be established properly. You may also need to enable an Interior Gateway Protocol (IGP), such as Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF). Commands are described in “Chapter 13, “MPLS Configuration,” Chapter 6, “OSPF Configuration,” and Chapter 10, “IS-IS Configuration.”

Use the **no** form of this command to disable LDP on the interface.

Examples

The following example enables an LDP, OSPF, and MPLS routing instance for the `local` context, and enables LDP, OSPF, and MPLS on the interface, `backbone1`:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#interface backbone1
[local]Redback(config-if)#ip address 10.1.2.3 255.255.255.0
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#router ospf 1
[local]Redback(config-ospf)#area 1
[local]Redback(config-ospf-area)#interface backbone1
[local]Redback(config-ospf-interface)#exit
[local]Redback(config-ospf-area)#exit
[local]Redback(config-ospf)#exit
[local]Redback(config-ctx)#router mpls 1
```



```
[local]Redback(config-mpls)#interface backbone1  
[local]Redback(config-mpls-if)#exit  
[local]Redback(config-mpls)#exit  
[local]Redback(config-ctx)#router ldp  
[local]Redback(config-ldp)#interface backbone1
```

Related Commands

explicit-null
hello holdtime
label-binding
router-id—LDP router configuration mode
router ldp

label-binding

[neighbor ip-addr] label-binding prefix-list pl-name {in | out}

no [neighbor ip-addr] label-binding prefix-list pl-name {in | out}

Purpose

Applies an IP prefix list to filter Label Distribution Protocol (LDP) label advertisements.

Command Mode

LDP router configuration

Syntax Description

neighbor ip-addr	Optional. Neighbor IP address. Filters label advertisements to and from the specified neighbor. If this construct is omitted, the prefix list is applied to all neighbors.
prefix-list pl-name	Prefix list name. Applies the filters in the specified prefix list to label advertisements. In doing so, restricts label advertisements to or from a Forwarding Equivalency Class (FEC), or set of destinations, that are identified in the prefix list.
in	Applies the prefix list to incoming label advertisements.
out	Applies the prefix list to outgoing label advertisements.

Default

Labels of directly connected interfaces and labels learned from LDP neighbors are advertised.

Usage Guidelines

Use the **label-binding** command to apply an IP prefix list to filter LDP label advertisements.

If the LDP neighbor's transport IP address differs from its router ID, the IP address specified in the **neighbor ip-addr** construct must be the LDP neighbor's transport IP address.

A typical application is to apply a prefix list that restricts LDP to advertise labels for only loopback interface IP addresses. Limiting LDP label advertisements to loopback interfaces provides fast and reliable transportation of label binding information, and streamlines the efforts to build LSPs.

To filter label advertisements, you must first configure the IP prefix list through the **ip prefix-list** command in context configuration mode. For more information, see Chapter 12, "Routing Policy Configuration."

Use the **no** form of this command to remove LDP label advertisement filtering.

Examples

The following example configures the LDP instance running in the `local` context to send LDP label advertisements over loopback interface addresses only:

```
[local]Redback(config)#context local  
[local]Redback(config-ctx)#ip prefix-list loopback-only  
[local]Redback(config-prefix-list)#permit 0.0.0.0/0 eq 32  
[local]Redback(config-prefix-list)#exit  
[local]Redback(config-ctx)#router ldp  
[local]Redback(config-ldp)#label-binding prefix-list loopback-only out
```

Related Commands

- explicit-null**
- hello holdtime**
- interface**—LDP router configuration mode
- ip prefix-list**
- router-id**—LDP router configuration mode
- router ldp**

neighbor password

neighbor *ip-addr* **password** *password*

no neighbor *ip-addr* **password**

Purpose

Assigns an encrypted Message Digest 5 (MD5) password to a Label Distribution Protocol (LDP) neighbor.

Command Mode

LDP router configuration

Syntax Description

<i>ip-addr</i>	Neighbor IP address in the form <i>A.B.C.D</i> .
<i>password</i>	Alphanumeric string consisting of up to 80 characters.

Default

MD5 password is disabled.

Usage Guidelines

Use the **neighbor password** command to assign an encrypted MD5 password to an LDP neighbor.

Note For an LDP session to be established, the MD5 password must be the same on both the router and its neighbor.

Use the **no** form of this command to remove the password from an LDP neighbor.

Examples

The following example assigns the password, *secret*, to LDP neighbor, *10.1.1.1*:

```
[local]Redback(config-ctx)#router ldp  
[local]Redback(config-ldp)#neighbor 10.1.1.1 password secret
```

Related Commands

neighbor targeted
router ldp

neighbor targeted

neighbor *ip-addr* **targeted**

no neighbor *ip-addr* **targeted**

Purpose

Configures a remote Label Distribution Protocol (LDP) neighbor and enables extended LDP discovery of the specified neighbor.

Command Mode

LDP router configuration

Syntax Description

ip-addr IP address of the remote LDP neighbor in the form *A.B.C.D*.

Default

Extended LDP discovery is disabled.

Usage Guidelines

There are two types of LDP neighbor discovery mechanisms: basic LDP discovery and extended LDP discovery. Basic LDP discovery is used to discover immediate neighbors; extended LDP discovery is used to discover neighbors that can be multiple hops away.

There are two types of LDP Hello messages: link Hello messages and targeted Hello messages. Link Hello messages are multicast on an interface to immediate neighbors. Link Hello messages are used in basic LDP discovery. Targeted Hello messages are unicast directly to remote neighbors, and are used in extended LDP discovery. Two LDP speaking label-switched routers (LSRs) can form LDP adjacencies after discovering each other. LDP adjacencies discovered by link Hello messages are link Hello adjacencies. LDP adjacencies discovered by targeted Hello messages are targeted Hello adjacencies.

Use the **neighbor targeted** command to configure a remote LDP neighbor and enable extended LDP discovery of the specified neighbor. Targeted Hello messages can be transmitted or accepted to or from the specified neighbor.

Use the **no** form of this command to remove a configured remote LDP neighbor, and to disable extended LDP discovery of the specified neighbor.

Examples

The following example configures a remote neighbor of address 10.1.1.1:

```
[local]Redback(config-ctx)#router ldp  
[local]Redback(config-ldp)#neighbor 10.1.1.1 targeted
```

Related Commands

neighbor password
router ldp
targeted-hello holdtime
targeted-hello interval

router-id

router-id *ip-addr*

no router-id *ip-addr*

Purpose

Configures the interface to be used as the Label Distribution Protocol (LDP) router ID.

Command Mode

LDP router configuration

Syntax Description

ip-addr IP address in the form *A.B.C.D*.

Default

By default, the SmartEdge router determines the LDP router ID in the following sequence:

1. If a fixed LDP router ID configured through the **router-id** command in LDP configuration mode, it is used.
2. If an LDP router ID is not configured, and a system router ID is configured through the **router-id** command in context configuration mode, the system router ID is used.
3. If neither router ID is configured, the configured loopback interface with the highest IP address is used as the LDP router ID.
4. If a loopback interface is not configured, the operational IS-IS or OSPF interface with the highest IP address is used as the LDP router ID.

Usage Guidelines

Use the **router-id** command to configure the interface to be used as the LDP router ID.



Caution Risk of traffic interruption. Because the router ID is used as the transport IP address for establishing a Transmission Control Protocol (TCP) connection, changing the router ID causes an active LDP session to be torn down, and then re-established. To reduce the risk, do not change the router ID when an LDP session is active.

Note We recommend that you configure a loopback interface that is advertised by the Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS) routing instance to ensure that the LDP router ID is always reachable.

Use the **no** form of this command to return the system to its default behavior.

Examples

The following example configures the interface, `ldp-routerID`, as the LDP router ID:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#router isis isis-backbone
[local]Redback(config-isis)#net 49.2222.0010.0100.1001.00
[local]Redback(config-isis)#exit
[local]Redback(config-ctx)#interface ldp-routerID
[local]Redback(config-ctx)#ip address 10.1.1.1 255.255.255.0
[local]Redback(config-if)#isis router isis-backbone
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#router ldp
[local]Redback(config-ldp)#router-id 10.1.1.1
```

Related Commands

- explicit-null**
- hello holdtime**
- interface**—LDP router configuration mode
- label-binding**
- router ldp**

router ldp

router ldp

no router ldp

Purpose

Enables a Label Distribution Protocol (LDP) routing instance for a context and enters LDP router configuration mode.

Command Mode

context configuration

Syntax Description

This command has no keywords or arguments.

Default

LDP routing is disabled.

Usage Guidelines

Use the **router ldp** command to enable an LDP routing instance for context, and to enter LDP router configuration mode. Our implementation of LDP follows the LDP specification as described in RFC 3036, *LDP Specification*.

For the context in which you configure LDP, you must also:

- Configure an Multiprotocol Label Switching (MPLS) routing instance.
- Enable MPLS on the interface on which you plan to enable LDP.

You may also need to enable an Interior Gateway Protocol (IGP), such as Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS), on the interface.

To ensure that the LDP router ID is always reachable, we recommend that you also configure a loopback interface that is advertised by the IGP, such as OSPF or IS-IS, routing instance.

Note For the commands used to configure an IGP routing instance and interface, such as IS-IS or OSPF, see either Chapter 6, “OSPF Configuration,” or Chapter 10, “IS-IS Configuration.” For MPLS commands, see Chapter 13, “MPLS Configuration.”

Use the **no** form of this command to disable LDP routing for the context.

Examples

The following example enables an LDP routing instance for the `local` context and enters LDP router configuration mode:

```
[local]Redback(config)#context local  
[local]Redback(config-ctx)#router ldp
```

[local]Redback(config-ldp)#

Related Commands

explicit-null

hello holdtime

interface—LDP router configuration mode

label-binding

router-id—LDP router configuration mode

targeted-hello holdtime

targeted-hello holdtime *seconds*

default targeted-hello holdtime

Purpose

Configures the time for which Label Distribution Protocol (LDP) targeted Hello adjacency is maintained in the absence of targeted Hello messages from an LDP neighbor.

Command Mode

LDP router configuration

Syntax Description

<i>seconds</i>	Number of seconds before LDP adjacency is deleted if LDP targeted Hello messages from an LDP neighbor are not received. The range of values is 15 to 3,600.

Default

The default LDP targeted Hello adjacency holdtime is 45 seconds.

Usage Guidelines

Use the **targeted-hello holdtime** command to configure the time for which LDP targeted Hello adjacency is maintained in the absence of targeted Hello messages from an LDP neighbor.

If LDP targeted Hello messages from an LDP neighbor are not received after the specified Hello holdtime, the LDP adjacency is deleted. If this is the last adjacency between the local LDP instance and an LDP neighbor, the LDP session to that LDP neighbor is torn down.

The locally configured targeted Hello holdtime as specified by the **targeted-hello holdtime** command is included in the targeted Hello messages sent to remote LDP neighbors. The negotiated holdtime used to timeout a targeted Hello adjacency is the minimum of the time value specified by the **targeted-hello holdtime** command and the Hello holdtime received in targeted Hello messages from the LDP neighbor of the adjacency.

Use the **hello holdtime** command in LDP router configuration mode to change the locally configured LDP link hello holdtime.

Use the **targeted-hello interval** command in LDP router configuration mode to change the locally configured LDP targeted hello interval.

Use the **default** form of this command to return to the default value of 45 seconds.

Examples

The following example configures a Hello holdtime of 60 seconds:

```
[local]Redback(config-ctx)#router ldp  
[local]Redback(config-ldp)#targeted-hello holdtime 60
```

Related Commands

hello holdtime
neighbor targeted
router ldp
targeted-hello interval

targeted-hello interval

targeted-hello interval *seconds*

no targeted-hello interval *seconds*

default targeted-hello interval *seconds*

Purpose

Configures the interval between consecutive LDP targeted Hello messages used in extended LDP discovery.

Command Mode

LDP router configuration

Syntax Description

<i>seconds</i>	Number of seconds between consecutive LDP targeted Hello messages. The range of values is 5 to 3,600.

Default

The default LDP targeted Hello interval is 15 seconds.

Usage Guidelines

Use the **targeted-hello interval** command to configure the interval between consecutive LDP targeted Hello messages used in extended LDP discovery.

If the targeted Hello interval is explicitly configured, then the specified value is used to control targeted Hello interval regardless of the targeted Hello holdtime; however, if the targeted Hello interval is not explicitly configured, the targeted Hello interval used is the negotiated LDP targeted Hello holdtime divided by three. The negotiated LDP targeted Hello holdtime is the lesser of the received LDP targeted Hello holdtime and the locally configured LDP targeted Hello holdtime.

Use the **targeted-hello holdtime** command in LDP router configuration mode to change the locally configured LDP targeted Hello holdtime.

Use the **hello holdtime** command in LDP router configuration mode to change the locally configured LDP link Hello holdtime.

Use the **no** form of this command to use the negotiated LDP targeted Hello holdtime divided by three as the targeted-hello interval.

Use the **default** form of this command to return to the default value of 15 seconds.

Examples

The following example configures a targeted Hello interval of 10 seconds:

```
[local]Redback(config-ctx)#router ldp  
[local]Redback(config-ldp)#targeted-hello interval 10
```

Related Commands

hello holdtime
hello interval
router ldp
targeted-hello holdtime

track-igp-metric

track-igp-metric

no track-igp-metric

Purpose

Enables Label Distribution Protocol (LDP) label-switched paths (LSPs) to inherit the Intermediate System-to-Intermediate System (IS-IS) routing metric for Border Gateway Protocol (BGP) to use when selecting a path.

Command Mode

LDP router configuration

Syntax Description

This command has no keywords or arguments.

Default

By default, inheriting the IS-IS routing metric is disabled.

Usage Guidelines

Use the **track-igp-metric** command to enable LDP LSPs to inherit the IS-IS routing metric for BGP to use when selecting a path.

Use the **no** form of this command to disable LDP LSPs from inheriting the IS-IS metric.

Examples

The following example enables LDP LSPs to inherit the IS-IS routing metric for BGP to use when selecting a path:

```
[local]Redback(config-ctx)#router ldp  
[local]Redback(config-ldp)#track-igp-metric
```

Related Commands

None

transport address

transport address *ip-addr*

Purpose

Configures the transport address advertised in Label Distribution Protocol (LDP) Hello messages.

Command Mode

LDP router configuration

Syntax Description

<i>ip-addr</i>	IP address to be advertised as the transport address. The IP address must be reachable.

Default

The label-switched router (LSR) router ID is used as the transport address.

Usage Guidelines

Use the **transport address** command to configure the transport address advertised in LDP Hello messages. Transport addresses are advertised in LDP Hello messages and are exchanged among LDP neighbors. LDP uses the local transport address as the source, and the received transport address as the destination when trying to establish a Transmission Control Protocol (TCP) connection to a neighbor. Therefore, transport addresses must be reachable. LDP also uses transport addresses to determine which of the two LSRs should perform active open.

If a transport address is not explicitly configured, the LSR router ID is used as the transport address. In this case, the router ID must be reachable; however, if a transport address is explicitly configured, then the specified value is used. In this case, the router ID is not required to be reachable.

Examples

The following example configures a transport address of 20.1.1.1:

```
[local]Redback(config-ctx)#router ldp  
[local]Redback(config-ldp)#transport address 20.1.1.1
```

Related Commands

router ldp

VPLS Configuration

This chapter provides an overview of Virtual Private LAN Services (VPLS) and describes the tasks and commands used to configure VPLS features through the SmartEdge® OS.

For information about the tasks and commands used to monitor, troubleshoot, and administer VPLS, see the “VPLS Operations” chapter in the *Routing Protocols Operations Guide* for the SmartEdge OS.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples
- Command Descriptions

Overview

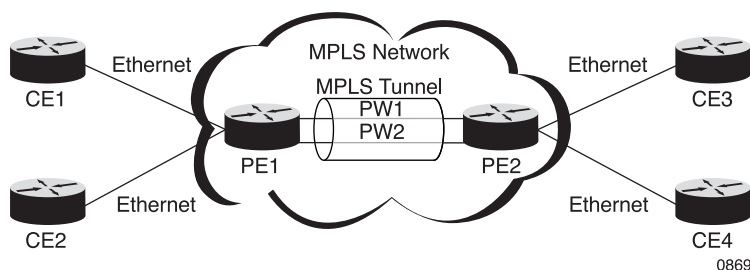
VPLS enables networks at separate geographical locations to communicate with each other across a wide area network (WAN) as if they were directly attached to each other in a LAN. The WAN becomes transparent, which is achieved by creating VPLS pseudo-wires.

