AX3800S/AX3650S Software Manual

Configuration Guide Vol. 3

For Version 11.10

AX38S-S003X-40

AlaxalA

■ Relevant products

This manual applies to the models in the AX3800S and AX3650S series of switches. It also describes the functionality of versions 11.10 of the software. The described functionality is that supported by the software OS-L3SA-A/OS-L3SA and OS-L3SL-A/OS-L3SL, and by optional licenses.

■ Export restrictions

In the event that any or all ALAXALA products (including technologies, programs and services) described or contained herein are controlled under any of applicable export control laws and regulations (including the Foreign Exchange and Foreign Trade Law of Japan and United States export control laws and regulations), such products shall not be exported without obtaining the required export licenses from the authorities concerned in accordance with the above laws.

■ Trademarks

■ Reading and storing this manual

Before you use the equipment, carefully read the manual and make sure that you understand all safety precautions.

After reading the manual, keep it in a convenient place for easy reference.

■ Notes

Information in this document is subject to change without notice.

■ Editions history

December 2012 (Edition 5) AX38S-S003X-40

■ Copyright

History of Amendments

[For version 11.9]

Summary of amendments

Item	Changes
Policy-based Routing (IPv4)	• AX3800S series switches supported policy-based routing.

In addition to the above change, minor editorial corrections have been made.

[For version 11.7]

Summary of amendments

Item	Changes
Policy-based Routing (IPv4)	• This chapter was added.
Notes about network configurations	• *(b) Load due to receiving unnecessary multicast packets from multiple VLANs* was added in *(1) Notes applying to both PIM-SM and PIM-SSM*.

[For version 11.6]

This manual contains descriptions of the AX3650S that were in the manual *AX3600S Software Manual for Version 11.5*.

Summary of amendments

Item	Changes
Load balancing specifications	• Descriptions for AX3800S series switches were added.
Load balancing specifications	• Descriptions for AX3800S series switches were added.

Preface

Applicable products and software versions

This manual applies to the models in the AX3800S and AX3650S series of switches. It also describes the functionality of version 11.10 of the software. The described functionality is that supported by the software OS-L3SA-A/OS-L3SA and OS-L3SL-A/OS-L3SL, and by optional licenses.

Before you operate the equipment, carefully read the manual and make sure that you understand all instructions and cautionary notes. After reading the manual, keep it in a convenient place for easy reference.

Unless otherwise noted, this manual describes the functions applicable to both the AX3800S and AX3650S series of switches, and functionalities common to each software. For functionalities that are not common to both AX3800S and AX3650S series switches, and functionalities not common to OS-L3SA-A/OS-L3SA and OS-L3SL-A/OS-L3SL are indicated as follows:

[AX3800S]:

The description applies to AX3800S switches.

[AX3650S]:

The description applies to AX3650S switches.

[OS-L3SA]:

The description applies to OS-L3SA-A/OS-L3SA for the AX3800S and AX3650S series of switches.

The functions supported by optional licenses are indicated as follows:

[OP-DH6R]:

The description applies to the OP-DH6R optional license.

[OP-OTP]:

The description applies to the OP-OTP optional license.

[OP-VAA]:

The description applies to the OP-VAA optional license.

Corrections to the manual

Corrections to this manual might be contained in the Release Notes and Manual Corrections that come with the software.

Intended readers

This manual is intended for system administrators who wish to configure and operate a network system that uses the Switch.

Readers must have an understanding of the following:

- The basics of network system management

Manual URL

You can view this manual on our website at:

http://www.alaxala.com/en/

Reading sequence of the manuals

The following shows the manuals you need to consult according to your requirements determined from the following workflow for installing, setting up, and starting regular operation of the Switch.

● **Unpacking the switch and the basic settings for initial installation**

Quick Start Guide (AX36S-Q001X)

● **Determining the hardware facility conditions and how to handle the hardware**

Hardware Instruction Manual (AX36S-H001X)

● **Understanding the software functions, configuration settings, and use of the operation commands**

Configuration Guide Vol.1 (AX38S-S001X)

Vol.2 (AX38S-S002X)

Vol.3 (AX38S-S003X)

● **Learning the syntax of configuration commands and the details of command parameters**

Configuration Command Reference Vol. 1 (AX38S-S004X)

Vol.2 (AX38S-S005X)

● **Learning the syntax of operation commands and the details of command parameters**

Operation Command Reference Vol. 1 (AX38S-S006X)

Vol.2 (AX38S-S007X)

● **Understanding messages and logs**

Message and Log Reference (AX38S-S008X)

● **Understanding the MIB**

MIB Reference (AX38S-S009X)

● **How to troubleshoot when a problem occurs**

Troubleshooting Guide (AX36S-T001X)

Conventions: The terms "Switch" and "switch"

The term Switch (upper-case "S") is an abbreviation for any or all of the following models:

AX3800S series switch

AX3650S series switch

The term switch (lower-case "s") might refer to a Switch, another type of switch from the current vendor, or a switch from another vendor. The context decides the meaning.

Abbreviations used in the manual

AC	Alternating Current
ACK	ACKnowledge
ADSL	Asymmetric Digital Subscriber Line
ALG	Application Level Gateway
ANSI	American National Standards Institute
ARP	Address Resolution Protocol
AS	Autonomous System
AUX	Auxiliary
BGP	Border Gateway Protocol
BGP4	Border Gateway Protocol - version 4
BGP4+	Multiprotocol Extensions for Border Gateway Protocol - version 4
bit/s	bits per second (can also appear as bps)
BPDU	Bridge Protocol Data Unit
BRI	Basic Rate Interface
CC	Continuity Check
CDP	Cisco Discovery Protocol
CFM	Connectivity Fault Management
CIDR	Classless Inter-Domain Routing
CIR	Committed Information Rate
CIST	Common and Internal Spanning Tree
CLNP	ConnectionLess Network Protocol
CLNS	ConnectionLess Network System
CONS	Connection Oriented Network System
CRC	Cyclic Redundancy Check
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
CSNP	Complete Sequence Numbers PDU
CST	Common Spanning Tree
DA	Destination Address
DC	Direct Current
DCE	Data Circuit terminating Equipment
DHCP	Dynamic Host Configuration Protocol
DIS	Draft International Standard/Designated Intermediate System
DNS	Domain Name System
DR	Designated Router
DSAP	Destination Service Access Point
DSCP	Differentiated Services Code Point
DTE	Data Terminal Equipment
DVMRP	Distance Vector Multicast Routing Protocol
E-Mail	Electronic Mail
EAP	Extensible Authentication Protocol
EAPOL	EAP Over LAN
EFM	Ethernet in the First Mile
ES	End System
FAN	Fan Unit
FCS	Frame Check Sequence
FDB	Filtering DataBase
FQDN	Fully Qualified Domain Name
FTTH	Fiber To The Home
GBIC	GigaBit Interface Converter
GSRP	Gigabit Switch Redundancy Protocol
HMAC	Keyed-Hashing for Message Authentication
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
ICMPv6	Internet Control Message Protocol version 6
ID	Identifier
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers, Inc.
IETF	the Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IP	Internet Protocol
IPCP	IP Control Protocol
IPv4	Internet Protocol version 4

IPv6	Internet Protocol version 6
IPV6CP	IP Version 6 Control Protocol
IPX	Internetwork Packet Exchange
ISO	International Organization for Standardization
ISP	Internet Service Provider
IST	Internal Spanning Tree
L2LD	Layer 2 Loop Detection
LAN	Local Area Network
LCP	Link Control Protocol
LED	Light Emitting Diode
LLC	Logical Link Control
LLDP	Link Layer Discovery Protocol
LLQ+3WFQ	Low Latency Queueing + 3 Weighted Fair Queueing
LSP	Label Switched Path
LSP	Link State PDU
LSR	Label Switched Router
MA	Maintenance Association
MAC	Media Access Control
MC	Memory Card
MD5	Message Digest 5
MDI	Medium Dependent Interface
MDI-X	Medium Dependent Interface crossover
MEP	Maintenance association End Point
MIB	Management Information Base
MIP	Maintenance domain Intermediate Point
MRU	Maximum Receive Unit
MSTI	Multiple Spanning Tree Instance
MSTP	Multiple Spanning Tree Protocol
MTU	Maximum Transfer Unit
NAK	Not AcKnowledge
NAS	Network Access Server
NAT	Network Address Translation
NCP	Network Control Protocol
NDP	Neighbor Discovery Protocol
NET	Network Entity Title
NLA ID	Next-Level Aggregation Identifier
NPDU	Network Protocol Data Unit
NSAP	Network Service Access Point
NSSA	Not So Stubby Area
NTP	Network Time Protocol
OADP	Octpower Auto Discovery Protocol
OAM	Operations, Administration, and Maintenance
OSPF	Open Shortest Path First
OUI	Organizationally Unique Identifier
packet/s	packets per second (can also appear as pps)
PAD	PADding
PAE	Port Access Entity
PC	Personal Computer
PCI	Protocol Control Information
PDU	Protocol Data Unit
PICS	Protocol Implementation Conformance Statement
PID	Protocol IDentifier
PIM	Protocol Independent Multicast
PIM-DM	Protocol Independent Multicast-Dense Mode
PIM-SM	Protocol Independent Multicast-Sparse Mode
PIM-SSM	Protocol Independent Multicast-Source Specific Multicast
PoE	Power over Ethernet
PRI	Primary Rate Interface
PS	Power Supply
PSNP	Partial Sequence Numbers PDU
QoS	Quality of Service
QSFP+	Quad Small Form factor Pluggable Plus
RA	Router Advertisement
RADIUS	Remote Authentication Dial In User Service
RDI	Remote Defect Indication
REJ	REJect
RFC	Request For Comments

RIP	Routing Information Protocol
RIPng	Routing Information Protocol next generation
RMON	Remote Network Monitoring MIB
RPF	Reverse Path Forwarding
RQ	ReQuest
RSTP	Rapid Spanning Tree Protocol
SA	Source Address
SD	Secure Digital
SDH	Synchronous Digital Hierarchy
SDU	Service Data Unit
SEL	NSAP SELector
SFD	Start Frame Delimiter
SFP	Small Form factor Pluggable
SFP+	Enhanced Small Form factor Pluggable
SMTP	Simple Mail Transfer Protocol
SNAP	Sub-Network Access Protocol
SNMP	Simple Network Management Protocol
SNP	Sequence Numbers PDU
SNPA	Subnetwork Point of Attachment
SPF	Shortest Path First
SSAP	Source Service Access Point
STP	Spanning Tree Protocol
TA	Terminal Adapter
TACACS+	Terminal Access Controller Access Control System Plus
TCP/IP	Transmission Control Protocol/Internet Protocol
TLA ID	Top-Level Aggregation Identifier
TLV	Type, Length, and Value
TOS	Type Of Service
TPID	Tag Protocol Identifier
TTL	Time To Live
UDLD	Uni-Directional Link Detection
UDP	User Datagram Protocol
UPC	Usage Parameter Control
UPC-RED	Usage Parameter Control - Random Early Detection
VAA	VLAN Access Agent
VLAN	Virtual LAN
VPN	Virtual Private Network
VRF	Virtual Routing and Forwarding/Virtual Routing and Forwarding Instance
VRRP	Virtual Router Redundancy Protocol
WAN	Wide Area Network
WDM	Wavelength Division Multiplexing
WFQ	Weighted Fair Queueing
WRED	Weighted Random Early Detection
WS	Work Station
WWW	World-Wide Web
XFP	10 gigabit small Form factor Pluggable

Conventions: KB, MB, GB, and TB

This manual uses the following conventions: 1 KB (kilobyte) is 1024 bytes. 1 MB (megabyte) is 1024^2 bytes. 1 GB (gigabyte) is 1024^3 bytes. 1 TB (terabyte) is 1024^4 bytes.

Contents

Preface

PART 2: IPv4 Routing Protocols

13. Route Filtering (IPv4) 259

14. Description of IPv4 Multicasting 307

15. Settings and Operation for IPv4 Multicasting 351

16. IPv4 Multicast Route Filtering [OS-L3SA] 373

PART 3: IPv6 Packet Relaying

PART 4: IPv6 Routing Protocols

25. RIPng 489

26. OSPFv3 [OS-L3SA] 507

Chapter

1. Description of IP, ARP, and ICMP

In IPv4 networks, the Switch defines routes to the destinations of IP packets and forwards the IP packets based on communication protocols. This chapter describes IPv4 addressing and IPv4 packet forwarding.

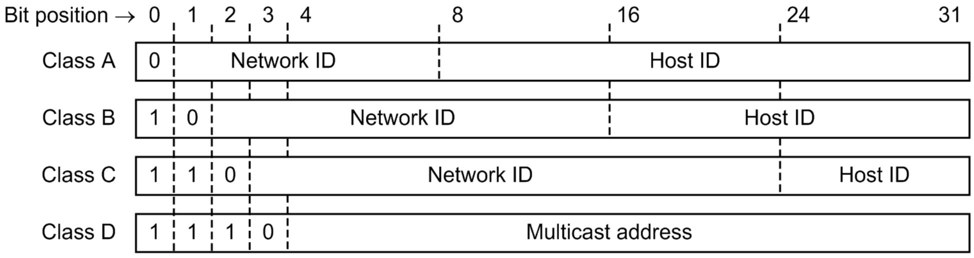
1.1 Addressing

This section provides an overview of the IPv4 addressing used by the Switch.

1.1.1 IP address

The Switch supports Class A, B, C, and D IP addresses. Class D addresses are used by routing protocols for multicast. Depending on the routing protocol being used, addresses defined in accordance with Classless Inter-Domain Routing (CIDR) are also supported. The following figure shows the format of an IP address.

Figure 1-1: Format of an IP address



Note that in a network broadcast address or subnetwork broadcast address, the host ID is binary and consists of all 0s or all 1s. You can select which type of host ID you want to use in the configuration for each interface. For details about interface, see *1.2.2 Items assigned IP addresses*.

The IP addresses in the following ranges can be assigned to the Switch:

- Network IDs

  The following network IDs are allowed:

  - Class A: 1.*x*.*x*.*x* to 126.*x*.*x*.*x*
  - Class B: 128.1.*x*.*x* to 191.254.*x*.*x*
  - Class C: 192.0.1.*x* to 223.255.254.*x* (*x* = host ID)

- Host IDs

  The following host IDs are allowed:

  - Class A: *y*.0.0.1 to *y*.255.255.254
  - Class B: *y*.*y*.0.1 to *y*.*y*.255.254
  - Class C: *y*.*y*.*y*.1 to *y*.*y*.*y*.254 (*y* = network ID)

1.1.2 Subnet mask

Regardless of the boundaries between network IDs and host IDs in the Class A, B, and C addresses shown in *Figure 1-1: Format of an IP address*, you can use subnet masks to specify boundaries at any location between network IDs and host IDs.

For example, if you obtain a Class B network ID and divide the network into 256 subnets, you would apply subnet mask 255.255.255.0. Or, if you want to employ CIDR and obtain two consecutive Class C network IDs (for example, 192.0.0.*x* and 192.0.1.*x*) to create a single subnetwork, you would apply subnet mask 255.255.254.0.

Specify a subnet mask in the configuration for each interface in left-adjusted format (specify consecutive 1s from the most significant digit in binary notation).

For example, you can specify 255.255.192.0 as a subnet mask, but you cannot specify 255.255.96.0 as a subnet mask.

1.2 IP layer functionality

1.2.1 Forwarding function

The Switch forwards received IP packets based on routing tables. This forwarding processing can be roughly divided into the three functions described below.

- Communication protocols

  Communication protocols are used to send and receive IP packets at the IP layer.

- Forwarding function

  The forwarding function forwards IP packets based on routing tables.

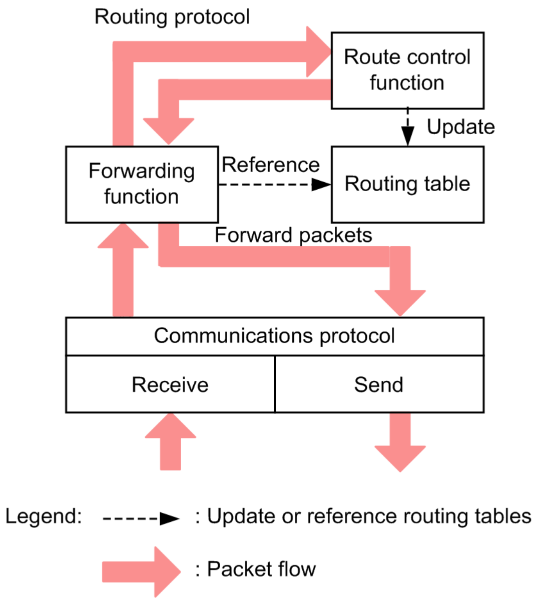
- Route control function

  The route control function exchanges routing information with other routers, determines routes, and creates routing tables.

For the AX3650S series switches, you can disable forwarding for a particular VLAN interface and use that VLAN interface to manage remote operation terminals.

The following figure provides an overview of IPv4 routing.

Figure 1-2: Overview of IPv4 routing



1.2.2 Items assigned IP addresses

In the Switch, you can assign IP addresses to VLANs. Multihomed connections, in which multiple IP addresses are assigned to a single VLAN, are also available. The switch broadcasts packets to all nodes in a network.

1.3 Communication protocols

This section describes the communication protocols used to forward IPv4 packets. The following IPv4 communication protocols are available:

- IP
- ICMP
- ARP

1.3.1 Internet Protocol (IP)

(1) Format of an IP packet

The format and settings of IP packets sent by the switch conform to RFC 791.

(2) Checking the validity of IP packet headers

The Switch checks the validity of the headers in IP packets when IP packets are received. The following table describes the IP packet header items that are checked.

Table 1-1: Items checked in an IP packet header

IP packet header field	Check item	Discards packet if error detected	Sends an ICMP message for a discarded packet
Version	The version must be 4.	Y	N
Header length	The header length must be equal to or greater than 5.	Y	N
TOS	Not checked.	--	--
Total length	The total length must be equal to or greater than the header length multiplied by 4.	Y	N
Packet identifier	Not checked.	--	--
Flag	Not checked.	--	--
Fragment offset	Not checked.	--	--
TTL	The TTL of the packets when the destination is the local device: Not checked.	--	--
	The TTL of the packets to be forwarded: TTL-1 must be greater than 0.	Y	Y#
Protocol	Not checked.	--	--
Header checksum	The header checksum must be correct.	Y	N
Source address	Not checked.	--	--
Destination address	All of the following requirements apply: 1. Class A, Class B, Class C, or Class D 2. The network number is a value other than 127 (127 is the internal loopback address). 3. The network number is a value other than 0 (except 0.0.0.0).	Y	N

Legend: Y: Action performed, N: Action not performed, --: Action not applicable
#: An ICMP Time Exceeded message is sent.

(3) Supported IP options

The following table describes the IP options supported by the Switch.

Table 1-2: Supported IP options

IP option	Type of IP packet		
	Packet sent from the Switch	Packet sent to the Switch	Packet forwarded by the Switch
End of Option List	Y	--	--
No Operation	Y	--	--
Loose Source Routing	Y	Y	Y
Strict Source Routing	N	Y	Y
Record Route	Y	Y	Y
Internet Timestamp	N	Y	Y

Legend: Y: Supported, N: Not supported, --:Not applicable

1.3.2 ICMP

(1) Format of an ICMP message

The format and settings of ICMP messages sent by the Switch conform to RFC 792.

(2) Supported ICMP message types

The following table describes the supported ICMP message types.

Table 1-3: Supported ICMP message types (code values are in decimal notation)

ICMP message				Supporte d
Type		Detailed type		
--	Code	--	Co de	
Destination Unreachable	3	Net Unreachable	0	N
		Host Unreachable	1	Y
		Protocol Unreachable	2	Y
		Port Unreachable	3	Y
		Fragmentation Needed and DF Set	4	Y
		Source Route Failed	5	Y
		Destination Network Unknown	6	N
		Destination Host Unknown	7	N
		Network Unreachable for Type of Service	11	N
		Host Unreachable for Type of Service	12	N

ICMP message				Supported
Type		Detailed type		
--	Code	--	Code	
		Communication Administratively Prohibited	13	Y
		Host Precedence Violation	14	N
		Precedence Cutoff in Effect	15	N
Source Quench	4	--	0	N
Redirect	5	Redirect Datagrams for the Network	0	N
		Redirect Datagrams for the Host	1	Y
		Redirect Datagrams for the Type of Service and Network	2	N
		Redirect Datagrams for the Type of Service and Host	3	N
Time Exceeded	11	Time to Live Exceeded in Transit	0	Y
		Fragment Reassembly Time Exceeded	1	N
Parameter Problem	12	--	0	Y
Echo Request	8	--	0	Y
Echo Reply	0	--	0	Y
Timestamp Request	13	--	0	N
Timestamp Reply	14	--	0	Y#
Information Request	15	--	0	N
Information Reply	16	--	0	N
Address Mask Request	17	--	0	N
Address Mask Reply	18	--	0	Y#

Legend: Y: Supported, N: Not supported, --: Not applicable

#: When a Request message is received, a Reply message is returned.

(3) Sending an ICMP Redirect message

When the receiving interface and the sending interface are the same for a forwarded packet, the hardware determines that a decision about whether an ICMP Redirect message can be sent is required, and the software determines whether an ICMP Redirect message can be sent. The software sends an ICMP Redirect message when the following conditions are met:

- The packet source and the next hop router belong to the same segment (the subnetwork address in the source IP address of the received IP packet matches the subnetwork address of the next hop router to which the IP packet is to be forwarded).

  An exception is a subnetwork address in the received IP packet that matches the default route.

- The received IP packet is not an ICMP packet.

- Sending is enabled in the IP routing information in the configuration.

(4) Sending an ICMP Time Exceeded message

The switch sends an ICMP Time Exceeded message when the following conditions are met:

- The TTL is 1 in the received IP packet to be forwarded.
- The received IP packet is not an ICMP packet (ICMP Echo packets excepted).

1.3.3 ARP

(1) Format of an ARP frame

The format and settings of ARP frames sent by the Switch conform to RFC 826.

(2) Checking the validity of ARP frames

The Switch checks the validity of received ARP frames. The following table describes the ARP frame items that are checked.

Table 1-4: Items checked in an ARP frame

ARP frame field	Check item	Discards frame
Hardware type	(For Ethernet) Hardware type: 1 (Ethernet)	Y
Protocol type	Must be 0800H (IP). Must be 1000H (trailer packet).#	Y
Hardware address length	Not checked.	--
Protocol address length	Not checked.	--
Operation code	Must be 1 (request). If a value other than 1 is set, the value is treated as 2 (reply).	--
Sender hardware address	The value must be a value other than the following: • The hardware address of the local device	Y
Sender protocol address	The value must be a value other than the following: • Multicast address • The protocol address of the local device • 0.0.0.0	Y
Target hardware address	• Must be the hardware address of the local device. • Must be a broadcast address.	Y
Target protocol address	• Must be the protocol address of the local device.	Y

Legend: Y: Frames are discarded if an error is detected, --: Not applicable

#

The switch does not initiate the sending of trailer packets. However, if a request is made, the switch returns a reply and learns the protocol type.

(3) ProxyARP

You can run Proxy ARP on all interfaces in the Switch. Set whether to enable local Proxy ARP in the configuration. When the Switch receives an ARP request packet that satisfies all of the following conditions, it sends an ARP reply packet on behalf of the target protocol.

- The target protocol address in the ARP request packet is not a broadcast address.
- The subnetwork number in the sender protocol address of the ARP request packet differs from

the subnetwork number in the target protocol address.

- The target protocol address in the ARP request packet exists in the routing table and the packet can reach the destination.

(4) Local Proxy ARP

You can run local Proxy ARP on all interfaces in the Switch. Set whether to enable local Proxy ARP in the configuration.

The differences between Proxy ARP and local Proxy ARP are as follows:

- Proxy ARP mainly responds to ARP requests directed to subnets connected to interfaces that differ from the ARP receiving interface for terminals that do not support routing.
- Local Proxy ARP responds to ARP requests directed to subnets connected to the receiving interface.

Use local Proxy ARP for subnets containing terminals that cannot directly communicate with one another for security reasons or for subnets where broadcast is prohibited. To provide an environment for running local Proxy ARP on the Switch, execute the `l2-isolation` configuration command. Local Proxy ARP allows the Switch to forward traffic between terminals on the same subnet. Note that using local Proxy ARP increases the number of ICMP Redirect messages. We therefore suggest that you disable ICMP Redirect messages.

The Switch sends an ARP reply packet on behalf of the target protocol when it receives an ARP request packet that satisfies all of the following conditions:

- The target protocol address in the ARP request packet is not a broadcast address.
- The subnetwork number in the target protocol address in the ARP request packet matches the subnetwork number of the receiving interface.
- The sender protocol address and the target protocol address are different.

(5) Aging timer

You can specify the aging time for ARP information for each interface in minutes. The minimum specifiable value is 1 minute and the maximum specifiable value is 24 hours. The default is 4 hours.

(6) Setting ARP information

To connect the Switch to a product that does not use the ARP protocol, use the `arp` configuration command to associate MAC addresses and IP addresses (the ARP information).

(7) Referencing ARP information

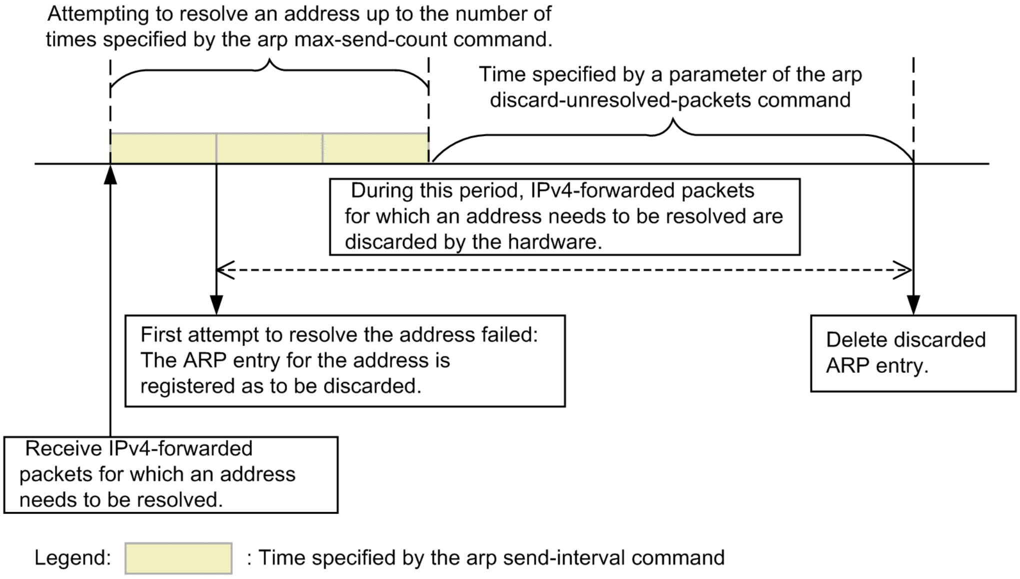
You can execute the `show ip arp` command on an operation terminal to check the ARP information. By checking the ARP information, you can determine the association between the IP address and MAC address for a specific interface.

(8) Hardware-based discard functionality for packets with unresolved addresses

If you continue to communicate with a non-existent terminal or via a non-existent router for a reason related to network configuration, packets with unresolved addresses are forwarded to the CPU, which might result in an increased CPU load. In such a case, you can reduce the CPU load by setting the `arp discard-unresolved-packets` configuration command to use the hardware to discard forwarding packets with unresolved addresses.

The following shows how the hardware discard functionality is performed for packets with unresolved addresses.

Figure 1-3: How the hardware discard functionality is used for packets with unresolved addresses


Attempting to resolve an address up to the number of times specified by the arp max-send-count command.
Time specified by a parameter of the arp discard-unresolved-packets command
During this period, IPv4-forwarded packets for which an address needs to be resolved are discarded by the hardware.
First attempt to resolve the address failed: The ARP entry for the address is registered as to be discarded.
Delete discarded ARP entry.
Receive IPv4-forwarded packets for which an address needs to be resolved.
Legend:
: Time specified by the arp send-interval command

When an interface with the `arp discard-unresolved-packets` command set fails to resolve an address for the first time, the ARP entry is temporarily registered to the hardware as an entry to be discarded. Because forwarding packets sent to the ARP entry or forwarding packets that use the ARP entry as the next hop are discarded by the hardware after address resolution attempts fails as many times as the number specified by the `arp max-send-count` configuration command until the time specified by the `arp discard-unresolved-packets` command has elapsed, the CPU load is reduced. If the next address resolution attempt is successful, normal communication is possible thereafter.

Use this functionality only in an abnormal situation where unresolved addresses persist.

1.4 Forwarding function

1.4.1 Forwarding IP packets

The forwarding function transfers received packets to the next router or host based on routing tables.

(1) Routing table contents

A routing table contains multiple entries, each of which consists of the items described below. To check the contents of the routing tables maintained by the Switch, execute the `show ip route` command.

Destination:

The destination network address and the bit length of the subnet mask for the destination network address. The subnet mask masks the destination IP address in a received IP packet when the routing table is searched. If the destination network is not divided into subnetworks, this field indicates the length of the mask bits corresponding to the network class of the network address (for example, 8 for Class A). This field indicates 32 when the IP packet is forwarded based on the host address.

Next Hop:

The IP address of the next router to which the IP packet is forwarded. When the Switch is using the multipath function, multiple next hops exist.

Interface: Name of the VLAN containing the next hop router

Metric: The distance to the destination network

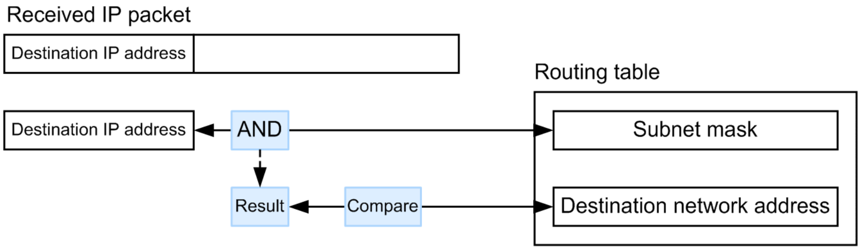
Protocol: The routing protocol used on the node that previously forwarded the packet

Age: The time, in seconds, since the route was confirmed or changed

(2) Searching a routing table

The forwarding function searches the routing table for the entry that matches the destination IP address in the received IP packet. The matching entry contains the destination network address, and is found when the Switch applies the subnet mask in the routing table in an AND operation with the destination IP address in the received IP packet and the result is the same as an entry in the routing table. The following figure shows how a routing table is searched to determine the destination network address.