A pseudo-wire is a mechanism that emulates the attributes and function of Ethernet connectivity over a WAN. Any required switching functionality or service translation is outside the scope of the pseudo-wire and of the transport network. Pseudo-wires are carried over Multiprotocol Label Switching (MPLS) tunnels on the network.

MPLS signaling protocols are used to automatically provision a service on a pseudo-wire end-to-end, so you can provision a pseudo-wire by pointing to its two endpoints, and MPLS automatically negotiates the path.

Figure 16-1 displays the network topology for a typical VPLS configuration.

Figure 16-1 Typical VPLS Network Topology



Customer edge (CE) routers, which are on the edge of geographically separate customer networks, are connected by Ethernet to provider edge (PE) routers on an MPLS provider network. A pseudo-wire is established for each pair of CE routers that are to be connected into a virtual private LAN. For example, the PW1 pseudo-wire is used to connect the CE1 and CE3 routers, and the PW2 pseudo-wire is used to connect the CE2 and CE4 routers.

To create pseudo-wires, a VPLS-enabled bridge must first be configured on each PE router, and then peering (neighbor) sessions can be established across that bridge. The pseudo-wire is the circuit across which the peering session occurs. A VPLS-enabled bridge can have multiple peering sessions.

Configuration Tasks

Note In this section, the command syntax in the task tables displays only the root command; for the complete command syntax, see the full description for the command in the “Command Descriptions” section.

Before VPLS can be configured, the following conditions must be met:

- MPLS core backbone configuration is up and running.
For more information on configuring MPLS, see Chapter 13, “MPLS Configuration.”
- Label Distribution Protocol (LDP) targeted discovery has been enabled between PE peers.
For more information on configuring LDP targeted discovery, see the “Targeted LDP” section in Chapter 15, “LDP Configuration.”

To configure VPLS, perform the tasks described in the following sections:

- Configuring a Bridge Profile
- Configuring a VPLS Profile
- Configuring a VPLS-Enabled Bridge

Configuring a Bridge Profile

You can assign a named bridge profile to a neighbor. When the subscriber circuit is bound to a bridged interface, the attribute values in the named bridge profile assigned to the neighbor override those in the default bridge profile for the circuit, unless the circuit is also assigned a named bridge profile.

To configure a bridge profile, perform the tasks described in Table 16-1. Enter all commands in bridge profile configuration mode, unless otherwise noted. For more information about the commands used to configure a bridge profile, see the “Bridging Configuration” chapter in the *Ports, Circuits, and Tunnels Configuration Guide* for the SmartEdge OS.

Table 16-1 Configure a Bridge Profile

Task	Root Command	Notes
Create a named or default bridge profile and access bridge profile configuration mode.	bridge profile	Enter this command in global configuration mode.
Set the rate and burst tolerance for broadcast traffic on any VPLS pseudo-wire circuit to which you assign this bridge profile.	broadcast rate-limit	
Specify the maximum number of medium access control (MAC) addresses for the VPLS pseudo-wire circuit to which you assign this bridge profile.	mac-limit	
Set the rate and burst tolerance for multicast traffic on any VPLS pseudo-wire circuit to which you assign this bridge profile.	multicast rate-limit	
Set the rate and burst tolerance for traffic to unknown destinations on any VPLS pseudo-wire circuit to which you assign this bridge profile.	unknown-dest rate-limit	

Configuring a VPLS Profile

A VPLS profile contains one or more neighbors, with each neighbor defining the attributes necessary to establish a separate peer instance (pseudo-wire) to a remote PE device. When a VPLS profile is assigned to a VPLS-enabled bridge, the bridge uses the neighbors in the profile to establish the peer instances and enable bridging over the pseudo-wires.

To configure a VPLS profile (with one or more neighbors), perform the tasks described in Table 16-2. Enter all commands in VPLS profile neighbor configuration mode, unless otherwise noted.

Table 16-2 Configure a VPLS Profile

Task	Root Command	Notes
Create a new VPLS profile, or select an existing one for modification, and enter VPLS profile configuration mode.	vpls profile	Enter this command in global configuration mode. VPSL profiles are used to configure one or more neighbors to which a VPLS instance can establish peering connections. All neighbors configured within a VPLS profile are referenced by the VPLS profile name. The VPLS profile name is unique in the system. The VPLS profile is referenced from the VPLS instance configuration. Multiple VPLS instances can apply (share) the same VPLS profile. If a profile is updated then all instances of its usage use the changed attributes. Conflicts arising due the updated VPLS profile in the VPLS instances does not result in rejecting the VPLS profile or the updates; the individual VPLS instances handle these conditions.
Create a new neighbor, or select an existing one for modification, and enter VPLS profile neighbor configuration mode.	neighbor	Enter this command in VPLS profile configuration mode. The neighbor is identified by the IP address of the remote PE device. It is used along with the pseudo-wire ID from the VPLS instance configuration to establish a pseudo-wire between the local and remote PE devices. Multiple peering sessions (created by VPLS profiles) can be established to the same PE device; different profiles can reference the same remote PE IP address.
Assign an existing named bridge profile to the neighbor.	bridge profile	For more information about this command, see the "Bridging Configuration" chapter in the <i>Ports, Circuits, and Tunnels Configuration Guide</i> for the SmartEdge OS.
Enable circuit statistics for VPLS circuits.	counters	When enabled, packet receive and transmit statistics are collected for each pseudo-wire circuit associated with this neighbor. Use the no form of this command to disable circuit statistics for VPLS circuits.
Associate a description with the neighbor.	description	This command does not affect the neighbor, but is used only as a note in the configuration. The neighbor is identified by the IP address of the remote PE device.
Set the local mode of operation for the neighbor connection.	local-mode	This command applies only if a spoke connection type is configured for the neighbor. With a spoke connection type, one end of the connection must be set to MTU-s mode and the other must be set to PE-rs mode. For proper VPLS operation ensure that the local mode at both ends is set correctly.

Table 16-2 Configure a VPLS Profile (*continued*)

Task	Root Command	Notes
Set the connection type used between the local and remote PE devices.	pe-type	Currently, hub and spoke connection types are supported. For proper VPLS peering, both ends of the peer must be configured with the same connection type.
Specify the pseudo-wire encapsulation type.	pw-encap	Ethernet or Ethernet VLAN encapsulation can be specified.
Enable a neighbor as a standby neighbor for a primary neighbor.	standby-for	A neighbor can serve as a standby for only one primary neighbor. This method of configuring a standby neighbor to reference a primary neighbor allows for establishing the primary and standby pseudo-wires using independent sets of attributes. Before a standby neighbor can be enabled, the following conditions must be met: • A spoke connection type must be set for the neighbor. • Local mode must be set to MTU-s. • No other standby neighbor in the VPLS profile can reference the same primary neighbor IP address.

Configuring a VPLS-Enabled Bridge

A VPLS-enabled bridge is used to establish peer instances to neighbors.

To configure a VPLS-enabled bridge, perform the tasks described in Table 16-3. Enter all commands in VPLS configuration mode, unless otherwise noted.

Table 16-3 Configure a VPLS-Enabled Bridge

Task	Root Command	Notes
Create a bridge or select one for modification and enter bridge configuration mode.	bridge	Enter this command in context configuration mode. For more information about this command, see the "Bridging Configuration" chapter in the <i>Ports, Circuits, and Tunnels Configuration Guide</i> for the SmartEdge OS.
Enable VPLS on a bridge and enter VPLS configuration mode.	vpls	Enter this command in bridge configuration mode.
Disable the operation of an enabled VPLS instance.	disable	If the VPLS instance has been disabled, you can use the no form of this command to enable it.
Apply an existing VPLS profile to a VPLS instance.	profile	When a VPLS profile is applied, a VPLS peer instance is created for each neighbor defined in the profile, and a pseudo-wire connection is established using the attributes defined for the neighbor. A VPLS profile must be configured using the vpls profile command (in global configuration mode) before it can be applied. Multiple VPLS profiles can be applied to the same VPLS instance. If two or more profiles reference the same neighbor (same IP address), then the neighbor from the first profile is used. The same profile cannot be applied multiple times even if the pseudo-wire IDs are different.

Table 16-3 Configure a VPLS-Enabled Bridge (*continued*)

Task	Root Command	Notes
Configure a default pseudo-wire number for use with all the pseudo-wires signaled by the VPLS instance.	pw-id	The default pseudo-wire number is used for VPLS profiles that do not have a pseudo-wire ID (number or name) specified. Remote PE devices use the pseudo-wire ID and the local IP address to identify the pseudo-wire and the associated VPLS instance. A VPLS instance can have only one default pseudo-wire ID, either a number or a name. If a default pseudo-wire ID (name or number) has been configured for a VPLS instance and a new one is configured, the previous pseudo-wire ID is replaced with the new one.
Configure a default pseudo-wire name for use with all the pseudo-wires signaled by the VPLS instance.	pw-name	The default pseudo-wire name is used for VPLS profiles that do not have a pseudo-wire ID (number or name) specified. Remote PE devices use the pseudo-wire ID and the local IP address to identify the pseudo-wire and the associated VPLS instance. A VPLS instance can have only one default pseudo-wire ID, either a number or a name. If a default pseudo-wire ID (name or number) has been configured for a VPLS instance and a new one is configured, the previous pseudo-wire ID is replaced with the new one.

Configuration Examples

The VPLS configuration examples assume that the following conditions are true:

- MPLS core backbone configuration is up and running.

For more information on configuring MPLS, see Chapter 13, “MPLS Configuration.”

- LDP targeted discovery has been enabled between PE peers.

For more information on configuring LDP targeted discovery, see the “Targeted LDP” section in Chapter 15, “LDP Configuration.”

The following configuration example creates a VPLS bridge to two VPLS neighbors. This configuration is broken down into the following sections:

- Bridge Profile
- VPLS Profile
- VPLS-Enabled Bridge

Bridge Profile

The following configuration example creates two bridge profiles, **100Mbps-bc** and **120Mbps-mc**. The **100Mbps-bc** bridge profile sets a rate limit of 125 Mbps (12,500 kbps) for broadcast traffic on the VPLS pseudo-wire circuit to which this bridge profile is assigned. The **120Mbps-mc** bridge profile sets a rate limit of 150 Mbps (15,000 kbps) for multicast traffic on the VPLS pseudo-wire circuit to which this bridge profile is assigned. The attributes of these bridge profiles will be applied to VPLS neighbor configurations.

```
[local]Redback#config
[local]Redback(config)#bridge profile 100Mbps-bc
[local]Redback(config-bridge-profile)#broadcast rate-limit 12500000
[local]Redback(config-bridge-profile)#exit
[local]Redback(config)#bridge profile 120Mbps-mc
[local]Redback(config-bridge-profile)#multicast rate-limit 15000000
[local]Redback(config-bridge-profile)#end
```

VPLS Profile

The following configuration example creates a VPLS profile, **vprofile1**, and two neighbors, **64.10.192.112** and **110.32.164.5**. The attributes from the bridge profile, **100Mbps-bc**, are applied to the neighbor given the description, **dallas-to-nyc**. The attributes from the bridge profile, **120Mbps-mc**, are applied to the neighbor given the description, **dallas-to-sfo**. The neighbor attributes in this bridge profile will be applied to VPLS-enabled bridge instance.

```
[local]Redback#config
[local]Redback(config)#vpls profile vprofile1
[local]Redback(config-vpls-profile)#neighbor 64.10.192.112
[local]Redback(config-vpls-profile-neighbor)#description dallas-to-nyc
[local]Redback(config-vpls-profile-neighbor)#bridge-profile 100Mbps-bc
[local]Redback(config-vpls-profile-neighbor)#exit
[local]Redback(config-vpls-profile)#neighbor 110.32.164.5
[local]Redback(config-vpls-profile-neighbor)#description dallas-to-sfo
[local]Redback(config-vpls-profile-neighbor)#bridge-profile 120Mbps-mc
[local]Redback(config-vpls-profile-neighbor)#end
```

VPLS-Enabled Bridge

The following configuration example creates a VPLS-enabled bridge instance, **truecom.net**, configures a default pseudo-wire number, **100**, for this instance, and applies the attributes from the VPLS profile, **vprofile1**, to this instance.

```
[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config-ctx)#bridge truecom.net
[local]Redback(config-bridge)#vpls
[local]Redback(config-vpls)#pw-id 100
[local]Redback(config-vpls)#profile vprofile1
[local]Redback(config-vpls)#end
```

Command Descriptions

This section describes the syntax and usage guidelines for the commands used to configure VPLS features. The commands are presented in alphabetical order.

counters	pw-encap
description	pw-id
disable	pw-name
local-mode	standby-for
neighbor	vpls
pe-type	vpls profile
profile	

counters

counters

no counters

Purpose

Enables circuit statistics for Virtual Private LAN Services (VPLS) circuits.

Command Mode

VPLS profile neighbor configuration

Syntax Description

This command has no keywords or arguments.

Default

VPLS pseudo-wire circuit counters are disabled.

Usage Guidelines

Use the **counters** command to enable circuit statistics for VPLS circuits.

When enabled, packet receive and transmit statistics are collected for each pseudo-wire circuit associated with this neighbor.

Use the **show circuit counters vpls** command (in any mode) to display packet counter information for VPLS circuits. For more information about the **show circuit counters vpls** command, see the “VPLS Operations” chapter in the *Routing Protocols Operations Guide* for the SmartEdge OS.

Use the **no** form of this command to disable circuit statistics for VPLS circuits.

Examples

The following example enables circuit statistics for VPLS circuits:

```
[local]Redback#config
[local]Redback(config)#vpls profile foo
[local]Redback(config-vpls-profile)#neighbor 10.10.10.1
[local]Redback(config-vpls-profile-neighbor)#counters
[local]Redback(config-vpls-profile-neighbor)#
```

Related Commands

description
local-mode
neighbor
pe-type
standby-for

description

description *text*

no description

Purpose

Associates a description with a neighbor.

Command Mode

VPLS profile neighbor configuration

Syntax Description

text Description of the neighbor (63 characters maximum).

Default

None

Usage Guidelines

Use the **description** command to associate a description with a neighbor. This command does not affect the neighbor, but is used only as a note in the configuration.

Note The neighbor is identified by the IP address of the remote PE device.

Use the **no** form of this command to remove a description from the neighbor. Because there can be only one description for a neighbor, when you use the **no** form of this command, it is not necessary to include the *text* argument.

Examples

The following example provides the description, `test-peer`, for the neighbor, `10.10.10.1`:

```
[local]Redback#config
[local]Redback(config)#vpls profile foo
[local]Redback(config-vpls-profile)#neighbor 10.10.10.1
[local]Redback(config-vpls-profile-neighbor)#description test-peer
[local]Redback(config-vpls-profile-neighbor)#
```

Related Commands

counters
local-mode
neighbor
pe-type
standby-for

disable

disable
no disable

Purpose

Disables the operation of an enabled Virtual Private LAN Services (VPLS) instance.

Command Mode

VPLS configuration

Syntax Description

This command has no keywords or arguments.

Default

VPLS instances are enabled.

Usage Guidelines

Use the **disable** command to disable the operation of an enabled VPLS instance. When the VPLS instance is disabled, the following actions occur:

- The bridge continues to learn medium access control (MAC) addresses and forwards traffic on all the associated bridge circuits.
- All pseudo-circuits associated with the pseudo-wires are marked down.

Use the **no** form of this command to enable a previously disabled VPLS instance.

Examples

The following example disables the VPLS instance on the to-pe4 bridge:

```
[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config-ctx)#bridge to-pe4
[local]Redback(config-bridge)#vpls
[local]Redback(config-vpls)#disable
[local]Redback(config-vpls)#
```

The following example enables the previously disabled VPLS instance on the to-pe4 bridge:

```
[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config-ctx)#bridge to-pe4
[local]Redback(config-bridge)#vpls
[local]Redback(config-vpls)#no disable
[local]Redback(config-vpls)#
```

Related Commands

profile
pw-id
pw-name
vpls

local-mode

local-mode {mtu-s | pe-rs}
{no | default} **local-mode**

Purpose

Sets the local mode of operation for the neighbor connection.

Command Mode

VPLS profile neighbor configuration

Syntax Description

mtu-s	Sets the local mode to multitenant unit switch (MTU-s). This mode is used when the local router is participating in hierarchical Virtual Private LAN Services (VPLS) by using a pseudo-wire connected to a core provider edge routers (PE-rs) device, and when the local VPLS instance does not have a mesh of pseudo-wire to all the core PE devices.
pe-rs	Sets the local mode to PE-rs. This mode is used at a core VPLS PE device that is providing hierarchical VPLS connectivity to other MTU-s routers.

Default

The PE-rs mode is set.

Usage Guidelines

Use the **local-mode** command to set the local mode of operation for the neighbor connection. This command applies only if a spoke connection type is configured for the neighbor. With a spoke connection type, one end of the connection must be set to MTU-s mode and the other must be set to PE-rs mode.

Note For proper VPLS operation, ensure that the local mode at both ends is set correctly.

Use the **no** or **default** form of this command to return the local mode of operation to PE-rs.

Examples

The following example sets the local mode to mtu-s:

```
[local]Redback#config
[local]Redback(config)#vpls profile foo
[local]Redback(config-vpls-profile)#neighbor 10.10.10.1
[local]Redback(config-vpls-profile-neighbor)#local-mode mtu-s
[local]Redback(config-vpls-profile-neighbor)#
```

Related Commands

counters	pe-type
description	standby-for
neighbor	

neighbor

neighbor *ip-addr*

{**no** | **default**} **neighbor** *ip-addr*

Purpose

Creates a new neighbor, or selects an existing one for modification, and enters Virtual Private LAN Services (VPLS) profile neighbor configuration mode.

Command Mode

VPLS profile configuration

Syntax Description

ip-addr Neighbor IP address, in the form *A.B.C.D*.

Default

None

Usage Guidelines

Use the **neighbor** command to create a new neighbor, or select an existing one for modification, and enter VPLS profile neighbor configuration mode.

The neighbor is identified by the IP address of the remote provider edge (PE) device. It is used along with the pseudo-wire ID from the VPLS instance configuration to establish a pseudo-wire between the local and remote PE devices. Multiple peering sessions (created by VPLS profiles) can be established to the same PE device; different profiles can reference the same remote PE IP address.

Use the **no** or **default** form of this command to remove a configured neighbor.

Examples

The following example creates a new VPLS neighbor with the IP address, 10.10.10.1:

```
[local]Redback#config
[local]Redback(config)#vpls profile foo
[local]Redback(config-vpls-profile)#neighbor 10.10.10.1
[local]Redback(config-vpls-profile-neighbor)#
```

Related Commands

counters
description
local-mode
pe-type
standby-for
vpls profile

pe-type

pe-type {hub | spoke}
{no | default} pe-type

Purpose

Specifies the connection type used between the local and remote provider edge (PE) devices.

Command Mode

VPLS profile neighbor configuration

Syntax Description

hub	Hub connection type. This connection type is used if the Virtual Private LAN Services (VPLS) topology is enabled using a full mesh of pseudo-wire. Packets received on a hub link pseudo-wire are not forwarded on other hub connections (split horizon).
spoke	Spoke connection type. This connection type is used for enabling hierarchical VPLS topologies between multitenant unit switch (MTU-s) and PE routers (PE-rs), or when a full mesh of pseudo-wires is not used. Forwarding is unrestricted on spoke links.

Default

The hub connection type is used.

Usage Guidelines

Use the **pe-type** command to specify the connection type used between the local and remote PE devices. Currently, hub and spoke connection types are supported. For proper VPLS peering, both ends of the peer must be configured with the same connection type.

Use the **no** or **default** form of this command to specify the default connection type.

Examples

The following example sets the connection type to spoke:

```
[local]Redback#config
[local]Redback(config)#vpls profile foo
[local]Redback(config-vpls-profile)#neighbor 10.10.10.1
[local]Redback(config-vpls-profile-neighbor)#pe-type spoke
[local]Redback(config-vpls-profile-neighbor)#
```

Related Commands

counters
description
local-mode
neighbor
standby-for

profile

profile *prof-name* [**pw-id** *pw-num* | **pw-name** *pw-name*]

no profile *prof-name*

Purpose

Applies an existing Virtual Private LAN Services (VPLS) profile to a VPLS instance.

Command Mode

VPLS configuration

Syntax Description

<i>prof-name</i>	Name of the VPLS profile that contains the neighbor attributes for establishing the pseudo-wires (maximum 40 characters).
pw-id <i>pw-num</i>	Optional. Pseudo-wire number. The value of the <i>pw-num</i> argument is a 4-byte number. The remote provider edge (PE) device uses the pseudo-wire number and the local IP address to identify the pseudo-wire and the associated VPLS instance.
pw-name <i>pw-name</i>	Optional. Pseudo-wire name. The remote PE device uses the pseudo-wire name and the local IP address to identify the pseudo-wire and the associated VPLS instance.

Default

None

Usage Guidelines

Use the **profile** command to apply an existing VPLS profile to a VPLS instance. When a VPLS profile is applied, a VPLS peer instance is created for each neighbor defined in the profile, and a pseudo-wire connection is established using the attributes defined for the neighbor.

A VPLS profile must be configured using the **vpls profile** command (in global configuration mode) before it can be applied.

Use the **pw-id** *pw-num* construct or **pw-name** *pw-name* construct to optionally specify a pseudo-wire ID (number or name) to signal the ID for pseudo-wires to the neighbor defined in the profile. If a pseudo-wire ID is not configured for a VPLS profile, then the VPLS instance-level default pseudo-wire ID is used.

Multiple VPLS profiles can be applied to the same VPLS instance. If two or more profiles reference the same PE (same IP address), then the neighbor from the first profile is used. The same profile cannot be applied multiple times even if the pseudo-wire IDs are different.

Use the **no** form of this command to delete a VPLS profile.

Examples

The following example applies the `foo` VPLS profile to the VPLS instance on the `to-pe4` bridge:

```
[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config-ctx)#bridge to-pe4
[local]Redback(config-bridge)#vpls
[local]Redback(config-vpls)#profile foo pw-id 20
[local]Redback(config-vpls)#
```

Related Commands

- `disable`
- `pw-id`
- `pw-name`
- `vpls`
- `vpls profile`

pw-encap

pw-encap {**ether** | **vlan**}
{**no** | **default**} **pw-encap**

Purpose

Specifies the pseudo-wire encapsulation type.

Command Mode

VPLS profile neighbor configuration

Syntax Description

ether	Specifies the encapsulation type as Ethernet encapsulation.
vlan	Specifies the encapsulation type as Ethernet virtual LAN (VLAN) encapsulation.

Default

The default pseudo-wire encapsulation type is Ethernet encapsulation.

Usage Guidelines

Use the **pw-encap** command to specify the pseudo-wire encapsulation type.

Use the **no** or **default** form of this command to specify the default encapsulation type.

Examples

The following example specifies the pseudo-wire encapsulation type as Ethernet VLAN encapsulation:

```
[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config-ctx)#bridge to-pe4
[local]Redback(config-bridge)#vpls
[local]Redback(config-vpls)#pw-id 1234
[local]Redback(config-vpls)#
```

pw-id

pw-id *pw-num*

no pw-id *pw-num*

Purpose

Configures a default pseudo-wire number for use with all the pseudo-wires signaled by the Virtual Private LAN Services (VPLS) instance.

Command Mode

VPLS configuration

Syntax Description

<i>pw-num</i>	Default pseudo-wire number, used to identify the pseudo-wire endpoints when signaling using Label Distribution Protocol (LDP). Valid values are 1 to 4,294,967,295.

Default

None

Usage Guidelines

Use the **pw-id** command to configure a default pseudo-wire number for use with all the pseudo-wires signaled by the VPLS instance. The default pseudo-wire number is used for VPLS profiles that do not have a pseudo-wire ID (number or name) specified.

Remote provider edge (PE) devices use the pseudo-wire ID and the local IP address to identify the pseudo-wire and the associated VPLS instance.

A VPLS instance can have only one default pseudo-wire ID, either a number or a name. If a default pseudo-wire ID (name or number) has been configured for a VPLS instance and a new one is configured, the previous pseudo-wire ID is replaced with the new one.

Use the **no** form of this command to remove the default pseudo-wire number.

Examples

The following example configures the default pseudo-wire number, 1234, for use with all the pseudo-wires signaled by the VPLS instance:

```
[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config-ctx)#bridge to-pe4
[local]Redback(config-bridge)#vpls
[local]Redback(config-vpls)#pw-id 1234
[local]Redback(config-vpls)#
```

Related Commands

disable
profile
pw-name
vpls

pw-name

pw-name *pw-name*

no pw-name *pw-name*

Purpose

Configures a default pseudo-wire name for use with all the pseudo-wires signaled by the Virtual Private LAN Services (VPLS) instance.

Command Mode

VPLS configuration

Syntax Description

<i>pw-name</i>	Name of the default pseudo-wire, used to identify the pseudo-wire endpoints when signaling using Label Distribution Protocol (LDP).

Default

None

Usage Guidelines

Use the **pw-name** command to configure a default pseudo-wire name for use with all the pseudo-wires signaled by the VPLS instance. The default pseudo-wire name is used for VPLS profiles that do not have a pseudo-wire ID (number or name) specified.

Remote provider edge (PE) devices use the pseudo-wire ID and the local IP address to identify the pseudo-wire and the associated VPLS instance.

A VPLS instance can have only one default pseudo-wire ID, either a number or a name. If a default pseudo-wire ID (name or number) has been configured for a VPLS instance and a new one is configured, the previous pseudo-wire ID is replaced with the new one.

Use the **no** form of this command to remove the default pseudo-wire name.

Examples

The following example configures the default pseudo-wire name, `pw-foo`, for use with all the pseudo-wires signaled by the VPLS instance:

```
[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config-ctx)#bridge to-pe4
[local]Redback(config-bridge)#vpls
[local]Redback(config-vpls)#pw-name pw-foo
[local]Redback(config-vpls)#
```


Related Commands

disable
profile
pw-id
vpls

standby-for

standby-for *ip-addr*

{**no** | **default**} **standby-for**

Purpose

Enables a neighbor as a standby neighbor for a primary neighbor.

Command Mode

VPLS profile neighbor configuration

Syntax Description

ip-addr

IP address, in the form *A.B.C.D*, of the primary neighbor for which the standby neighbor is being configured.

Default

No standby neighbor is configured.

Usage Guidelines

Use the **standby-for** command to enable a neighbor as a standby neighbor for a primary neighbor. A neighbor can serve as a standby for only one primary neighbor. This method of configuring a standby neighbor to reference a primary neighbor allows for establishing the primary and standby pseudo-wires using independent sets of attributes.

Before a standby neighbor can be enabled, the following conditions must be met:

- A primary neighbor must be configured in the same profile.
- A spoke connection type must be set for the neighbor.
- Local mode must be set to multitenant unit switch (MTU-s).
- No other standby neighbor in the Virtual Private LAN Services (VPLS) profile can reference the same primary neighbor IP address.

Use the **no** or **default** form of this command to disable a neighbor from being a standby neighbor for a primary neighbor.

Examples

The following example creates a standby neighbor, 10.10.10.1, for the primary neighbor, 20.20.5.5:

```
[local]Redback#config
[local]Redback(config)#vpls profile foo
[local]Redback(config-vpls-profile)#neighbor 10.10.10.1
[local]Redback(config-vpls-profile-neighbor)#standby-for 20.20.5.5
[local]Redback(config-vpls-profile-neighbor)#
```

Related Commands

<code>counters</code>	<code>neighbor</code>
<code>description</code>	<code>pe-type</code>
<code>local-mode</code>	

vpls

vpls
no vpls

Purpose

Enables Virtual Private LAN Services (VPLS) on a bridge and enters VPLS configuration mode.

Command Mode

bridge configuration

Syntax Description

This command has no keywords or arguments.

Default

VPLS is not enabled on the bridge.

Usage Guidelines

Use the **vpls** command to enable VPLS on a bridge and enter VPLS configuration mode.

Use the **no** form of this command to disable VPLS on the bridge.

Examples

The following example enables VPLS on the to-pe4 bridge and enter VPLS configuration mode:

```
[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config-ctx)#bridge to-pe4
[local]Redback(config-bridge)#vpls
[local]Redback(config-vpls)#
```

Related Commands

disable
profile
pw-id
pw-name

vpls profile

vpls profile *prof-name*

no vpls profile *prof-name*

Purpose

Creates a new Virtual Private LAN Services (VPLS) profile, or selects an existing one for modification, and enters VPLS profile configuration mode.

Command Mode

global configuration

Syntax Description

prof-name Name of the VPLS profile (maximum of 40 characters).

Default

None

Usage Guidelines

Use the **vpls profile** command to create a new VPLS profile, or select an existing one for modification, and enter VPLS profile configuration mode. VPLS profiles are used to configure one or more neighbors to which a VPLS instance can establish peering connections. All neighbors configured within a VPLS profile are referenced by the VPLS profile name, which is unique in the system.

The VPLS profile is referenced from the VPLS instance configuration. Multiple VPLS instances can apply (share) the same VPLS profile. If a profile is updated, then all instances of its usage use the changed attributes. Conflicts arising, due to the updated VPLS profile in the VPLS instances, do not result in rejecting the VPLS profile or the updates; the individual VPLS instances handle these conditions.

Examples

The following example creates the foo VPLS profile and enter VPLS profile configuration mode:

```
[local]Redback#config
[local]Redback(config)#vpls profile foo
[local]Redback(config-vpls-profile)#
```

Related Commands

neighbor
profile

A

- ABR (area border router), 6-4
- access control list configuration mode, described, 1-9
- ACL condition configuration mode, described, 1-9
- address families
 - IPv4
 - BGP instances, 8-9
 - BGP neighbors, 8-14
 - BGP peer groups, 8-18
 - IPv6
 - BGP instances, 8-10
 - BGP neighbors, 8-15
 - BGP peer groups, 8-18
 - IS-IS instances, 10-3
 - IS-IS interfaces, 10-7
- administrative distance
 - RIP, 5-2
 - RIPng, 5-4
- advertise interval
 - VRRP backup router, 4-3
 - VRRP owner router, 4-2
- aggregate addresses
 - IPv4, 8-9
 - IPv6, 8-10
- anycast RP
 - described, 11-5
 - enabling, 11-10
- areas, OSPF
 - backbone, 6-3
 - normal, 6-3
 - stub, 6-3
- area type
 - OSPF, 6-10
 - OSPFv3, 6-15
- AS (autonomous system)
 - BGP, 8-3
 - local, BGP neighbors, 8-12
 - OSPF, 6-3
 - remote, BGP neighbors, 8-13

- ASBR (autonomous system boundary router), 6-4
- ASNs (autonomous system numbers)
 - BGP neighbors
 - IPv4, 8-14
 - IPv6, 8-15
 - BGP peer groups
 - IPv4, 8-18
 - IPv6, 8-18
- AS path list configuration mode, described, 1-9
- AS path lists
 - BGP neighbors
 - IPv4, 8-14
 - IPv6, 8-15
 - BGP peer groups
 - IPv4, 8-18
 - IPv6, 8-18
 - configuration examples
 - complex, 12-13
 - simple, 12-13
 - creating, 12-2
 - described, 12-2
 - matching, 12-8
 - permit or deny, 12-2
 - resequence, 12-3
- AS paths
 - detecting loops, 9-10
 - overriding attributes, 9-10
- ATM DS-3 configuration mode, described, 1-9
- ATM OC configuration mode, described, 1-9
- ATM PVC configuration mode, described, 1-9
- attached bits, 10-5
- attribute-based accounting
 - configuration example, 12-15
 - enabling, 12-11
 - table maps, 12-11
 - traffic index values, 12-11
- attributes
 - route target, 9-4
 - site of origin, 9-4