Figure 1-4: Searching a routing table

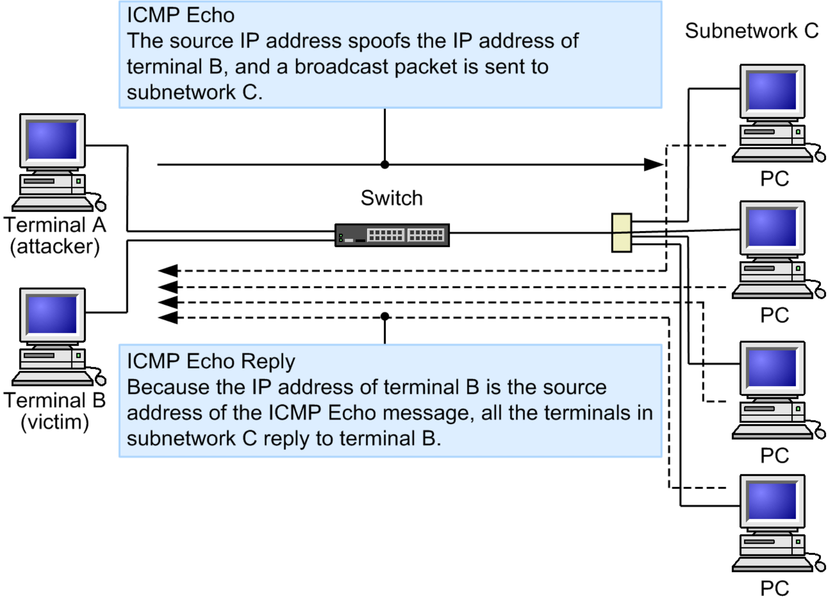


1.4.2 Forwarding broadcast packets

In the Switch, you can use configuration commands to set whether to forward broadcast packets to a network or subnetwork that is directly connected to the Switch via IP forwarding (referred to as directed broadcast hereafter). To set the operation of the receiving interface, use the `ip subnet-broadcast` command. To set the operation of the sending interface for each subnet, use the `directed-broadcast` parameter in the `ip address` command.

Directed broadcasts are not forwarded by default when the above configuration commands are not executed. If you set forwarding of directed broadcasts, you need to be aware that attacks might be directed against terminals as shown in the following figure.

Figure 1-5: Example of an attack with broadcast packets against a subnetwork


ICMP Echo
The source IP address spoofs the IP address of terminal B, and a broadcast packet is sent to subnetwork C.
Subnetwork C
PC
Switch
Terminal A
(attacker)
PC
Terminal B
(victim)
ICMP Echo Reply
Because the IP address of terminal B is the source address of the ICMP Echo message, all the terminals in subnetwork C reply to terminal B.
PC
PC

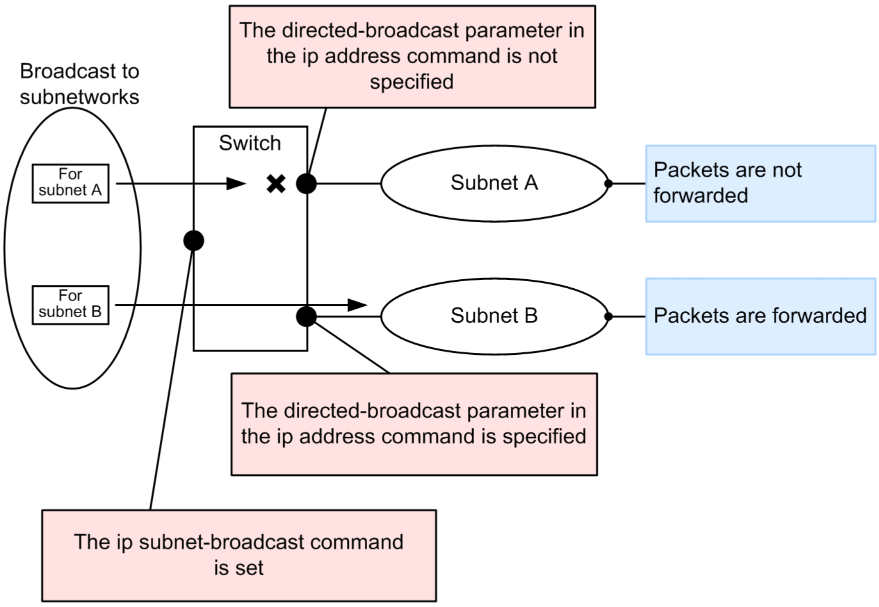
The switch forwards directed broadcast packets when the `ip subnet-broadcast` command has been set and the `directed-broadcast` parameter has been specified in the `ip address` command. The following table describes the relationship between these commands and the parameter. The figure after the table shows an example of executing these commands.

Table 1-5: Command settings and behavior

ip subnet-broadcast command	**ip address command**	
	directed-broadcast specified	**directed-broadcast not specified**
When default settings are used and ip subnet-broadcast is set	Y	N
When no ip subnet-broadcast is set	N	N

Legend: Y: Forwarded. N: Not forwarded.

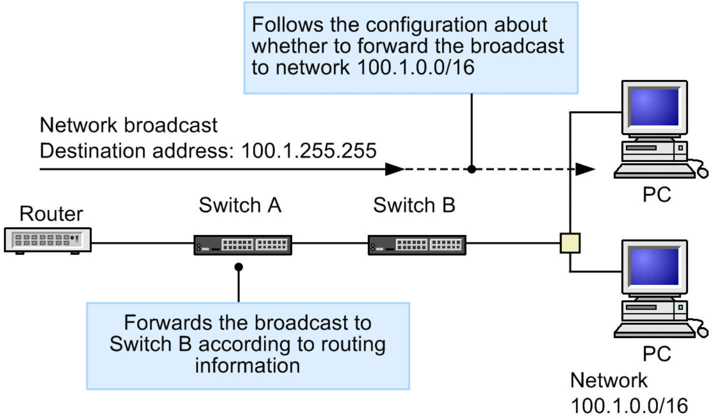
Figure 1-6: Example of executing the commands


The directed-broadcast parameter in the ip address command is not specified
Broadcast to subnetworks
Switch
For subnet A
For subnet B
Subnet A
Subnet B
Packets are not forwarded
Packets are forwarded
The directed-broadcast parameter in the ip address command is specified
The ip subnet-broadcast command is set

(1) Network broadcasts

A network broadcast is a broadcast to a network that is not divided into subnetworks. Suppose a network broadcast IP packet for destination 100.1.255.255 is sent to network 100.1.0.0/16. When the Switch is directly connected to network 100.1.0.0/16, the switch follows the switch settings for forwarding broadcasts in the configuration to determine whether to forward the network broadcast IP packet to the devices connected to the switch. The following figure shows how a network broadcast works.

Figure 1-7: Network broadcasts

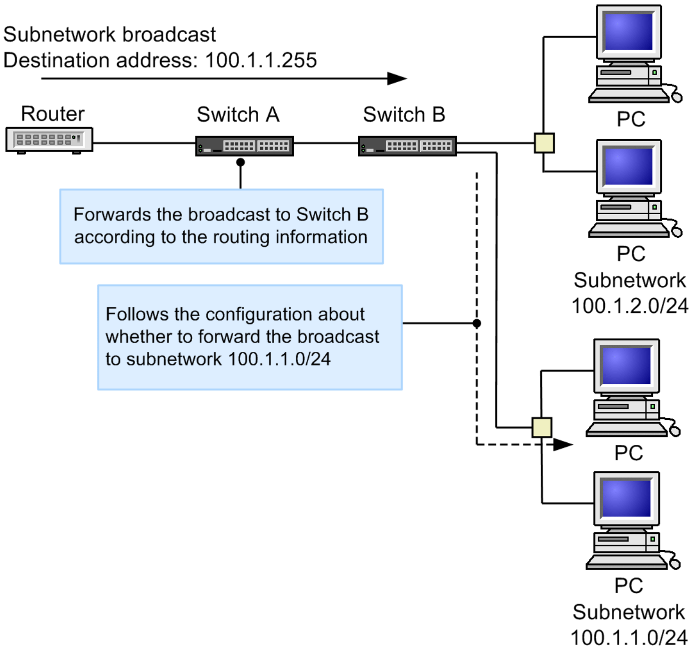

Follows the configuration about whether to forward the broadcast to network 100.1.0.0/16
Network broadcast
Destination address: 100.1.255.255
Router
Switch A
Switch B
PC
PC
Forwards the broadcast to Switch B according to routing information
Network 100.1.0.0/16

(2) Subnetwork broadcasts

A subnetwork broadcast is a broadcast to a single subnetwork.

For example, consider a case in which network 100.1.0.0/16 is divided into two subnetworks, 100.1.1.0/24 and 100.1.2.0/24, and a subnet broadcast IP packet (broadcast to subnetwork 100.1.1.0/24) is sent to destination 100.1.1.255. If the Switch is directly connected to subnetwork 100.1.1.0/24, the switch follows the switch settings for forwarding broadcasts in the configuration to determine whether to forward the subnetwork broadcast IP packet to the devices connected to the switch. The following figure shows how a subnetwork broadcast works.

Figure 1-8: Subnetwork broadcasts

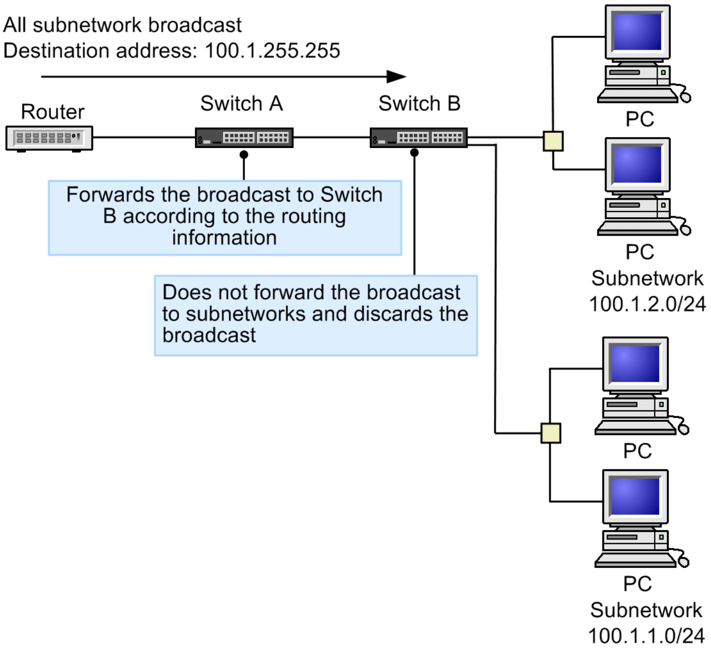

Subnetwork broadcast
Destination address: 100.1.1.255
Router
Switch A
Switch B
PC
PC
Subnetwork
100.1.2.0/24
Forwards the broadcast to Switch B according to the routing information
Follows the configuration about whether to forward the broadcast to subnetwork 100.1.1.0/24
PC
PC
Subnetwork
100.1.1.0/24

(3) All-subnetwork broadcasts

An all-subnetwork broadcast refers to a broadcast to all subnetworks. The Switch uses an all-subnetwork broadcast as a normal path.

If, for example, network 100.1.0.0/16 is divided into two subnetworks, 100.1.1.0/24 and 100.1.2.0/24, and an all-subnetwork broadcast IP packet is sent to destination 100.1.255.255, the packet reaches the Switch directly connected to subnetworks 100.1.1.0/24 and 100.1.2.0/24. However, the packet is not forwarded to subnetworks 100.1.1.0/24 and 100.1.2.0/24, and the switch discards the packet. If there is another corresponding path other than the default path, IP packets are sent via that path. The following figure shows how an all-subnetwork broadcast works.

Figure 1-9: All-subnetwork broadcasts


All subnetwork broadcast
Destination address: 100.1.255.255
Router
Switch A
Switch B
PC
PC
Subnetwork
100.1.2.0/24
Forwards the broadcast to Switch B according to the routing information
Does not forward the broadcast to subnetworks and discards the broadcast
PC
PC
Subnetwork
100.1.1.0/24

1.4.3 MTU and fragmentation

When forwarding IP packets, the Switch divides an IP packet that exceeds the maximum transfer unit (MTU) into smaller packets that do not exceed the MTU. This is called fragmentation. Packets that do not exceed the MTU are processed by the hardware. A packet that has been fragmented, however, is forwarded by the software in fragments, adversely affecting forwarding performance.

(1) Determining the MTU

- Determining the MTU for VLAN interfaces

Of the MTUs for the Ethernet interfaces belonging to a VLAN, the system MTU information, and the IP MTU information, the smallest MTU is used as the MTU for the VLAN interface.

In IPv4/IPv6 communication, the Switch uses the MTU for VLAN interfaces.

The following table describes how the MTU is determined for a VLAN interface.

Table 1-6: Determining the MTU for a VLAN interface

Settings	1	2	3	4	5	6	7	8
System MTU information	Set	Set	Set	Set	Omitted	Omitted	Omitted	Omitted
IP MTU information	Set	Set	Omitted	Omitted	Set	Set	Omitted	Omitted
Port MTU information	Set	Omitted	Set	Omitted	Set	Omitted	Set	Omitted
MTU value	A2	A1	A4	A1	A2	A3	A4	A5

Legend:

A1: The setting in the system MTU information and the setting in the IP MTU information are compared, and the smaller value is selected.

A2: The setting in the IP MTU information and the smallest value of the MTU values set for ports in the port MTU information are compared, and the smaller value is selected.

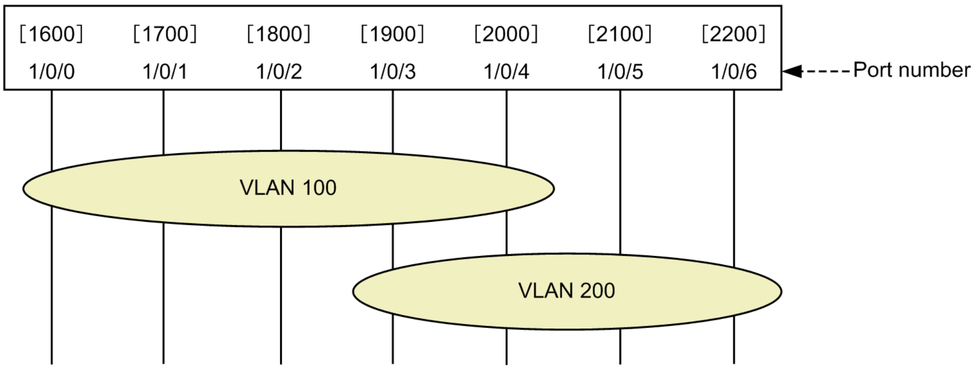
A3: The setting in the IP MTU information and 1500 are compared, and the smaller value is selected.

A4: Smallest value of the MTU values set for ports in the port MTU information

A5: 1500

Note: If the line type is 10BASE-T (full or half duplex) or 100BASE-TX (half duplex), the MTU is 1500 regardless of the settings.

Figure 1-10: Example of setting the MTU for VLAN interfaces



- When the IP MTU information does not exist

Determination of MTUs

MTU value for VLAN 100: 1600

MTU value for VLAN 200: 1900

- When the IP MTU information exists

When `ip mtu 1000` is set for VLAN 100 and `ip mtu 3000` is set for VLAN 200

Determination of MTUs

MTU value for VLAN 100: 1000

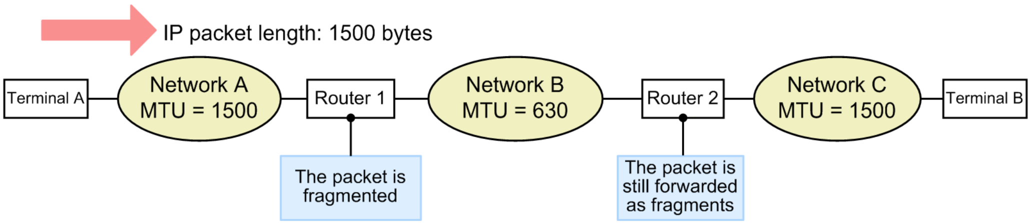
MTU value for VLAN 200: 1900

(2) MTU and fragmentation

A network can contain subnetworks with different MTUs. When a large IP packet passes through a network with a smaller MTU, the IP packet is fragmented before being forwarded.

The following figure illustrates fragmentation. In the figure, when a packet sent from network A is forwarded to network B, the packet is split into fragments because the MTU for network A is 1500 and the MTU for network B is 630.

Figure 1-11: Fragmentation example

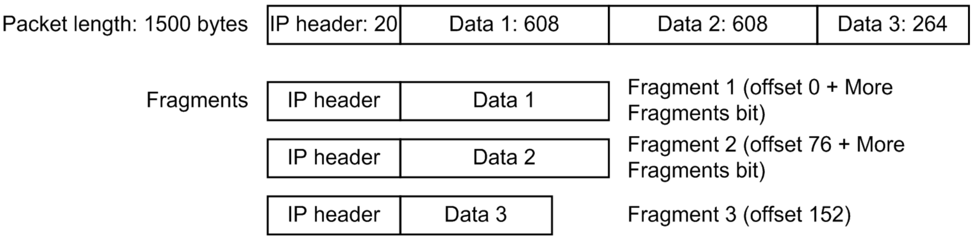


(3) Generating fragments

The data portion of an IP packet (the packet without the IP header) exceeding the MTU is divided into fragments that are multiples of 8.

Because the MTU of network B is 630, the size of a packet must be 610, excluding the IP header. Because the largest multiple of 8 that is less than 610 is 608, a packet is divided into 608-byte fragments, and each fragment is given an IP header. The following figure shows the fragmentation of a packet.

Figure 1-12: Fragmenting a packet



When an IP packet is split into fragments to fit the MTU, the fragmentation is described by the Fragment Offset field and the More Fragments bit in the IP header. The switch sets the same Identification field in all the IP headers and recalculates the checksum. The offset is calculated by dividing the data length from the beginning by 8.

(4) Reassembling the fragments

The fragments of an IP packet are reassembled at the end point based on the Identification and Fragment Offset fields and the More Fragments bit in the IP headers. Intermediate routers do not reassemble fragments. This is because routers are intended to route the individual fragments to the end point. If an intermediate router were to store fragments for reassembly into an IP packet, the stored fragments would be discarded if any fragments did not pass through the router.

1.5 Notes on using IPv4

(1) Notes on multihomed configurations

When you assign multiple IPv4 addresses to an interface, if terminals that belong to the same broadcast domain as the interface use different subnet addresses to communicate with one another, IPv4 forwarding can occur via the Switch.

If forwarding does occur, the hardware forwards the packets to the software so that the software can determine whether an ICMP Redirect message can be sent. Because this process significantly increases the load on the CPU of the Switch, note the following ways to prevent this problem:

- When terminals in the same broadcast domain are allowed to directly communicate with each other, use the same subnet for all terminals.
- If you need to assign different subnets to the terminals in the same broadcast domain for security reasons, we recommend that you avoid high CPU load by using the appropriate configuration command to stop the hardware from making a determination regarding whether ICMP Redirect messages can be sent.

Chapter

2. Settings and Operation for IP, ARP, and ICMP

This chapter describes how to configure an IPv4 network and how to check the network's status.

2.1 Configuration

2.1.1 List of configuration commands

The following table describes the configuration commands for IPv4.

Table 2-1: List of configuration commands

Command name	Description
arp	Creates a static ARP table.
arp discard-unresolved-packets	Reduces the CPU load by using the hardware to discard IPv4 forwarding packets with unresolved addresses.
arp max-send-count	Specifies the maximum number of times an ARP request frame is sent.
arp send-interval	Specifies the retry interval for sending an ARP request frame.
arp timeout	Specifies the aging time for an ARP cache table.
ip address	Specifies an IPv4 address for an interface.
ip icmp rate-limit unreachable	Specifies the interval for sending an ICMP error message.
ip local-proxy-arp	Specifies whether a local Proxy ARP reply can be returned.
ip mtu	Specifies the MTU length of IP packets sent on the interface.
ip proxy-arp	Specifies whether a Proxy ARP reply is possible.
ip redirects (global)	Specifies whether ICMP and ICMPv6 Redirect messages can be sent for the entire Switch.
ip redirects (interface)	Specifies whether an ICMP Redirect message can be sent for an interface.
ip routing	`no ip routing` invalidates IPv4 and IPv6 forwarding.
ip source-route	Specifies whether an IPv4 packet with the Source Route option can be forwarded.
ip subnet-broadcast	Specifies whether a subnet broadcast IPv4 packet can be forwarded. To forward broadcast packets, the `directed-broadcast` parameter must also be set in the `ip address` command.

2.1.2 Configuring an interface

Points to note

The example below shows how to set an IPv4 address for a VLAN. To do so, switch to interface configuration mode.

Command examples

1.
```
(config)# interface vlan 100
```
Switches to interface configuration mode for VLAN ID 100.

2.
```
(config-if)# ip address 192.168.1.1 255.255.255.0
```
Sets IPv4 address 192.168.1.1 and subnet mask 255.255.255.0 for VLAN ID 100.

2.1.3 Configuring a multihomed interface

Points to note

The example below shows how to set multiple IPv4 addresses for a VLAN. For the second and later IPv4 addresses, you need to specify the `secondary` parameter.

Command examples

1. `(config)# interface vlan 100`

   Switches to interface configuration mode for VLAN ID 100.

2. `(config-if)# ip address 192.168.1.1 255.255.255.0`

   Sets primary IPv4 address 192.168.1.1 and subnet mask 255.255.255.0 for VLAN ID 100.

3. `(config-if)# ip address 170.1.1.1 255.255.255.0 secondary`

   Sets secondary IPv4 address 170.1.1.1 and subnet mask 255.255.255.0 for VLAN ID 100.

2.1.4 Configuring directed broadcast forwarding

Points to note

To enable directed broadcast forwarding, you need to enable the `directed-broadcast` parameter for the `ip address` command. When subnet broadcast packet forwarding is disabled by using the `no ip subnet-broadcast` command, execute the `ip subnet-broadcast` command to enable subnet broadcast packet forwarding.

Command examples

1. `(config)# interface vlan 100`

   Switches to interface configuration mode for VLAN ID 100.

2. `(config-if)# ip subnet-broadcast`

   Enables the subnet broadcast packet forward option (this setting is required only when the `no ip subnet-broadcast` command has been executed).

3. `(config-if)# ip address 170.10.10.1 255.255.255.0 directed-broadcast`

   Sets primary IP address 170.10.10.1 and subnet mask 255.255.255.0 for VLAN ID 100, and enables direct broadcast IPv4 packet forwarding.

2.1.5 Configuring the loopback interface

Points to note

The example below shows how to set IPv4 addresses for identifying the Switches. Interface number 0 is dedicated to the global network. You can configure only one address.

Command examples

1. `(config)# interface loopback 0`

   Switches to interface configuration mode for the loopback interface.

2.
```
(config-if)# ip address 192.168.1.1
```

   Sets IP address 192.168.1.1 for the loopback interface.

2.1.6 Configuring static ARP

Points to note

The example below shows how to configure static ARP on the Switch.

You will need to specify an interface.

Command examples

1.
```
(config)# arp 123.10.1.1 interface vlan 100 0012.e240.0a00
```

   Sets the next hop IPv4 address 123.10.1.1 and sets destination MAC address 0012.e240.0a00 for VLAN ID 100 to configure static ARP.

2.2 Operation

2.2.1 List of operation commands

The following table describes the operation commands for IP, ARP, and ICMP.

Table 2-2: List of operation commands

Command name	Description
show ip-dual interface	Shows the status of IPv4 and IPv6 interfaces.
show ip interface	Shows the status of IPv4 interfaces.
show ip arp	Shows the information in the ARP entries.
clear arp-cache	Deletes dynamic ARP information.
show netstat (netstat)	Shows the status of a network.
clear netstat	Clears the network statistics counter.
clear tcp	Ends a TCP connection.
ping	Performs an Echo test.
traceroute	Shows a route.

2.2.2 Checking the up/down states for an IPv4 interface

After setting an IPv4 address for a line or a port on a line of the Switch connected to an IPv4 network, execute the `show ip interface` command to confirm that the state of the IPv4 interface is `UP`.

Figure 2-1: Example of displaying the IPv4 interface status

```
> show ip interface summary
vlan100 : UP 158.215.100.1/24
vlan200 : UP 123.10.1.1/24

>
```

2.2.3 Checking the reachability to the destination

Execute the `ping` command for an interface in the Switch connected to the IPv4 network to determine whether the destination device is reachable.

Figure 2-2: Results of executing the ping command (when the destination is reachable)

```
> ping 192.168.0.1
PING 192.168.0.1 (192.168.0.1): 56 data bytes
64 bytes from 192.168.1.51: icmp_seq=0 ttl=255 time=0.286 ms
64 bytes from 192.168.1.51: icmp_seq=1 ttl=255 time=0.271 ms
64 bytes from 192.168.1.51: icmp_seq=2 ttl=255 time=0.266 ms
^C
--- 192.168.0.1 PING Statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max = 0.266/0.274/0.286 ms
>
```

Figure 2-3: Results of executing the ping command (when the destination is not reachable)

```
> ping 192.168.0.1
PING 192.168.0.1 (192.168.0.1): 56 data bytes
^C
--- 192.168.0.1 PING Statistics ---
3 packets transmitted, 0 packets received, 100.0% packet loss
```

```
>
```

2.2.4 Checking the route to the destination

Execute the `traceroute` command to check the routers between the interface of the Switch connected to the IPv4 network and the destination.

Figure 2-4: Results of executing the traceroute command

```
> traceroute 192.168.0.1 numeric
traceroute to 192.168.0.1 (192.168.0.1), 30 hops max, 40 byte packets
1  192.168.2.101  0.612 ms  0.541 ms  0.532 ms
2  192.168.1.51  0.905 ms  0.816 ms  0.807 ms
3  192.168.0.1  1.325 ms  1.236 ms  1.227 ms
>
```

2.2.5 Checking ARP information

After setting an IPv4 address for a line or a port on a line in the Switch connected to the IPv4 network, execute the `show ip arp` command to check whether the addresses are resolved between the Switch and the neighboring devices (that is, whether ARP entries exist).

Figure 2-5: Results of executing the show ip arp command

```
> show ip arp interface vlan 100
Date 20XX/10/25 14:00 UTC
Total: 3 entries
 IP Address    Linklayer Address  Netif      Expire    Type
 192.168.2.101 0012.e240.0a00     VLAN0100   Static    arpa
 192.168.1.51  0012.e240.0a01     VLAN0100   Static    arpa
 192.168.0.1   0012.e240.0a02     VLAN0100   3h30m0s   arpa
```

Chapter
3. Null Interface (IPv4)

This chapter describes the null interface for an IPv4 network and how to use it.

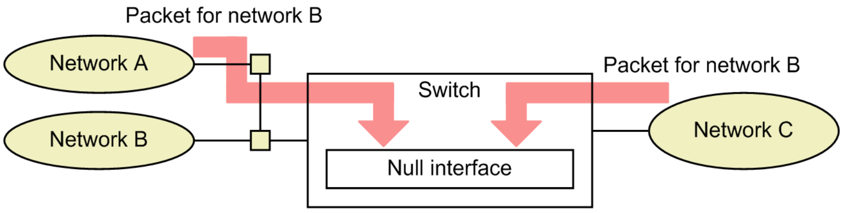
3.1 Description

The null interface is a virtual interface for discarding packets that does not depend on physical lines. You can discard packets by specifying the null interface as the output destination of specific traffic.

The null interface is always enabled and neither forwards nor receives traffic. The null interface does not report the discarding of packets to the source by sending an ICMP Unreachable message. The null interface is not used to discard multicast packets.

With the null interface, you can restrict traffic to a specific network or terminal that passes through the Switch. In the following figure, all the traffic sent to network B via the Switch is directed to the null interface so that it will be discarded.

Figure 3-1: Network configuration with the null interface



The null interface function is part of static routing. When you use the null interface to discard packets, you need to specify a static route to set the null interface as the output destination.

When the Switch searches for a route, the Switch internally discards and does not forward packets it determines are directed to the null interface (packets that are routed to the null interface based on a static route).

For details about static routing and route control, see *8. Static Routing (IPv4)* through *12. BGP4 [OS-L3SA]*.

The switch also provides filtering to discard packets based on conditions specified for each interface. However, the null interface is useful because it allows the Switch to discard unnecessary packets all at once when you specify simply a static route to the null interface for specific traffic.

The following table describes the function that discards packets when both the null interface and filtering are used.

Table 3-1: Function that discards packets when both the null interface and filtering are used

Routing information	Filtering settings	Operation	Discard mechanism
Directed to the null interface	Forward	Discard	Null interface
	Discard	Discard	Filtering
Directed to a non-null interface	Forward	Forward	--
	Discard	Discard	Filtering

Legend: --: Not applicable

3.2 Configuration

3.2.1 List of configuration commands

The following table describes the configuration commands for the null interface (IPv4).

Table 3-2: List of configuration commands

Command name	Description
interface null	Accesses the null interface.
ip route#	Generates an IPv4 static route.

#

See *10. Static Routing (IPv4)* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

3.2.2 Configuring a route to the null interface

Points to note

The example below shows how to set the null interface to discard the packets sent to a specific network or terminal via the Switch.

Command examples

1. `(config)# interface null 0`

   Sets the null interface.

2. `(config)# ip route 10.0.0.0 255.0.0.0 null 0`

   Specifies the null interface as the next hop of static route 10.0.0.0/8. When the packets sent to this network pass through the Switch, the packets are not forwarded and instead are sent to the null interface for discarding.

3.3 Operation

3.3.1 List of operation commands

The following table describes the operation command for the null interface (IPv4).

Table 3-3: List of operation commands

Command name	Description
show ip route#	Shows routing information stored in the routing table.

#

See *6. IPv4 Routing Protocols* in the manual *Operation Command Reference Vol. 2 For Version 11.10*.

3.3.2 Checking the null interface

You can check the following information when you use the null interface in the Switch.

(1) Check after configuration

(a) Checking the routing information

Execute the `show ip route` command to check whether the routing information you set by using the `static` configuration command is correct.

Figure 3-2: Displaying the routing information related to the null interface

```
> show ip route static
Total: 1 routes
Destination        Next Hop        Interface     Metric  Protocol  Age
172.16.251.89/32   ----            null0         0/0     Static     1m  9s
>
```

Chapter

4. Policy-based Routing (IPv4) [OS-L3SA]

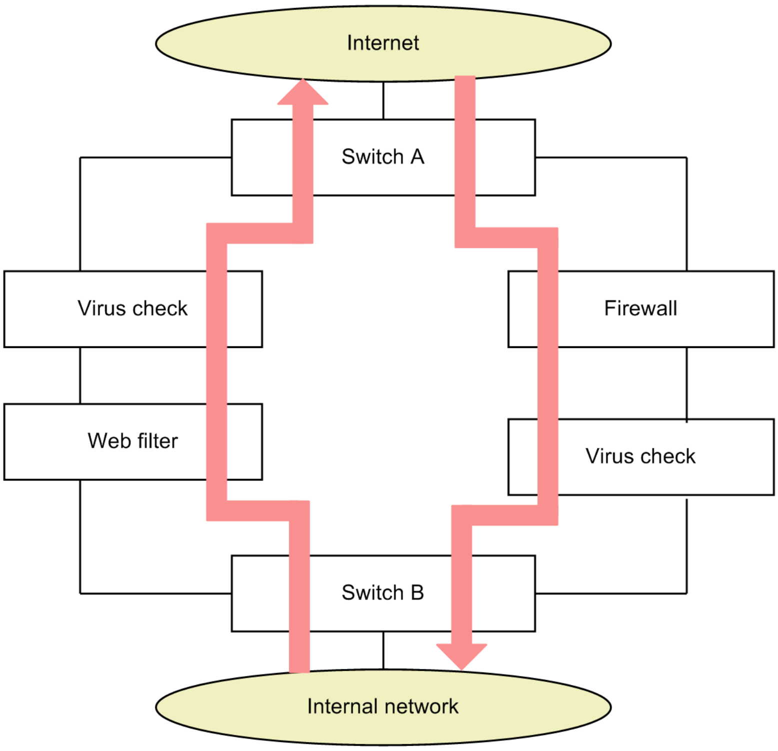
This chapter describes policy-based routing (IPv4) and its use.