- AU-3 configuration mode, described, 1-9
- audience, for this guide, xxiii
- authentication
 - IS-IS, 10-4
 - OSPF interface, 6-11
 - RIP, 5-3
 - sham link, 6-12
 - virtual link, 6-13
 - VRRP backup router, 4-3
 - VRRP owner router, 4-2
- auto cost
 - OSPF, 6-8
 - OSPFv3, 6-14

B

- backbone
 - OSPF areas, 6-3
 - routers, 6-4
- backup
 - designated router, 6-4
 - router, 4-3
- basic IP routing
 - commands, described, 2-6
 - configuration examples, 2-5
 - configuration tasks
 - additional parameters, 2-5
 - static IP routes, 2-4
 - intercontext static routes, 2-5
 - martian addresses, 2-5
 - maximum routes, 2-5
 - MTU, 2-5
 - multicast RPF, 2-5
 - overview, 2-1
 - router identifier, 2-5
 - route selection process, 2-3
 - verify RPF, 2-5
- BFD (Bidirectional Forwarding Detection)
 - commands, described, 7-5
 - configuration examples
 - BFD interface, 7-5
 - BFD neighbor, 7-4
 - disabling BFD, 7-5
 - configuration tasks
 - BFD interface, 7-3
 - BFD neighbor, 7-2
 - disabling BFD, 7-4
 - enabling BFD, 7-4
 - instances, 7-2
 - interfaces
 - creating, 7-3
 - detection multiplier, 7-3
 - receive interval, minimum, 7-3
 - transmit interval, minimum, 7-3

- neighbors
 - creating, 7-2
 - detection multiplier, 7-2
 - receive interval, minimum, 7-2
 - transmit interval, minimum, 7-2
- overview, 7-1
- BFD interface configuration mode, described, 1-9
- BFD neighbor configuration mode, described, 1-9
- BFD router configuration mode, described, 1-9
- BGP (Border Gateway Protocol)
 - AS path lists
 - configuration example, complex, 12-13
 - configuration example, simple, 12-13
 - creating, 12-2
 - described, 12-2
 - matching, 12-8
 - permit or deny, 12-2
 - resequence, 12-3
 - attribute-based accounting
 - configuration example, 12-15
 - enabling, 12-11
 - table maps, 12-11
 - traffic index values, 12-11
 - commands, described, 8-25
 - community lists
 - configuration example, complex, 12-14
 - configuration example, simple, 12-14
 - creating, 12-3
 - described, 12-3
 - matching, 12-9
 - permit or deny, 12-4
 - resequence, 12-4
 - confederations, 8-5
 - configuration examples
 - eMBGP peer configuration, 8-22
 - eMBGP peer groups configuration, 8-23
 - iMBGP peer configuration, 8-20
 - iMBGP peer groups configuration, 8-21
 - minimum configuration, 8-19
 - destination-based QoS
 - configuration example, 12-16
 - DSCP destinations, 12-11
 - DSCP values, 12-11
 - table maps, 12-11
 - extended community lists
 - creating, 12-5
 - described, 12-5
 - matching, 12-9
 - permit or deny, 12-5
 - resequence, 12-5
 - graceful restart
 - maximum update delays, instances, 8-11
 - restart time, neighbors, 8-16
 - restart times, instances, 8-11

- retain routes, 8-16
- retain times, instances, 8-11
- retain times, neighbors, 8-16
- instances
 - aggregate addresses, IPv4, 8-9
 - aggregate addresses, IPv6, 8-10
 - client-to-client route reflectors, 8-11
 - cluster IDs, 8-11
 - comparing MED paths, 8-8
 - confederation IDs, 8-11
 - confederation peers, 8-11
 - creating, 8-8
 - dampening, IPv4, 8-9
 - dampening, IPv6, 8-10
 - distance, IPv4, 8-9
 - distance, IPv6, 8-10
 - fast resets, 8-8
 - flap statistics, IPv4, 8-9
 - flap statistics, IPv6, 8-10
 - holdtimes, 8-9
 - IPv4 address families, 8-9
 - IPv6 address families, 8-10
 - keepalive, 8-9
 - local preferences, 8-9
 - logging neighbors resets, 8-9
 - multipath load balancing, 8-9
 - networks, IPv4, 8-10
 - networks, IPv6, 8-10
 - redistributing routes, IPv4, 8-10
 - redistributing routes, IPv6, 8-10
 - router IDs, 8-9
 - timers, 8-9
 - traffic index counters, 8-10
- neighbors
 - advertisement intervals, 8-12
 - ASNs, IPv4 address families, 8-14
 - ASNs, IPv6 address families, 8-15
 - AS path lists, IPv4, 8-14
 - AS path lists, IPv6, 8-15
 - community attributes, 8-13
 - creating, 8-12
 - default routes, IPv4, 8-14
 - default routes, IPv6, 8-15
 - described, 8-12
 - enforcing TTLs, 8-12
 - filters, 8-12
 - holdtimes, 8-13
 - IPv4 address families, 8-14
 - IPv6 address families, 8-15
 - keepalive, 8-13
 - local AS, 8-12
 - logging resets, 8-9
 - maximum prefixes, IPv4, 8-14
 - maximum prefixes, IPv6, 8-15
 - MPLS labels, 8-13
 - multihops, 8-12
 - next hops, 8-12
 - passwords, 8-12
 - peer group, 8-13
 - peer groups, IPv4, 8-14
 - peer groups, IPv6, 8-15
 - prefix lists, IPv4, 8-14
 - prefix lists, IPv6, 8-15
 - remote AS, 8-13
 - route maps, IPv4, 8-14
 - route maps, IPv6, 8-15
 - route reflectors, IPv4, 8-14
 - route reflectors, IPv6, 8-15
 - shutdown, 8-13
 - timer password, 8-13
 - timers, 8-13
 - update source, 8-13
- peer groups
 - advertisement intervals, 8-16
 - applying attributes, 8-19
 - ASNs, IPv4 address families, 8-18
 - ASNs, IPv6 address families, 8-18
 - AS path lists, IPv4, 8-18
 - AS path lists, IPv6, 8-18
 - community attributes, 8-17
 - creating, 8-16
 - dampening sessions, 8-17
 - default routes, IPv4, 8-18
 - default routes, IPv6, 8-18
 - description, 8-17
 - enforcing TTL, 8-17
 - holdtime, 8-17
 - IPv4 address families, 8-18
 - IPv6 address families, 8-18
 - keepalive, 8-17
 - maximum prefixes, IPv4, 8-18
 - maximum prefixes, IPv6, 8-18
 - multihops, 8-17
 - next hops, 8-17
 - passwords, 8-17
 - prefix lists, IPv4, 8-18
 - prefix lists, IPv6, 8-18
 - route maps, IPv4, 8-18
 - route maps, IPv6, 8-19
 - route reflectors, IPv4, 8-18
 - route reflectors, IPv6, 8-19
 - shutdown, 8-17
 - timers, 8-17
 - update source, 8-17
- route aggregation, 8-6

- route reflectors
 - client-to-client, 8-11
 - cluster IDs, 8-11
 - defined, 8-4
 - supported IETF drafts and RFCs, 8-1
- BGP/MPLS VPN over GRE, 9-4
- BGP/MPLS VPNs (Border Gateway Protocol/Multiprotocol Label Switching Virtual Private Networks)
 - address families
 - BGP routing instances, 9-7
 - enabling, 9-7
 - configuration examples
 - AS path loop detection, 9-31
 - AS path override, 9-31
 - backbone connectivity, 9-11
 - BGP/MPLS VPN over GRE, 9-28
 - GRE over MPLS, 9-26
 - hub-and-spoke, 9-22
 - local import, 9-19
 - route origin, 9-33
 - typical configuration, 9-16
 - VPN using eBGP, 9-15
 - VPN using OSPF, 9-14
 - VPN using RIP, 9-14
 - VPN using static routing, 9-13
 - multipath load balancing, 9-8
 - next-hop reachability check, 9-9
 - PE-to-CE routes
 - AS path loops, 9-10
 - OSPF instances, 9-10
 - overriding AS path attributes, 9-10
 - route origins, 9-10
 - route targets
 - exporting, 9-9
 - filtering, 9-9
 - importing, 9-9
 - soft GRE tunnels, 9-11
 - VPN contexts
 - BGP instances, 9-8
 - creating, 9-7
 - servicing multiple contexts, 9-7
- BGP address family configuration mode, described, 1-9
- BGP neighbor address family configuration mode, described, 1-9
- BGP neighbor configuration mode, described, 1-9
- BGP peer group address family configuration mode, described, 1-9
- BGP peer group configuration mode, described, 1-9
- BGP router configuration mode, described, 1-9
- block flooding
 - IS-IS, 10-9
 - OSPF, 6-11
 - OSPFv3, 6-16
- bridge configuration mode, described, 1-9

- bridge profile configuration mode, described, 1-9
- BSR (bootstrap router)
 - border, 11-10
 - candidate, 11-11
 - described, 11-4

C

- CE (customer edge)
 - PE-to-CE route distribution, 9-3
 - routers, 9-2
- characters, in command syntax, xxv
- checksums, optional, 10-7
- CIDR (Classless InterDomain Routing), 8-6
- circuit MTUs, 10-7
- circuit types, IS-IS, 10-7
- CLI (command-line interface) syntax, 1-9
- command modes, access commands and prompts, 1-9
- command modes, conventions for, xxiv
- command privilege, conventions for, xxiv
- command syntax
 - conventions, xxiv
 - special characters, xxv
 - terminology, xxiv
 - text formats, xxv
- community list configuration mode, described, 1-9
- community lists
 - configuration examples
 - complex, 12-14
 - simple, 12-14
 - creating, 12-3
 - described, 12-3
 - matching, 12-9
 - permit or deny, 12-4
 - resequence, 12-4
- confederations
 - described, 8-5
 - IDs, 8-11
 - peers, 8-11
- context configuration mode, described, 1-9
- conventions, used in this guide, xxiv
 - command modes, xxiv
 - command privilege, xxiv
- cost
 - OSPF interface, 6-11
 - OSPFv3 interface, 6-16
 - RIP interface, 5-3
 - RIPng interface, 5-5
 - sham link, 6-12
 - static routes, 2-4
- CSNP (complete sequence number protocol data unit)
 - intervals, 10-7
 - on P2P interfaces, 10-7

D

- dampening
 - IPv4, 8-9
 - IPv6, 8-10
- database description packets, 6-4
- default metric
 - OSPF, 6-8
 - OSPFv3, 6-14
 - RIP, 5-2
 - RIPng, 5-4
- default routes
 - BGP neighbors
 - IPv4, 8-14
 - IPv6, 8-15
 - BGP peer groups
 - IPv4, 8-18
 - IPv6, 8-18
 - described, 6-3
 - originating
 - RIP instances, 5-2
 - RIP interfaces, 5-3
 - RIPng instances, 5-4
 - RIPng interfaces, 5-4
 - OSPF, 6-10
 - OSPFv3, 6-15
- demand circuit
 - OSPF, 6-11
 - OSPFv3, 6-16
- dense mode
 - described, 11-3
 - enabling, 11-10
- designated router, 6-4
- destination-based QoS
 - configuration example, 12-16
 - DSCP destinations, 12-11
 - DSCP values, 12-11
 - table maps, 12-11
- detection multiplier
 - BFD interfaces, 7-3
 - BFD neighbors, 7-2
- distance
 - BGP IPv4, 8-9
 - BGP IPv6, 8-10
 - DVSR, 3-3
 - IS-IS, 10-4
 - OSPF, 6-9
 - OSPFv3, 6-14
 - RIP, 5-2
 - RIPng, 5-4
- distribution list
 - RIP, 5-2
 - RIPng, 5-4
- dot1q PVC configuration mode, described, 1-9

- DR (designated router)
 - described, 11-2
 - priority, 11-11
- DS-0 configuration mode, described, 1-9
- DS-1 configuration mode, described, 1-9
- DS-3 configuration mode, described, 1-9
- DVSR (dynamically verified static routing)
 - commands, described, 3-6
 - configuration examples
 - anycast, 3-4
 - customer multihoming, 3-5
 - minimum configuration, 3-3
 - configuration tasks, 3-2
 - distance value, 3-3
 - overview, 3-1
 - profile, 3-3
 - source IP address, 3-3
 - tag value, 3-3
 - TTL value, 3-3
 - verify-set values, 3-3
- DVSR profile configuration mode, described, 1-9
- dynamic hostnames, 10-4
- dynamic routing, 1-2

E

- E1 configuration mode, described, 1-9
- E3 configuration mode, described, 1-9
- eBGP (external BGP), 8-3
- EGP (Exterior Gateway Protocol)
 - BGP, 1-4
 - described, 1-2
- election priority, VRRP backup router, 4-3
- exec mode, 1-9
- explicit null label, 13-7
- explicit routes
 - creating, 13-10
 - next hops, 13-10
- exporting route targets, 9-9
- extended community lists
 - creating, 12-5
 - described, 12-5
 - matching, 12-9
 - permit or deny, 12-5
 - resequence, 12-5

F

- fast convergence, 10-5
- fast hello
 - OSPF, 6-11
 - OSPFv3, 6-16
- fast reroute
 - link protection, 13-4
 - node protection, 13-4

- fast resets, 8-8
- flap statistics
 - IPv4, 8-9
 - IPv6, 8-10
- flash update threshold
 - RIP, 5-2
 - RIPng, 5-4
- flood reduction
 - OSPF, 6-11
 - OSPFv3, 6-16
- Frame Relay PVC configuration mode, described, 1-10

G

- global configuration mode, described, 1-10
- graceful restart
 - BGP instances
 - maximum update delays, 8-11
 - restart times, 8-11
 - retain times, 8-11
 - BGP neighbors
 - restart time, 8-16
 - retain routes, 8-16
 - retain times, 8-16
 - OSPF, 6-9
 - OSPFv3, 6-14
 - PIM, 11-14
 - RSVP
 - enabling, 13-12
 - hello intervals, 13-12
 - hello keep multipliers, 13-12
- group bandwidth, 11-8
- group membership, 11-8

H

- hello interval
 - LDP, 15-6
 - OSPF
 - interface, 6-11
 - sham link, 6-12
 - virtual link, 6-13
 - OSPFv3
 - interface, 6-16
 - virtual link, 6-17
 - PIM, 11-11
- hello packets, 6-4
 - intervals, 10-8
 - multipliers, 10-8
 - padding, 10-8
- holdtime timers
 - BGP instances, 8-9
 - BGP neighbors, 8-13
 - BGP peer groups, 8-17
 - LDP instance, 15-5
- hostnames, dynamic, 10-4

I

- iBGP (internal BGP)
 - confederations, 8-5
 - described, 8-3
 - route reflectors, 8-4
- ID (identifier)
 - BGP, 8-9
 - OSPF, 6-9
 - OSPFv3, 6-14
 - router, 2-5
 - VRRP backup router, 4-3
 - VRRP owner router, 4-2
- IGMP (Internet Group Management Protocol)
 - configuration tasks, 11-8
 - group bandwidth, 11-8
 - group membership, 11-8
 - join group, 11-8
 - last member query interval, 11-8
 - maximum bandwidth, 11-8
 - mtrace prohibit, 11-8
 - overview, 11-2
 - query interval, 11-8
 - query maximum response time, 11-8
 - robustness, 11-8
 - service profile
 - creating, 11-9
 - enabling, 11-9
 - instant leave, 11-9
 - maximum groups, 11-9
 - multicast destination, 11-9
 - priority, 11-9
 - static group, 11-10
 - sticky groups, 11-10
 - version, 11-8
- IGMP membership tracking
 - general, 11-2
 - IGMPv2, 11-3
 - IGMPv3, 11-3
- IGMP service profile configuration mode, described, 1-10
- IGP (Interior Gateway Protocol)
 - defined, 1-2
 - IS-IS, 1-5
 - OSPF, 1-4
- IGP shortcuts
 - RSVP instances, 13-8
 - RSVP LSPs, 13-8
- importing route targets, 9-9
- instances
 - BGP, 8-8
 - BGP, PE routers, 9-7
 - BGP VPN, 9-8
 - IS-IS, 10-3
 - OSPF, 6-8
 - OSPFv3, 6-14

-
- RIP, 5-2
 - RIPng, 5-4
 - interarea distribution, 10-4
 - interarea range
 - OSPF, 6-10
 - OSPFv3, 6-15
 - interconnection, L2VPN
 - ATM RFC 1483 bridged to dot1q, 14-21
 - ATM RFC 1483 bridged to Ethernet, 14-22
 - intercontext static routes, 2-5
 - interface configuration mode, described, 1-10
 - interface metrics, 10-9
 - internal router, 6-4
 - intervals, IS-IS LSP, 10-9
 - IP prefix list configuration mode, described, 1-10
 - IP prefix lists
 - configuration examples
 - complex, 12-12
 - simple, 12-12
 - creating, 12-6
 - described, 12-6
 - matching, 12-9
 - permit or deny, 12-6
 - resequence, 12-6
 - IP routing
 - route selection process, 1-6
 - supported protocols, 1-2
 - IPv4 (IP Version 4)
 - aggregate addresses, 8-9
 - ASNs
 - BGP neighbors, 8-14
 - BGP peer groups, 8-18
 - AS path lists
 - BGP neighbors, 8-14
 - BGP peer groups, 8-18
 - dampening, 8-9
 - default routes
 - BGP neighbors, 8-14
 - BGP peer groups, 8-18
 - distance, 8-9
 - enabling address families
 - BGP instances, 8-9
 - BGP neighbors, 8-14
 - BGP peer groups, 8-18
 - flap statistics, 8-9
 - maximum prefixes
 - BGP neighbors, 8-14
 - BGP peer groups, 8-18
 - networks, 8-10
 - peer groups, 8-14
 - prefix lists
 - BGP neighbors, 8-14
 - BGP peer groups, 8-18
 - redistributing routes, 8-10
 - route maps
 - BGP neighbors, 8-14
 - BGP peer groups, 8-18
 - route reflectors
 - BGP neighbors, 8-14
 - BGP peer groups, 8-18
 - traffic index counters, 8-10
- IPv6 (IP Version 6)
 - aggregate addresses, 8-10
 - ASNs
 - BGP neighbors, 8-15
 - BGP peer groups, 8-18
 - AS path lists
 - BGP neighbors, 8-15
 - BGP peer groups, 8-18
 - dampening, 8-10
 - default routes
 - BGP neighbors, 8-15
 - BGP peer groups, 8-18
 - distance, 8-10
 - enabling address families
 - BGP instances, 8-10
 - BGP neighbors, 8-15
 - BGP peer groups, 8-18
 - flap statistics, 8-10
 - maximum prefixes
 - BGP neighbors, 8-15
 - BGP peer groups, 8-18
 - networks, 8-10
 - peer groups, 8-15
 - prefix lists
 - BGP neighbors, 8-15
 - BGP peer groups, 8-18
 - redistributing routes, 8-10
 - route maps
 - BGP neighbors, 8-15
 - BGP peer groups, 8-19
 - route reflectors
 - BGP neighbors, 8-15
 - BGP peer groups, 8-19
 - traffic index counters, 8-10
- IPv6 prefix list configuration mode, described, 1-10
- IPv6 prefix lists
 - creating, 12-7
 - described, 12-7
 - matching, 12-9
 - permit or deny, 12-7
 - resequence, 12-7
- IS-IS (Intermediate System-to-Intermediate System)
 - configuration examples
 - minimum configuration, 10-10
 - multitopology IS-IS, 10-16
 - P2P-over-LAN circuit, 10-12

- three routers, 10-13
- two routers, 10-11
- features supported, 10-1
- hello packets
 - intervals, 10-8
 - multipliers, 10-8
 - padding, 10-8
- instances
 - address families, 10-3
 - attached bits, 10-5
 - authentication, 10-4
 - distances, 10-4
 - enabling, 10-3
 - fast convergence, 10-5
 - hostnames, dynamic, 10-4
 - interarea distribution, 10-4
 - levels, 10-3
 - maximum redistribute, 10-5
 - metric types, 10-4
 - NET, 10-3
 - overload bits, 10-5
 - paths, maximum, 10-5
 - route redistribution, 10-4
 - summary addresses, 10-4
 - traffic engineering, 10-5
- interfaces
 - address families, 10-7
 - circuit MTUs, 10-7
 - circuit types, 10-7
 - CSNP, intervals, 10-7
 - CSNP, on P2P, 10-7
 - enabling, 10-7
 - metrics, 10-9
 - optional checksums, 10-7
 - passive, 10-7
 - priorities, 10-7
- LSP
 - blocking flooding, 10-9
 - intervals, 10-9
 - lifetime, 10-6
 - receive only mode, 10-9
 - refresh intervals, 10-6
 - regeneration intervals, 10-6
 - retransmit intervals, 10-9
- packets, 10-2
- protocol data units, 10-2
- SPF
 - delay, 10-6
 - minimum intervals, 10-6
- start-on-demand, 10-10
- IS-IS (Intermediate System-to-Intermediate System),
 - features supported, 1-5
- IS-IS address family configuration mode, described, 1-10

- IS-IS interface address family configuration mode,
 - described, 1-10
- IS-IS interface configuration mode, described, 1-10
- IS-IS router configuration mode, described, 1-10

J

- join group, 11-8

K

- keepalive timers
 - BGP instances, 8-9
 - BGP neighbors, 8-13
 - BGP peer groups, 8-17

L

- L2 (Layer 2) circuits, enabling
 - 802.1Q PVCs, 14-5
 - ATM PVCs, 14-5
 - Ethernet ports, 14-5
 - Frame Relay PVCs, 14-5
- L2VPN configuration mode, described, 1-10
- L2VPN LDP configuration mode, described, 1-10
- L2VPN over GRE
 - configuration example, 14-23
 - described, 14-4
- L2VPNs (Layer 2 Virtual Private Networks)
 - configuration examples
 - ATM RFC 1483 bridged to dot1q
 - interconnection, 14-21
 - ATM RFC 1483 bridged to Ethernet
 - interconnection, 14-22
 - CE router, RFC 1483 bridged encapsulation, 14-14
 - dot1q bit propagation, 14-20
 - EXP bits, 14-18
 - interoperability with Extreme Networks, 14-14
 - L2VPN over GRE, 14-23
 - LDP L2VPNs, 14-7
 - QoS metering, 14-18
 - QoS rate limiting, 14-17
 - static L2VPNs, 14-7
 - cross-connections
 - LDP, 14-5
 - static, 14-6
 - encapsulation interconnectivity, supported, 14-3
 - encapsulation types, supported
 - ATM AAL5, 14-3
 - Ethernet, 14-3
 - Ethernet VLAN, 14-3
 - Frame Relay Martini, 14-2
 - implementation, described, 14-1
 - L2 circuits, enabling
 - 802.1Q PVCs, 14-5
 - ATM PVCs, 14-5

- Ethernet ports, 14-5
 - Frame Relay PVCs, 14-5
 - L2VPN over GRE, 14-4
 - QoS policies for, 14-4
 - soft GRE, enabling, 14-6
- L2VPN static configuration mode, described, 1-10
- label action, 13-2
- last member query interval, 11-8
- LDP (Label Distribution Protocol)
 - configuration examples
 - basic configuration, 15-6
 - targeted LDP, 15-8
 - explicit null, 15-3
 - graceful restart, 15-3
 - Hello
 - holdtime, 15-5
 - interval, 15-6
 - messages, described, 15-2
 - implementation, described, 15-1
 - instance, 15-3
 - interface, 15-3
 - label advertisement messages, 15-2
 - LSP, 15-1
 - LSR, 15-1
 - neighbor
 - password, 15-4
 - targeted, 15-4
 - prefix list filtering, 15-3
 - pseudo-circuits, 15-3
 - router ID, 15-4
 - session, 15-2
 - targeted
 - hello holdtime, 15-5
 - hello interval, 15-6
 - tracking IGP metric, 15-4
 - transport address, 15-4
- LDP L2VPNs (Label Distribution Protocol Layer 2 Virtual Private Networks)
 - configuration examples
 - ATM DS-3 encapsulation, 14-12
 - ATM OC encapsulation, 14-13
 - Ethernet encapsulation, 14-11
 - Ethernet VLAN encapsulation, 14-10
 - Frame Relay Martini encapsulation, 14-8
 - cross-connections, 14-5
- LDP router configuration mode, described, 1-10
- levels, IS-IS, 10-3
- lifetimes, 10-6
- link-state packets
 - acknowledgment, 6-5
 - request, 6-5
 - update, 6-5
- listen, RIPng packets, 5-5
- listen, RIP packets, 5-3
- load balancing, multipath
 - BGP, 8-9
 - BGP /MPLS VPNs, 9-8
- local mode, VPLS, 16-4
- local preferences, 8-9
- local protection, 13-9
- LSA (link-state advertisement)
 - AS-external-LSA, 6-6
 - fast origination, 6-9
 - network-LSA, 6-5
 - NSSA-external-LSA, 6-6
 - router-LSA, 6-5
 - summary-LSA
 - networks, 6-5
 - routers, 6-6
- LSP (label-switched path)
 - LDP, described, 15-1
- RSVP
 - backup, 13-8
 - bandwidth, 13-8
 - bypass, link protection, 13-9
 - bypass, node protection, 13-9
 - configuration example, 13-13
 - described, 13-2, 13-8
 - disabling, 13-9
 - egress, 13-9
 - fast reroute, 13-10
 - IGP shortcuts, 13-8
 - ingress, 13-9
 - link protection, 13-4
 - local protection, 13-9
 - node protection, 13-4
 - recording routes, 13-9
 - setup priority, 13-9
 - source path, 13-9
 - standard, 13-8
- static
 - configuration example, 13-12
 - creating, 13-7
 - described, 13-7
 - egress, 13-7
 - next hops, 13-7
 - outgoing labels, 13-7
 - overview, 13-2
- VPN route distribution, 9-3
- LSP (link-state protocol data unit)
 - blocking flooding, 10-9
 - intervals, 10-9
 - lifetimes, 10-6
 - receive only mode, 10-9
 - refresh intervals, 10-6
 - regeneration intervals, 10-6
 - retransmit intervals, 10-9

LSR (label-switched router)

- LDP, described, 15-1

MPLS

- described, 13-2
- egress, RSVP, 13-9
- egress, static, 13-7
- ingress, 13-9
- label action, 13-6
- local protection, 13-9

M

martian addresses, 2-5

maximum

paths

- IS-IS, 10-5
- RIP, 5-2
- RIPng, 5-4
- redistribute, 10-5
- redistribution quantum
 - OSPF, 6-10
 - OSPFv3, 6-15
- route redistribution
 - OSPF, 6-10
 - OSPFv3, 6-15

maximum bandwidth, 11-8

maximum prefixes

- BGP neighbors
 - IPv4, 8-14
 - IPv6, 8-15
- BGP peer groups
 - IPv4, 8-18
 - IPv6, 8-18

maximum routes, 2-5

maximum update delays, 8-11

MD5 (Message Digest 5) authentication

OSPF

- configuration example, 6-21
- configuring, 6-28

RIP, 5-7

VRRP, 4-10

MDTs (multicast domain trees)

- default group, specifying, 11-14
- encapsulation type, 11-14

MEDs (multi-exit discriminators), 8-8

membership tracking

- with IGMPv2, 11-3
- with IGMPv3, 11-3

mesh groups, 11-12

metric types, 10-4

minimum

- receive interval
 - BFD interfaces, 7-3
 - BFD neighbors, 7-2

transmit interval

- BFD interfaces, 7-3
- BFD neighbors, 7-2

mode access commands and prompts, 1-9

MPLS (Multiprotocol Label Switching)

configuration examples

- signaled LSP tunnel, 13-13
- static LSP tunnel, 13-12

instances

- creating, 13-5
- TTL, 13-5

interfaces, enabling, 13-5

overview, 13-1

static instances, creating, 13-6

static interfaces

- enabling, 13-6
- label action, 13-6

static LSPs

- creating, 13-7
- described, 13-7
- egress, 13-7
- next hops, 13-7
- outgoing labels, 13-7

TTL

- decrementing, 13-5
- propagating MPLS to TTL, 13-6
- propagating TTL to MPLS, 13-5

MPLS interface configuration mode, described, 1-10

MPLS router configuration mode, described, 1-10

MPLS static interface configuration mode, described, 1-10

MPLS static LSP configuration mode, described, 1-10

MPLS static router configuration mode, described, 1-10

MSDP (Multicast Source Discovery Protocol)

configuration example, 11-17

default peer, 11-12

enabling, 11-12

mesh groups, 11-12

originating RP address, 11-12

originating RP SA filter, 11-12

overview, 11-5

peer

- AS number, 11-12
- creating, 11-12
- description, 11-12
- disabling, 11-12
- SA filter, 11-12

MSDP peer configuration mode, described, 1-10

MSDP router configuration mode, described, 1-10

mtrace prohibit, 11-8

MTU negotiation, 2-5

multicast boundary, 11-10

- multicast routing
 - commands, described, 11-30
 - configuration examples
 - MSDP, 11-17
 - PIM-SM, 11-16
 - configuration tasks
 - IGMP, 11-8
 - MSDP, 11-12
 - MSDP peer, 11-12
 - multicast NPNs, 11-14
 - PIM-DM, 11-10
 - PIM-SM, 11-10
 - RMR, 11-15
 - service profile, 11-9
 - SSM, 11-14
 - subscribers, 11-13
 - overview
 - anycast RP, 11-5
 - general, 11-1
 - IGMP, 11-2
 - MSDP, 11-5
 - PIM, 11-3
 - RMR, 11-7
 - SSM, 11-4
 - subscribers
 - multicast receive permission, 11-13
 - multicast send, 11-13
 - service profile, 11-13
- multicast VPNs
 - default MDT group, 11-14
 - enabling, 11-14
 - MDT encapsulation type, 11-14
- multipath load balancing
 - BGP, 8-9
 - BGP/MPLS VPNs, 9-8
- mutual VRRP
 - different subnets, 4-5
 - multiple subnets, 4-6
 - same subnet, 4-4

N

- neighbors
 - advertisement intervals, 8-12
 - BFD, 7-2
 - community attributes, 8-13
 - creating, 8-12
 - described, 8-12
 - enforcing TTLs, 8-12
 - filters, 8-12
 - local AS, 8-12
 - MPLS labels, 8-13
 - multihops, 8-12
 - next hops, 8-12

- OSPF, 6-11
- OSPFv3, 6-16
- passwords, 8-12
- peer group, 8-13
- remote AS, 8-13
- shutdown, 8-13
- update source, 8-13
- VPLS, 16-4
- NET (network entity title), 10-3
- networks, BGP
 - IPv4, 8-10
 - IPv6, 8-10
- network type
 - OSPF, 6-12
 - OSPFv3, 6-16
- next-hop fast reroute
 - link protection, 13-4
 - node protection, 13-4
- next-hop reachability check, BGP/MPLS VPNs, 9-9
- NSSA (not so stubby area), range
 - OSPF, 6-10
 - OSPFv3, 6-15

O

- offset list, 5-3
- organization, of this guide, xxiii
- originating default route
 - OSPF instances, 6-9
 - OSPFv3 instances, 6-14
 - RIP instances, 5-2
 - RIPng instances, 5-4
- OSPF (Open Shortest Path First)
 - ABR, 6-4
 - area
 - creating, 6-10
 - default route, 6-10
 - interarea range, 6-10
 - NSSA range, 6-10
 - type, 6-10
 - ASBR, 6-4
 - backbone
 - area, 6-3
 - routers, 6-4
 - commands, described, 6-23
 - configuration examples
 - base configuration, 6-18
 - MD5 authentication, 6-21
 - route redistribution, 6-20
 - simple key chain configuration, 6-22
 - configuration tasks
 - area, 6-10
 - interface, 6-11
 - route redistribution, 6-10