4.1 Description

Policy-based routing forwards packets over Layer 3 to user-defined destination devices, instead of following routing information registered in a routing protocol or by configuration commands.

The following figure shows a policy-based routing configuration example.

Figure 4-1: Policy-based routing configuration example



Switch A forwards all packets received from the Internet to the firewall and antivirus security devices. Switch B forwards all packets received from the internal network to filtering and antivirus devices. Different security checks can be used for sent and received packets. This configuration can also be applied to load distribution and authentication.

Thus, policy-based routing can forward packets according to user-defined routes irrespective of routing information.

4.1.1 Policy-based routing control

Policy-based routing is used as part of the filter functionality.

Packets that match the filtering flow detection conditions set for the receiving-side VLAN interface are forwarded according to the policy-based routing settings specified in the filter entry.

Policy-based routing is enabled when layer3-6 is specified for the receiving-side flow detection mode. Packets that are forwarded over Layer 3 are subject to policy-based routing.

To enable policy-based routing, specify policy-based routing list information that contains routing information registered for access list operation. The functionality enabled by this specification is called a policy-based routing group.

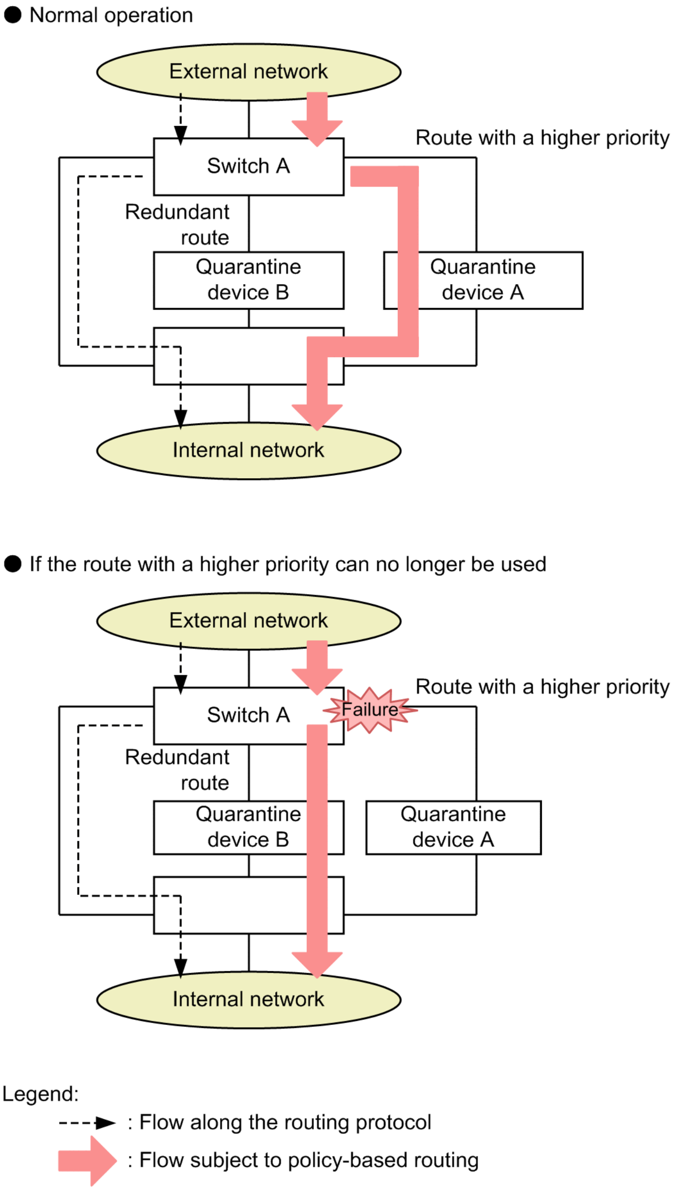
4.1.2 Policy-based routing group

A policy-based routing group allows you to define a group of routes. Priority is assigned to each route according to the specified application order. This allows a route with a higher priority to be dynamically selected from multiple routes according to the status of the destination interface and forwarding destination route. Information for a group of routes is called policy-based routing list information.

Specifying multiple routes in policy-based routing list information provides redundant routes. If the highest priority route can no longer be used for forwarding due to a failure, operation continues by switching to the next highest priority route in the same policy-based routing list information.

The following figure shows an example of a policy-based routing group configuration.

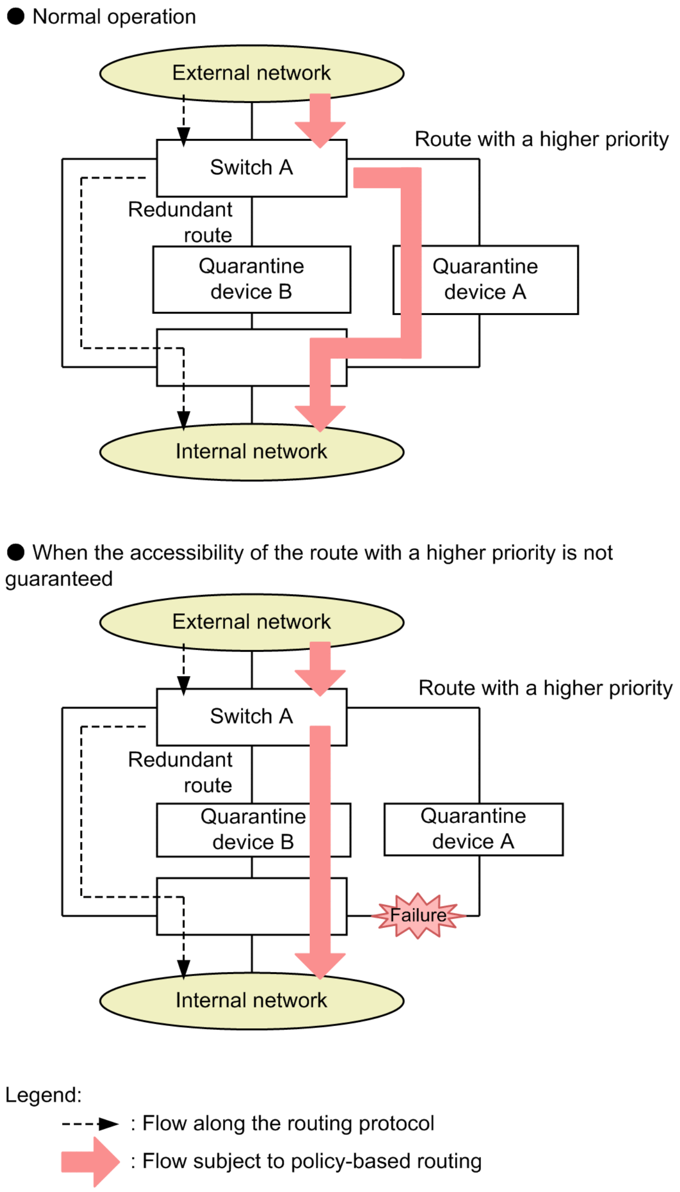
Figure 4-2: Policy-based routing group configuration example



The policy-based routing group can also be linked with the polling monitoring tracking function, to monitor routes to a location subject to polling monitoring. Polling monitoring of the tracking function monitors whether communication with devices on the network is possible. Monitoring results are used to determine the route to be selected based on the policy-based routing group. This allows route switching in response to failures that occur between the Switch and neighboring devices and in other routes.

The following figure shows a configuration example when a policy-based routing group is linked with the tracking functionality.

Figure 4-3: Configuration example when policy-based routing group is linked with the tracking functionality



(1) Route selection by a policy-based routing group

The policy-based routing group selects a route from the routes registered in the policy-based routing list information based on the following information:

- Monitoring results to determine priority and whether forwarding is possible
- Default operation specifications
- Switch-back operation specifications

(a) Monitoring results to determine priority and whether forwarding is possible

The results of the following monitored items are used to determine whether the routes registered in the policy-based routing list information can be used for forwarding:

- Monitoring of the destination interface VLAN status

- Polling monitoring by the tracking functionality

The route with the highest priority is selected from the routes that can be used for forwarding.

■ Monitoring of the destination interface VLAN status

If the forwarding destination route is specified by using the following configuration command, the possibility of forwarding is determined based on the status of the destination interface VLAN:

- `policy-interface` command

  VLAN ID (`vlan` parameter) and next-hop address (`next-hop` parameter) of the destination interface

Forwarding is possible only when the destination interface VLAN status is `Up`.

■ Polling monitoring based on tracking functionality

If the forwarding destination route is specified by using the following configuration command, the possibility of forwarding is determined based on the results of polling monitoring by the tracking functionality in addition to the status of the destination interface VLAN:

- `policy-interface` command

  VLAN ID (`vlan` parameter), next-hop address (`next-hop` parameter), and track ID (`track-object` parameter) of the destination interface

Forwarding is determined to be possible only when both the destination interface VLAN status and the result of polling monitoring are `Up`.

For details about the tracking functionality, see *4.1.5 Policy-based routing tracking functionality*.

■ Determination by priority

Route selection is based on the results of monitoring the destination interface VLAN status or polling monitoring by the tracking functionality. The route with the highest priority is selected based on the application order specified in the configuration, from the routes that can be used for forwarding in the policy-based routing list information.

(b) Default operation specification

The default operation is performed when none of the routes registered in the policy-based routing list information can be used for forwarding, or no routes are registered. The default operation can be specified in the `default` configuration command. The following table describes the possible default operation specifications.

Table 4-1: Default operation specification

Specification in the configuration command	Default operation	Description
`permit`	Forward normally	The target packet is forwarded over Layer 3 according to the routing protocol.
`deny`	Discard	The target packet is discarded.
Not specified	Discard	The target packet is discarded.

Packets that are forwarded over Layer 3 according to the routing protocol or discarded by the default operation are counted in statistics for the access list of the specified target policy-based routing list information.

The following packets cannot be discarded by the default operation:

- An IPv4 packet exceeding the MTU
- A packet whose TTL is set to 1

- An IPv4 packet with an unknown destination

(c) Path switch-back operation specification

If a route with a higher priority can no longer be used for forwarding, a route with a lower priority is used instead. The operation performed when the route with a higher priority becomes available for forwarding again is called the path switch-back operation. The path switch-back operation can be specified in the `recover` configuration command. The following table describes the possible path switch-back operation specifications.

Table 4-2: Path switch-back operation specification

Specification in the configuration command	Path switch-back operation	Description
`on`	Switch back	If the route with a higher priority becomes available for forwarding, the route is switched back.
`off`	Do not switch back	Even if the route with a higher priority becomes available for forwarding, the route is not switched back.
Not specified	Switch back	If the route with a higher priority becomes available for forwarding, the route is switched back.

If `Switch back` is specified for the path switch-back operation, the route with the highest priority is always selected from the forwarding-enabled routes in the policy-based routing list information.

If `Do not switch back` is specified for the path switch-back operation, the route is not switched back even if a route with a higher priority than the currently selected route becomes available for forwarding. If the selected route can no longer be used for forwarding, the route is always switched to a route with a lower priority. If none of the routes registered in the policy-based routing list information can be used for forwarding, the default operation is performed instead of switching back the route. However, the route with the highest priority among the forwarding-enabled routes in the policy-based routing list information is selected again in the following cases:

- The `reset policy-list` operation command is executed.
- The `recover` configuration command is used to change the path switch-back operation to `Switch back`.
- The policy-based program is restarted.

(2) Policy-based routing group at startup

When the Switch starts or restarts, monitoring of forwarding availability and route switching are stopped for a certain period of time after the policy-based program starts. This is because the results of monitoring of the forwarding availability are unstable for the following reasons until the device state after startup has been collected:

- The VLAN interface is not in the `Up` state.
- The result of polling monitoring by the tracking functionality is not `Up`.

The state of a route after the policy-based program starts until the policy-based routing group starts monitoring the forwarding availability is called the starting state. In the starting state, any packets subject to policy-based routing are discarded.

If you use any of the following configuration commands to change the policy-based routing list information while in the starting state, the change is applied when monitoring of the forwarding availability starts:

- default (policy-list)

- policy-interface (policy-list)
- policy-list
- recover (policy-list)

You can use the `policy-list default-init-interval` configuration command to change the time that can elapse from when the policy-based program starts, until the start of forwarding availability monitoring (no-forwarding availability monitoring period). Specify the time required to obtain stable results of forwarding availability monitoring after the program starts. The following table describes the transitions of the starting state and describes the transition conditions.

Table 4-3: Transition of starting state and transition conditions

State transition	Transition conditions
Starting state exited	The starting state exits when the no-forwarding availability monitoring period expires. This state also exits when an operation to interrupt the startup is performed.
Starting state interrupted	The starting state is interrupted when any of the following is performed: • The `reset policy-list` operation command is executed. • The no-forwarding availability monitoring period is changed to be shorter than the currently elapsed time in the configuration. • The policy-based program is restarted.
Starting state extended	The starting state is extended when the no-forwarding availability monitoring period is changed to a longer time in the configuration. In this case, forwarding availability monitoring starts when the time obtained by subtracting the currently elapsed time from the new time period has elapsed.

If the starting state exits or is interrupted, the route with the highest priority is selected from the routes that can be used for forwarding in the policy-based routing list information.

4.1.3 Packets subject to policy-based routing

The following table describes the packets subject to policy-based routing.

Table 4-4: Packets subject to policy-based routing

Packet type	Address type	Applicable?
IPv4 packet	Unicast address	Y
	Multicast address	N
	Restricted broadcast address#1	N
	Subnet broadcast address	N#2
	Incoming IP packet address	Y
	Outgoing IP packet address	N
Other than IPv4 packets		N

Legend: Y: Applicable, N: Not applicable

#1

Indicates an IP broadcast address of the format 255.255.255.255 or 0.0.0.0.

#2

The packet is subject to policy-based routing if a flow that contains a subnet broadcast packet is detected.

4.1.4 Address types that can be set as a next hop

The following table describes the IP address types that can be set as a next hop for policy-based routing.

Table 4-5: IP address types that can be set as a next hop for policy-based routing

Address type	Setting
Unicast address (including the receiving interface)	Y#
IP address set for the destination interface	N
Multicast address	N
Restricted broadcast address	N
Direct broadcast address to the network connected to the destination interface	N
Internal loopback address (127.0.0.0)	N

Legend: Y: Can be set, N: Cannot be set

#: Only the addresses in the same network as the address set for the destination interface can be set.

4.1.5 Policy-based routing tracking functionality

The policy-based routing tracking functionality sends polling packets to the devices to be tracked on the network, and sets the track state to `Up` if communication is possible.

The Switch sends a polling packet at regular intervals to a device to be tracked, and monitors whether a response packet returns. If a response packet is received, polling is assumed to be successful. If polling is successful for a predefined consecutive number of times, the track state is set to `Up`. If a response packet is not received, polling is assumed to have failed. If polling fails a predefined number of times, the track state is set to `Down`.

(1) IPv4 ICMP polling monitoring

For IPv4 ICMP polling monitoring, an IPv4 address is used to specify a device to be tracked on the network. This function sends an IPv4 ICMP Echo packet as a polling packet to the IPv4 address to be tracked, and monitors whether an IPv4 ICMP Echo Reply packet returns as a polling response packet.

(2) Polling results and verification sequence

Polling monitoring periodically sends a polling packet, and then waits the specified period of time for a response packet. If a response packet is received within the wait time, polling is successful. If the response wait time expires before a packet is received, polling fails.

However, in the network, packets might be temporarily discarded although communication is possible, or communication might be temporarily permitted even in a state that disables communication. If you apply polling monitoring to such a network and reflect the polling results directly to the track state, the track state might be unstable.

Therefore, polling monitoring of the Switch has a verification period before the polling results are applied to the track state. During the verification period, the current track state is retained, while the polling results are monitored to determine whether the track state can be changed. The verification period prevents the track state from changing unexpectedly when communication is unstable. Note that you can adjust the verification period by specifying the number of times polling is performed and the polling interval. The number of times polling is performed and polling interval required to change the track state can be specified for each track.

The following table describes the items that can be set for polling monitoring tracks. Each item is specified by a configuration command parameter.

Table 4-6: Items that can be specified for polling monitoring tracks

Item	Description	Default value
Response wait time	Time to wait after a polling packet is sent until a response packet is received	2 seconds
Polling interval	Interval for sending a polling packet being used for a purpose other than verification	6 seconds
Successful polling count for setting the track state `Up`	Required number of times polling is successful to determine that the track state is `Up` during failure recovery verification	4 times
Maximum number of polling attempts during failure recovery verification	Maximum number of times polling is attempted to continue failure recovery verification	5 times
Polling attempt interval for failure recovery verification	Interval for sending polling packets during failure recovery verification	2 seconds
Failed polling count for setting the track state `Down`	Required number of times polling failed in order to determine that the track state is `Down` during failure verification	4 times
Number of polling attempts during failure verification	Maximum number of times polling is attempted to continue failure verification	5 times
Polling attempt interval for failure verification	Interval for sending polling packets during failure verification	2 seconds

In addition to the track state, the operating status of the track is provided. The track operating status during the verification period is called `Checking`, and any other status (except for the status after the Switch is started until track monitoring starts) is called `Running`.

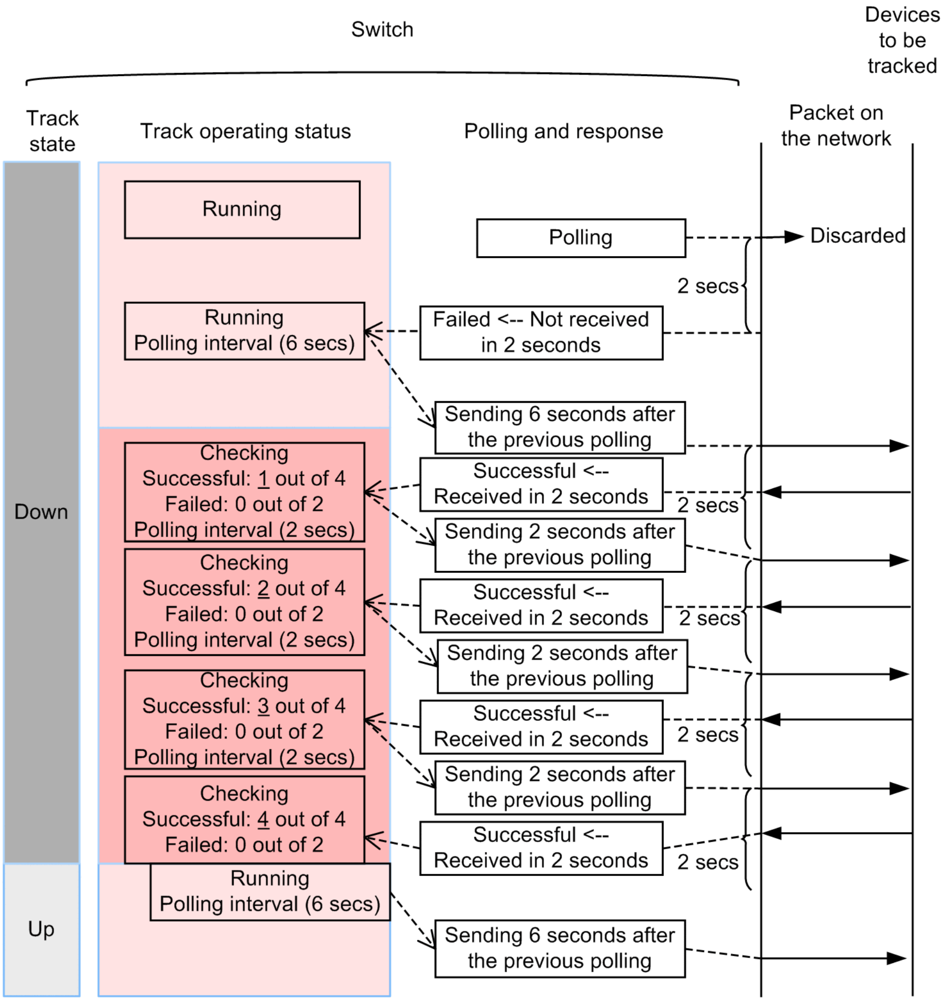
(a) During failure recovery verification

If the track state is `Down`, the track operating status is `Running` as long as the same response (failed) returns as a result of polling. While in the `Running` status, polling packets are sent at the specified polling interval.

If polling is successful when the track state is `Down`, verification starts to determine whether to change the track state to `Up`. This is called failure recovery verification.

The following figure shows the failure recovery verification sequence. This figure and its description use the default values for all items.

Figure 4-4: Failure recovery verification sequence (example when "Up" is determined as a result of verification)


Switch
Devices to be tracked
Track state
Track operating status
Polling and response
Packet on the network
Running
Polling
Discarded
2 secs
Running Polling interval (6 secs)
Failed <-- Not received in 2 seconds
Sending 6 seconds after the previous polling
Down
Checking Successful: 1 out of 4 Failed: 0 out of 2 Polling interval (2 secs)
Successful <-- Received in 2 seconds
2 secs
Sending 2 seconds after the previous polling
Checking Successful: 2 out of 4 Failed: 0 out of 2 Polling interval (2 secs)
Successful <-- Received in 2 seconds
2 secs
Sending 2 seconds after the previous polling
Checking Successful: 3 out of 4 Failed: 0 out of 2 Polling interval (2 secs)
Successful <-- Received in 2 seconds
2 secs
Sending 2 seconds after the previous polling
Checking Successful: 4 out of 4 Failed: 0 out of 2
Successful <-- Received in 2 seconds
2 secs
Up
Running Polling interval (6 secs)
Sending 6 seconds after the previous polling

When the track state remains `Down`, the track operating status changes to `Checking` and failure recovery verification starts. The polling packet sending interval during failure recovery verification is set to 2 seconds (the default value).

During failure recovery verification, if polling is successful 4 consecutive times (the default successful polling count), the track state changes to `Up`, the track operating status changes to `Running`, and failure recovery verification terminates. Note that the successful polling count includes the successful polling that triggered failure recovery verification.

If polling fails 2 times (a value that is obtained as a result of 5 - 4 + 1 = $2^{\#}$), the track state remains `Down`, the track operating status changes to `Running`, and failure recovery verification terminates.

#: This expression means subtracting the successful polling count for setting the track state `Up` (4) from the number of polling attempts during failure recovery verification (5), and then adding 1 to the result.

Thus, irrespective of the changes in the track state, failure recovery verification terminates before polling exceeds 5 times (which is the default maximum number of polling attempts during failure recovery verification).

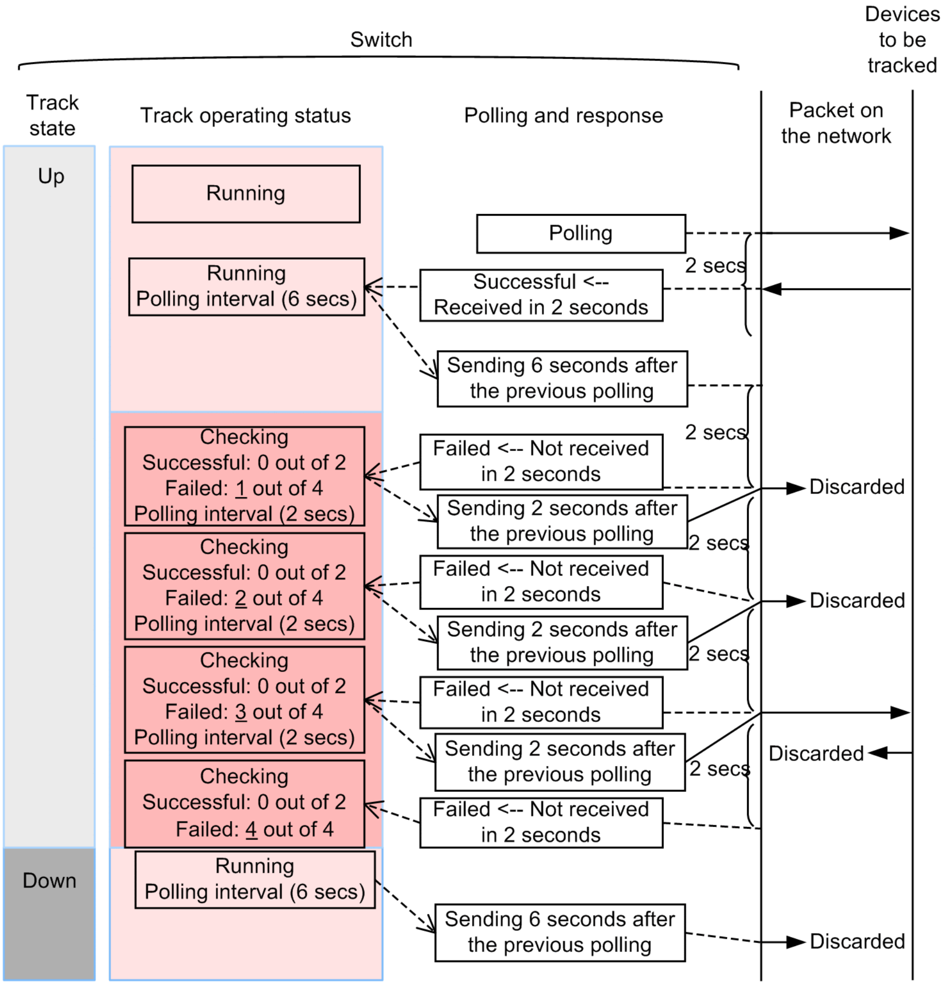
(b) During failure verification

If the track state is `Up`, the track operating status is `Running` as long as the same response (successful) returns as a result of polling. While in the `Running` status, polling packets are sent at the specified polling interval.

If polling fails when the track state is `Up`, verification starts to determine whether to change the track state to `Down`. This is called failure verification.

The following figure shows the failure verification sequence. This figure and its description use the default values for all items.

Figure 4-5: Failure verification sequence (example when "Down" is determined as a result of verification)


Switch
Devices to be tracked
Track state
Track operating status
Polling and response
Packet on the network
Up
Running
Polling
Running
Polling interval (6 secs)
Successful <-- Received in 2 seconds
2 secs
Sending 6 seconds after the previous polling
2 secs
Checking
Successful: 0 out of 2
Failed: 1 out of 4
Polling interval (2 secs)
Failed <-- Not received in 2 seconds
Discarded
Sending 2 seconds after the previous polling
2 secs
Checking
Successful: 0 out of 2
Failed: 2 out of 4
Polling interval (2 secs)
Failed <-- Not received in 2 seconds
Discarded
Sending 2 seconds after the previous polling
2 secs
Checking
Successful: 0 out of 2
Failed: 3 out of 4
Polling interval (2 secs)
Failed <-- Not received in 2 seconds
Discarded
Sending 2 seconds after the previous polling
2 secs
Checking
Successful: 0 out of 2
Failed: 4 out of 4
Failed <-- Not received in 2 seconds
Down
Running
Polling interval (6 secs)
Sending 6 seconds after the previous polling
Discarded

When the track state remains `Up`, the track operating status changes to `Checking`, and failure verification starts. The polling packet sending interval during failure recovery verification is set to 2 seconds (the default value).

During failure verification, if polling failed 4 consecutive times (the default failed polling count), the track state changes to `Down`, the track operating status changes to `Running`, and failure verification terminates. Note that the failed polling count includes the failed polling that triggered failure verification.

If polling is successful 2 times (a value that is obtained as a result of 5 - 4 + 1 = 2[#]), the track state remains `Up`, the track operating status changes to `Running`, and failure verification terminates.

#: This expression means subtracting the failed polling count for setting the track state `Down` (4) from the number of polling attempts during failure verification (5), and then adding 1 to the result.

Thus, irrespective of the changes in the track state, failure verification terminates before polling exceeds 5 times (which is the default maximum number of polling attempts during failure verification).

(3) Notes on polling monitoring

- When you use tracks linked with the policy-based routing group, we recommend that you specify a value other than 1 for the failed polling count for setting the track state `Down`. This is because linking the tracks for which the failed polling count is set to 1 might make control unstable due to network conditions.

  Specify 1 for this failed polling count only when you want to use operation logs or traps to notify the network administrator, even if polling fails only once.

- Make sure that the polling interval, polling attempt interval during failure recovery verification, and polling attempt interval during failure verification are longer than the response wait time. This is because the next polling packet is not sent until the previous polling result is determined as either a failure or success.

  Even if the specified polling interval is shorter than the response wait time, the next polling packet is not sent until a response packet is received (if polling is successful) or the response wait time elapses (if polling fails).

- The sum of the maximum frequency for sending polling packets for all polling monitoring tracks is 100 packets per second (pps). Considering that the polling interval varies while the track operating status is `Checking`, set up the configuration so that the sending frequency does not exceed 100 pps.

  If more than 100 packets are to be sent per second, packets exceeding 100 are held until the next second. In a configuration that allows more than 100 pps, the polling intervals for all tracks are increased so that the sending frequency is within 100 pps.

4.1.6 Tracks for tracking functionality

The policy-based routing tracking functionality allows you to use the configuration commands to stop track operation and specify the track state of the stopped track. Thus, when adding or changing the track configuration, you can fix the track state to minimize the impact on route selection of the linked policy-based routing group, and start track operation at a time when route selection is less affected.

(1) Default track state

The default track state is applied when a track is stopped. You can use the `default-state` configuration command to apply the default track state to each track. The default track state is `Down` for tracks that are not specified by the configuration command.

The configuration of the default track state is valid even for stopped tracks and tracks with incomplete configuration. Therefore, if you specify the default track state in advance, you can change the track configuration without affecting route selection of the policy-based routing group linked with tracks.

(2) Stopping a track by using a configuration command

Specify the `disable` configuration command to stop operation for a track. The track state of the stopped track is the default track state.

(3) Tracks with incomplete configuration

Tracks with incomplete configuration are those tracks for which the track type is not specified, that is, tracks for which the `track-object` or `type icmp` configuration command value is not specified.

Even tracks with incomplete configuration can be linked with the policy-based routing group. If such a track is linked with the policy-based routing group, the default state of that track is used as the result of polling monitoring of the tracking functionality, to determine whether the route can be used for forwarding.

(4) Track operation at startup of the Switch

When the Switch starts or restarts, the polling monitoring track stops for a certain period of time. This is because monitoring by polling cannot be performed immediately after startup of the Switch for the following reasons:

- The VLAN interface is not in the `Up` state.
- The routes are not stable.

The track operating status after the Switch is started or restarted until polling monitoring track starts operation is called Starting. In the Starting status, the default track state is set.

The time that can elapse before the polling monitoring track starts operation after the Switch is started or restarted can be changed for each device by using the `track-object default-init-interval` configuration command. Specify the time required until communication using the polling monitoring track is stable after the Switch is started or restarted. The default is 180 seconds.

4.1.7 Notes on policy-based routing

(1) Setting forwarding destination routes for policy-based routing

If the ARP information for the next-hop address to be specified for policy-based routing has not been registered on the Switch, packets subject to policy-based routing will be discarded. To use the policy-based routing, perform either of the following:

- Set a static ARP for the next-hop address to be specified for policy-based routing and set the MAC address static entry.
- Link the routes registered in the policy-based routing list information with polling monitoring of the tracking functionality.

(2) Using policy-based routing with DHCP snooping

See *13.1.7 Notes on using DHCP snooping* in the manual *Configuration Guide Vol. 2 For Version 11.10*.

(3) Packets that cannot be forwarded by policy-based routing

The packets listed below are detected and counted as statistics by using an access list in which policy-based routing list information has been set. However, the following packets are discarded because they cannot be forwarded by policy-based routing:

- Frames discarded by Layer 2 authentication
- Frames discarded by DHCP snooping
- Packets discarded by flow control

(4) Packets not subject to policy-based routing

The following packets are not subject to policy-based routing because they cannot be detected by using an access list in which policy-based routing list information has been set:

- Frames that are discarded because the data transfer status of the VLAN port is `Blocking` (data transfer is stopped)
- Frames discarded when a receiving-side interface to which an access list with policy-based routing list information set is applied and the policy-based routing destination interface are blocked by the inter-port relay blocking functionality
- Untagged frames received when native VLANs are not set as VLANs that use a trunk port for receiving frames
- Tagged frames that are not set for VLANs that use a trunk port for receiving frames
- Tagged frames received at an access port, protocol port, or MAC port

- Frames discarded by the MAC address learning functionality
- Packets discarded due to a check error when the validity of the IP packet header is checked
- Packets discarded by hardware because their addresses cannot be resolved
- Packets discarded by a null interface
- Packets discarded because the `no ip routing` configuration command disabled the IPv4 and IPv6 forwarding functionality

(5) Using policy-based routing with sFlow statistics functionality

If packets are subject to both sFlow statistics and policy-based routing, the following information is collected in sFlow statistics as routing information for the forwarding destination based on the routing protocol (but not for the forwarding destination of policy-based routing):

- Router type formats `nexthop` and `dst_mask`
- Gateway type formats `dst_peer_as` and `dst_as`

(6) Using policy-based routing with flow control

If a packet subject to policy-based routing is detected by a QoS flow list, both forwarding by policy-based routing and flow control configured in the QoS flow list are enabled.

(7) If ICMP redirection packets are subject to policy-based routing

If ICMP redirection packets that are to be redirected to the forwarding destination of policy-based routing are subject to policy-based routing, CPU might be heavily loaded.

(8) Policy-based routing and MTU

If the MTU of a receiving-side interface to which an access list with policy-based routing list information set is applied is greater than the MTU of the policy-based routing destination interface, policy-based routing might be disabled. To use policy-based routing, make sure that the MTU of the receiving-side interface is no larger than the MTU of the sending-side interface.

4.2 Configuration

4.2.1 List of configuration commands

The following table describes the configuration commands for policy-based routing.

Table 4-7: List of configuration commands

Command	Description
default	Sets the default operation for policy-based routing list information.
policy-interface	Sets a route in policy-based routing list information.
policy-list	Sets policy-based routing list information.
policy-list default-init-interval	Sets the interval during which the following monitoring is stopped: the monitoring of whether the forwarding of policy-based routing (for example, during startup of a device) is possible.
policy-list resequence	Resets the value that controls the sequence in which policy-based routing routes are applied.
recover	Sets the path switch-back operation for policy-based routing list information.
access-list#	Sets an access list used as an IPv4 filter.
ip access-group#	Applies an IPv4 filter to a VLAN interface, and enables IPv4 filter functionality.
ip access-list extended#	Sets an access list used as an IPv4 packet filter.
permit#	Specifies conditions by which a filter permits access.

#

See *19. Access Lists* in the manual *Configuration Command Reference Vol. 1 For Version 11.10*.

The following table describes the configuration commands for the policy-based routing tracking functionality.

Table 4-8: List of configuration commands (policy-based routing tracking functionality)

Command	Description
default-state	Sets the default track state.
disable	Stops track operation.
failure detection	Sets the number of times polling is performed and the polling interval during failure verification.
interval	Sets the polling interval.
recovery detection	Sets the number of times polling is performed and the polling interval during failure recovery verification.
timeout	Sets the polling response wait time.
track-object	Sets tracks for the tracking functionality.
type icmp	Sets IPv4 ICMP polling monitoring as the track type.

4.2.2 Configuring policy-based routing

The following provides an example of configuring policy-based routing.

(1) Setting a policy-based routing group

The following example sets policy-based routing list information by using IPv4 packets as flow detection conditions.

Points to note

The example below shows how to use an access list to set policy-based routing list information.

Command examples

1.
```
(config)# mac-address-table static 0012.e200.1122 vlan 100
interface gigabitethernet 1/0/1
(config)# arp 192.168.1.1 interface vlan 100 0012.e200.1122
```
Sets the next-hop IPv4 address 192.168.1.1 and destination MAC address 0012.e200.1122 for VLAN 100 to configure static entry.

2.
```
(config)# mac-address-table static 0012.e200.3344 vlan 200
interface gigabitethernet 1/0/2
(config)# arp 192.168.2.1 interface vlan 200 0012.e200.3344
```
Sets the next-hop IPv4 address192.168.2.1 and destination MAC address 0012.e200.3344 for VLAN 200 to configure static entry.

3.
```
(config)# policy-list 10
```
Sets policy-based routing list information with list number 10. When this list is created, the command switches to policy-based routing list information mode.

4.
```
(config-pol)# policy-interface vlan 100 next-hop 192.168.1.1
```
Sets VLAN 100 and the next-hop address 192.168.1.1 as the route with the highest priority in the policy-based routing list information.

5.
```
(config-pol)# policy-interface vlan 200 next-hop 192.168.2.1
```
Sets VLAN 200 and the next-hop address 192.168.2.1 as a redundant route in the policy-based routing list information.

6.
```
(config-pol)# default permit
```
Sets normal forwarding as the default operation in the policy-based routing list information.

7.
```
(config-pol)# exit
```
Returns to global configuration mode from policy-based routing list information mode.

8.
```
(config)# ip access-list extended POLICY_GROUP
```

Creates `ip access-list (POLICY_GROUP)`. When this list is created, the command switches to IPv4 packet filtering mode.

9. `(config-ext-nacl)# permit tcp any any action policy-list 10`

   Sets the policy-based routing list information for enabling policy-based routing for IPv4 packets, and sets list number 10.

10. `(config-ext-nacl)# permit ip any any`

    Configures an IPv4 packet filter that forwards all frames.

11. `(config-ext-nacl)# exit`

    Returns to global configuration mode from IPv4 packet filtering mode.

12. `(config)# interface vlan 10`

    Switches to interface mode for VLAN 10.

13. `(config-if)# ip access-group POLICY_GROUP in`

    Enables the target `ip access-list (POLICY_GROUP)` on the receiving side.

(2) Setting the path switch-back operation to Do not switch back

The following example sets `Do not switch back` as the path switch-back operation in the policy-based routing list information in which a forwarding destination route has been set.

Points to note

If you set `Do not switch back` as the path switch-back operation, use the `show ip cache policy` operation command to make sure that the setting has been applied to the target policy-based routing list information.

Command examples

1. `(config)# policy-list 10`

   Switches to policy-based routing list information mode with list number 10.

2. `(config-pol)# recover off`

   Sets `Do not switch back` as the path switch-back operation. After setting, execute the `show ip cache policy 10` operation command.

(3) Configuring the tracking functionality

The following example configures an IPv4 ICMP polling monitoring track.

Points to note

To start polling after setting all parameters, we recommend that you use the commands and specify the parameters in the following order:

1. Use the `track-object` command to specify the track ID.

2. Use the `disable` command to stop track operation.
3. Specify all parameters.
4. Use the `no disable` command to cancel the setting that stops the track operation.

Note that if you set a source IPv4 address for IPv4 ICMP polling monitoring, a fixed destination address is set for response packets. This allows you to design the route for response packets more easily.

Command examples

1.
```
(config)# track-object 1000
```
Specifies the track ID to be configured.

2.
```
(config-track-object)# disable
```
Stops operation of the track being configured.

3.
```
(config-track-object)# default-state up
```
Specifies `Up` as the default track state. After that, the track state is `Up` after the track operation starts until the track state changes to `Down`.

4.
```
(config-track-object)# type icmp 192.0.2.2 nexthop 192.158.1.1
source 198.51.100.1
(config-track-object)# timeout 5
(config-track-object)# interval 10
(config-track-object)# failure detection 4 trial 5 interval 10
(config-track-object)# recovery detection 4 trial 5 interval 10
```
Specifies the track as an IPv4 ICMP polling monitoring track that monitors 192.0.2.2, and specifies the polling packet source address as 198.51.100.1.

Then, the command specifies the track's response wait time, normal polling interval, the number of times polling is performed and polling interval during failure verification, the number of times polling is performed and the polling interval during failure recovery verification.

5.
```
(config-track-object)# no disable
```
Deletes the configuration that stops track operation. When the configuration is deleted, track operation starts.

6.
```
(config-track-object)# exit
```
Returns to global configuration mode from tracking functionality mode.

7.
```
(config)# policy-list 10
```
Sets policy-based routing list information with list number 10. When this list is created, the command switches to policy-based routing list information mode.

8. `(config-pol)# policy-interface vlan 100 next-hop 192.168.1.1 track-object 1000`

   Sets VLAN 100, next hop address 192.168.1.1, and track ID 1000 as the route for policy-based routing list information.

9. `(config-pol)# default permit`

   Sets normal forwarding as the default operation in the policy-based routing list information.

10. `(config-pol)# exit`

    Returns to global configuration mode from policy-based routing list information mode.

11. `(config)# ip access-list extended POLICY_GROUP`

    Creates `ip access-list (POLICY_GROUP)`. Switches to IPv4 packet filtering mode when this list is created.

12. `(config-ext-nacl)# permit tcp any any action policy-list 10`

    Sets the policy-based routing list information for enabling policy-based routing for IPv4 packets, and sets list number 10.

13. `(config-ext-nacl)# permit ip any any`

    Configures an IPv4 packet filter that forwards all frames.

14. `(config-ext-nacl)# exit`

    Returns to global configuration mode from IPv4 packet filtering mode.

15. `(config)# interface vlan 10`

    Switches to interface mode for VLAN 10.

16. `(config-if)# ip access-group POLICY_GROUP in`

    Enables the target `ip access-list (POLICY_GROUP)` on the receiving side.

4.2.3 Configuring an extranet with policy-based routing

To implement an extranet for network partitioning, set policy-based routing.

To enable communication between VRFs, set two VRFs first, and set policy-based routing for each of the VRFs (VLAN).

(1) Setting two VRFs

Points to note

Set two VRFs, and then set different VLAN for each of the VRFs.

Command examples

1.

```
(config)# vrf definition 2
(config-vrf)# exit
(config)# interface vlan 20
(config-if)# vrf forwarding 2
(config-if)# ip address 192.168.20.1 255.255.255.0
(config-if)# exit
```

Sets VRF2, and then sets VRF2, IPv4 address 192.168.20.1, and subnet mask 255.255.255.0 for VLAN20.

2.

```
(config)# vrf definition 3
(config-vrf)# exit
(config)# interface vlan 30
(config-if)# vrf forwarding 3
(config-if)# ip address 192.168.30.1 255.255.255.0
(config-if)# exit
```

Sets VRF3, and then sets VRF3, IPv4 address 192.168.30.1, and subnet mask 255.255.255.0 for VLAN30.

(2) Setting policy-based routing between VRFs

Points to note

Set policy-based routing between VLANs with different VRFs. Set policy-based routing by using the access list.

Command examples

1.

```
(config)# mac-address-table static 0012.e200.5566 vlan 30 interface gigabitethernet 1/0/2
(config)# arp 192.168.20.2 interface vlan 30 0012.e200.5566
```

Sets the next-hop IPv4 address 192.168.20.2 and destination MAC address 0012.e200.5566 for VLAN30 to configure a static entry.

2.

```
(config)# policy-list 20
```

Sets policy-based routing list information with list number 20. When this list is created, the command switches to policy-based routing list information mode.

3.

```
(config-pol)# policy-interface vlan 30 next-hop 192.168.30.2
```

Sets VLAN30 and the next-hop address 192.168.30.2 as a route in the policy-based routing list information.

4.

```
(config-pol)# default permit
```

Sets normal forwarding as the default operation in the policy-based routing list information.

5. `(config-pol)# exit`

   Returns to global configuration mode from policy-based routing list information mode.

6. `(config)# ip access-list extended EXTRA_NET_POLICY_VLAN_20_TO_30`

   Creates `ip access-list` (`EXTRA_NET_POLICY_VLAN_20_TO_30`), and then switches to IPv4 packet filtering mode.

7. `(config-ext-nacl)# permit ip any 192.168.30.0 0.0.0.255 action policy-list 20`

   Sets the policy-based routing list information for enabling policy-based routing for IPv4 packets. Sets list number 20.

8. `(config-ext-nacl)# permit ip any any`

   `(config-ext-nacl)# exit`

   Configures an IPv4 packet filter that forwards all frames and returns to global configuration mode.

9. `(config)# interface vlan 20`

   `(config-if)# ip access-group EXTRA_NET_POLICY_VLAN_20_TO_30 in`

   Enables the target `ip access-list (EXTRA_NET_POLICY_VLAN_20_TO_30)` on the receiving side of the VLAN20.

10. `(config-if)# exit`

    Returns to global configuration mode.

11. `(config)# mac-address-table static 0012.e200.7788 vlan 20 interface gigabitethernet 1/0/3`

    `(config)# arp 192.168.30.2 interface vlan 20 0012.e200.7788`

    Sets the next-hop IPv4 address192.168.30.2 and destination MAC address 0012.e200.7788 for VLAN20 to configure a static entry.

12. `(config)# policy-list 30`

    Sets policy-based routing list information with list number 30. When this list is created, the command switches to policy-based routing list information mode.

13. `(config-pol)# policy-interface vlan 20 next-hop 192.168.20.2`

    Sets VLAN20 and the next-hop address 192.168.20.2 as a route in the policy-based routing list information

14. `(config-pol)# default permit`

Sets normal forwarding as the default operation in the policy-based routing list information.

15. `(config-pol)# exit`

    Returns to global configuration mode from policy-based routing list information mode.

16. `(config)# ip access-list extended EXTRA_NET_POLICY_VLAN_30_TO_20`

    Creates `ip access-list (EXTRA_NET_POLICY_VLAN_30_TO_20)`, and then switches to IPv4 packet filtering mode.

17. `(config-ext-nacl)# permit ip any 192.168.20.0 0.0.0.255 action policy-list 30`

    Sets the policy-based routing list information for enabling policy-based routing for IPv4 packets. Sets list number 30.

18.
    ```
    (config-ext-nacl)# permit ip any any
    (config-ext-nacl)# exit
    ```

    Configures an IPv4 packet filter that forwards all frames and returns to global configuration mode.

19.
    ```
    (config)# interface vlan 30
    (config-if)# ip access-group EXTRA_NET_POLICY_VLAN_30_TO_20 in
    ```

    Enables the target `ip access-list (EXTRA_NET_POLICY_VLAN_30_TO_20)` on the receiving side of the VLAN30.

4.3 Operation

4.3.1 List of operation commands

The following table describes the operation commands for policy-based routing.

Table 4-9: List of operation commands

Command	Description
show ip policy	Shows VLAN ID and access list information about the VLAN interfaces for which IPv4 policy-based routing is enabled.
show ip cache policy	Shows the routing information and status for the specified policy-based routing list information.
reset policy-list	Reselects the routing information.
dump policy	Outputs to a file event trace information and control table information collected by the policy-based program.
restart policy	Restarts the policy-based program.
show access-filter#	Shows the statistics for the access list (`access-list` or `ip access-list`) configured by using the access group command (`ip access-group`).
clear access-filter#	Clears the statistics for the access list (`access-list` or `ip access-list`) configured by using the access group command (`ip access-group`).

#

See *23. Filters* in the manual *Operation Command Reference Vol.1 For Version 11.10.*

The following table describes the operation commands for the policy-based routing tracking functionality.

Table 4-10: List of operation commands (policy-based routing tracking functionality)

Command	Description
show track-object	Shows track information for the tracking functionality.
dump protocols track-object	Outputs trace information or debug information collected by the track object program to a file.
restart track-object	Restarts the track object program.

4.3.2 Checking policy-based routing

(1) Checking the policy-based routing group

The following describes how to check the operation of the policy-based routing group.

Execute the `show ip policy` command, and then check the VLAN interface number to make sure that information about the access list for which policy-based routing list information has been set is displayed.

Figure 4-6: Results of executing the show ip policy command

```
> show ip policy
Date 20XX/01/01 12:00:00 UTC
VLAN ID  Access List Name/Number            Sequence  Policy List
     10  POLICY_GROUP                             10           10
```

Execute the `show access-filter` command to check the operation of the access list for which policy-based routing list information has been set. Make sure that the filter information for the specified VLAN interface displays `Extended IP access-list:POLICY_GROUP` and `action policy-list 10`, and that the value of the `matched packets` is incremented.

Figure 4-7: Results of executing the show access-filter command

```
> show access-filter interface vlan 10 POLICY_GROUP in
Date 20XX/01/01 12:00:00 UTC
Using Interface:vlan 10 in
Extended IP access-list:POLICY_GROUP
      remark "permit Policy Group policy"
      permit tcp(6) any any action policy-list 10
         matched packets       :  74699826
      permit ip any any
         matched packets       :    264176
      implicitly denied packets:         0
```

Execute the `show ip cache policy` command to check the route selected in the policy-based routing list information. Make sure that all routes configured in the specified policy-based routing list information are displayed, and that the selected route is displayed with `*>`.

Figure 4-8: Results of executing the show ip cache policy command (checking routes)

```
> show ip cache policy 10
Date 20XX/01/01 12:00:00 UTC
Policy Base Routing Default Init Interval :  200
   Start Time : 20XX/01/01 00:00:00
   End Time   : 20XX/01/01 00:03:20
Policy Base Routing List :  10
   Default : Permit
   Recover : On
   Priority    Sequence  VLAN ID  Status  Next Hop         Track Object ID
 *>       1          10      100  Up      192.168.1.1                    -
          2          20      200  Up      192.168.2.1                    -
```

(2) Checking the path switch-back operation

Execute the `show ip cache policy` command to check the path switch-back operation set in the policy-based routing list information.

Figure 4-9: Results of executing the show ip cache policy command (checking path switch-back operation)

```
> show ip cache policy 10
Date 20XX/01/01 12:00:00 UTC
Policy Base Routing Default Init Interval :  200
   Start Time : 20XX/01/01 00:00:00
   End Time   : 20XX/01/01 00:03:20
Policy Base Routing List :  10
   Default : Permit
   Recover : Off                                                        ...1
   Priority    Sequence  VLAN ID  Status  Next Hop         Track Object ID
 *>       1          10      100  Up      192.168.1.1                    -
          2          20      200  Up      192.168.2.1                    -
```

1. If `Recover : Off` is displayed, the path switch-back operation is set to `Do not switch back`.

(3) Checking the tracking functionality

The following describes how to check the tracking functionality operation.

Execute the `show track-object` command to display the track state. See the information below `State` to check the track state of each track.

Figure 4-10: Results of executing the show track-object command

```
> show track-object
```

```
Date 20XX/01/01 12:00:00 UTC
Track State          Type        Target
101   UP(Active)     ICMP        172.16.1.1
102   UP(Transit)    ICMP        172.16.2.1
201   DOWN(Transit)  ICMP        172.16.3.1
>
```

Execute the `show track-object` command with a rack ID specified to display detailed track information about the specified track. See the information under `State` to check the track state. See the information under `Last Change` to check the time when the track state changed.

Figure 4-11: Results of executing the show track-object (track ID specified)

```
> show track-object 101
Date 20XX/01/01 12:00:00 UTC
Track: 101
   State: UP(Active),   Last Change: 20XX/12/30 18:11:23
   Type: ICMP
      Destination: 172.16.1.1
      Source: 172.16.1.100, Nexthop: 172.16.1.200
      TOS: max-reliability(2), Precedence: flash(3)
      Interval: 6sec, Timeout: 2sec
>
```

Chapter

5. DHCP and BOOTP Relay Agent Functionality

This chapter describes the DHCP and BOOTP relay agents, and how to configure and check it.

5.1 Description

The DHCP or BOOTP relay agent forwards the DHCP or BOOTP packets broadcast by a DHCP or BOOTP client (abbreviated hereafter to client) to the DHCP or BOOTP server (abbreviated hereafter to server) when the server and the client exist in different subnets.

To forward a DHCP or BOOTP packet to the server, the DHCP or BOOTP relay agent inserts the server IP address set in the configuration, or the helper address, in the destination IP address of the packet. The helper address is the IP address of the router that can forward the packet to the subnet containing the server.

5.1.1 Supported specifications

The following table describes the specifications of the DHCP and BOOTP relay agents on the Switch.

Table 5-1: Specifications of the DHCP and BOOTP relay agents

Item	Specifications
Connection	• DHCP clients are connected to the DHCP relay agent. • DHCP clients are connected to a DHCP server via a DHCP relay agent.
BOOTP	Supported
VRF-enabled	Can be forwarded in the same VRF or between VLANs in an extranet#

#

Exchange routes between VRFs when using the DHCP or BOOTP relay agent in an extranet.

5.1.2 Items checked when a DHCP or BOOTP packet is received

The following table describes the items that are checked when the DHCP or BOOTP relay agent receives DHCP or BOOTP packets.

Table 5-2: Items checked when a DHCP or BOOTP packet is received

DHCP and BOOTP packet header field	Check item	Handling if a packet fails the check	
		From client to server	From server to client
BOOTP REQUEST HOPS	The number of hops must be smaller than the value specified in the configuration.	Discarded	Not discarded
Relay agent address	The address must be the address of the Switch being used.	Discarded	Discarded
TTL in the IP header	The value must be 1 or greater.	Discarded	Discarded
Source address in the IP header	The network number must be a value other than 0.	Not discarded	Discarded

5.1.3 Settings for forwarding

The following table describes the settings that enable the DHCP and BOOTP relay agents to forward DHCP and BOOTP packets.

Table 5-3: Settings for forwarding DHCP and BOOTP packets

Packet header field	Condition for setting	Required setting when the condition is met	
		From client to server	From server to client
Relay agent address in the DHCP or BOOTP header	The address is 0.0.0.0.	• If the receiving interface is not configured to be multihomed, replace the existing address with the IP address of the receiving interface. • If the receiving interface is configured to be multihomed, replace the existing address with the IP address displayed by the `show dhcp giaddr` operation command.	--
Broadcast flag in the DHCP or BOOTP header	The flag is 1.	--	Set a restricted broadcast address# as the destination IP address.
	The flag is 0.	--	Set the client IP address as the destination IP address. Set the client hardware address as the destination MAC address.
BOOTP request hops in the DHCP or BOOTP header	A DHCP or BOOTP request packet is forwarded to the DHCP or BOOTP server.	Increase the number of hops by one.	--
Source address in the IP header	A DHCP or BOOTP request packet is forwarded to the DHCP or BOOTP server.	Replace the existing address with the IP address of the sending interface.	--
	A DHCP or BOOTP reply packet is forwarded to a client.	--	Replace the existing address with the IP address of the sending interface.
Destination address in the IP header	The original address is a restricted broadcast address#.	Replace the existing address with the helper address.	--

Legend: --: Not applicable

#

IP broadcast address 255.255.255.255 or 0.0.0.0

5.1.4 Notes on using the DHCP or BOOTP relay agent

1. When you run the DHCP or BOOTP relay agent and VRRP on the same interface, you need to set the virtual router address set for the Switch on the DHCP or BOOTP server. This virtual router address is the DHCP or BOOTP client gateway address (router option).
2. The Switch can forward packets if the IP packet size is 1500 bytes or less and if the packets are not fragmented.

5.2 Configuration

5.2.1 List of configuration commands

The following table describes the configuration commands for the DHCP and BOOTP relay agents.

Table 5-4: List of configuration commands

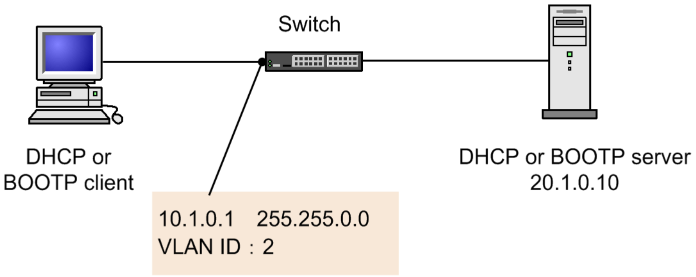
Command name	Description
ip bootp-hops	Sets the threshold for the number of hops.
ip helper-address	Sets the address of the destination to which the DHCP relay agent forwards packets. For *5.2.2 Configuring a basic configuration* and *5.2.3 Configuring a multihomed configuration*, this command is used to specify the IP address of the DHCP or BOOTP server as the helper address.
ip relay-agent-address	Sets the address of the relay agent interface to which a DHCP or BOOTP client connects (`giaddr`). For *5.2.3 Configuring a multihomed configuration*, this command is used to specify the IP address of network A as the address of the relay agent.

5.2.2 Configuring a basic configuration

Points to note

The example below shows how to, on the DHCP relay agent, set the helper address, which is the address of the destination to which BOOTP request packets are forwarded.

Figure 5-1: Basic configuration (one relay agent between the DHCP or BOOTP server and a DHCP or BOOTP client)



Command examples

1.
```
(config)# vlan 2
(config-vlan)# exit
(config)# interface gigabitethernet 1/0/5
(config-if)# switchport mode access
(config-if)# switchport access vlan 2
(config-if)# exit
(config)# interface vlan 2
(config-if)# ip address 10.1.0.1 255.255.0.0
(config-if)# exit
```

   Sets the VLAN ID, line, access port, VLAN interface, and IP address in advance.

2.
```
(config)# vlan 3
(config-vlan)# exit
(config)# interface gigabitethernet 1/0/7
(config-if)# switchport mode access
(config-if)# switchport access vlan 3
(config-if)# exit
(config)# interface vlan 3
(config-if)# ip address 20.1.0.1 255.255.0.0
(config-if)# exit
```
Similar to step 1, sets the VLAN ID, line, access port, and IP address for the interface from which packets are forwarded to the DHCP or BOOTP server.

3.
```
(config)# interface vlan 2
(config-if)# ip helper-address 20.1.0.10
(config-if)# exit
```
Sets the IP address of the DHCP or BOOTP server as the helper address.

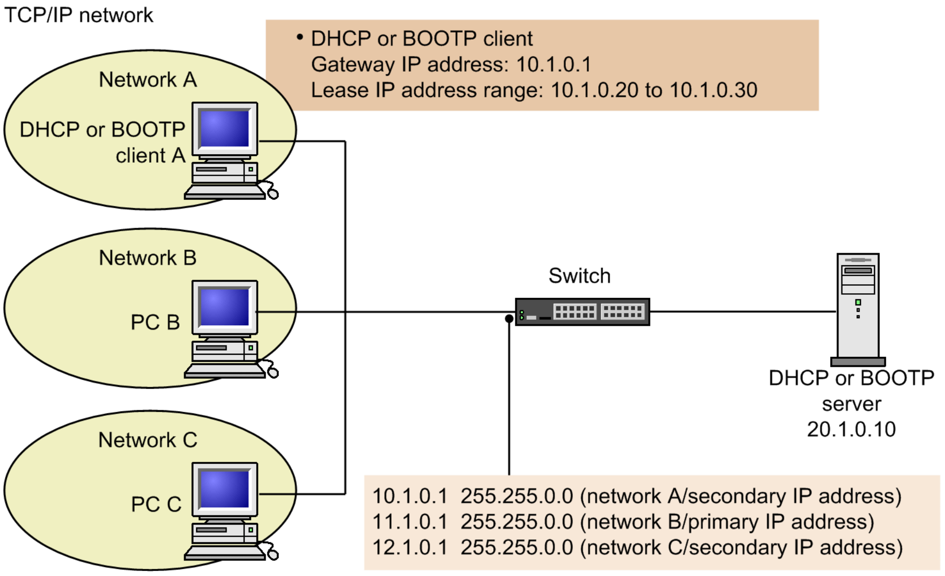
5.2.3 Configuring a multihomed configuration

Points to note

In a multihomed configuration, the primary IP address is usually used as the IP address of the input interface. However, you can use the `ip relay-agent-address` command to assign a secondary IP address as the IP address of the input interface.

In the following example, networks B and C do not run DHCP or BOOTP.

Figure 5-2: Multihomed configuration


TCP/IP network
• DHCP or BOOTP client
Gateway IP address: 10.1.0.1
Lease IP address range: 10.1.0.20 to 10.1.0.30
Network A
DHCP or BOOTP client A
Network B
PC B
Network C
PC C
Switch
DHCP or BOOTP server 20.1.0.10
10.1.0.1 255.255.0.0 (network A/secondary IP address)
11.1.0.1 255.255.0.0 (network B/primary IP address)
12.1.0.1 255.255.0.0 (network C/secondary IP address)

Command examples

1.
```
(config)# vlan 2
(config-vlan)# exit
(config)# interface gigabitethernet 1/0/5
(config-if)# switchport mode access
(config-if)# switchport access vlan 2
(config-if)# exit
```
Sets the VLAN ID, line, and access port in advance.

2.
```
(config)# interface vlan 2
(config-if)# ip address 11.1.0.1 255.255.0.0
(config-if)# ip address 10.1.0.1 255.255.0.0 secondary
(config-if)# ip address 12.1.0.1 255.255.0.0 secondary
(config-if)# exit
```
Sets the IP address of network B as the primary IP address, and the IP addresses of networks A and C as secondary IP addresses.

3.
```
(config)# vlan 3
(config-vlan)# exit
(config)# interface gigabitethernet 1/0/7
(config-if)# switchport mode access
(config-if)# switchport access vlan 3
(config-if)# exit
(config)# interface vlan 3
(config-if)# ip address 20.1.0.1 255.255.0.0
(config-if)# exit
```
Similar to steps 1 and 2, sets the VLAN ID, line, access port, and IP address for the interface from which packets are forwarded to the DHCP or BOOTP server.

4.
```
(config)# interface vlan 2
(config-if)# ip helper-address 20.1.0.10
```
Sets the IP address of the DHCP or BOOTP server as the helper address.

5.
```
(config-if)# ip relay-agent-address 10.1.0.1
(config-if)# exit
```
Sets the IP address of network A as the address of the relay agent.

Notes

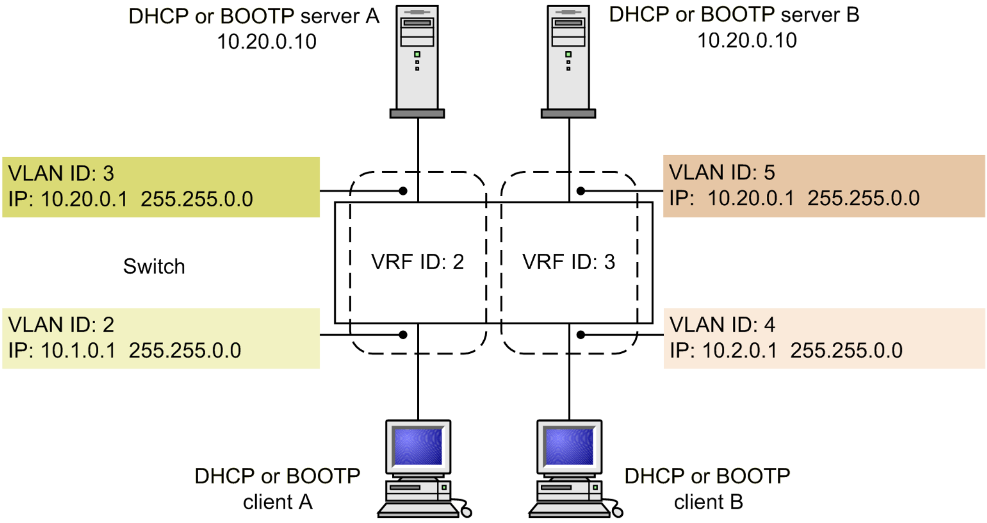
If you omit the `ip relay-agent-address` command, the address of the relay agent is the primary IP address assigned to the applicable interface.