- routing instance, 6-8
 - sham link, 6-12
 - virtual link, 6-13
- designated router, described, 6-4
- instance
 - auto cost, 6-8
 - capabilities, 6-8
 - creating, 6-8
 - default metric, 6-8
 - distance, 6-9
 - fast LSA origination, 6-9
 - graceful restart, 6-9
 - logging neighbor, 6-9
 - MPLS shortcuts, 6-9
 - originating default route, 6-9
 - redistributing routes, 6-10
 - router ID, 6-9
 - SPF timers, 6-9
 - stub router, 6-9
 - TE metrics, 6-9
- interface
 - authentication, 6-11
 - block flooding, 6-11
 - cost, 6-11
 - demand circuit, 6-11
 - enabling, 6-11
 - fast hello, 6-11
 - flood reduction, 6-11
 - hello interval, 6-11
 - neighbor, 6-11
 - network type, 6-12
 - passive, 6-12
 - retransmit interval, 6-12
 - router dead interval, 6-12
 - router priority, 6-12
 - transmit delay, 6-12
- internal router, 6-4
- LSAs
 - AS-external-LSA, 6-6
 - network-LSA, 6-5
 - router-LSA, 6-5
 - summary-LSA, networks, 6-5
 - summary-LSA, routers, 6-6
- maximum route redistribution, 6-10
- overview, 6-1
- packet header, 6-5
- sham link
 - authentication, 6-12
 - cost, 6-12
 - creating, 6-12
 - hello interval, 6-12
 - retransmit interval, 6-12
 - router dead interval, 6-13
 - transmit delay, 6-13
- summarizing external routes, 6-10
- supported IETF drafts and RFCs, 6-1
- virtual link
 - authentication, 6-13
 - creating, 6-13
 - hello interval, 6-13
 - retransmit interval, 6-13
 - router dead interval, 6-13
 - transmit delay, 6-13
- OSPF3 area configuration mode, described, 1-10
- OSPF3 interface configuration mode, described, 1-10
- OSPF3 router configuration mode, described, 1-10
- OSPF area configuration mode, described, 1-10
- OSPF interface configuration mode, described, 1-10
- OSPF router configuration mode, described, 1-10
- OSPF sham link configuration mode, described, 1-10
- OSPFv3 (Open Shortest Path First Version 3)
 - area
 - creating, 6-15
 - default route, 6-15
 - interarea range, 6-15
 - NSSA range, 6-15
 - type, 6-15
 - configuration tasks
 - area, 6-15
 - interface, 6-15
 - route redistribution, 6-15
 - routing instance, 6-14
 - virtual link, 6-17
 - instance
 - auto cost, 6-14
 - creating, 6-14
 - default metric, 6-14
 - distance, 6-14
 - graceful restart, 6-14
 - logging neighbor, 6-14
 - originating default route, 6-14
 - redistributing routes, 6-15
 - router ID, 6-14
 - SPF timers, 6-14
 - stub router, 6-14
 - interface
 - block flooding, 6-16
 - cost, 6-16
 - demand circuit, 6-16
 - enabling, 6-16
 - fast hello, 6-16
 - flood reduction, 6-16
 - hello interval, 6-16
 - neighbor, 6-16
 - network type, 6-16
 - passive, 6-17
 - retransmit interval, 6-17
 - router dead interval, 6-17

-
- router priority, 6-17
 - transmit delay, 6-17
 - maximum route redistribution, 6-15
 - summarizing external routes, 6-15
 - virtual link
 - creating, 6-17
 - hello interval, 6-17
 - retransmit interval, 6-17
 - router dead interval, 6-17
 - transmit delay, 6-17
 - OSPF virtual link configuration mode, described, 1-10
 - output delay
 - RIP, 5-3
 - RIPng, 5-4
 - overload bits, 10-5
 - owner router, 4-2
- P**
- packet types, OSPF, 6-4
 - passive interfaces, 10-7
 - PE (provider edge)
 - route distribution, 9-3
 - routers, 9-2
 - VPN topology, 9-2
 - peer groups
 - advertisement intervals, 8-16
 - applying attributes, 8-19
 - BGP neighbors, 8-13
 - community attributes, 8-17
 - creating, 8-16
 - dampening sessions, 8-17
 - description, 8-17
 - enforcing TTL, 8-17
 - IPv4 neighbor address families, 8-14
 - IPv6 neighbor address families, 8-15
 - multihops, 8-17
 - next hops, 8-17
 - passwords, 8-17
 - shutdown, 8-17
 - update source, 8-17
 - PE-to-CE routes
 - AS path loops, 9-10
 - overriding AS path attributes, 9-10
 - PIM (Protocol Independent Multicast)
 - dense mode, enabling, 11-10
 - graceful restart, 11-14
 - overview
 - anycast RP, 11-5
 - dense mode, 11-3
 - general, 11-3
 - sparse mode, 11-4
 - sparse mode
 - accept RP, 11-10
 - anycast RP, 11-10
 - BSR border, 11-10
 - BSR candidate, 11-11
 - configuration example, 11-16
 - DR priority, 11-11
 - enabling, 11-10
 - filtering neighbor, 11-11
 - hello interval, 11-11
 - multicast boundary, 11-10
 - operation mode, 11-11
 - RP address, 11-11
 - RP candidate, 11-11
 - SPT threshold infinity, 11-11
 - static group, 11-11
 - port configuration mode, described, 1-10
 - preempt, VRRP backup router, 4-3
 - prefix lists
 - BGP neighbors
 - IPv4, 8-14
 - IPv6, 8-15
 - BGP peer groups
 - IPv4, 8-18
 - IPv6, 8-18
 - priorities, IS-IS interfaces, 10-7
 - protocol precedences, default values, 2-3
 - prune messages, 11-3
 - pseudo-circuits, LDP LSPs, 15-3
 - pseudo-wire, VPLS
 - described, 16-1
 - name, 16-6
 - number, 16-6
 - publications, related to this guide, xxii
- Q**
- query interval, 11-8
 - query maximum response time, 11-8
- R**
- receive interval, minimum
 - BFD interfaces, 7-3
 - BFD neighbors, 7-2
 - receive only modes, 10-9
 - redback, 4-10
 - redistributing routes
 - BGP
 - IPv4, 8-10
 - IPv6, 8-10
 - IS-IS, 10-4
 - OSPF, 6-10
 - OSPFv3, 6-15

-
- RIP, 5-3
 - RIPng, 5-4
 - refresh intervals, 10-6
 - regeneration intervals, 10-6
 - related publications, xxii
 - reservation state lifetimes
 - keep multiplier, 13-11
 - refresh interval, 13-11
 - restart times, graceful
 - BGP instances, 8-11
 - BGP neighbors, 8-16
 - retaining routes, 8-16
 - retain times, graceful restart
 - BGP instances, 8-11
 - BGP neighbors, 8-16
 - retransmit interval
 - OSPF
 - interface, 6-12
 - sham link, 6-12
 - virtual link, 6-13
 - OSPFv3
 - interface, 6-17
 - virtual link, 6-17
 - retransmit intervals, 10-9
 - RIP (Routing Information Protocol)
 - commands, described, 5-6
 - configuration examples, 5-5
 - configuration tasks
 - RIP instance, 5-2
 - RIP interface, 5-3
 - instance
 - administrative distance, 5-2
 - create, 5-2
 - default metric, 5-2
 - distribution list, 5-2
 - flash update threshold, 5-2
 - maximum paths, 5-2
 - offset list, 5-3
 - originating default route, 5-2
 - output delay, 5-3
 - redistribute routes, 5-3
 - timers, 5-3
 - interface
 - authentication, 5-3
 - cost value, 5-3
 - enable, 5-3
 - listen, 5-3
 - originate default route, 5-3
 - split horizon, 5-3
 - summary address, 5-3
 - supply, 5-3
 - timers, 5-3
 - overview, 5-1
 - RIP interface configuration mode, described, 1-10
 - RIPng (Routing Information Protocol next generation)
 - configuration tasks
 - RIPng instance, 5-4
 - RIPng interface, 5-4
 - instance
 - administrative distance, 5-4
 - create, 5-4
 - default metric, 5-4
 - distribution list, 5-4
 - flash update threshold, 5-4
 - maximum paths, 5-4
 - originating default route, 5-4
 - output delay, 5-4
 - redistribute routes, 5-4
 - timers, 5-4
 - interface
 - cost value, 5-5
 - enable, 5-4
 - listen, 5-5
 - originate default route, 5-4
 - split horizon, 5-5
 - summary address, 5-5
 - supply, 5-5
 - timers, 5-5
 - RIPng interface configuration mode, described, 1-10
 - RIPng router configuration mode, described, 1-11
 - RIP router configuration mode, described, 1-10
 - RMR (remote multicast replication)
 - destination, 11-9
 - enabling, 11-15
 - output interface, 11-15
 - overview, 11-7
 - robustness, 11-8
 - route aggregation, BGP, 8-6
 - route distribution
 - among PE routers, 9-3
 - PE-to-CE, 9-3
 - route map configuration mode, described, 1-11
 - route maps
 - BGP neighbors
 - IPv4, 8-14
 - IPv6, 8-15
 - BGP peer groups
 - IPv4, 8-18
 - IPv6, 8-19
 - configuration examples
 - complex, 12-15
 - simple, 12-14
 - creating, 12-8
 - resequencing, 12-8
 - route origins, 9-10
-

- router dead interval
 - OSPF
 - interface, 6-12
 - sham link, 6-13
 - virtual link, 6-13
 - OSPFv3
 - interface, 6-17
 - virtual link, 6-17
- route redistribution
 - BGP
 - IPv4, 8-10
 - IPv6, 8-10
 - IS-IS, 10-4
 - OSPF, 6-10
 - OSPFv3, 6-15
 - RIP, 5-3
 - RIPng, 5-4
- route reflectors
 - BGP/MPLS VPNs, 9-2
 - BGP neighbors
 - IPv4, 8-14
 - IPv6, 8-15
 - BGP peer groups
 - IPv4, 8-18
 - IPv6, 8-19
 - client-to-client, 8-11
 - cluster IDs, assigning, 8-11
 - described, 8-4
- router functions, 6-4
- router priority
 - OSPF, 6-12
 - OSPFv3, 6-17
- route selection process, 6-4
- route target attribute, 9-4
- route targets
 - exporting, 9-9
 - filtering, 9-9
 - importing, 9-9
- routing policies
 - BGP AS path lists
 - configuration example, complex, 12-13
 - configuration example, simple, 12-13
 - creating, 12-2
 - described, 12-2
 - matching, 12-8
 - permit or deny, 12-2
 - resequence, 12-3
 - BGP attribute-based accounting
 - configuration example, 12-15
 - enabling, 12-11
 - table maps, 12-11
 - traffic index values, 12-11
 - BGP community lists
 - configuration example, complex, 12-14
 - configuration example, simple, 12-14
 - creating, 12-3
 - described, 12-3
 - matching, 12-9
 - permit or deny, 12-4
 - resequence, 12-4
 - BGP destination-based QoS
 - configuration example, 12-16
 - DSCP destinations, 12-11
 - DSCP values, 12-11
 - table maps, 12-11
 - BGP extended community lists
 - creating, 12-5
 - described, 12-5
 - matching, 12-9
 - permit or deny, 12-5
 - resequence, 12-5
 - IP prefix lists
 - configuration example, complex, 12-12
 - configuration example, simple, 12-12
 - creating, 12-6
 - described, 12-6
 - matching, 12-9
 - permit or deny, 12-6
 - resequence, 12-6
 - IPv6 prefix lists
 - creating, 12-7
 - described, 12-7
 - matching, 12-9
 - permit or deny, 12-7
 - resequence, 12-7
 - matching
 - BGP AS path lists, 12-8
 - BGP community lists, 12-9
 - BGP extended community lists, 12-9
 - IP prefix lists, 12-9
 - IPv6 prefix lists, 12-9
 - metric values, 12-9
 - next hops, IP prefix lists, 12-9
 - next hops, IPv6 prefix lists, 12-9
 - route types, 12-9
 - tag values, 12-9
 - route maps
 - configuration example, complex, 12-15
 - configuration example, simple, 12-14
 - creating, 12-8
 - match conditions, 12-8
 - resequencing, 12-8
 - set conditions, 12-9
 - setting
 - advertisement scope, 12-10
 - AS paths, 12-9

- BGP community attributes, 12-9
- BGP community lists, 12-9
- BGP extended community attributes, 12-10
- degree of preference, 12-10
- DSCP values, 12-11
- IP next hops, 12-10
- IPv6 next hops, 12-10
- local preferences, 12-10
- metric types, 12-10
- metric values, 12-10
- MPLS labels, 12-10
- route dampening, 12-10
- route origins, 12-10
- tag values, 12-10
- traffic index values, 12-11
- routing tables
 - BGP, 8-6
 - OSPF, 6-4
 - protocol precedence defaults, 2-3
 - static IP entries, 2-4
 - upper limit, 2-5
- routing tables, protocol precedence defaults, 1-7
- RP (rendezvous point)
 - accepting, 11-10
 - anycast, 11-10
 - candidate, 11-11
 - described, 11-4
 - IP address, 11-11
 - originating
 - IP address, 11-12
 - SA filter, 11-12
- RPF (reverse path forwarding)
 - static routes, 2-5
 - verifying source, 2-5
- RRO (record route object), 13-8
- RSVP (Resource Reservation Protocol)
 - instances
 - creating, 13-7
 - explicit null label, 13-7
 - explicit routes, 13-10
 - IGP shortcuts, 13-8
 - next hops, 13-10
 - RRO prefix types, 13-8
 - RSVP-INFO messages, 13-8
- interfaces
 - authenticating, 13-11
 - enabling, 13-11
 - graceful restart, enabling, 13-12
 - hello intervals, 13-12
 - hello keep multipliers, 13-12
 - keep multiplier, 13-11
 - refresh interval, 13-11
 - reservation state lifetimes, 13-11

- LSPs
 - backup, 13-8
 - bandwidth, 13-8
 - bypass, link protection, 13-9
 - bypass, node protection, 13-9
 - described, 13-8
 - disabling, 13-9
 - egress, 13-9
 - fast reroute, 13-10
 - IGP shortcuts, 13-8
 - ingress, 13-9
 - link protection, 13-4
 - local protection, 13-9
 - node protection, 13-4
 - recording routes, 13-9
 - setup priority, 13-9
 - source path, 13-9
 - standard, 13-8
- RSVP explicit route configuration mode, described, 1-11
- RSVP-INFO messages, 13-8
- RSVP interface configuration mode, described, 1-11
- RSVP LSP configuration mode, described, 1-11
- RSVP router configuration mode, described, 1-11

S

- service profile
 - enabling, 11-9
 - subscribers, 11-13
- servicing multiple contexts, 9-7
- sham link
 - authentication, 6-12
 - cost, 6-12
 - creating, 6-12
 - hello interval, 6-12
 - retransmit interval, 6-12
 - router dead interval, 6-13
 - transmit delay, 6-13
- site of origin attribute, 9-4
- soft GRE
 - BGP/MPLS VPNs, 9-4
 - configuration example, 14-23
 - described, 14-4
 - enabling, 14-6
 - tunnels, 9-11
- source IP address, DVSR profile, 3-3
- sparse mode
 - described, 11-4
 - enabling, 11-10
- special characters, in command syntax, xxiv
- SPF (Shortest Path First)
 - delay, 10-6
 - minimum intervals, 10-6

- timers
 - OSPF, 6-9
 - OSPFv3, 6-14
- split horizon
 - RIP interfaces, 5-3
 - RIPng interfaces, 5-5
- SPT (shortest-path tree)
 - defined, 11-3
 - threshold infinity, 11-11
- SSM (source-specific multicast)
 - defined, 11-4
 - enabling, 11-14
- static group
 - IGMP service profile, 11-10
 - PIM, 11-11
- static IPv6 routes, 2-4
- static L2VPNs
 - configuration example, 14-7
 - cross-connections, 14-6
- static routes
 - cost value, 2-4
 - intercontext, 2-5
 - multicast RPF, 2-5
 - unicast, 2-4
- static versus dynamic routing, 2-2
- STM-1 configuration mode, described, 1-11
- stub areas, 6-3
- stub router
 - OSPF, 6-9
 - OSPFv3, 6-14
- subscriber configuration mode, described, 1-11
- subscribers
 - multicast receive permission, 11-13
 - multicast send, 11-13
 - service profile, 11-13
- summarizing external routes
 - OSPF, 6-10
 - OSPFv3, 6-15
- summary addresses
 - IS-IS, 10-4
 - RIP, 5-3
 - RIPng, 5-5
- supply, send
 - RIPng packets, 5-5
 - RIP packets, 5-3