5.2.4 Configuring a VRF configuration [OS-L3SA]

Points to note

In this configuration, DHCP or BOOTP servers are prepared for each VRF.

Figure 5-3: VRF configuration



Command examples

1.
```
(config)# vlan 2
(config-vlan)# exit
(config)# interface gigabitethernet 1/0/5
(config-if)# switchport mode access
(config-if)# switchport access vlan 2
(config-if)# exit
(config)# vrf definition 2
(config-vrf)# exit
(config)# interface vlan 2
(config-if)# vrf forwarding 2
(config-if)# ip address 10.1.0.1 255.255.0.0
(config-if)# exit
```

Sets the VLAN ID, line, access port, VRF, VLAN interface, VRF ID, and IP address in advance.

2.
```
(config)# vlan 3
(config-vlan)# exit
(config)# interface gigabitethernet 1/0/7
(config-if)# switchport mode access
(config-if)# switchport access vlan 3
```

```
(config-if)# exit
(config)# interface vlan 3
(config-if)# vrf forwarding 2
(config-if)# ip address 10.20.0.1 255.255.0.0
(config-if)# exit
```

As in step 1, sets the VLAN ID, line, access port, VRF ID, and IP address for the interface from which packets are forwarded to the DHCP or BOOTP server.

3.

```
(config)# vlan 4
(config-vlan)# exit
(config)# interface gigabitethernet 1/0/9
(config-if)# switchport mode access
(config-if)# switchport access vlan 4
(config-if)# exit
(config)# vrf definition 3
(config-vrf)# exit
(config)# interface vlan 4
(config-if)# vrf forwarding 3
(config-if)# ip address 10.2.0.1 255.255.0.0
(config-if)# exit
(config)# vlan 5
(config-vlan)# exit
(config)# interface gigabitethernet 1/0/11
(config-if)# switchport mode access
(config-if)# switchport access vlan 5
(config-if)# exit
(config)# interface vlan 5
(config-if)# vrf forwarding 3
(config-if)# ip address 10.20.0.1 255.255.0.0
(config-if)# exit
```

As in steps 1 and 2, sets the VLAN ID, line, access port, VRF, VRF ID, and IP address for each interface on the VRF ID 3 side.

4.

```
(config)# interface vlan 2
(config-if)# ip helper-address 10.20.0.10
```

Sets the IP address of the DHCP or BOOTP server on the VRF ID 2 side as the helper address.

5.

```
(config)# interface vlan 4
```

```
(config-if)# ip helper-address 10.20.0.10
```

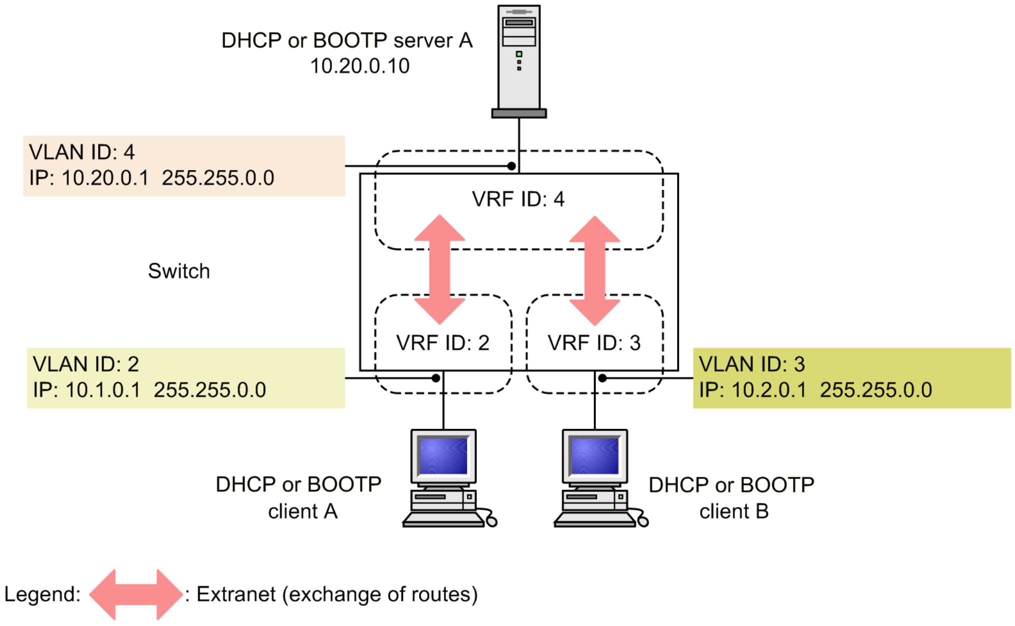
Sets the IP address of the DHCP or BOOTP server on the VRF ID 3 side as the helper address (since the VRF ID differs from the VLAN ID 2, the forwarding destination helper address is treated as another destination).

5.2.5 Configuring an extranet configuration [OS-L3SA]

Points to note

Sets exchange routing between VRFs in an extranet configuration.

Figure 5-4: Extranet configuration



Command examples

1.
```
(config)# vlan 2
(config-vlan)# exit
(config)# interface gigabitethernet 1/0/5
(config-if)# switchport mode access
(config-if)# switchport access vlan 2
(config-if)# exit
(config)# vrf definition 2
(config-vrf)# exit
(config)# interface vlan 2
(config-if)# vrf forwarding 2
(config-if)# ip address 10.1.0.1 255.255.0.0
(config-if)# exit
```

Sets the VLAN ID, line, access port, VRF, VLAN interface, VRF ID, and IP address in advance.

2.
```
(config)# vlan 4
(config-vlan)# exit
(config)# interface gigabitethernet 1/0/1
(config-if)# switchport mode access
(config-if)# switchport access vlan 4
(config-if)# exit
(config)# vrf definition 4
(config-vrf)# exit
(config)# interface vlan 4
(config-if)# vrf forwarding 4
(config-if)# ip address 10.20.0.1 255.255.0.0
(config-if)# exit
```

Sets the VLAN ID, line, access port, VRF, VRF ID, and IP address for the interface from which packets are forwarded to the DHCP or BOOTP server.

3.
```
(config)# vlan 3
(config-vlan)# exit
(config)# interface gigabitethernet 1/0/7
(config-if)# switchport mode access
(config-if)# switchport access vlan 3
(config-if)# exit
(config)# vrf definition 3
(config-vrf)# exit
(config)# interface vlan 3
(config-if)# vrf forwarding 3
(config-if)# ip address 10.2.0.1 255.255.0.0
(config-if)# exit
```

As in step 1, sets the VLAN ID, line, access port, VRF, VRF ID, and IP address for the interface on the VRF ID 3 side.

4.
```
(config)# route-map VRF4PERMIT permit 10
(config-route-map)# match vrf 4
(config-route-map)# exit
(config)# vrf definition 2
(config-vrf)# import inter-vrf VRF4PERMIT
(config-vrf)# exit
(config)# vrf definition 3
(config-vrf)# import inter-vrf VRF4PERMIT
```

```
(config-vrf)# exit
```

Creates a filter that permits the VRF ID 4 route to use the VRF ID 4 route in VRF IDs 2 and 3.

5.
```
(config)# route-map VRF2AND3PERMIT permit 10
(config-route-map)# match vrf 2 3
(config-route-map)# exit
(config)# vrf definition 4
(config-vrf)# import inter-vrf VRF2AND3PERMIT
(config-vrf)# exit
```

Creates a filter that permits the routes of VRF IDs 2 and 3 to use the routes in VRF ID 4.

6.
```
(config)# interface vlan 2
(config-if)# ip helper-address 10.20.0.10
```

Sets the IP address of the DHCP or BOOTP server on the VRF ID 4 side as the helper address.

7.
```
(config)# interface vlan 3
(config-if)# ip helper-address 10.20.0.10
```

Sets the IP address of the DHCP or BOOTP server on the VRF ID 4 side as the helper

5.3 Operation

5.3.1 List of operation commands

The following table describes the operation commands for the DHCP and BOOTP relay agents.

Table 5-5: List of operation commands

Command name	Description
show dhcp traffic	Shows the statistics for DHCP or BOOTP relay agents.
clear dhcp traffic	Clears the relay agent statistics.
show dhcp giaddr	Shows the IP address on which DHCP or BOOTP packets from the DHCP or BOOTP server are received.

5.3.2 Checking the IP address of the receiver of DHCP and BOOTP packets

Execute the `show dhcp giaddr` command to check whether the displayed IP address matches the IP address of the interface on the Switch to which the DHCP or BOOTP client is connected.

Figure 5-5: Results of executing the show dhcp giaddr command

```
>show dhcp giaddr all
Date 20XX/10/15 12:00:00 UTC
DHCP GIADDR <vlan 2>: 10.1.0.1
```

Chapter

6. DHCP Server Functionality

A DHCP server dynamically assigns IP addresses and optional information to DHCP clients. This chapter describes the DHCP server and how to configure it.

6.1 Description

A DHCP server dynamically assigns IP addresses and optional information to DHCP clients. This section describes the specifications of a DHCP server on the Switch and how it operates.

6.1.1 Supported specifications

The table below describes the specifications of a DHCP server on the Switch. A DHCP server and clients can be directly connected on the same network or connected via a DHCP relay agent.

Table 6-1: DHCP server specifications

Item	Specifications
Connection	• DHCP clients are directly connected to a DHCP server. • DHCP clients are connected to a DHCP server via a DHCP relay agent.
BOOTP server	Not supported
Linking dynamic DNS	Supported The Switch supports dynamic DNS servers that perform DNS updates as defined in RFC 2136.
Dynamic and static IP address distribution	Supported

6.1.2 Distributing information to clients

The table below describes the types of information that the Switch can distribute to clients. Optional information is not distributed even if option numbers are specified on the Switch unless clients request optional information by submitting an option request list.

Table 6-2: Information distributed by the Switches to clients

Information	Overview
IP address	IP address that can be used by a client
IP address lease time (optional)	Lease time for a distributed IP address. In the Switch, the lease time is determined based on the values of the `default-lease-time` and `max-lease-time` parameters and the request from the client. (Option No. 51)
Subnet mask	Subnet mask length indicating a network address specified in the configuration. (Option No. 1)
Router (optional)	A list of the IP addresses of the routers on the subnet containing the target client. Routers are listed according to priority from highest to the lowest. One of the IP addresses in this list is used as the gateway address for the client. (Option No. 3) If you do not specify this item in the configuration, the Switches do not distribute a list of router IP addresses to the client when the client requests a router IP address. However, the Switches insert the IP address set for the client in the router IP address field and distribute the information to the client.
DNS (optional)	A list of IP addresses of the domain name servers that a client can use. Domain name servers are listed according to priority from highest to the lowest. (Option No. 6)
Host name (optional)	Host name of the client determined by the server. The host name might be restricted by the local domain name. A host name is specified as a character string. (Option No. 12)
Domain name (optional)	Domain name to be used when the IP address of the client is converted to an FQDN by the Domain Name System. (Option No. 15)

Information	Overview
NetBIOS over TCP/IP name server (optional)	A list of the IP addresses of the NetBIOS name servers (WINS servers) referenced by a client. NetBIOS name servers are listed according to priority from highest to the lowest. (Option No. 44)
NetBIOS over TCP/IP node type (optional)	Node type of the NetBIOS over TCP/IP client (NetBIOS name resolution method). (Option No. 46) • Code 1: B-node (broadcast) • Code 2: P-node (peer to peer; WINS only) • Code 4: M-node (mixed; WINS is used when the IP address is not found by a broadcast) • Code 8: H-node (hybrid; broadcasting is used when the IP address is not found by WINS)

6.1.3 Linking dynamic DNS

The DHCP server of the Switch distributes IP addresses and adds entry records to a dynamic DNS server (DNS updates). For DNS updates to take place, you need to specify the target zone and the target DNS server on the DHCP server, and configure the DNS server to receive record updates from the Switch.

Record updates are permitted either based on IP address or by using an HMAC-MD5 authentication key. For permission based on IP address, the DNS server simply permits access from the IP addresses of the clients or networks connected to the DNS server. For permission using an authentication key, you need to set the key specified for the DNS server in the DNS authentication key information on the DHCP server.

Notes on linking dynamic DNS

- The DHCP server of the Switch only performs DNS updating for the IP addresses that are dynamically assigned. If the DHCP server distributes static addresses, you need to add the corresponding records to the DNS server beforehand.
- To enable DNS updates, a DHCP client must return its FQDN to the DHCP server when it receives a distributed IP address. If the required information does not exist, the DHCP server does not perform DNS updating for the leased IP addresses. For details about the settings, see the configuration of the devices used as clients.
- When you use an authentication key for DNS updates, the DNS server and the Switch must have the same time information. In many cases, the time difference must be no more than five minutes in UTC. Use NTP to synchronize the time information.

6.1.4 Preventing duplicate distribution of IP addresses

If the Switch is restarted after the DHCP server of the Switch has assigned addresses to DHCP clients, all of the DHCP address pools on the Switch become empty. When the Switch resumes assigning IP addresses to DHCP clients, the switch sends an ICMP Echo Request packet for a previously assigned IP address. The Switch checks whether a reply packet is returned and makes sure that no client is using the previously assigned IP address, avoiding a duplicate assignment of IP addresses. At the same time, the Switch attempts to assign to a client the same IP address it had previously assigned. Accordingly, the communication of the client is not affected.

If a reply is returned for the ICMP Echo Request packet (a network terminal is using the IP address in question), the incident is displayed on the screen as a detected address conflict when the `show ip dhcp conflict` command is executed.

6.1.5 Notes on using a DHCP server

The following are notes on using a DHCP server.

(1) IP address of an input interface when a multihomed connection is established

In a multihomed configuration, the primary IP address is usually used for the IP address of the input interface. The IP addresses in the DHCP address pool set for the subnet are assigned to DHCP clients.

(2) Number of simultaneously connected clients when the lease time is short

If the lease time is 10 seconds, make sure that the maximum number of connected clients is no more than 200. If the lease time is 20 seconds, the maximum number of connected clients must be no more than 400. If the lease time is 30 seconds, the maximum number of connected clients must be no more than 600.

6.2 Configuration

6.2.1 List of configuration commands

The following table describes the configuration commands for a DHCP server.

Table 6-3: List of configuration commands

Command name	Description
client-name	Specifies the host name for a client (optional information). The host name specified by this command is used by the client when the DHCP server distributes a static IP address to the client.
default-router	Specifies the router for a client (optional information). From this list, the server selects the IP address of the default router available for the client on the subnet and distributes it to the client. For details, see *6.2.2 Configuring the distribution of IP addresses to clients* and set the IP address for the router used by the client.
dns-server	Specifies the list of the IP addresses of the DNS servers for a client (optional information). From this list, the server selects the IP address of a DNS server available for the client and distributes it to the client. For details, see *6.2.4 Configuring settings when linking to a dynamic DNS server* and set the IP address for the DNS server used by the client.
domain-name	Specifies the domain name for a client (optional information). The domain name specified by using this command is used by the client as the preferred domain name and DNS resolves it to the IP address distributed to the client. For details, see *6.2.4 Configuring settings when linking to a dynamic DNS server* and set the domain name used for solving the host name by the client.
hardware-address	Specifies the MAC address of a client when a static IP address is distributed to the client. This command is used together with the `host` command. For details, see *6.2.3 Configuring the distribution of static IP addresses to clients* and set the MAC address for the client.
host	Specifies the static IP address to be assigned to a client when a static IP address is distributed to the client. This command is used together with the `hardware-address` command. For details, see *6.2.3 Configuring the distribution of static IP addresses to clients* and set the IP address used by the client.
ip dhcp dynamic-dns-update	Specifies whether to link dynamic DNS when distributing IP addresses. For details, see *6.2.4 Configuring settings when linking to a dynamic DNS server* and enable dynamic DNS linkage.
ip dhcp excluded-address	Specifies the range of IP addresses in the DHCP address pool specified by using the `network` command that are to be excluded from distribution. For details, see *6.2.2 Configuring the distribution of IP addresses to clients* and within the network address range, set the IP addresses that are to be excluded from being distributed.
ip dhcp key	Sets the authentication key to be used for authentication on the DNS server when dynamic DNS is used.
ip dhcp pool	Sets DHCP address pool information.
ip dhcp zone	Sets the information about the zone where DNS updating is performed when a dynamic DNS server is linked. For details, see *6.2.4 Configuring settings when linking to a dynamic DNS server* and set the zone information for the domain that performs linkage.
lease	Specifies the default lease time for the IP address distributed to a client. For details, see *6.2.2 Configuring the distribution of IP addresses to clients* and set the lease time for the IP address used by the client.

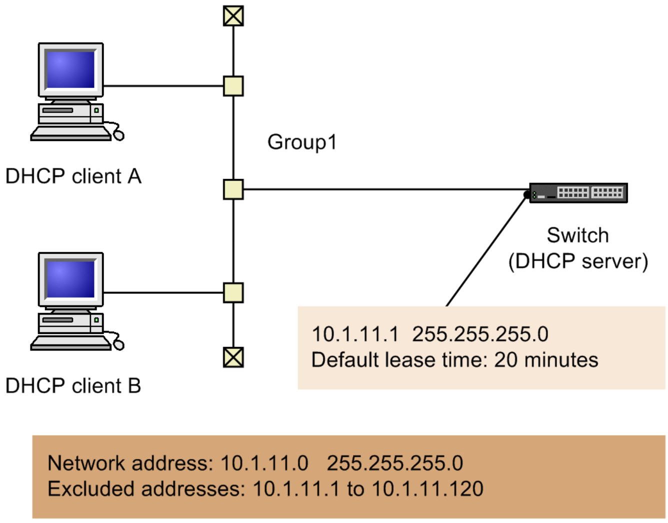
Command name	Description
max-lease	Specifies the maximum lease time allowed when a client requests an IP address with a specific lease time.
netbios-name-server	Specifies the list of the IP addresses of the NetBIOS name servers (NBNS/WINS servers) for a client. From this list, the DHCP server selects the IP address of the NetBIOS name server available for the client and distributes it to the client.
netbios-node-type	Specifies the NetBIOS node type for a client (optional information). A NetBIOS node type indicates the name resolution method used by the client when NetBIOS over TCP/IP is used.
network	Specifies the subnet of the network to which an IP address is dynamically distributed by DHCP. IP addresses whose host name portion is set to all 0s or all 1s are not included in the DHCP address pool. For details, see *6.2.2 Configuring the distribution of IP addresses to clients* and set the network to which an IP address is distributed by DHCP.
service dhcp	Specifies the interface on which a DHCP server is enabled. Only the VLAN interface with this configuration receives DHCP packets. For details, see *6.2.2 Configuring the distribution of IP addresses to clients* and set the VLAN interface to which the DHCP client is connected.

6.2.2 Configuring the distribution of IP addresses to clients

Points to note

Specify the IP addresses that you want to be excluded from assignment to DHCP clients, and then create a DHCP address pool and use it to dynamically distribute IP addresses to DHCP clients.

Figure 6-1: Client-server configuration (when IP addresses are dynamically distributed)


DHCP client A
Group1
Switch
(DHCP server)
DHCP client B
10.1.11.1 255.255.255.0
Default lease time: 20 minutes
Network address: 10.1.11.0 255.255.255.0
Excluded addresses: 10.1.11.1 to 10.1.11.120

Command examples

1.
```
(config)# interface vlan 10
(config-if)# ip address 10.1.11.1 255.255.255.0
(config-if)# exit
```

Sets the necessary VLAN interface and IP address in advance.

2. `(config)# service dhcp vlan 10`

   Specifies the name of the VLAN interface on which the DHCP server is enabled.

3. `(config)# ip dhcp excluded-address 10.1.11.1 10.1.11.120`

   Sets the IP addresses that are to be excluded from assignment to DHCP clients by the DHCP server.

4. `(config)# ip dhcp pool Group1`

   Creates a DHCP address pool.

   Switches to DHCP configuration mode.

5. `(dhcp-config)# network 10.1.11.0 255.255.255.0`

   Sets the network address of the DHCP address pool.

6. `(dhcp-config)# lease 0 0 20`

   Sets 20 minutes as the default lease time for the DHCP address pool.

7. `(dhcp-config)# default-router 10.1.11.1`

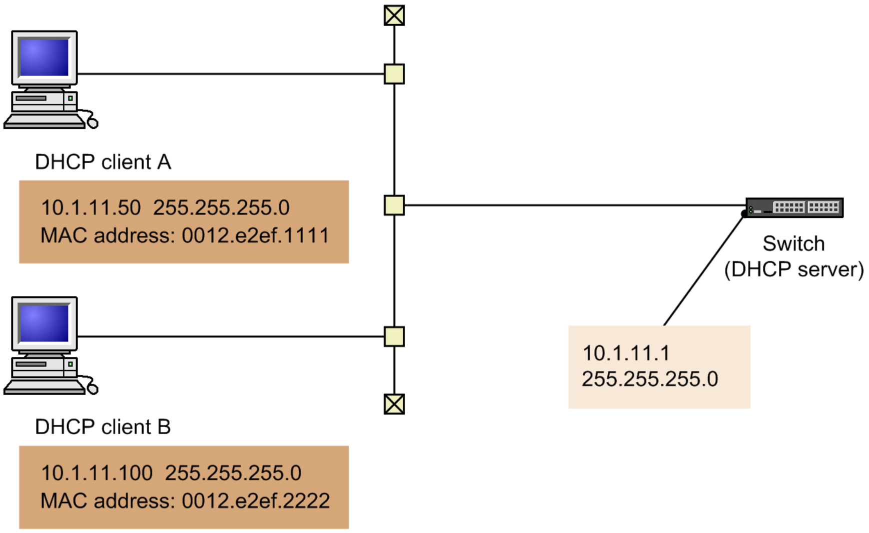
   Sets the IP address of the router on the subnet.

6.2.3 Configuring the distribution of static IP addresses to clients

Points to note

The example below shows how to distribute static IP addresses to DHCP clients and pair the IP addresses with the corresponding MAC addresses.

Figure 6-2: Client-server configuration (when static IP addresses are distributed)



Command examples

1.
```
(config)# interface vlan 10
(config-if)# ip address 10.1.11.1 255.255.255.0
(config-if)# exit
```
Sets the necessary VLAN interface and IP address in advance.

2.
```
(config)# service dhcp vlan 10
```
Specifies the name of the VLAN interface on which the DHCP server is enabled.

3.
```
(config)# ip dhcp pool Client1
```
Sets the name of the DHCP address pool for DHCP client A.

Switches to DHCP configuration mode.

4.
```
(dhcp-config)# host 10.1.11.50 255.255.255.0
```
Sets a static IP address for DHCP client A in the DHCP address pool.

5.
```
(dhcp-config)# hardware-address 0012.e2ef.1111 ethernet
```
Sets the MAC address for DHCP client A in the DHCP address pool.

6.
```
(dhcp-config)# default-router 10.1.11.1
(dhcp-config)# exit
```
Sets the IP address of the router on the subnet.

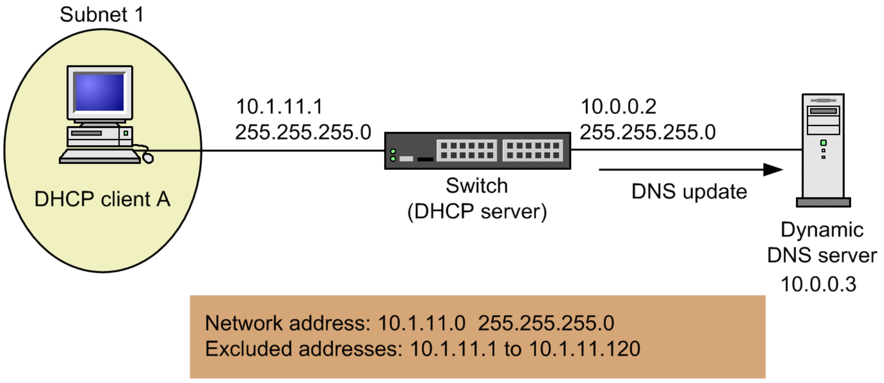
7.
```
(config)# ip dhcp pool Client2
(dhcp-config)# host 10.1.11.100 255.255.255.0
(dhcp-config)# hardware-address 0012.e2ef.2222 ethernet
(dhcp-config)# default-router 10.1.11.1
```
Repeats steps 3 through 6 to set the name of the DHCP address pool for DHCP client B, and to set a static IP address and MAC address for DHCP client B in the Client2 DHCP address pool.

6.2.4 Configuring settings when linking to a dynamic DNS server

Points to note

The example below shows how to specify zone information and enable dynamic DNS server linkage so that DNS records associated with a client can be reported to the dynamic DNS server when an IP address is distributed to the client.

Figure 6-3: Connection configuration when a dynamic DNS server is linked



Command examples

1.
```
(config)# interface vlan 10
(config-if)# ip address 10.1.11.1 255.255.255.0
(config-if)# exit
```
Sets a VLAN interface and IP address for subnet 1 in advance.

2.
```
(config)# interface vlan 20
(config-if)# ip address 10.0.0.2 255.255.255.0
(config-if)# exit
```
As in step 1, sets a VLAN interface and IP address for the dynamic DNS server.

3.
```
(config)# service dhcp vlan 10
(config)# ip dhcp excluded-address 10.1.11.1 10.1.11.120
(config)# ip dhcp pool Group1
(dhcp-config)# network 10.1.11.0 255.255.255.0
(dhcp-config)# default-router 10.1.11.1
```
Sets the IP address as in *6.2.2 Configuring the distribution of IP addresses to clients*.

4.
```
(dhcp-config)# domain-name example.net
```
Sets the domain name to be used by the client when DNS is used to resolve the IP address assigned to the client to the domain name.

5.
```
(dhcp-config)# dns-server 10.0.0.3
```
Sets the IP address of the DNS server that the client can use.

6.
```
(dhcp-config)# exit
```
Switches from DHCP configuration mode to global configuration mode.

7. `(config)# ip dhcp zone example.net. primary 10.0.0.3`

   Sets a zone for the `example.net.` domain for forward DNS resolution, and specifies the DNS server defined in step 5 (10.0.0.3) as the dynamic DNS server.

8. `(config)# ip dhcp zone 11.1.10.in-addr.arpa. primary 10.0.0.3`

   Sets a zone for the `11.1.10.in-addr.arpa.` domain for reverse DNS resolution, and specifies the DNS server defined in step 5 (10.0.0.3) as the dynamic DNS server.

9. `(config)# ip dhcp dynamic-dns-update`

   Enables dynamic DNS linkage.

6.3 Operation

6.3.1 List of operation commands

The following table describes the operation commands for the DHCP server.

Table 6-4: List of operation commands

Command name	Description
show ip dhcp binding	Shows the binding information on the DHCP server.
clear ip dhcp binding	Deletes the binding information from the DHCP server database.
show ip dhcp import	Shows the values of options and parameters that are set in the configuration of a DHCP server.
show ip dhcp conflict	Shows an IP address conflict detected by the DHCP server. An IP address conflict refers to when an IP address is indicated as available in the DHCP address pool on the DHCP server but is already assigned to a terminal on the network. Before the DHCP server assigns the IP address to a DHCP client, the DHCP server detects an IP address conflict by checking for a response to a sent ICMP packet.
clear ip dhcp conflict	Clears the IP address conflict information from the DHCP server.
show ip dhcp server statistics	Shows statistics about the DHCP server.
clear ip dhcp server statistics	Resets statistics on the DHCP server.
restart dhcp	Restarts the DHCP server daemon process.
dump protocols dhcp	Outputs the server log data and the packet sending and receiving log data collected by the DHCP server program to a file.
dhcp server monitor	Starts collection of sending and receiving log data for packets which are sent and received by the DHCP server.
no dhcp server monitor	Stops collection of the sending and receiving log data for packets on the DHCP server.

6.3.2 Checking the number of IP addresses that can be assigned

To check the number of IP addresses that can be assigned to clients, execute the `show ip dhcp server statistics` command and check the `address pools` field in the displayed list. Make sure that the number displayed here is greater than the number of IP addresses you want to assign to clients.

Figure 6-4: Results of executing the show ip dhcp server statistics command

```
> show ip dhcp server statistics
Date 20XX/10/15 12:00:00 UTC
   < DHCP Server use statistics >
     address pools          :19
     automatic bindings     :170
     manual bindings        :1
     expired bindings       :3
     over pools request     :0
     discard packets        :0
   < Receive Packets >
     BOOTREQUEST            :0
     DHCPDISCOVER           :178
     DHCPREQUEST            :178
     DHCPDECLINE            :0
     DHCPRELEASE            :1
     DHCPINFORM             :0
```

```
    < Send Packets >
      BOOTREPLY              :0
      DHCPOFFER              :178
      DHCPACK                :172
      DHCPNAK                :6
>
```

6.3.3 Checking the distributed IP addresses

To check the IP addresses that are actually assigned to DHCP clients, execute the `show ip dhcp binding` command. The displayed list shows the IP addresses that have not expired yet.

Figure 6-5: Results of executing the show ip dhcp binding command

```
> show ip dhcp binding
Date 20XX/10/15 12:00:00 UTC
<IP address>      <MAC address>       <Lease expiration>   <Type>
10.1.11.1         0012.e2ef.1111      XX/10/15 19:39:20    Automatic
10.1.11.50        0012.e2ef.2222                           Manual
>
```

Chapter
7. IPv4 Routing Protocol Overview

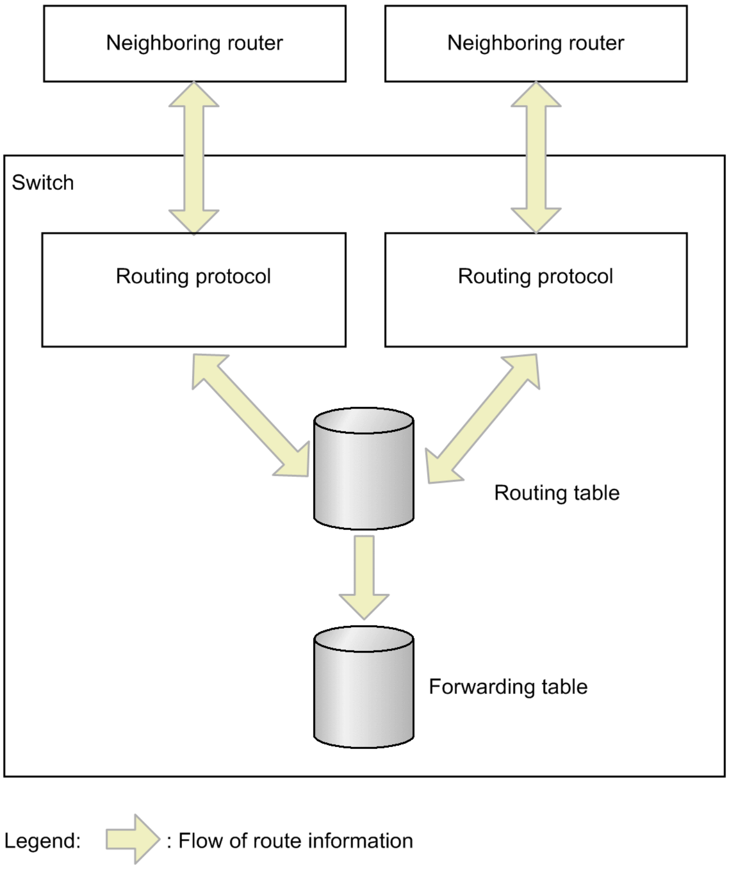
This chapter provides an overview of IPv4 routing protocols.