T

- tag value, DVSR profile, 3-3
- TCP (Transmission Control Protocol), MTU, 2-5
- text formats, in command syntax, xxv
- threshold, flash update
 - RIP, 5-2
 - RIPng, 5-4

- timers
 - holdtime
 - BGP instances, 8-9
 - BGP neighbors, 8-13
 - BGP peer groups, 8-17
 - keepalive
 - BGP instances, 8-9
 - BGP neighbors, 8-13
 - BGP peer groups, 8-17
 - RIP, 5-3
 - RIPng
 - instance, 5-4
 - interface, 5-5
- traffic engineering, 10-5
- traffic index counters
 - IPv4, 8-10
 - IPv6, 8-10
- transmit delay
 - OSPF
 - interface, 6-12
 - sham link, 6-13
 - virtual link, 6-13
 - OSPFv3
 - interface, 6-17
 - virtual link, 6-17
- transmit interval, minimum
 - BFD interfaces, 7-3
 - BFD neighbors, 7-2
- TTL (time to live)
 - DVSR profile, 3-3
 - enforcing
 - BGP neighbors, 8-12
 - GBP peer groups, 8-17

V

- verify-set values, DVSR profile, 3-3
- version, IGMP, 11-8
- virtual IP address
 - VRRP backup router, 4-3
 - VRRP owner router, 4-2
- virtual link
 - OSPF
 - authentication, 6-13
 - creating, 6-13
 - hello interval, 6-13
 - retransmit interval, 6-13
 - router dead interval, 6-13
 - transmit delay, 6-13
 - OSPFv3
 - creating, 6-17
 - hello interval, 6-17
 - retransmit interval, 6-17

- router dead interval, 6-17
- transmit delay, 6-17
- VPLS (Virtual Private LAN Services)
 - bridges
 - creating, 16-5
 - disabling VPLS, 16-5
 - enabling VPLS, 16-5
 - pseudo-wire name, 16-6
 - pseudo-wire number, 16-6
 - VPLS profile, applying, 16-5
 - commands, described, 16-8
 - configuration examples
 - bridge profile, 16-7
 - VPLS-enabled bridge, 16-7
 - VPLS profile, 16-7
 - configuration tasks
 - bridge profile, 16-3
 - VPLS-enabled bridge, 16-5
 - VPLS profile, 16-4
 - neighbors
 - bridge profiles, 16-4
 - connection type, 16-5
 - counters, 16-4
 - creating, 16-4
 - description, 16-4
 - encapsulation type, 16-5
 - local mode, 16-4
 - standby, 16-5
 - overview, 16-1
 - profiles
 - applying, 16-5
 - creating, 16-4
 - neighbors, 16-4
- VPLS configuration mode, described, 1-11
- VPLS profile configuration mode, described, 1-11
- VPLS profile neighbor configuration mode, described, 1-11
- VPN (Virtual Private Network)
 - contexts
 - creating, 9-7
 - multiple, 9-2
 - described, 9-2
 - topology, 9-2
- VPN-IPv4
 - address family, 9-3
 - route target attribute, 9-4
- VRRP (Virtual Router Redundancy Protocol)
 - backup router
 - advertise interval, 4-3
 - authentication, 4-3
 - election priority, 4-3
 - ID, 4-3
 - preempt, 4-3
 - virtual IP address, 4-3
 - commands, described, 4-8
 - configuration examples
 - basic, 4-3
 - MD5 authentication, 4-7
 - mutual VRRP, different subnets, 4-5
 - mutual VRRP, multiple subnets, 4-6
 - mutual VRRP, same subnet, 4-4
 - configuration tasks
 - backup router, 4-3
 - owner router, 4-2
 - overview, 4-1
 - owner router, 4-2
 - advertise interval, 4-2
 - authentication, 4-2
 - virtual IP address, 4-2
 - VRRP configuration mode, described, 1-11

Commands

A

- accept filter prefix-list, 8-26
- address-family, IS-IS
 - instance, 10-18
 - interface, 10-18
- address-family ipv4
 - BGP neighbors, 8-28
 - BGP peer groups, 8-28
 - BGP routers, 8-28
- address-family ipv4 vpn, BGP
 - neighbors, 9-50
 - peer groups, 9-50
 - routers, 9-50
- address-family ipv6 unicast
 - BGP neighbors, 8-30
 - BGP peer groups, 8-30
 - BGP routers, 8-30
- advertise-interval, VRRP, 4-9
- advertisement-interval, BGP
 - neighbors, 8-32
 - peer groups, 8-32
- aggregate-address, 8-34
- area, 6-24
- area-type, 6-26
- art, 13-23
- asloop-in, 8-36
- as-override, 8-38
- as-path-list
 - BGP
 - neighbor address family, 8-40
 - peer group address family, 8-40
 - context, 12-19
- attached-bit, 10-20
- authentication
 - IS-IS, 10-22
 - OSPF
 - interfaces, 6-28
 - sham links, 6-28
 - virtual links, 6-28

- RIP, 5-7

- RSVP, 13-15

- VRRP, 4-10

- auto-cost, 6-30

B

- bandwidth, 13-16
- bestpath med always-compare, 8-42
- bfd detection, 7-6
- BGP attribute-based accounting
 - table-map, 8-105
 - traffic-index accounting, 12-76
- block-flooding, 6-31

C

- capabilities, 6-32
- circuit mtu, 10-24
- circuit type, 10-25
- client-to-client reflection, 8-43
- cluster-id, 8-44
- community-list, 12-21
- confederation identifier, 8-45
- confederation peers, 8-46
- context vpn-rd, 9-52
- cost, OSPF
 - interfaces, 6-34
 - sham links, 6-34
- counters, 16-9
- create-lsp-circuit, 15-10
- csnp interval, 10-26
- csnp periodic-on-ptp, 10-28

D

- dampening, 8-47
- decrement ttl, 13-17
- default-information originate, 5-9

- default-metric
 - OSPF, 6-35
 - RIP, 5-11
- default-originate, BGP
 - neighbor address family, 8-49
 - peer group address family, 8-49
- default-peer, 11-31
- default-route, 6-36
- demand-circuit, 6-38
- deny
 - AS path lists, 12-42
 - community lists, 12-42
 - IP prefix lists, 12-42
- description, 16-10
 - AS path lists, 12-22
 - BGP
 - neighbors, 8-51
 - peer groups, 8-51
 - community lists, 12-22
 - IP prefix lists, 12-22
 - MPLS static LSP, 13-18
 - MSDP peers, 11-33
 - RSVP LSP, 13-18
- detection-multiplier, 7-7
- disable, 16-11
- distance
 - BGP, 8-52
 - DVSR profiles, 3-7
 - IS-IS, 10-29
 - OSPF, 6-40
 - RIP, 5-12
- distribute-list, 5-13
- dvsvr-profile, 3-8
- dynamic-hostname, 10-31

E

- ebgp-multihop, 8-53
- egress, MPLS
 - signaled LSP, 13-19
 - static LSP, 13-19
- enforce ttl, 8-54
- explicit-null
 - LDP, 15-11
 - RSVP, 13-20
- explicit-route, 13-21
- export route-target, 9-54
- ext-community-list, 12-23

F

- fast-convergence, 10-33
- fast-hello, 6-41
- fast-lsa-origination, 6-43
- fast-reroute, 13-22

- fast-reset, 8-56
- flap-statistics, 8-57
- flash-update-threshold, 5-14
- flood-reduction, 6-44

G

- graceful-restart
 - LDP, 15-13
 - OSPF, 6-45
 - RSVP, 13-23

H

- hello holdtime, 15-14
- hello interval
 - IS-IS, 10-34
 - LDP, 15-16
 - RSVP, 13-24
- hello-interval, OSPF
 - interfaces, 6-46
 - sham links, 6-46
 - virtual links, 6-46
- hello keep-multiplier, 13-26
- hello multiplier, 10-36
- hello padding, 10-38

I

- igmp access-group, 11-34
- igmp group-bandwidth, 11-35
- igmp join-group, 11-36
- igmp last-member-query-interval, 11-37
- igmp maximum-bandwidth, 11-38
- igmp mtrace-prohibit, 11-40
- igmp query-interval, 11-41
- igmp query-max-response-time, 11-42
- igmp robust, 11-43
- igmp service-profile, 11-44
- igmp version, 11-46
- import route-target, 9-56
- ingress, 13-29
- instant-leave, 11-47
- interarea-distribute, 10-39
- interface
 - BFD, 7-8
 - IS-IS, 10-41
 - LDP, 15-18
 - MPLS, 13-30
 - OSPF, 6-48
 - RIP, 5-15
 - RSVP, 13-30
- interface-cost, 5-17
- ip igmp service-profile, 11-48
- ip martian, 2-7
- ip maximum-routes, 2-9

- ip mstatic, 2-11
- ip multicast boundary, 11-49
- ip multicast receive, 11-50
- ip multicast send, 11-52
- ip prefix-list, 12-25
- ip route, 2-12
- ip soft-gre
 - L2VPN over GRE, 14-25
 - MPLS over GRE, 9-58
- ipv6 prefix-list, 12-26
- ipv6 route, 2-15
- ip verify unicast source, 2-17
- is type, 10-43

K

- keep-multiplier, 13-32

L

- l2vpn, 14-27
- l2vpn-cct-bindings ldp, 14-28
- l2vpn-cct-bindings static, 14-29
- l2vpn ctx-name
 - ATM PVCs, 14-30
 - dot1q PVCs, 14-30
 - Frame Relay PVCs, 14-30
 - ports, 14-30
- label-action, 13-33
- label-binding, 15-20
- listen, 5-18
- local-as, 8-58
- local-mode, 16-13
- local-preference, 8-60
- local-protection, 13-35
- log-lsp-up-down, 13-36
- log-neighbor-changes, 8-61
- log-neighbor-up-down, 6-50
- lsp, 13-37
- lsp block-flooding, 10-45
- lsp gen-interval, 10-46
- lsp interval, 10-47
- lsp max-lifetime, 10-48
- lsp receive-only-mode, 10-49
- lsp refresh-interval, 10-50
- lsp retransmit-interval, 10-51

M

- mark dscp destination, 12-27
- match as-path-list, 12-29
- match community-list, 12-30
- match ext-community-list, 12-32
- match ip address prefix-list, 12-34
- match ip next-hop prefix-list, 12-35
- match ipv6 address prefix-list, 12-36

- match ipv6 next-hop prefix-list, 12-37
- match metric, 12-38
- match route-type, 12-39
- match tag, 12-41
- max-groups, 11-54
- maximum paths, 10-52
- maximum-paths, 5-19
- maximum prefix, BGP
 - neighbor address family, 8-62
 - peer group address family, 8-62
- maximum redistribute
 - IS-IS, 10-53
 - OSPF, 6-51
- maximum redistribute-quantum, 6-52
- maximum restart-time, BGP
 - neighbors, 8-64
 - routers, 8-64
- maximum retain-time, BGP
 - neighbors, 8-65
 - routers, 8-65
- maximum update-delay, 8-67
- mdt default-group, 11-56
- mdt encapsulation, 11-57
- mesh-group, 11-58
- metric, 10-54
- metric-style, 10-56
- minimum receive-interval, 7-9
- minimum transmit-interval, 7-10
- mpls shortcuts, 6-53
- mpls traffic-engineering, 6-54
- multicast destination, 11-59
- multicast output, 11-61
- multi-paths, 8-68
- multi-paths eibgp, 9-60

N

- neighbor, 16-15
 - BFD, 7-11
 - BGP, 8-70
 - OSPF, 6-55
- neighbor password, 15-22
- neighbor targeted, 15-23
- net, 10-58
- network, 8-72
- network-type, 6-57
- next-hop
 - MPLS explicit routes, 13-39
 - MPLS static LSPs, 13-39
- next-hop-self, BGP
 - neighbors, 8-74
 - peer groups, 8-74
- nssa-range, 6-59

O

- optional-checksums, 10-59
- originate-default, 6-61
- originating-rp, 11-63
- originating-rp sa-filter, 11-64
- out-label, 13-40
- output-delay, 5-21

P

- passive, 6-63
- passive-interface, 10-60
- password, BGP
 - neighbors, 8-76
 - peer groups, 8-76
- peer, 11-65
- peer-as, 11-66
- peer-group, BGP
 - neighbor address family, 8-77
 - neighbors, 8-77
 - routers, 8-77
- permit
 - AS path lists, 12-42
 - community lists, 12-42
 - IP prefix lists, 12-42
- pe-type, 16-17
- pim accept-rp, 11-67
- pim anycast-rp, 11-69
- pim bsr-border, 11-70
- pim bsr-candidate, 11-71
- pim dense-mode, 11-72
- pim dr-priority, 11-73
- pim hello-interval, 11-76
- pim neighbor-filter, 11-77
- pim operation-mode, 11-78
- pim rp-address, 11-79
- pim rp-candidate, 11-80
- pim sparse-mode, 11-81
- pim spt-threshold infinity, 11-82
- pim ssm, 11-83
- pim static group, 11-84
- preempt, 4-11
- prefix-list, BGP
 - neighbor address family, 8-80
 - peer group address family, 8-80
- priority
 - IGMP, 11-85
 - IS-IS, 10-61
 - VRRP, 4-12
- profile, 16-19
- propagate ttl ip-to-mpls, 13-41
- propagate ttl mpls-to-ip, 13-42
- pw-encap, 16-21
- pw-id, 16-22

- pw-name, 16-24

R

- range, 6-64
- record-route, 13-43
- redistribute
 - BGP, 8-82
 - IS-IS, 10-63
 - OSPF, 6-65
 - RIP, 5-22
- refresh-interval, 13-44
- remote-as, 8-84
- remove-private-as, BGP
 - neighbor address family, 8-85
 - peer group address family, 8-85
- resequence as-path-list, 12-46
- resequence community-list, 12-47
- resequence ext-community-list, 12-48
- resequence ip prefix-list, 12-49
- resequence ipv6 prefix-list, 12-50
- resequence route-map, 12-51
- retain-ibgp-routes, 8-86
- retransmit-interval, OSPF
 - interfaces, 6-68
 - sham links, 6-68
 - virtual links, 6-68
- route-map
 - BGP
 - neighbor address family, 8-87
 - peer group address family, 8-87
 - routing policies, 12-52
- route-origin, 8-89
- router bfd, 7-12
- router bgp, 8-91
- router bgp vpn, 9-62, 9-64
- router-dead-interval, OSPF
 - interfaces, 6-69
 - sham links, 6-69
- router-dead-interval, OSPF virtual links, 6-69
- route-reflector-client, BGP
 - neighbor address family, 8-92
 - peer group address family, 8-92
- router-id
 - BGP, 8-94
 - contexts, 2-19
 - LDP, 15-25
 - OSPF, 6-71
- router isis, 10-65
- router ldp, 15-27
- router mpls, 13-45
- router mpls-static, 13-46
- router msdp, 11-86
- router ospf, 6-72

- router ospf3, 6-73
- router-priority, 6-74
- router rip, 5-24
- router ripng, 5-25
- router rsvp, 13-47
- route-target filter, 9-66
- rro-prefix-type, 13-48

S

- sa-filter, 11-87
- send community, BGP
 - neighbors, 8-95
 - peer groups, 8-95
- send ext-community, BGP
 - neighbors, 8-96
 - peer groups, 8-96
- send filter prefix-list, 8-98
- send label, 8-100
- service inter-context routing, 2-20
- session-dampening, 8-102
- set as-path, 12-54
- set community, 12-56
- set community-list, 12-58
- set dampening, 12-59
- set dscp, 12-61
- set ext-community, 12-62
- set ip next-hop, 12-64
- set ipv6 next-hop, 12-65
- set label, 12-66
- set level, 12-67
- set local-preference, 12-69
- set metric, 12-70
- set metric-type, 12-71
- set origin, 12-72
- set-overload-bit, 10-66
- set tag, 12-73
- set traffic-index, 12-74
- setup-priority
 - MPLS routers, 13-49
 - MPLS signaled LSPs, 13-49
- set weight, 12-75
- sham-link, 6-75
- shutdown
 - BGP
 - neighbors, 8-104
 - peer groups, 8-104
 - MSDP peers, 11-89
 - RSVP LSP, 13-50
- source-address, 3-9
- source-path, 13-51
- spf holddown, 10-68
- spf interval, 10-69
- spf-timers, 6-77