7.1 Description of IPv4 routing

7.1.1 Overview of routing

Routing protocols define how routers exchange routing information with each other. A router stores the routing information it learns from each routing protocol in a routing table, and then registers preferred routes for packet forwarding in a forwarding table. Packet relay takes place based on the contents of the forwarding table.

Figure 7-1: Routing overview



7.1.2 Static routing and dynamic routing

A switch must create a routing table before it can relay packets. The switch uses static routing and dynamic routing to create routing tables.

- Static routing

  A user sets routing information manually using configuration commands.

- Dynamic routing

  The Switch determines how to relay packets based on routing information it receives from other routers in the network. The Switch supports versions 1 and 2 of the RIP protocol (abbreviated hereafter to RIP-1 and RIP-2), version 2 of the OSPF protocol (abbreviated hereafter to OSPF), and version 4 of the BGP protocol (abbreviated hereafter to BGP4).

7.1.3 Routing information

The following table describes the routing information handled by the Switch (the address types subject to routing).

Table 7-1: Routing information

Routing information		Description
Standard routes	Default route	A route that matches every network destination (destination address: 0.0.0.0, network mask: 0.0.0.0).
	Natural mask route	A network mask route derived from the address class (network mask: 8 bits for class A, 16 bits for class B, 24 bits for class C).
	Subnet route	A route to a specific subnet (having a longer network mask than the network mask derived from the address class).
	Host route	A route to a specific host (with a 32-bit network mask).
	Variable length subnet mask	Performs routing via a VLSM (Variable Length Subnet Mask). This technique allows you to use subnet masks of different lengths within the same network address space.
CIDR routes	Supernet route	A supernet uses a network mask with fewer bits than the classful equivalent. For example, routing information for the four class-C network addresses 192.168.8.0/24, 192.168.9.0/24, 192.168.10.0/24, and 192.168.11.0/24 can be aggregated as a single supernet route 192.168.8.0/22.
	0 subnet route	Network addresses whose subnet is 0 are handled as a single subnetwork. For example, the switch can apply the 0 subnet route to the class-B network address 172.16.0.0/24.
	-1 subnet route	Network addresses whose subnet is -1 (all 1s) are handled as a single subnetwork. For example, the switch can apply the -1 subnet route to the class-B network address 172.16.255.0/24.
	Inclusive subnets	Routes whose network addresses present an inclusive relationship between multiple pieces of routing information can be handled separately. For example, the switch can apply separate routing information to the class-B network addresses 172.16.3.0/24 and 172.16.2.0/23.

7.1.4 Scope of individual routing protocols

The following table provides an overview of the routing information and functionality offered by the Switch for each supported routing protocol.

Table 7-2: Scope of individual routing protocols

Routing information		Routing				
		Static	Dynamic			
			RIP-1	RIP-2	OSPF	BGP4
Routing information	Default route	Y	Y	Y	Y	Y
	Natural mask route	Y	Y	Y	Y	Y
	Subnet route	Y	Y	Y	Y	Y
	Host route	Y	Y	Y	Y	Y

Routing information		Routing				
		Static	Dynamic			
			RIP-1	RIP-2	OSPF	BGP4
	Variable length subnet mask	Y	N	Y	Y	Y
	CIDR compatibility	Y	P	Y	Y	Y
	Multipath (max 16 paths)	Y	N	N	Y	Y
Route selection		--	Metric (hop count)		Cost (hop count and line speed)	AS path attribute
Routing loop prevention		--	Split horizon		Y	Y
Authentication functionality		--	N	N	Y	Y

Legend:

Y: Supported

P: Partially supported (supported for 0 subnet and -1 subnet routes)

N: Not supported

--: Not applicable

7.1.5 Concurrent use of routing protocols

You can implement various static and dynamic routing protocols concurrently on the Switch.

(1) Determining priority for learned routes

In an environment running more than one routing protocol concurrently, each protocol uses its own route selection algorithm to select the best route to a given destination. Both summarized routes and directly connected routes are treated as a protocol route as with routes learned by routing protocols. This can result in the Switch learning multiple different routes to the same destination. In this case, the switch compares the distance of each route and applies the routing information with the highest priority.

In the Switch, you can use configuration commands to set the default distance (priority) that applies to each static route and to each piece of routing information generated by dynamic routing protocols (such as RIP). A route with a smaller distance has a higher priority. The following table describes the default distances for each protocol.

Table 7-3: Default distances

Route	Default distance
Directly connected route	0 (fixed value)
Static route	2
BGP4 route learned from external peer	20
OSPF (internal AS route)	110
OSPF (external AS route)	110
RIP route	120
Summarized route	130

Route	Default distance
BGP4 route learned from internal peer	200
BGP4 route learned from inter-AS peer	200
Route imported from another VRF or a global network	210

(2) Route advertisement

In an environment running more than one routing protocol, each routing protocol only advertises the routing information learned by that protocol. It does not advertise routing information it learns from other routing protocols.

In the Switch, you can use route filtering to configure a routing protocol to advertise routing information it learns from other routing protocols, or to exclude specific routes from advertisement. Information of inactive routes cannot be advertised in other routing protocols.

For details about route filtering, see *13. Route Filtering (IPv4)*.

(a) RIP route advertisement

RIP-1 and RIP-2 implement the same routing protocol. Each advertises the routes it learns from the other.

(b) OSPF route advertisement

The OSPF routing protocol operates independently in each OSPF domain. For this reason, a number of internal and external AS routes taken from different OSPF domains might exist for a given destination address. When OSPF routes share the same distance, the route with the smaller domain number takes priority. You can change the default distances for internal and external AS OSPF routes (intra-area and inter-area routes).

Under normal circumstances, routes are not advertised between the various OSPF domains configured on the Switch. However, you can configure route filtering to enable advertisement of internal and external OSPF routes to other OSPF domains.

(c) BGP4 route advertisement

When route filtering is disabled, the switch advertises the BGP4 routes it learns from a given AS to other ASs. In this case, the switch advertises the best BGP4 route as selected by the BGP4 routing protocol even if another routing protocol defines a route to the same destination.

If route filtering is enabled, the switch advertises the routing information that represents the route with the highest priority, selected according to its distance.

7.1.6 Notes on concurrent use of routing protocols

(1) Concurrent use of RIP-1 with OSPF or RIP-2

The routing protocols OSPF and RIP-2 use variable-length subnet masking that disregards the class (A, B, or C) of the IP address. In contrast, RIP-1 cannot use variable-length subnet masking because it recognizes the class of the destination address. For this reason, caution is required when deploying these routing protocols concurrently in the same network, as illustrated below. This section uses an example of the relationship between OSPF and RIP-1, but applies equally to the relationship between RIP-2 and RIP-1.

(a) When RIP-1 does not advertise a subnet route learned by OSPF

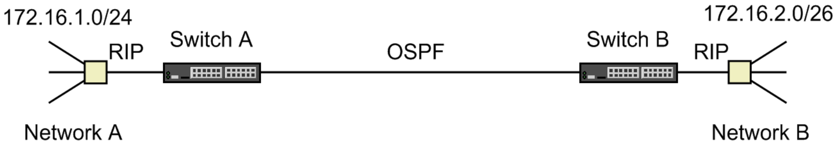
Note that RIP-1 does not advertise routes to a subnetted network when either of the following conditions is met:

1. The route is to a subnet whose subnet mask length differs from the network address of the interface using RIP.
2. The route is to a subnet whose network address differs from that of the interface using RIP.

■ Connections between subnets with different subnet mask lengths

In the following figure, Switch A registers the route to network B in its own routing table. At this time, because network B matches the first of the conditions given above, Switch A does not advertise its routes to network A by the RIP protocol.

Figure 7-2: Connections between subnets with different subnet mask lengths

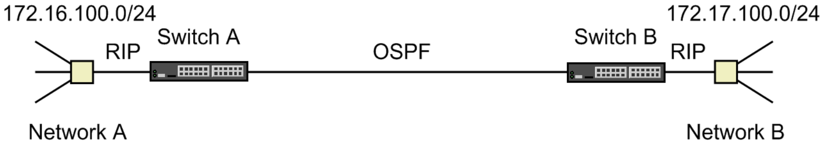


In *Figure 7-5: Example of connection between subnets*, because network A and network B represent subnets with the same subnet length in the same network, Switch A advertises routing information for both networks.

■ Connections between subnets with different network addresses

In the following figure, Switch A registers the route to network B in its own routing table, but because network B meets the second of the conditions given above, Switch A does not advertise the routes of network B to network A by the RIP protocol.

Figure 7-3: Connections between subnets with different network addresses



In *Figure 7-5: Example of connection between subnets*, because network A and network B represent subnets with the same subnet length in the same network, Switch A advertises routing information for both networks.

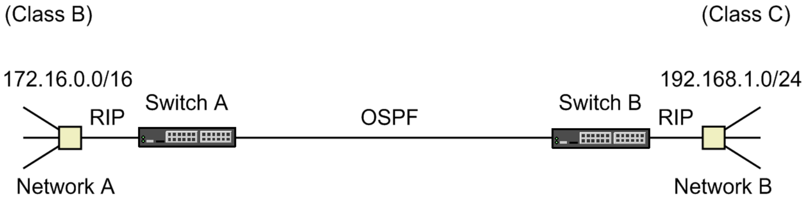
(b) Establishing an OSPF connection between RIP networks

You can use one of the following configurations to provide an OSPF connection between networks running RIP.

■ Using a configuration without subnets

In the example in the following figure, routing information to Network A and Network B is advertised to Network B and network A, respectively.

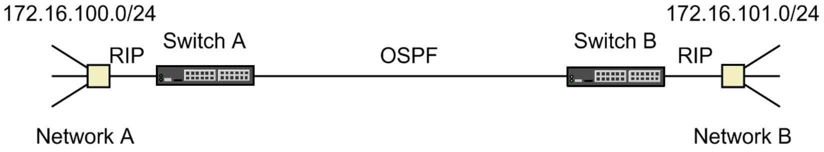
Figure 7-4: Example without subnets



■ Using an OSPF connection between subnets with the same subnet length in the same network

In the example in the following figure, routing information to Network A and Network B is advertised to Network B and network A, respectively.

Figure 7-5: Example of connection between subnets

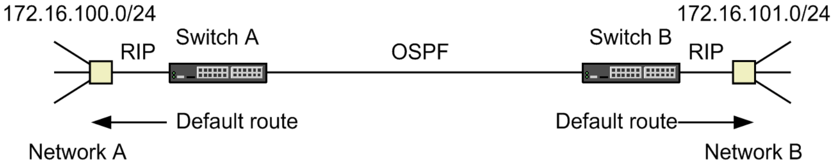


■ Advertising the default route

Configure static routes with the default address as their destination in Switch A and Switch B, and advertise the routes to the networks running RIP.

In the example in the following figure, packets whose destination address does not match the local network reach Switch A and Switch B via the default routes, and are distributed to the other network via OSPF routes.

Figure 7-6: Example of advertising default routes

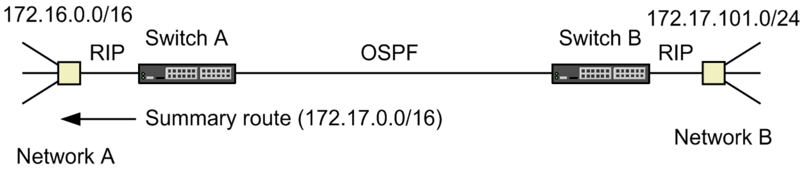


■ Advertising summarized routes

Switch A can summarize the routes to network B that it learns by OSPF/OSPF-ASE (OSPF route from external AS) into one natural mask route for advertisement to networks running RIP.

In the example in the following figure, by advertisement of the summarized route, packets destined for Network B reach Switch A and are distributed to Network B via OSPF/OSPF-ASE routes.

Figure 7-7: Example of advertising summarized routes

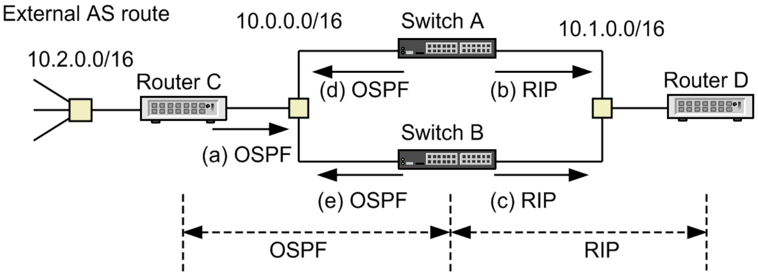


(2) When multiple protocols learn routes to the same destination

In certain network topologies, routing loops can occur when more than one protocol learns a route to the same destination. To ensure that routing loops do not occur, use route filtering if required by your network topology.

In the network topology shown in the following figure, network 10.0.0.0 uses OSPF and network 10.1.0.0 uses RIP.

Figure 7-8: Example network configuration



In this topology, the routes generated for network 10.2.0.0 can be categorized as follows:

1. External AS route advertised by router C ((a) in the figure)
2. Routes advertised from OSPF to RIP ((b) and (c) in the figure)
3. Routes advertised from RIP to OSPF ((d) and (e) in the figure)

In this example, a routing loop develops if Switch B selects (d) and Switch A selects (c), or if Switch A selects (e) and Switch B selects (b). That is, each router specifies the other as its next hop. In such a scenario, you need to configure route filtering such that routes to 10.2.0.0 advertised from OSPF to RIP by Switch A and Switch B are not learned as external AS routes advertised from RIP to OSPF.

7.1.7 Considerations when setting or changing routing protocol configurations

When you set or change the configuration of a unicast routing protocol, the protocol re-evaluates every one of its routes according to the new configuration. During this re-evaluation process, operation commands that apply to the unicast routing protocol might take a long time to execute, and MIB information might take a long time to gather via SNMP.

7.2 General IPv4 routing operations

7.2.1 List of operation commands

The following table describes the operation commands for IPv4 routing.

Table 7-4: List of operation commands

Command name	Description
show ip route	Shows routing information stored in the routing table.
clear ip route	Clears the IPv4 forwarding entries stored in the Switch and re-registers the routing entries.
show ip interface ipv4-unicast	Shows information about the IPv4 interfaces on the Switch recognized by the unicast routing program.
debug ip	Shows the packets being routed by IPv4 routing protocols in real time.
show system#1	Shows operating status.
show ip interface#2	Shows the status of IPv4 interfaces.
show netstat (netstat) (IPv4)#2	Shows the network status and statistics.
ping#2	Tests connectivity by sending a test packet to an IPv4 address associated with a specific device.
traceroute#2	Shows the route an IPv4 datagram travels to reach a destination host.
show processes cpu unicast#3	Shows the CPU usage of a unicast routing program.
restart unicast#3	Restarts the unicast routing program.
debug protocols unicast#3	Starts the operation message display for event log information output by a unicast routing program.
no debug protocols unicast#3	Stops the operation message display for event log information output by a unicast routing program.
dump protocols unicast#3	Outputs the trace information and control table information collected by the unicast routing program to a file.
erase protocol-dump unicast#3	Deletes the file of trace information and control table information generated by the unicast routing program.

#1

See *9. Checking Software Versions and Device Statuses* in the manual *Operation Command Reference Vol.1 For Version 11.10.*

#2

See *2. IPv4, ARP, and ICMP* in the manual *Operation Command Reference Vol. 2 For Version 11.10.*

#3

See *8. Routing Protocol Common to IPv4 and IPv6* in the manual *Operation Command Reference Vol. 2 For Version 11.10.*

7.2.2 Checking routes to destination addresses

If you configure IPv4 unicast routing information in the Switch, use the `show ip route` command to check whether a route to a specific destination address exists.

Figure 7-9: Example of show ip route command output

```
> show ip route
Date 20XX/07/14 12:00:00 UTC
Total: 13 routes
Destination        Next Hop        Interface      Metric   Protocol  Age
172.16/16          192.168.1.100   VLAN0010       2/0      RIP       8s ...1
192.168.1/24       192.168.1.1     VLAN0010       0/0      Connected  8s
     :
     :
>
```

1. Check whether the command output contains a route corresponding to the destination address.

7.3 Network design considerations

This section describes considerations for the design of an IPv4 network.

7.3.1 Address design

If your local addressing scheme provides more than enough addresses for every device on the network, you can achieve a relatively simple network design and avoid associated pitfalls by keeping the following considerations in mind:

1. Avoid crossing network address space boundaries by using subnet address spaces of a single large (class A or class B) network address space, rather than multiple network address spaces.
2. Use the same number of bits for each subnet mask (avoid creating variable-length subnet masks).

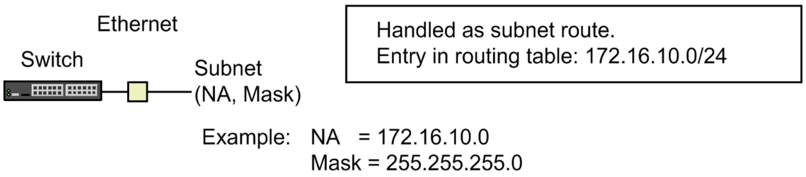
If you implement RIP-1 routing in a manner that fails to satisfy these addressing conditions, you must exercise due caution when setting the conditions for route advertisement.

7.3.2 Handling of direct connections

When addressing routing information to a directly connected broadcast interface, the Switch uses a combination of a network address (NA) and subnet mask (Mask).

The following figure illustrates the handling of direct connections.

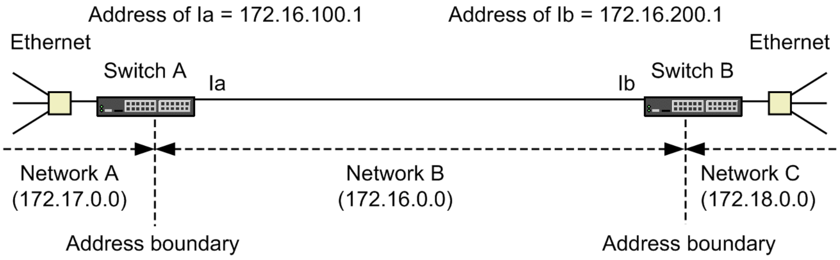
Figure 7-10: Handling direct connections



7.3.3 Designing address boundaries

When using multiple network address spaces, ensure that the address boundaries are located on the Switch itself, as shown in the figure below. An address boundary is the boundary between network addresses associated with a given natural mask, and differs from the boundary between address classes.

Figure 7-11: Example of standard network boundary design



7.4 Description of load balancing

7.4.1 Overview of load balancing

Load balancing is the practice of distributing increased traffic loads among multipaths to the same destination. It is implemented by routing control in the IP layer. This functionality limits the need for capacity upgrades by increasing bandwidth use through the aggregation of existing capacity.

This section covers load balancing at Layer 3.

The figures below illustrate load balancing using multipath routing, for scenarios involving one or many neighboring routers. In these figures, the switch uses hardware processing to relay packets from Network A to the servers in Network B over four paths at high speed.

Figure 7-12: Load balancing with multipath routing (single neighboring router)

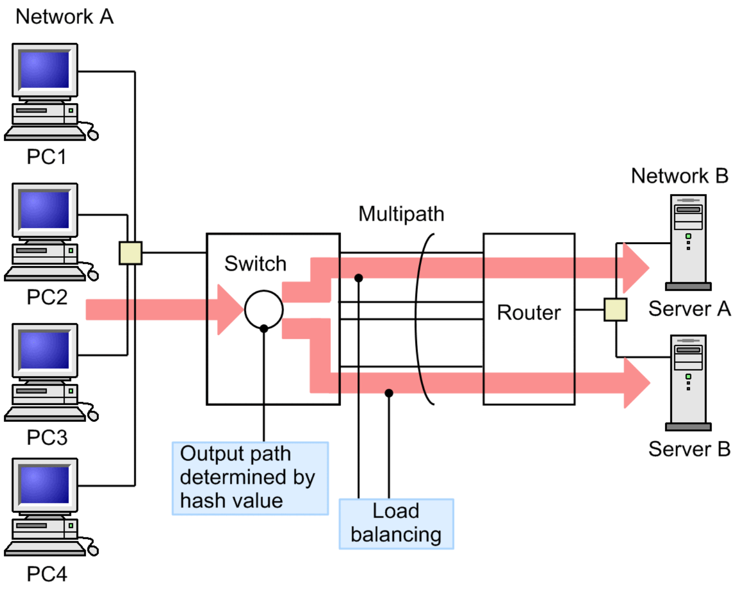
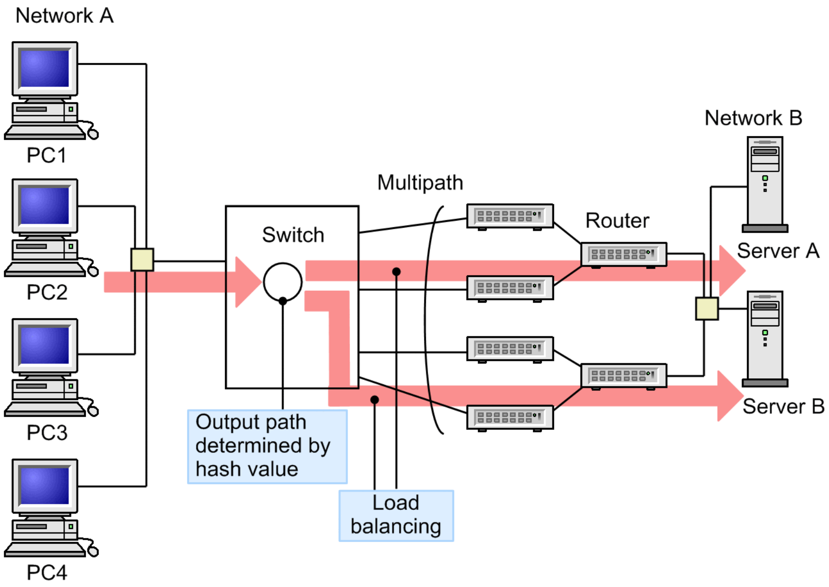


Figure 7-13: Load balancing with multipath routing (multiple neighboring routers)



7.4.2 Load balancing specifications

The following table describes the specifications for multipath routing on the Switch.

Table 7-5: Multipath specifications

Item	Specifications	Remarks
Number of multipaths for a single destination network	2 to 16	--
Maximum number of multipaths that can be specified in the configuration	1 to 16 (Multipaths are not generated if 1 is specified.)	The maximum number of multipaths is specified for each routing protocol.
Maximum number of multipath routes	For AX3800S series switches: 256, 512, or 1024 For AX3650S series switches: 128, 256, 512, or 1024	The value differs depending on the maximum number of multipaths handled by the switch. For details, see *Table 7-6: Maximum number of multipath routes*.
Routing protocols capable of generating multipaths	• Static (IPv4) • OSPF • BGP4	--
Number of multipaths in a default configuration	• Static (IPv4): 6 • OSPF: 4 • BGP4: 1 (multipaths are not generated)	--
Connection configuration	Can be used with any combination of line type and interface type. Concurrent use of different types of lines or interfaces is possible.	Multipaths between VRFs are not supported.

Legend: --: Not applicable

Table 7-6: Maximum number of multipath routes

Model	Maximum number of multipaths specified in switch configuration#1	Maximum number of multipaths handled by the switch#2	Maximum number of multipath routes the switch can handle#2, #3
AX3800S	1 to 4	4	1024#4
	5 to 8	8	512
	9 to 16, or multipath disabled#5	16	256
AX3650S	1 to 2	2	1024#4
	3 to 4	4	512
	5 to 8	8	256
	9 to 16, or multipath disabled#5	16	128

#1

The maximum number of multipaths specified for static routing (IPv4/IPv6), OSPF/OSPFv3, or BGP4/BGP4+, whichever is largest. For example, if the device configuration specifies a maximum number of multipaths of 6 for static routing and 3 for OSPF, the larger value of 6 applies.

Each routing protocol can generate a number of multipaths no greater than the maximum number defined for that protocol in the switch configuration. If a change is made to the maximum number of multipaths that affects how many multipaths the switch can handle, the changes will not take effect until the switch is restarted.

#2

The maximum value is determined at startup. If you change the maximum number of multipaths for a unicast protocol after the switch starts, it remains unchanged from the value determined at startup. To change the maximum multipath count, restart the switch after making the appropriate changes in the device configuration.

#3

The maximum number of multipath routes applies to the combined total of IPv4 and IPv6 routes.

#4

For a single path, the maximum number of paths is determined by the capacity limit for the number of table entries. For multipath, the values in the table apply.

#5

When not using static routing (IPv4/IPv6), OSPF/OSPFv3, or BGP4/BGP4+, the switch does not handle multipath routes. In this case, the maximum values in the table apply.

Using static routing setting for the AX3650S switches as an example, the following table describes how the maximum multipath values change as you change the device configuration or restart the device.

Table 7-7: Changes in maximum multipath values (for static routing)

Seq.	Status	Maximum number of multipaths for static routing	Maximum number of multipaths handled by the switch	Maximum number of multipath routes the switch can handle
1	The switch starts without static routing configured.	--	16	128
2	You add a static route.	6#1	16	128
3	The switch restarts.	6	8	256
4	You set the maximum multipath count for static routing to 3.	3	8	256
5	The switch restarts.	3	4	512
6	You set the maximum multipath count for static routing to 5.	4#2	4	512
7	The switch restarts.	5#2	8	256

Legend: --: Not applicable

#1

If you configure static routes without specifying a maximum multipath value, the default value applies to the number of static multipaths. For details, see *7.4.3 Notes on using load balancing*.

#2

You cannot generate statically routed multipaths that exceed the maximum number of multipaths the switch can handle. However, after you restart the switch, the maximum number of multipaths handled by the switch changes, and the value specified for static routing multipaths takes effect.

The following table describes the load balancing specifications implemented by the Switch.

Table 7-8: Load balancing specifications

Item	Specifications	Remarks
Path selection in multipaths	A value (hash value) is calculated and allocated to the designated output path. This hash value is derived from the following four fields: • Source IP address • Destination IP address • Source TCP/UDP port number • Destination TCP/UDP port number The transmission order is guaranteed within each session.	--
Multipath information in routing table	Hash values are assigned on a substantially equal basis to the output interfaces in the routing table.	See notes *1* and *2* in *7.4.3 Notes on using load balancing*.
Weighting of paths	Not available	See note *1* in *7.4.3 Notes on using load balancing*.
Processing of packets exceeding output bandwidth	Not reallocated to another path. The packets are retained within the switch as long as bandwidth continues to be exceeded. However, any overflow will be discarded.	--

Legend: --: Not applicable

7.4.3 Notes on using load balancing

1. The switch selects one of 16 paths based on the packet's hash value. For this reason, packets are not necessarily distributed equally among the paths to a given destination network.
2. Because the switch applies no weighting to individual paths, line speed is not taken into account when distributing packets. However, you can give greater weighting to faster lines by using a multihomed connection. In this case, you must use a redundant configuration to guard against failures.
3. If the switch attempts to send packets in a manner that continuously exceeds the bandwidth of a path selected on the basis of hash values, these packets will be discarded without being re-allocated to another path.
4. Note the following when using the `traceroute` command to check the paths selected for load balancing:
   - Although the response to the `traceroute` command bears the IP address of the interface that received the command in its source field, the response does not necessarily originate from that interface.
   - If the `traceroute` command is received by a multihomed interface, it cannot determine the address to which the neighboring device sent the command. The interface responds by using one of the addresses in the multihomed environment as the source IP address.
5. When using load balancing, relay performance may decline dramatically if traffic is

concentrated on a specific route (gateway). In such a case, configure a static ARP entry for all gateways.

6. When multipaths for a BGP4 route include a null interface as a result of next hop resolution performed on the basis of an IGP route that specifies a null interface, the switch is unable to use this route to relay packets. In such a case, use the BGP configuration command `bgp nexthop` to ensure that the IGP route that specifies the null interface is not used for next hop resolution of the BGP4 route.

   If a next hop to which the Switch is not directly connected is included in the multipath static route, and that next hop is resolved in a route in which a null interface is used as a next hop, that route cannot be used for forwarding.

7. If you configure a unicast routing protocol without specifying a maximum number of multipaths, the following default values apply:

   - Static (IPv4): 6
   - OSPF: 4
   - BGP4: 1 (does not generate multipaths)

8. You cannot change the maximum number of multipath routes after the Switch has begun operation. To change this setting, modify the maximum number of multipaths in the configuration for each unicast routing protocol (static routing, OSPF, BGP4), and then restart the switch.

9. Although you can configure a static route in which multiple VRFs exist in the next hop, the multipath of the created route consists only of one VRF.

   The path is selected from among the next hops in a VRF in which the next hop with the highest weight exists, using that next hop as a basis. **[OS-L3SA]**

7.5 Load balancing configuration

7.5.1 List of configuration commands

The following table describes the configuration commands for load balancing.

Table 7-9: List of configuration commands

Command name	Description
ip route static maximum-paths#1	Specifies the maximum number of paths (maximum number of next hops) that the routing protocol will generate for a static IPv4 route.
maximum-paths (OSPF)#2	Specifies the maximum number of paths in the route when multiple paths (next hops) of equal cost exist for an OSPF-generated route.
maximum-paths (BGP4)#3	Generates multipaths using the specified value as the maximum number of paths when multiple routing information entries of equal cost to a given destination exist.

#1

See *10. Static Routing (IPv4)* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

#2

See *12. OSPF [OS-L3SA]* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

#3

See *13. BGP4 [OS-L3SA]* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

7.5.2 Configuring the maximum number of multipaths handled by the Switch

The maximum number of multipaths used for each protocol in the Switch determines the maximum number of multipaths that the Switch handles and the maximum number of multipath routes that the Switch can handle.

The maximum number of multipaths is determined at startup, and any changes made by configuration commands do not take effect until the switch is restarted. If you use a configuration command in a manner that affects the maximum number of multipaths, a warning-level operation message appears prompting you to restart the switch. The new values for the maximum number of multipaths and multipath routes will take effect after the Switch restarts.

Points to note

By default, the switch handles a maximum of 16 multipaths. For multipath routes, AX3800S series switches handle a maximum of 256 and AX3650S series switches handle a maximum of 128. To change the maximum number of multipaths that the switch handles, you must restart the Switch after setting the maximum number of multipaths in the configuration of a unicast routing protocol. For this reason, we recommend that you set the maximum number of multipaths when you first deploy the switch.

The configuration below uses an example of IPv4 static routing.

Command examples

1.
```
(config)# ip route static maximum-paths 2
```

In configuration mode, sets 2 as the maximum number of multipaths for IPv4 static routes.

2.
```
(config)# ip route 192.168.2.0 255.255.255.0 172.16.1.100
noresolve
(config)# ip route 192.168.2.0 255.255.255.0 172.16.2.100
noresolve
```
In configuration mode, establishes a static IPv4 multipath route (192.168.2.0/24).

3.
```
(config)# save
(config)# exit
```
Saves the settings and switches from configuration mode to administrator mode.

4.
```
# reload
```
Restarts the Switch.

7.5.3 Load balancing using static routes

See *8.2.4 Configuring a multipath route*.

7.5.4 Load balancing using OSPF [OS-L3SA]

See *10.2.6 Configuring multipath*.

7.5.5 Load balancing using BGP4 [OS-L3SA]

See *12.5.3 Configuring BGP4 multipath* and *(2) Configuring BGP4 multipath*.

7.6 Load balancing operation

7.6.1 Checking the maximum number of multipaths handled by the Switch

Use the `show system` command to check the maximum number of multipaths handled by the Switch.

Figure 7-14: Checking the maximum number of multipaths handled by the Switch

```
>show system
           :
           :
    Device resources
        Current selected swrt_table_resource: l3switch-2
        Current selected swrt_multicast_table: On
        Current selected unicast multipath number: 8
           :
           :
>
```

7.6.2 Checking selected paths

(1) Checking the routing information

Execute the `show ip route` command to see whether the multipath route settings have been applied correctly.

Figure 7-15: Displaying multipath routing information

```
> show ip route
Date 20XX/07/14 12:00:00 UTC
Total: 13 routes
Destination        Next Hop        Interface      Metric   Protocol  Age
192.168.1/24       192.168.1.1     VLAN0010       0/0      Connected 19m 46s
192.168.1.1/32     192.168.1.1     VLAN0010       0/0      Connected 19m 46s
192.168.2/24       192.168.2.1     VLAN0020       0/0      Connected 19m 46s
192.168.2.1/32     192.168.2.1     VLAN0020       0/0      Connected 19m 46s
192.168.3/24       192.168.3.1     VLAN0030       0/0      Connected 19m 46s
192.168.3.1/32     192.168.3.1     VLAN0030       0/0      Connected 19m 46s
172.16/16          192.168.1.200   VLAN0010       0/0      Static      9s
                   192.168.2.200   VLAN0020       -        -         -
                   192.168.3.200   VLAN0030       -        -         -
     :
     :
>
```

(2) Checking reachability of specific destination addresses

Use the `ping` *<IPv4 Address>* `specific-route source` *<Source Address>* command at each interface of the Switch used for load balancing to confirm that the interface can communicate with the remote system. As *<Source Address>* of the `ping` command, specify the local IPv4 address on the Switch of the interface performing load balancing.

7.7 Description of route summarization

7.7.1 Overview

Route summarization is a method of representing routing information from one or more source routes by generating routing information that is shorter than the network masks that include information about the corresponding paths. If the information for multiple routes is condensed into one set of routing information containing the information for the multiple routes, and then that summarized information is reported to (for example) neighboring routers, the number of items of routing information on the network is reduced. For example, the switch can generate the summarized route 172.16.0.0/16 having learned routing information for addresses 172.16.178.0/24 and 172.16.179.0/24.

You must explicitly enable route summarization by executing the `ip summary-address` configuration command. You can assign a distance to a summarized route. If you do not specify a distance, the default of 130 is used. The switch cannot generate a summarized route without first learning the routing information that serves as its source.

7.7.2 Transferring summarized routes

Summarized routes are configured as reject routes. Packets that lack a higher-priority route are discarded.

Summarized routes are configured in this way to prevent routing loops. When the Switch advertises a summarized route, packets destined for that route are relayed to the Switch. If the Switch then transfers a packet destined to a route that is not a summarization source route via the designated next-best route (such as the default route), a routing loop can occur between the Switch and the destination system of the default route. Summarized routes are designated as reject routes to prevent this scenario.

However, summarized routes configured with the `noinstall` parameter specified do not discard packets. If there is a default route or other next-best route, packets will be forwarded via that route. Use the `noinstall` parameter when you wish to advertise a summarized route but would rather it forward packets via the next-best route than discard them.

7.7.3 Summarizing AS_PATH attributes

When a summarized route includes a BGP4 route among its source routes, the summarized route is tagged with the path attribute of the BGP4 route. If the source routes include more than one BGP4 route, the path attributes are summarized among those routes. The `AS_PATH` and `COMMUNITIES` attributes of summarized routes are edited as follows:

(1) AS_PATH attribute

The initial AS path segment within the `AS_SEQUENCE` type of the `AS_PATH` attribute which is common to the summarization source routes is assigned as the `AS_SEQUENCE` type of the `AS_PATH` attribute of the summarized route. All other AS paths in the `AS_SEQUENCE` type and those outside the `AS_SEQUENCE` type appear in the `AS_SET` type of the `AS_PATH` attribute of the summarized route only if you execute the `ip summary-address` configuration command with the `as_set` parameter specified.

(2) COMMUNITIES attribute

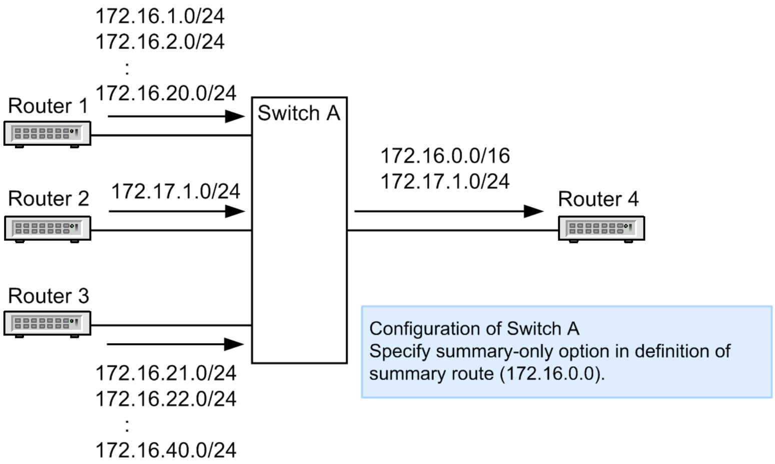
Every community in the summarization source BGP4 routes is set to the `COMMUNITIES` attribute of the summarized route.

7.7.4 Suppressing advertisement of summarization source route

After you summarize a route, it is possible to advertise the summarized route but exclude its source routes from being advertised. For example, you can advertise RIP routes that have not been summarized while excluding those that have.

You can suppress the advertisement of source routes on an individual basis or for all summarized routes. To suppress advertisement for an individual summarized route, specify the `summary-only` parameter when you execute the `ip summary-address` configuration command. The following figure shows an example in which advertisement is suppressed for a summarized route.

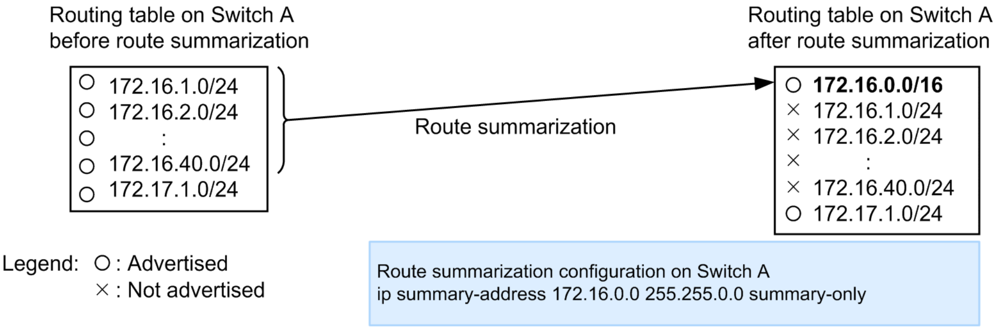
Figure 7-16: Example of suppressing summarization source route advertisement



Configuration of Switch A
Specify summary-only option in definition of summary route (172.16.0.0).

Switch A receives advertisements and learns routes for packets destined for the address range 172.16.1.0/24 to 172.16.20.0/24 from Router 1 and 172.17.1.0/24 from Router 2, and the address range 172.16.21.0/24 to 172.16.40.0/24 from Router 3. Switch A configures the advertised route filter to advertise the summarized route 172.16.0.0/16 and learned route 172.17.1.0/24 to Router 4. At this time, if you specified the `summary-only` parameter when you configured the switch to generate the summarized route 172.16.0.0/16 from learned routes, you do not need to configure the advertised route filter to prevent the switch from advertising the summary source routes. The following figure shows an example configuration for route summarization, and the entries in the routing table before and after summarization.

Figure 7-17: Example of route summarization configuration and routing entries before and after summarization



Route summarization configuration on Switch A
ip summary-address 172.16.0.0 255.255.0.0 summary-only

7.8 Route summarization configuration

7.8.1 List of configuration commands

The following table describes the configuration commands for route summarization.

Table 7-10: List of configuration commands

Command name	Description
ip summary-address	Generates a summarized IPv4 route.
redistribute (BGP4)#	Sets the protocol types of routes advertised from BGP4.
redistribute (OSPF)#	Sets the protocol types of routes advertised from OSPF.
redistribute (RIP)#	Sets the protocol types of routes advertised from RIP.

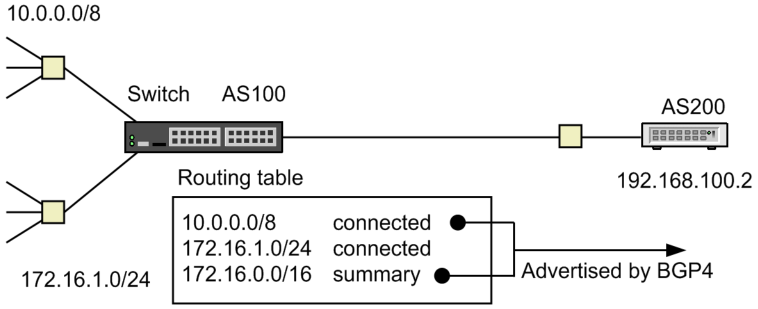
#

See *14. Route Filters (IPv4 and IPv6)* in the manual *Configuration Command Reference Vol. 2 For Version 11.10.*

7.8.2 Configuring route summarization and advertisement of summarized routes

Configure summarization of directly connected routes as summarization source routes. Also, configure the switch to re-advertise the summarized and directly connected routes by BGP4 without re-advertising directly connected routes that became summarization source routes.

Figure 7-18: Configuration for advertising summarized routes by BGP4



Points to note

To generate a summarized route, use the `ip summary-address` command. Use the `redistribute summary` command to configure BGP4 to advertise the summarized route.

Command examples

1.
   ```
   (config)# ip summary-address 172.16.0.0 255.255.0.0
   summary-only
   ```

   Configures the switch to generate the summarized route 172.16.0.0/16. By specifying summary-only, you suppress re-advertisement of the directly connected route 172.16.1.0/24 serving as the summarization source.

2.
   ```
   (config)# router bgp 100
   (config-router)# neighbor 192.168.100.2 remote-as 200
   ```

Establishes a BGP4 connection with neighboring router 192.168.100.2.

3. `(config-router)# redistribute summary`

   Uses BGP4 to re-advertise the summarized route.

4. `(config-router)# redistribute connected`

   Uses BGP4 to re-advertise the directly connected route.

7.9 Route summarization operations

7.9.1 List of operation commands

The following table describes the operation commands for route summarization (IPv4).

Table 7-11: List of operation commands

Command name	Description
show ip route	Shows routing information stored in the routing table.
show ip rip	Shows information about the RIP protocol.
show ip ospf	Shows information about the OSPF protocol.
show ip bgp	Shows information about the BGP protocol.
restart unicast#	Restarts the unicast routing program.
dump protocols unicast#	Outputs the trace information and control table information collected by the unicast routing program to a file.
erase protocol-dump unicast#	Deletes the file of trace information and control table information generated by the unicast routing program.

#

See *8. Routing Protocol Common to IPv4 and IPv6* in the manual *Operation Command Reference Vol. 2 For Version 11.10*.

7.9.2 Checking summarized routes

You can display information about summarized routes entered in the routing table. The following figure shows an example of displaying a summarized route.

Figure 7-19: Example of displaying a summarized route

```
> show ip route summary_routes
Date 20XX/07/14 12:00:00 UTC
Total: 1 routes
Destination        Next Hop        Interface      Metric   Protocol  Age
172.16/16          ----            -              0/0      Summary   50s
```

You can also display information about the active routes in a specific network (172.16.0.0/16). The following figure shows an example of displaying active routes.

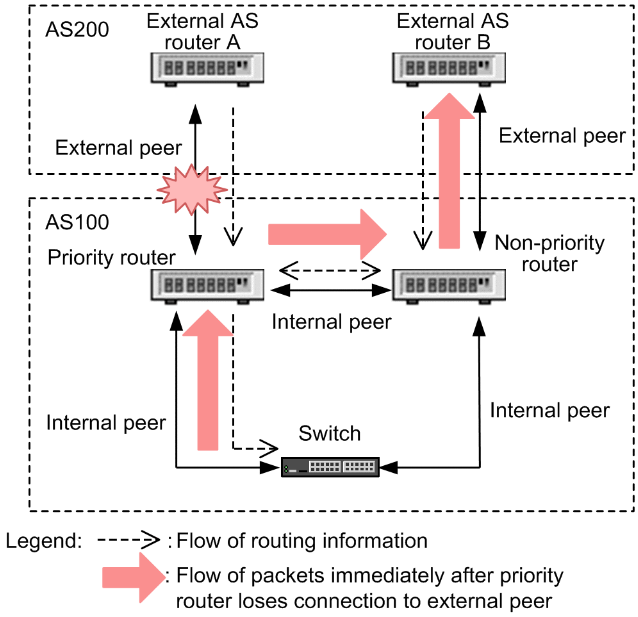
Figure 7-20: Example of displaying active routes

```
> show ip route 172.16.0.0/16 longer-prefixes
Date 20XX/07/14 12:00:00 UTC
Total: 3 routes
Destination        Next Hop        Interface      Metric   Protocol  Age
172.16/16          ----            -              0/0      Summary   56s
172.16.1/24        172.16.1.1      VLAN0010       0/0      Connected 365d
172.16.1.1/32      172.16.1.1      VLAN0010       0/0      Connected 365d
```

7.10 Route deletion delay functionality

The route deletion delay functionality delays the deletion from the routing table of routes that routing protocols have invalidated. Its use enables packet forwarding to be maintained over existing routes until an alternate route is available. The following figure shows an example of applying the route deletion delay functionality.

Figure 7-21: Application example of route deletion delay functionality



In the figure above, a peer disconnect occurs between the priority router and the external-AS router A. This causes the Switch to disable its BGP4 routes until they can be re-learned from a non-priority router. However, the route deletion delay functionality stops the switch from deleting the routing information from the routing table, and packet forwarding is maintained through the following route:

Priority router -> Non-priority router -> External-AS router B

To enable the route deletion delay functionality, execute the `routing options delete-delay` configuration command with a value from 5 to 4294967295 (seconds) specified as the timer value.

7.11 Description of a VRF [OS-L3SA]

Virtual Routing and Forwarding (VRF) is a technology that logically partitions the routing space. VRF allows a router to retain multiple instances of routing tables to perform concurrent transmission according to each routing table.

Because VRF separates the IP address spaces, each VRF instance can use the same IP address more than once. Routing protocols operate independently for each VRF.

7.11.1 Scope of support

The following table describes the IPv4 routing protocol functionality supported by VRF.

Table 7-12: Functionality supported in VRFs

Functionality		Supported
Static routing		Y
Dynamic routing	RIP-1	Y
	RIP-2	Y
	OSPF	Y
	BGP4	Y
Multipaths and load balancing		Y#1
Route summarization		Y
Route deletion delay functionality		Y
Graceful restart		Y#2
Limiting the number of routes		Y
Extranet	Route exchange between VRFs	Y
	Static routing across VRFs	Y
	Policy-based routing	Y

Legend: Y: Supported

#1: Multipaths between VRFs are not supported.

#2: Only OSPF helper router functionality and BGP4 receiving router functionality are supported.

7.11.2 Limiting the number of routes

You can limit the number of routes specified for each VRF.

(1) Suppressing the addition of routes

When the number of routes per VRF exceeds the specified maximum number of routes, the Switch suppresses subsequently learned routes to be added to the forwarding table.

The suppressed routes are retained in the routing table and will be added sequentially when the routes that have been added are deleted, freeing up space in the forwarding table.

(2) Outputting warning messages

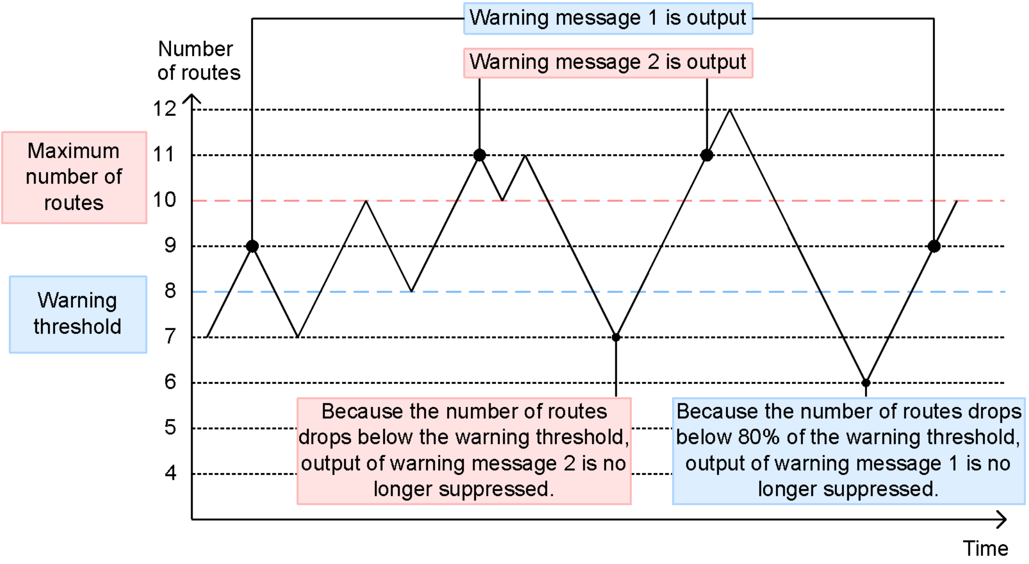
When the number of routes per VRF exceeds the specified warning threshold or the maximum number of routes, a warning message is output.

Additional output of warning message 1, which is output when the warning threshold is exceeded, is suppressed until the number of routes drops below 80% of the warning threshold.

Additional output of warning message 2, which is output when the maximum number of routes is exceeded, is suppressed until the number of routes drops below the warning threshold.

The following figure shows the relationship between the number of routes and the output of warning messages.

Figure 7-22: Number of routes and the output of warning messages



(3) Notes

If you change the maximum number of routes in the configuration to a value smaller than the number of routes registered in the forwarding table, the number of routes registered in the forwarding table is not immediately reduced to the new maximum number of routes.

To forcibly reduce the number of routes registered in the forwarding table to the maximum number of routes that you specified, execute the `clear ip route` operation command.

7.11.3 Extranet

The following two methods are available for implementing an extranet:

- Route exchange between VRFs
- Static routing across VRFs

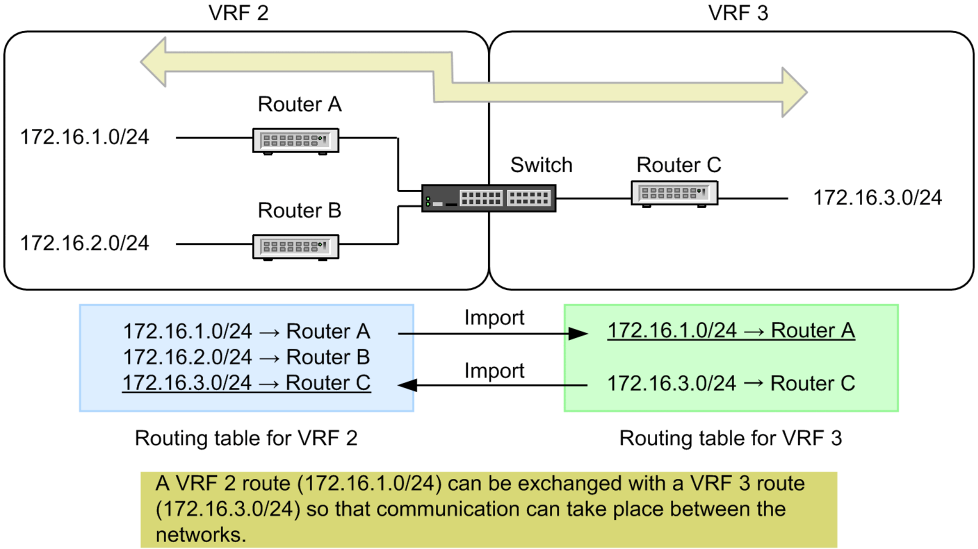
The following describes the exchange of routes between VRFs that uses a routing table and static routing across VRFs. Routes that can be imported between VRFs are also described.

(1) Route exchange between VRFs

An extranet can be implemented by exchanging routing information held by VRFs with other VRFs.

The following figure shows an example of exchanging routes between VRFs.

Figure 7-23: Route exchange between VRFs



172.16.1.0/24 → Router A
172.16.2.0/24 → Router B
172.16.3.0/24 → Router C

172.16.1.0/24 → Router A
172.16.3.0/24 → Router C

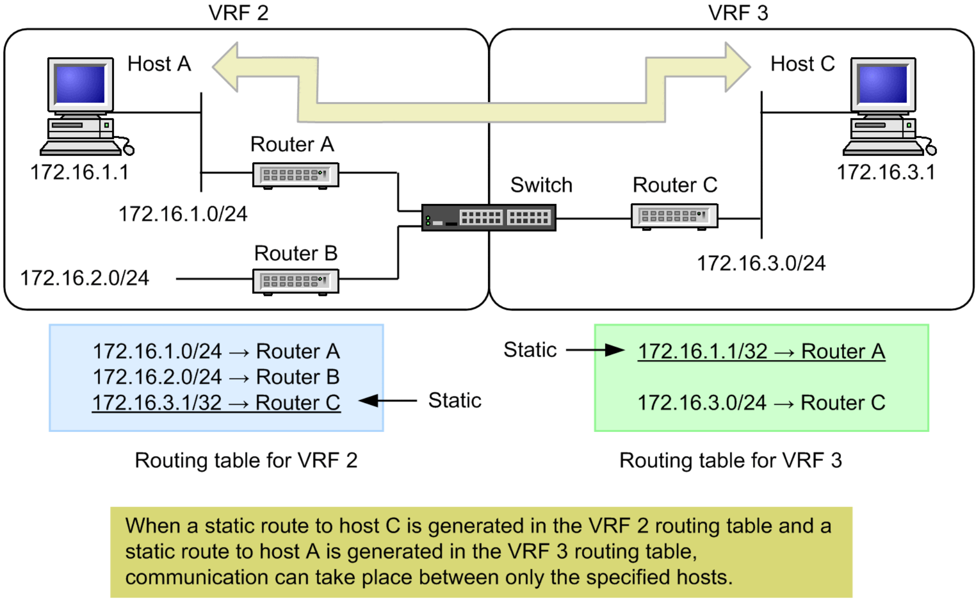
A VRF 2 route (172.16.1.0/24) can be exchanged with a VRF 3 route (172.16.3.0/24) so that communication can take place between the networks.

(2) Static routing across VRFs

An extranet can be implemented by creating a static route that uses a gateway for another VRF as the next hop.

The following figure shows static routing across VRFs.

Figure 7-24: Static routing across VRFs



172.16.1.0/24 → Router A
172.16.2.0/24 → Router B
172.16.3.1/32 → Router C

172.16.1.1/32 → Router A
172.16.3.0/24 → Router C

When a static route to host C is generated in the VRF 2 routing table and a static route to host A is generated in the VRF 3 routing table, communication can take place between only the specified hosts.

(3) Routes that can be imported between VRFs

The following table describes the types of routes that can be imported from another VRF or the global network.

Table 7-13: Types of routes that can be imported from another VRF or the global network

Route type	Importable?
Inactive route	N

Route type	Importable?
Route whose deletion is pending	N
Route imported for an extranet	N
Summarized route	Y
Route of an IPv4 device address specified for a loopback interface	Y
Routes directly connecting VLAN interfaces	Y
Route whose output interface is a VLAN interface	Y
Route whose output interface is a loopback interface	Y
Route whose output interface is a null interface	Y

Legend: Y: Imported, N: Not imported

If the routes to be imported match multiple route types, import is possible only if all the matching route types can be imported.

7.12 VRF configuration [OS-L3SA]

7.12.1 List of configuration commands

The following table describes the commands used to configure VRFs.

Table 7-14: List of configuration commands

Command name	Description
ip route[#1]	Generates an IPv4 static route.
match vrf[#2]	Configures a route-map to use a VRF as filtering conditions.
route-map[#2]	Configures a route-map.
import inter-vrf[#3]	Applies a filter to control which routes are imported from another VRF or the global network.
maximum routes[#3]	Sets the maximum number of routes for the VRF and the threshold for output of a warning message.
vrf definition[#3]	Configures a VRF.

#1

See *10. Static Routing (IPv4)* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

#2

See *14. Route Filters (IPv4 and IPv6)* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

#3

See *30. VRF [OS-L3SA]* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

7.12.2 Configuring the maximum number of routes

Set the maximum number of routes for the VRF and the threshold for output of a warning message.

Points to note

The example below shows how to use the `maximum routes` command to set the maximum number of routes and the threshold for output of a warning message.

Command examples

1.
```
(config)# vrf definition 2
```
Switches to VRF 2 configuration mode.

2.
```
(config-vrf)# maximum routes 1000 80
```
Sets 1000 as the maximum number of routes handled by VRF2. Also, sets the threshold for outputting a warning message to 80%.

7.12.3 Configuring extranets

For details about how to configure extranets, see *8.2.7 Configuring static routes across VRFs*

[OS-L3SA] and *13.2.8 Extranet [OS-L3SA]*.

7.13 VRF operation [OS-L3SA]

7.13.1 List of operation commands

The following table describes the operation commands for VRFs.

Table 7-15: List of operation commands

Command name	Description
show ip route	Shows routing information stored in the routing table.
show ip static	Shows information related to a static route.
show ip vrf	Shows the IPv4 information of a VRF.

7.13.2 Checking the maximum number of routes

Use the `show ip vrf` command to display the current number of routes registered in the VRF forwarding table and the maximum number of routes that can be specified.

Figure 7-25: Results of executing the show ip vrf command

```
> show ip vrf 2
Date 20XX/12/20 12:00:00 UTC
VRF              Routes        ARP
2                270/1000      7/50               ...1
>
```

1. The numerator represents the current number of routes, and the denominator represents the maximum number of routes.

Figure 7-26: Results of executing the show ip vrf detail command

```
> show ip vrf 2 detail
Date 20XX/12/20 12:00:00 UTC
VRF 2
  Maximum routes: 1000, Warn threshold: 80%, Current routes: 270     ...1
  Maximum ARP entries: 50, Current ARP entries: 7
  Import inter-vrf: -
Interface
Name           Local              Remote           Status
VLAN0010       192.168.10.1/24    192.168.10.255   Up
loopback2      2.2.2.2/32         2.2.2.2          Up
loopback2      127.0.0.1/8        127.0.0.1        Up
>
```

1. Information is displayed in the following order: the maximum number of routes, the threshold for the output of a warning message, and the current number of routes.

7.13.3 Checking extranets

For details about how to check extranets, see *13.3.10 Checking extranet [OS-L3SA]*.

Chapter

8. Static Routing (IPv4)

This chapter describes IPv4 static routing.

8.1 Description

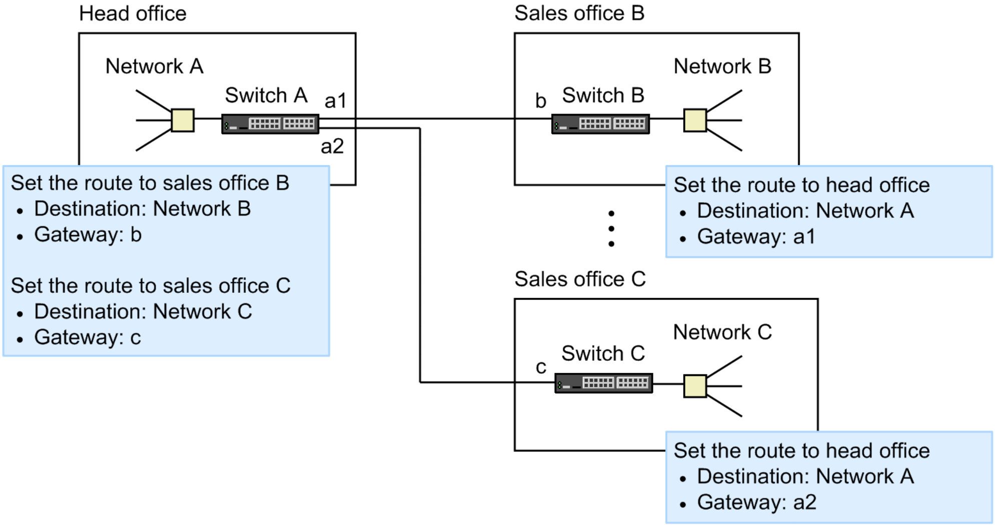
8.1.1 Overview

In static routing, packets are forwarded according to the routing information (static routes) set by using configuration commands.

In the Switch, you can set multiple forwarding routes, including a default route, to a particular destination subnetwork or host.

The figure below shows an example of a network configuration that uses static routing. Set the static route from the head office to each sales office, and then from the sales offices to the head office. In this example, the sales offices cannot communicate with each other.

Figure 8-1: Example of network configuration using static routing



Set the route to sales office B
- Destination: Network B
- Gateway: b

Set the route to sales office C
- Destination: Network C
- Gateway: c

Set the route to head office
- Destination: Network A
- Gateway: a1

Set the route to head office
- Destination: Network A
- Gateway: a2

8.1.2 Route selection conditions

In static routing, routes to the same destination network are grouped by distance, and a route from the group that has the smallest distance is selected.

When the maximum number of multipath routes is set to 2 or greater, the routes are configured according to the priorities given in the following table. When the maximum number of multipath routes is set to 1, the route with the highest priority is selected.

The default maximum number of multipaths is 6. You can change this value by using the `ip route static maximum-paths` configuration command.

Table 8-1: Route selection priority

Priority	Description
High	Selects the route with the greatest weight.
Low	Selects the route with the smallest next-hop address.

8.1.3 Specifying a forwarding route for static routing

For the forwarding route (gateway), you can specify a directly connected neighboring gateway or a remote gateway that is not directly connected to the switch. With a neighboring gateway, the status of the connected interface governs path generation and deletion. With a remote gateway, the presence or absence of a route to that gateway controls path generation and deletion. The Switch uses remote gateways as the default gateway type. When you set up a neighboring gateway using the `ip route` configuration command, specify the `noresolve` parameter.

Two additional parameters can be specified for the path to a neighboring or remote gateway specified in the command. Both parameters stop packets from being sent via that gateway. Packet transmission will also be disabled if you have specified a null interface for the gateway.

- `noinstall` parameter

  If you specify the `noinstall` parameter for a static route, that route will not be used for packet transfer. If there is a default route or other next-best route, packets will be forwarded via that route. Use the `noinstall` parameter when you want to set a static advertising route but have packets forwarded through a different route.