- split-horizon, 5-26
- standby-for, 16-26
- static-group, 11-90
- sticky-groups, 11-92
- stub-router, 6-78
- summary-address
 - IS-IS, 10-70
 - OSPF, 6-80
 - RIP, 5-28
- supply, 5-30

T

- table-map, 8-105
- tag, 3-10
- targeted-hello holdtime, 15-29
- targeted-hello interval, 15-31
- tcp path-mtu-discovery, 2-21
- timer password, 8-106
- timers, BGP
 - neighbors, 8-107
 - peer groups, 8-107
 - routers, 8-107
- timers basic
 - RIP instances, 5-31
 - RIP interfaces, 5-31
- track-igp-metric, 15-33
- traffic-engineering, 10-72
- traffic-index accounting, 12-76
- transmit-delay, OSPF
 - interfaces, 6-82
 - sham links, 6-82
 - virtual links, 6-82
- transport address, 15-34
- ttl, 3-11

U

- update-source, BGP
 - neighbors, 8-109
 - peer groups, 8-109

V

- verify-set, 3-12
- virtual-address, 4-14
- virtual-link, 6-83
- vpls, 16-28
- vpls profile, 16-29
- vpn, 9-67
- vrrp, 4-15

X

- xc vc-id, 14-32
- xc vpn-label, 14-35

A

- AS path list configuration mode
 - deny, 12-42
 - description, 12-22
 - permit, 12-42
- ATM PVC configuration mode
 - l2vpn ctx-name, 14-30

B

- BFD interface configuration mode
 - detection-multiplier, 7-7
 - minimum receive-interval, 7-9
 - minimum transmit-interval, 7-10
- BFD neighbor configuration mode
 - detection-multiplier, 7-7
 - minimum receive-interval, 7-9
 - minimum transmit-interval, 7-10
- BFD router configuration mode
 - interface, 7-8
 - neighbor, 7-11
- BGP address family configuration mode
 - aggregate-address, 8-34
 - dampening, 8-47
 - distance, 8-52
 - export route-target, 9-54
 - flap-statistics, 8-57
 - import route-target, 9-56
 - network, 8-72
 - redistribute, 8-82
 - route-origin, 8-89
 - route-target filter, 9-66
 - send label, 8-100
 - table-map, 8-105
- BGP neighbor address family configuration mode
 - as-path-list, 8-40
 - default-originate, 8-49
 - maximum prefix, 8-62
 - peer-group, 8-77
 - prefix-list, 8-80

- remove-private-as, 8-85
 - route-map, 8-87
 - route-reflector-client, 8-92
- BGP neighbor configuration mode
 - accept filter prefix-list, 8-26
 - address-family ipv4, 8-28
 - address-family ipv4 vpn, 9-50
 - address-family ipv6 unicast, 8-30
 - advertisement-interval, 8-32
 - asloop-in, 8-36
 - as-override, 8-38
 - description, 8-51
 - ebgp-multihop, 8-53
 - enforce ttl, 8-54
 - local-as, 8-58
 - maximum restart-time, 8-64
 - maximum retain-time, 8-65
 - next-hop-self, 8-74
 - password, 8-76
 - peer-group, 8-77
 - remote-as, 8-84
 - retain-ibgp-routes, 8-86
 - send community, 8-95
 - send ext-community, 8-96
 - send filter prefix-list, 8-98
 - session-dampening, 8-102
 - shutdown, 8-104
 - timers, 8-107
 - update-source, 8-109
- BGP peer group address family configuration mode
 - as-path-list, 8-40
 - default-originate, 8-49
 - maximum prefix, 8-62
 - prefix-list, 8-80
 - remove-private-as, 8-85
 - route-map, 8-87
 - route-reflector-client, 8-92

BGP peer group configuration mode
 address-family ipv4, 8-28
 address-family ipv4 vpn, 9-50
 address-family ipv6 unicast, 8-30
 advertisement-interval, 8-32
 description, 8-51
 ebgp-multihop, 8-53
 enforce ttl, 8-54
 next-hop-self, 8-74
 password, 8-76
 send community, 8-95
 send ext-community, 8-96
 session-dampening, 8-102
 shutdown, 8-104
 timers, 8-107
 update-source, 8-109

BGP router configuration mode
 address-family ipv4, 8-28
 address-family ipv4 vpn, 9-50
 address-family ipv6 unicast, 8-30
 bestpath med always-compare, 8-42
 client-to-client reflection, 8-43
 cluster-id, 8-44
 confederation identifier, 8-45
 confederation peers, 8-46
 fast-reset, 8-56
 local-preference, 8-60
 log-neighbor-changes, 8-61
 maximum restart-time, 8-64
 maximum retain-time, 8-65
 maximum update-delay, 8-67
 multi-paths, 8-68
 multi-paths eibgp, 9-60
 neighbor, 8-70
 peer-group, 8-77
 router-id, 8-94
 timer password, 8-106
 timers, 8-107

bridge configuration mode
 vpls, 16-28

C

community list configuration mode
 deny, 12-42
 description, 12-22
 permit, 12-42
context configuration mode
 as-path-list, 12-19
 community-list, 12-21
 dvsr-profile, 3-8
 ext-community-list, 12-23
 igmp group-bandwidth, 11-35
 igmp mtrace-prohibit, 11-40

igmp service-profile, 11-44
ip martian, 2-7
ip maximum-routes, 2-9
ip mstatic, 2-11
ip prefix-list, 12-25
ip route, 2-12
ip soft-gre
 L2VPN over GRE, 14-25
 MPLS over GRE, 9-58
ipv6 prefix-list, 12-26
ipv6 route, 2-15
l2vpn, 14-27
pim accept-rp, 11-67
pim anycast-rp, 11-69
pim bsr-candidate, 11-71
pim rp-address, 11-79
pim rp-candidate, 11-80
pim spt-threshold infinity, 11-82
pim ssm, 11-83
pim static group, 11-84
resequence as-path-list, 12-46
resequence community-list, 12-47
resequence ext-community-list, 12-48
resequence ip prefix-list, 12-49
resequence ipv6 prefix-list, 12-50
resequence route-map, 12-51
route-map, 12-52
router bfd, 7-12
router bgp, 8-91
router bgp vpn, 9-62, 9-64
router-id, 2-19
router isis, 10-65
router ldp, 15-27
router mpls, 13-45
router mpls-static, 13-46
router msdp, 11-86
router ospf, 6-72
router ospf3, 6-73
router rip, 5-24
router ripng, 5-25
router rsvp, 13-47
static-group, 11-90

D

dot1q PVC configuration mode
 l2vpn ctx-name, 14-30
DVSr profile configuration mode
 distance, 3-7
 source-address, 3-9
 tag, 3-10
 ttl, 3-11
 verify-set, 3-12

F

Frame Relay PVC configuration mode
l2vpn ctx-name, 14-30

G

global configuration mode
context vpn-rd, 9-52
service inter-context routing, 2-20
tcp path-mtu-discovery, 2-21
vpls profile, 16-29

I

IGMP service profile configuration mode
instant-leave, 11-47
max-groups, 11-54
multicast destination, 11-59
priority, 11-85
static-group, 11-90
sticky-groups, 11-92
interface configuration mode
igmp access-group, 11-34
igmp join-group, 11-36
igmp last-member-query-interval, 11-37
igmp query-interval, 11-41
igmp query-max-response-time, 11-42
igmp robust, 11-43
igmp service-profile, 11-44
igmp version, 11-46
ip multicast boundary, 11-49
ip verify unicast source, 2-17
mark dscp destination, 12-27
mdt default-group, 11-56
mdt encapsulation, 11-57
multicast ourput, 11-61
pim bsr-border, 11-70
pim dense-mode, 11-72
pim dr-priority, 11-73
pim hello-interval, 11-76
pim neighbor-filter, 11-77
pim operation-mode, 11-78
pim sparse-mode, 11-81
traffic-index accounting, 12-76
vrrp, 4-15
IP prefix list configuration mode
deny, 12-42
description, 12-22
permit, 12-42
IS-IS address family configuration mode
interarea-distribute, 10-39
redistribute, 10-63
IS-IS interface configuration mode
address-family, 10-18
authentication, 10-22

circuit mtu, 10-24
circuit type, 10-25
csnp interval, 10-26
csnp periodic-on-ptp, 10-28
hello interval, 10-34
hello multiplier, 10-36
hello padding, 10-38
lsp block-flooding, 10-45
lsp interval, 10-47
lsp receive-only-mode, 10-49
lsp retransmit-interval, 10-51
metric, 10-54
optional-checksums, 10-59
passive-interface, 10-60
priority, 10-61
IS-IS router configuration mode
address-family, 10-18
attached-bit, 10-20
authentication, 10-22
distance, 10-29
dynamic-hostname, 10-31
fast-convergence, 10-33
interface, 10-41
is type, 10-43
lsp gen-interval, 10-46
lsp max-lifetime, 10-48
lsp refresh-interval, 10-50
maximum paths, 10-52
maximum redistribute, 10-53
metric-style, 10-56
net, 10-58
set-overload-bit, 10-66
spf holddown, 10-68
spf interval, 10-69
summary-address, 10-70
traffic-engineering, 10-72

L

L2VPN configuration mode
l2vpn-cct-bindings ldp, 14-28
l2vpn-cct-bindings static, 14-29
L2VPN LDP configuration mode
xc vc-id, 14-32
L2VPN static configuration mode
xc vpn-label, 14-35
LDP router configuration mode
create-lsp-circuit, 15-10
explicit-null, 15-11
graceful-restart, 15-13
hello holdtime, 15-14
hello interval, 15-16
interface, 15-18
label-binding, 15-20

- neighbor password, 15-22
- neighbor targeted, 15-23
- router-id, 15-25
- targeted-hello holdtime, 15-29
- targeted-hello interval, 15-31
- track-igp-metric, 15-33
- transport address, 15-34

M

- MPLS router configuration mode
 - decrement ttl, 13-17
 - interface, 13-30
 - propagate ttl ip-to-mpls, 13-41
 - propagate ttl mpls-to-ip, 13-42
- MPLS static interface configuration mode
 - label-action, 13-33
- MPLS static LSP configuration mode
 - description, 13-18
 - egress, 13-19
 - next-hop, 13-39
 - out-label, 13-40
- MPLS static router configuration mode
 - interface, 13-30
 - lsp, 13-37
- MSDP peer configuration mode
 - description, 11-33
 - peer-as, 11-66
 - sa-filter, 11-87
 - shutdown, 11-89
- MSDP router configuration mode
 - default-peer, 11-31
 - mesh-group, 11-58
 - originating-rp, 11-63
 - originating-rp sa-filter, 11-64
 - peer, 11-65

O

- OSPF area configuration mode
 - area-type, 6-26
 - default-route, 6-36
 - interface, 6-48
 - nssa-range, 6-59
 - range, 6-64
 - sham-link, 6-75
 - virtual-link, 6-83
- OSPF interface configuration mode
 - authentication, 6-28
 - bfd detection, 7-6
 - block-flooding, 6-31
 - cost, 6-34
 - demand-circuit, 6-38
 - fast-hello, 6-41
 - flood-reduction, 6-44

- hello-interval, 6-46
- neighbor, 6-55
- network-type, 6-57
- passive, 6-63
- retransmit-interval, 6-68
- router-dead-interval, 6-69
- router-priority, 6-74
- transmit-delay, 6-82

- OSPF router configuration mode
 - area, 6-24
 - auto-cost, 6-30
 - capabilities, 6-32
 - default-metric, 6-35
 - distance, 6-40
 - fast-lsa-origination, 6-43
 - graceful-restart, 6-45
 - log-neighbor-up-down, 6-50
 - maximum redistribute, 6-51
 - maximum redistribute-quantum, 6-52
 - mpls shortcuts, 6-53
 - mpls traffic-engineering, 6-54
 - originate-default, 6-61
 - redistribute, 6-65
 - router-id, 6-71
 - spf-timers, 6-77
 - stub-router, 6-78
 - summary-address, 6-80
 - vpn, 9-67

- OSPF sham link configuration mode
 - authentication, 6-28
 - cost, 6-34
 - hello-interval, 6-46
 - retransmit-interval, 6-68
 - router-dead-interval, 6-69
 - transmit-delay, 6-82

- OSPF virtual link configuration mode
 - authentication, 6-28
 - hello-interval, 6-46
 - retransmit-interval, 6-68
 - router-dead-interval, 6-69
 - transmit-delay, 6-82

P

- port configuration mode
 - igmp maximum-bandwidth, 11-38
 - l2vpn ctx-name, 14-30

R

- RIP interface configuration mode
 - authentication, 5-7
 - default-information originate, 5-9
 - interface-cost, 5-17
 - listen, 5-18

-
- split-horizon, 5-26
 - summary-address, 5-28
 - supply, 5-30
 - timers basic, 5-31
 - RIP router configuration mode
 - default-information originate, 5-9
 - default-metric, 5-11
 - distance, 5-12
 - distribute-list, 5-13
 - flash-update-threshold, 5-14
 - interface, 5-15
 - maximum-paths, 5-19
 - output-delay, 5-21
 - redistribute, 5-22
 - timers basic, 5-31
 - route map configuration mode
 - match as-path-list, 12-29
 - match community-list, 12-30
 - match ext-community-list, 12-32
 - match ip address prefix-list, 12-34
 - match ip next-hop prefix-list, 12-35
 - match ipv6 address prefix-list, 12-36
 - match ipv6 next-hop prefix-list, 12-37
 - match metric, 12-38
 - match route-type, 12-39
 - match tag, 12-41
 - set as-path, 12-54
 - set community, 12-56
 - set community-list, 12-58
 - set dampening, 12-59
 - set dscp, 12-61
 - set ext-community, 12-62
 - set ip next-hop, 12-64
 - set ipv6 next-hop, 12-65
 - set label, 12-66
 - set level, 12-67
 - set local-preference, 12-69
 - set metric, 12-70
 - set metric-type, 12-71
 - set origin, 12-72
 - set tag, 12-73
 - set traffic-index, 12-74
 - set weight, 12-75
 - RSVP explicit route configuration mode
 - next-hop, 13-39
 - RSVP interface configuration mode
 - authentication, 13-15
 - hello interval, 13-24
 - hello keep-multiplier, 13-26
 - keep-multiplier, 13-32
 - refresh-interval, 13-44
 - RSVP LSP configuration mode
 - bandwidth, 13-16
 - description, 13-18
 - egress, 13-19
 - fast-reroute, 13-22
 - ingress, 13-29
 - local-protection, 13-35
 - record-route, 13-43
 - setup-priority, 13-49
 - shutdown, 13-50
 - source-path, 13-51
 - RSVP router configuration mode
 - explicit-null, 13-20
 - explicit-route, 13-21
 - graceful-restart, 13-23
 - interface, 13-30
 - log-lsp-up-down, 13-36
 - lsp, 13-37
 - rro-prefix-type, 13-48
- S**
- subscriber configuration mode
 - ip igmp service-profile, 11-48
 - ip multicast receive, 11-50
 - ip multicast send, 11-52
- V**
- VPLS configuration mode
 - disable, 16-11
 - profile, 16-19
 - pw-id, 16-22
 - pw-name, 16-24
 - VPLS profile configuration mode
 - neighbor, 16-15
 - VPLS profile neighbor configuration mode
 - counters, 16-9
 - description, 16-10
 - local-mode, 16-13
 - pe-type, 16-17
 - pw-encap, 16-21
 - standby-for, 16-26
 - VRRP configuration mode
 - advertise-interval, 4-9
 - authentication, 4-10
 - preempt, 4-11
 - priority, 4-12
 - virtual-address, 4-14
-