- `reject` parameter

  If you specify the `reject` parameter for a static route, that route becomes a reject route. Any packets that match the route are discarded. An ICMP Unreachable message notifies the source device that the packet has been discarded. Use the `reject` parameter when you want to set a static advertising route but want to discard packets for which a route with a higher priority has not been set in the Switch. You can also use this parameter to prevent packets destined for a particular address or destination from being forwarded via this route.

- Null interface

  If you specify only a null interface for a static route without specifying a gateway, all packets on that static route will be discarded. Unlike packets discarded by the `reject` parameter, no ICMP message is sent to the source device. Specify a null interface when you want to set the same behavior as the `reject` parameter but do not want ICMP packets to be returned. For details on null interface behavior, see *3. Null Interface (IPv4)*.

8.1.4 Dynamic monitoring

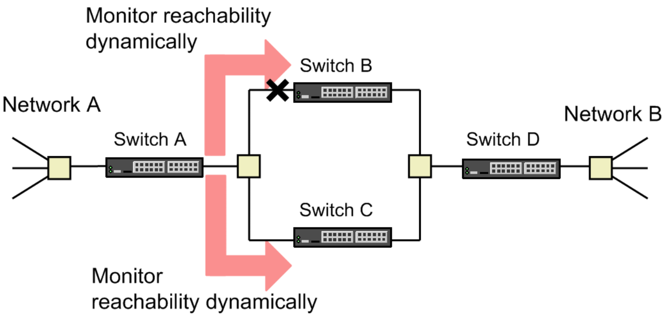
The generation and deletion of static routes is governed according to the status of the interface directly connected to the gateway, or the presence or absence of a route to that gateway. Consequently, there is no guarantee that packets will reach the gateway even if a route has been created. The Switch provides functionality for dynamically monitoring packet delivery by polling gateways at regular intervals using ICMPv4 Echo Request and Echo Reply messages. By using this monitoring functionality, you can control static route generation so that a route is created only when the route generation and deletion conditions described in *8.1.3 Specifying a forwarding route for static routing* are met and, moreover, the packets are reliably deliverable.

Even if a gateway that was unreachable becomes reachable, the route is not generated at that time. Rather, the route is generated after the reachability of the gateway is monitored for a set period and stability is confirmed.

(1) Path switching by dynamic monitoring of static routes

The following figure shows an example of monitoring static routes dynamically.

Figure 8-2: Example of dynamic monitoring of static routes


Monitor reachability dynamically
Switch B
Network A
Switch A
Network B
Switch D
Switch C
Monitor reachability dynamically

In this example, two static routes to network B have been set up in Switch A: a preferred route via Switch B and a next-best route via Switch C. Without dynamic monitoring, if an error occurs in the Switch B interface, the static route via Switch B would not be deleted because the interface on the Switch A side is still working normally. As a result, the path is not switched to the static route via Switch C, so communication between Switch A and network B ceases.

With dynamic monitoring, however, the monitoring functionality of Switch A detects that Switch B is unreachable although the interface on the Switch A side is normal. The static route through Switch B is deleted, and the path is switched to the static route via Switch C, assuring normal communication between Switch A and network B.

(2) Timing of static route generation, deletion, and restoration when using dynamic monitoring

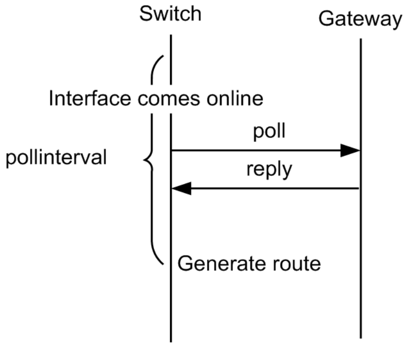
The timing for generating, deleting, or restoring a static route by dynamic monitoring depends on the values set in the `ip route static poll-interval` and `ip route static poll-multiplier` configuration commands.

In the following description, the relevant settings are `pollinterval` in `ip route static poll-interval`, and `invalidcount` and `restorecount` in `ip route static poll-multiplier`.

(a) Timing for generating a route

The Switch polls a gateway when its interface comes online or some other triggering event occurs. If a response is received, a static route through that gateway is created when the next polling time arrives (`pollinterval`). The following figure shows an example of route generation by dynamically monitoring static routes.

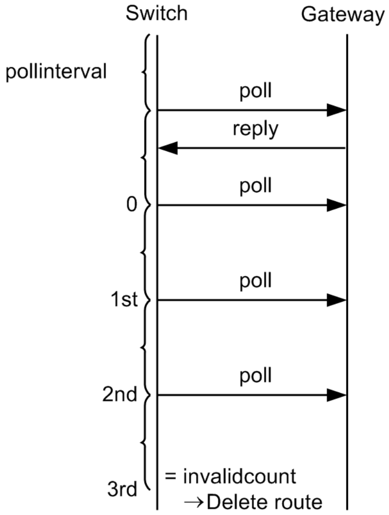
Figure 8-3: Example of generating a route by dynamic monitoring of static routes



(b) Timing for deleting a route

A static route through a gateway that is polled at regular intervals (`pollinterval`) will be deleted if no response is received after the number of consecutive poll attempts set in the `invalidcount` parameter. For example, if the `invalidcount` setting is 3, the route will be deleted if no response is received after three consecutive polls. If the interface goes down or some other trigger for route generation is lost, the static route will be deleted just as if the gateway were not being polled (`poll` parameter not specified). The following figure shows an example of route deletion by dynamically monitoring static routes.

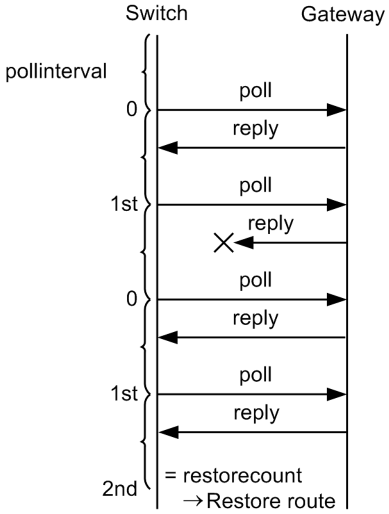
Figure 8-4: Example of deleting a static route by dynamic monitoring (invalidcount = 3)



(c) Timing for restoring a route

A static route deleted as a result of dynamic monitoring will be restored when the number of consecutive responses received from the polled gateway matches the number set in the `restorecount` parameter. For example, if the `restorecount` setting is 2, the route will be restored when responses are received for two consecutive polls. The following figure shows an example of restoring a static route as a result of dynamic monitoring.

Figure 8-5: Example of restoring a static route by dynamic monitoring (restorecount = 2)



8.2 Configuration

8.2.1 List of configuration commands

The following table describes the configuration commands for static routing (IPv4).

Table 8-2: List of configuration commands

Command name	Description
ip route	Generates an IPv4 static route.
ip route static poll-interval	Specifies the interval for polling the gateway.
ip route static poll-multiplier	Specifies the number of consecutive poll attempts and responses.

8.2.2 Configuring the default route

Set the default static route.

Points to note

Use the `ip route` command for setting a static route. To set a default static route, specify 0.0.0.0 as the destination address and 0.0.0.0 as the mask.

Command examples

1.
```
(config)# ip route 0.0.0.0 0.0.0.0 10.1.1.50
```
Specifies the remote gateway 10.1.1.50 as the next hop of the default route.

8.2.3 Configuring single-path routes

Set single-path static routes. The distances will determine path priority.

Points to note

For the static route you are setting as the alternate route, specify a distance larger than that of the preferred route.

Command examples

1.
```
(config)# ip route 192.168.1.0 255.255.255.0 10.1.1.100 100
```
Specifies the remote gateway 10.1.1.100 as the next hop of the static route 192.168.1.0/24. Also, specify 100 as the distance.

2.
```
(config)# ip route 192.168.1.0 255.255.255.0 172.16.1.100 200
noresolve
```
Specifies neighboring gateway 172.16.1.100 as the next hop of static route 192.168.1.0/24. Also, specify 200 as the distance. The Switch will use this path as the alternate route if the route to gateway 10.1.1.100 becomes invalid.

8.2.4 Configuring a multipath route

Set a multipath static route.

Points to note

Use the `ip route` command to configure a multipath route by omitting the distance or by specifying the same distance for static routes to the same destination.

Command examples

1.
```
(config)# ip route 192.168.2.0 255.255.255.0 172.16.1.100
noresolve
```

   Specifies neighboring gateway 172.16.1.100 as the next hop of static route 192.168.2.0/24.

2.
```
(config)# ip route 192.168.2.0 255.255.255.0 172.16.2.100
noresolve
```

   Specifies neighboring gateway 172.16.2.100 as the next hop of the static route 192.168.2.0/24. The static route 192.168.2.0/24 is therefore configured as a multipath route through neighboring gateways 172.16.1.100 and 172.16.2.100.

8.2.5 Applying the dynamic monitoring functionality

Before you can apply the dynamic monitoring functionality to static routes, you must set an appropriate interval for polling the gateway and adjust the timing for route deletion and generation.

Points to note

To set the polling interval and the number of consecutive poll failures or responses, use the `ip route static poll-interval` command and `ip route static poll-multiplier` command. To apply dynamic monitoring to a static route, set the `poll` parameter in the `ip route` command.

Command examples

1.
```
(config)# ip route static poll-interval 10
```

   Specifies 10 seconds as the polling interval for dynamic monitoring.

2.
```
(config)# ip route static poll-multiplier 4 2
```

   Specifies 4 as the number of consecutive failures (`invalidcount`) and 2 as the number of consecutive responses (`restorecount`) for dynamic monitoring.

3.
```
(config)# ip route 192.168.3.0 255.255.255.0 10.2.1.100 poll
(config)# ip route 192.168.4.0 255.255.255.0 10.2.1.101 poll
```

   Applies dynamic monitoring to static routes 192.168.3.0/24 and 192.168.4.0/24.

8.2.6 Configuring a static route in a VRF [OS-L3SA]

Set a static route for a VRF.

Points to note

Specifies a VRF by using the `vrf` parameter of the `ip route` command.

Command examples

1.
```
(config)# ip route vrf 2 172.16.2.0 255.255.255.0 10.2.1.100
noresolve
```

   Generates the static route 172.16.2.0/24 for VRF2. As the next hop, specifies neighboring gateway 10.2.1.100.

8.2.7 Configuring static routes across VRFs [OS-L3SA]

Set a static route between VRFs to configure an extranet between specific hosts.

Points to note

Specifies the partner VRF with the `vrf` parameter of the `ip route` command (after the next hop address).

Command examples

1.
```
(config)# ip route vrf 2 172.16.3.1 255.255.255.255 10.3.1.100
vrf 3 noresolve
```
   Generates the static route 172.16.3.1/32 for VRF2. As the next hop, specifies neighboring gateway 10.3.1.100 for VRF3.

2.
```
(config)# ip route vrf 3 172.16.1.1 255.255.255.255 10.1.1.100
vrf 2 noresolve
```
   Generates the static route 172.16.1.1/32 for VRF3. As the next hop, specifies neighboring gateway 10.1.1.100 for VRF2.

8.3 Operation

8.3.1 List of operation commands

The following table describes the operation commands for static routing (IPv4).

Table 8-3: List of operation commands

Command name	Description
show ip route	Shows routing information stored in the routing table.
clear ip route	Clears the IPv4 forwarding entries stored in the Switch and re-registers the routing entries.
show ip static	Shows information related to a static route.
clear ip static-gateway	Performs polling for the gateways on the routes that were disabled by dynamic monitoring of static routes, and generates routes if the gateway responded.
show ip vrf	Shows the IPv4 information of a VRF.
show ip interface ipv4-unicast	Shows information about the IPv4 interfaces on the Switch recognized by the unicast routing program.
restart unicast#	Restarts the unicast routing program.
dump protocols unicast#	Outputs the trace information and control table information collected by the unicast routing program to a file.
erase protocol-dump unicast#	Deletes the file of trace information and control table information generated by the unicast routing program.

#

See *8. Routing Protocol Common to IPv4 and IPv6* in the manual *Operation Command Reference Vol. 2 For Version 11.10*.

8.3.2 Checking routing information

Check information about static routes.

Figure 8-6: Results of executing the show ip static route command

```
> show ip static route
Date 20XX/07/14 12:00:00 UTC
Status Codes: * valid, > active, r RIB failure
   Destination        Next Hop       Distance Weight Status      Flag
*> 0.0.0.0/0          10.1.1.50       2       0      IFdown      -
*> 192.168.1/24       10.1.1.100      100     0      Act         -
*  192.168.1/24       172.16.1.100    200     0      Act         NoResolve
*> 192.168.2/24       172.16.1.100    2       0      Act         NoResolve
                      172.16.2.100    2       0      Act         NoResolve
*> 192.168.3/24       10.2.1.100      2       0      Act Reach   Poll
   192.168.4/24       10.2.1.101      2       0      UnReach     Poll
```

Notes

1. When a route is registered in the routing table, an asterisk and angle bracket (`*>`) appear in the leftmost `Status Codes` field.
2. For an alternate route not registered in the routing table, the angle bracket (`>`) is omitted but the asterisk (`*`) appears in the `Status Codes` field if the route is valid.

3. For an invalid route, neither the asterisk nor angle bracket appears in the `Status Codes` field, and a problem of some sort is indicated in the `Status` field. `IFdown` means that the route is invalid due to a failure in an interface. `UnReach` means that the dynamic monitoring functionality has not confirmed that the route is reachable.

8.3.3 Checking gateway information

Check information about a static route gateway.

Figure 8-7: Results of executing the show ip static gateway command

```
> show ip static gateway
Date 20XX/07/14 12:00:00 UTC
Gateway         Status  Success     Failure     Transition
10.1.1.50       IFdown  -           -           -
10.1.1.100      -       -           -           -
10.2.1.100      Reach   -           0/4         13m 39s
10.2.1.101      UnReach 1/2         -           21s
172.16.1.100    -       -           -           -
172.16.2.100    -       -           -           -
```

Notes

1. The `Status` field indicates whether a dynamically monitored gateway is reachable. `Reach` means that the route is confirmed to be reachable, and `UnReach` means that the route has not been confirmed as reachable.
2. If the dynamic monitoring functionality has not yet confirmed that the route is reachable (`UnReach` appears in the `Status` field), check the gateway's monitoring status from the `Success` counter. In the above results, gateway 10.2.1.101 has a Success counter of 1/2. This means that two consecutive responses are required to confirm reachability, and at present one successful response has been received.

Chapter

9. RIP

This chapter describes the IPv4 Routing Information Protocol (RIP).

9.1 Description

9.1.1 Overview

The Routing Information Protocol (RIP) is used between the routers connected on a network. By using RIP, each router generates its own routing information by exchanging information about the networks it can reach and the number of hops (metrics) to those networks.

The Switch supports RIP versions 1 and 2. RIP version 0 messages are discarded. Messages generated from RIP version 3 or higher are handled as version 2 messages.

The following table describes the functionality supported by RIP.

Table 9-1: RIP functionality

Functionality	RIP
triggered update	Y
Split horizon	Y
Route poisoning	Y
Poison reverse	N
Hold-downs	N
RIP automatic route summarization	Y
Route tagging	Y
Read specified next hop	Y
Plain-text password authentication	Y
Cryptographic authentication (Keyed-MD5)	Y

Legend: Y: Supported, N: Not supported

(1) Message type

RIP uses two types of messages: request messages and response messages. A request message is used to request routing information from another router. A response message is used to reply to a request from another router. Response messages are also used to notify other routers of Switch's routing information, either periodically or when there is a change in the network topology.

(2) Message processing

At startup, the Switch sends all of its request messages to neighboring routers, and requests neighboring routers to send all of their routing information. After startup, the Switch sends responses for the following purposes:

- To send relevant routing information in response to a request from a neighboring router
- To periodically report routing information. At 30-second intervals the Switch sends all of its routing information as responses to neighboring routers.
- To report a detected route change. The Switch sends routing information regarding a changed route as a response to neighboring routers.

On receipt of a response from a neighboring router, the Switch updates its routing information if a route change has been detected. Responses are also used to check the transmission status to a neighboring router. If no response is received for 180 seconds or longer, the neighboring router is deemed unreachable and the routing table is updated with an alternate route, if available. Otherwise, the unreachable route is deleted.

(3) Preventing routing loops

The Switch use split horizon logic to prevent loops in forwarding routes. Split horizon processing stops received information from being forwarded to the interface from which the information originated.

9.1.2 Route selection conditions

From information learned by each protocol about routes to the same destination, the Switch selects the best route according to the independent route selection procedures associated with each protocol. If there are a number of alternative routes in the generated information, their distances are compared and the routing information that has the highest priority is selected.

Under RIP, priority rules govern the selection of the best route to a given destination among the routes learned and advertised by different routers. The following table explains the priority rules for selecting the best route.

Table 9-2: Route selection priority

Priority	Description
High	Selects the route with the lowest metric.
↑	Selects the route whose aging time is within half a second of the timer value (when routes have the same metric).
	Selects the route with the smallest next-hop address.
↓	Selects the route whose next-hop address in the routing information matches the address of the gateway from which the information originated.#
Low	In all other cases, ignores the most recently learned route.

#: This condition applies when routing information that contains the same next-hop address is learned from different neighboring routers on the network.

When the routes to a given destination learned by each protocol (OSPF, BGP4, and static) result in multiple entries, their distances are compared, and the route that has the highest priority is set in the routing table.

(1) Generating secondary route

Using the `generate-secondary-route` configuration command, you can generate up to two route entries (primary route and secondary route) to the same destination learned by two different neighboring routers. The following table describes the conditions for generating a secondary route.

Table 9-3: Secondary route generation conditions

Conditions		Secondary route generation
Specification of the generate-secondary-route configuration command	Distance	
N	--	Not generated
Y	Different values for primary route and secondary route	Not generated
Y	Same values for primary route and secondary route	Generated

Legend: Y: Command entered, N: Command not entered, --: Not applicable

When secondary route generation is enabled, route priority to a given destination is determined as follows:

Table 9-4: Route selection priority when registration of a secondary route is specified

Priority	Description
High	Selects the route with the next lowest metric.
↑	Selects the route whose aging time is within half a second of the timer value (when routes have the same metric).
	Selects the route with the next smallest next-hop address.[#1]
	Selects the route whose next-hop address in the routing information matches the address of the gateway from which the information originated.[#2]
↓	Selects the route that has been the primary route until now.
Low	In all other cases, ignores the most recently learned route.

Note:

If routes have the same next-hop address, only a primary route is generated.

#1

This condition does not apply to a newly learned route if a secondary route has already been registered.

#2

This condition applies if routing information containing the same next-hop address is learned by different neighboring routers on the network.

9.1.3 Route advertisements

(1) Advertised routes

(a) Learning protocol

If filtering of advertised routes has not been specified, learned RIP routes and directly connected routes within the RIP network are advertised. If filtering is specified, the advertising behavior is governed by the filter conditions. The following table describes the learning protocol for route advertising via RIP.

Table 9-5: Learning protocol and advertising behavior

Learning protocol		Advertising behavior without route filtering	Order in which the advertising metrics are applied[#5]
Directly connected route[#1]	Route within the RIP network	Advertised	1. Setting for advertised route filtering 2. Default (metric: 1)
	Route outside the RIP network	Not advertised	
Summarized route		Not advertised	
Static route		Not advertised	1. Setting for advertised route filtering 2. Setting by `default-metric` 3. Default (metric: 1)
RIP[#2]		Advertised	1. Setting for advertised route filtering 2. Metric in the routing table

Learning protocol	Advertising behavior without route filtering	Order in which the advertising metrics are applied[#5]
OSPF	Not advertised	1. Setting for advertised route filtering 2. Metric in the routing table if `inherit-metric` is set[#3] 3. Setting by `default-metric`[#4]
BGP	Not advertised	
Route imported from another VRF or a global network	Not advertised	

#1

The secondary address is also advertised.

#2

Split horizon is applied.

#3

The route is not advertised if the metric in the routing table is 16 or higher.

#4

The route is not advertised if advertised route filtering is unspecified or if no metric is specified by `inherit-metric` or `default-metric`.

#5

If set, the `metric-offset out` setting is added to the selected metric. If this results in a metric of 16 or higher, the route is not advertised.

(b) Address types

The following table describes the types of addresses that can be advertised via RIP.

Table 9-6: Types of advertised addresses

Address types	Definition	Example	Advertised	
			RIP-1	RIP-2
Default routing information	Routing information about all destination networks	0.0.0.0/0	Y	Y
Natural mask routing information	Information about the network mask for the class of IP address (Class A: 8 bits) (Class B: 16 bits) (Class C: 24 bits)	172.16.0.0/16 • Class B • Network mask: 16 bits (255.255.0.0)	Y	Y
Subnet routing information	Routing information about a specific subnet destination	172.16.10.0/24 • Class B • Network mask: 24 bits (255.255.255.0)	S[#1, #2]	Y[#2]
Supernet routing information	Routing information that encompasses multiple networks	172.0.0.0/8 • Class B • Network mask: 8 bits (255.0.0.0)	N	Y

Address types	Definition	Example	Advertised	
			RIP-1	RIP-2
Host routing information	Routing information about a specific destination host	172.16.10.1/32 • Network mask: 32 bits (255.255.255.255)	Y	Y

Legend: Y: Can be advertised, N: Cannot be advertised, S: Some information can be advertised

#1: In RIP-1 there are limitations on the subnet routes that can be advertised. For details, see *9.1.5 RIP-1 (1) Route advertisements in RIP-1*.

#2: When the `auto-summary` configuration command is set, subnet routing information is automatically summarized and advertised as natural mask routing information. For details, see *(4) RIP automatic route summarization*.

(2) Destination of route advertisements

Route advertisements are sent to all neighboring routers on the network specified by the `network` configuration command. Alternatively, you can restrict the destination to a specific neighboring router by setting the `neighbor` configuration command. The following table describes the destination of route advertisements via RIP.

Table 9-7: Destination of route advertisements

Destination device	Destination address
RIP network#1, #2	Multicast address (RIP-2) or subnet broadcast address (RIP-1)
Specific neighboring router#3	Unicast address

#1: The distribution of advertisements to interfaces specified as `passive-interface` is suppressed.

#2: Also distributed to secondary addresses.

#3: The neighboring router must be included in the RIP network.

(3) Timing of route advertisements

The following table describes the functionality related to the timing of route advertisements distributed via RIP.

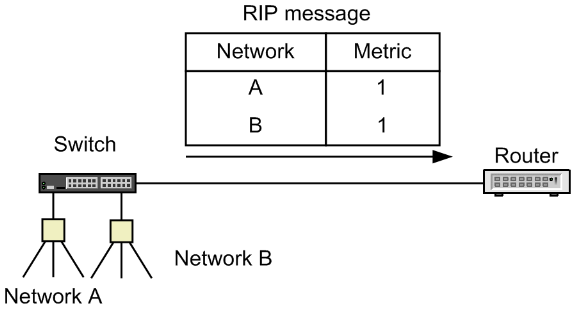
Table 9-8: Timing of route advertisements

Functionality	Description
Periodic route advertisement	Neighboring routers are reported periodically about routing information held by the switch.
triggered update	Any change in the routing information held by the switch is reported immediately without waiting for a periodic advertisement.
Response to a request from a neighboring router	The neighboring router that sent the request packet is notified.
Route poisoning	Neighboring routers are notified for a set duration of any deleted routing information.

(a) Periodic route advertisement

RIP periodically advertises routing information held by the local router to neighboring routers. The following figure shows an example of a periodic route advertisement.

Figure 9-1: Periodic route advertisement


RIP message

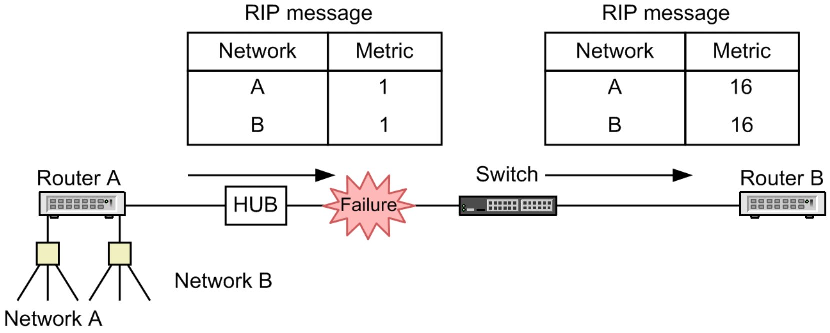
Network	Metric
A	1
B	1

The Switch advertises routing information about networks A and B to the router at 30-second intervals (advertisement timer).

(b) triggered update

Changes in the routing information held by the switch are distributed immediately without waiting for the periodic distribution cycle. The following figure shows a route advertisement distributed as a triggered update.

Figure 9-2: Route advertisement sent as a triggered update


RIP message

Network	Metric
A	1
B	1

RIP message

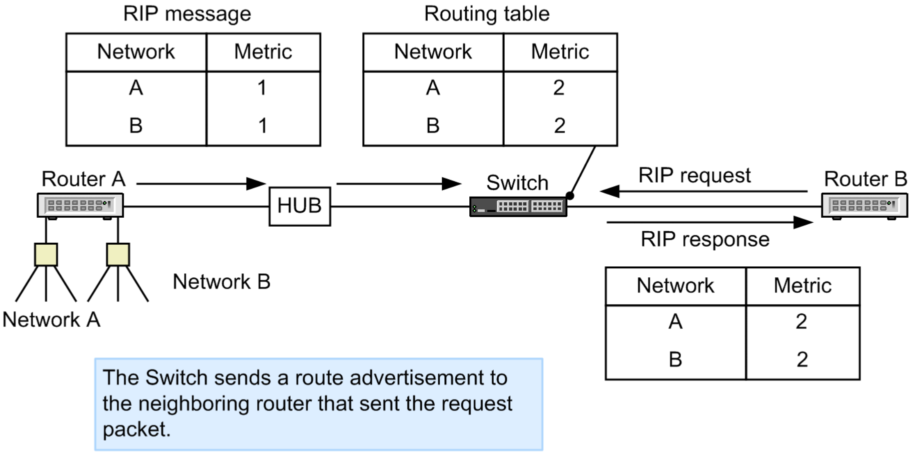
Network	Metric
A	16
B	16

If the Switch detects a failure between itself and the hub, the Switch deletes routing information about networks A and B from its routing table.
At the same time, the Switch sends a route advertisement to router B, giving networks A and B a metric of 16 (unreachable).

(c) Response to request packets

On receipt of a request packet, the Switch sends the requested information to the neighboring router that sent the packet. The following figure shows how routing information is advertised when a request packet is received.

Figure 9-3: Route advertisement sent on receipt of a request packet



RIP message

Network	Metric
A	1
B	1

Routing table

Network	Metric
A	2
B	2

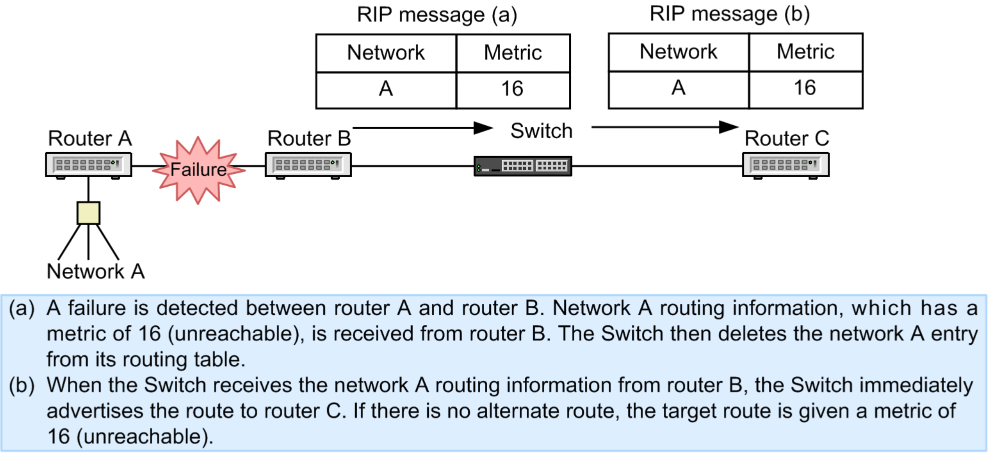
Network	Metric
A	2
B	2

The Switch sends a route advertisement to the neighboring router that sent the request packet.

(d) Route poisoning

When a route changes from reachable to unreachable status (on receipt of a metric-16 route advertisement or on deletion of a route learned from an interface that has since failed), the router advertises the metric-16 (unreachable) route to its neighboring routers for a set period (60 seconds: garbage collection timer). The following figure shows route poisoning.

Figure 9-4: Route poisoning



RIP message (a)

Network	Metric
A	16

RIP message (b)

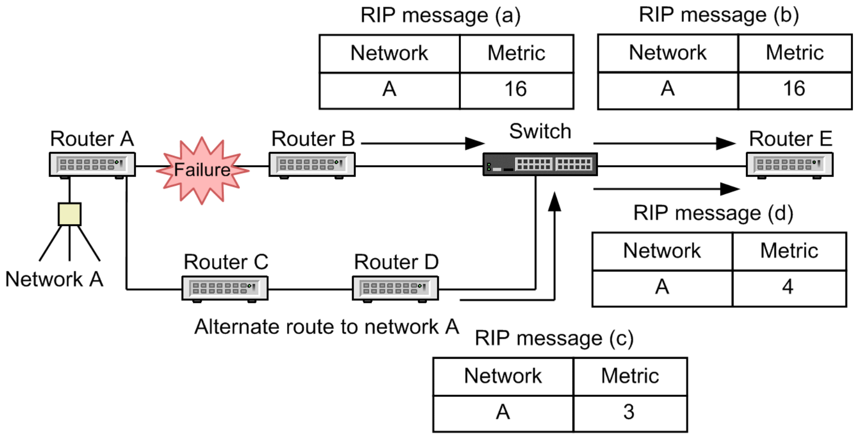
Network	Metric
A	16

(a) A failure is detected between router A and router B. Network A routing information, which has a metric of 16 (unreachable), is received from router B. The Switch then deletes the network A entry from its routing table.

(b) When the Switch receives the network A routing information from router B, the Switch immediately advertises the route to router C. If there is no alternate route, the target route is given a metric of 16 (unreachable).

If a new route to the affected destination is learned while route poisoning is in effect, the Switch advertises the new routing information. This is illustrated in the following figure.

Figure 9-5: Learning a new route during route poisoning



RIP message (a)

Network	Metric
A	16

RIP message (b)

Network	Metric
A	16

RIP message (c)

Network	Metric
A	3

RIP message (d)

Network	Metric
A	4

(a) A failure is detected between router A and router B. Network A routing information, which has a metric of 16 (unreachable), is received from router B. The Switch then deletes the network A entry from its routing table.
(b) At the same time, the Switch advertises the network A routing information, which has a metric of 16 (unreachable), to router E.
(c) The Switch receives the network A routing information in a periodic advertisement from router D. This information is added to its routing table (update timing is dependent on the periodic advertisement timer of router D).
(d) The Switch advertises the network A routing information to router E.

(4) RIP automatic route summarization

By setting the `auto-summary` configuration command, you can advertise multiple subnet routing entries to neighboring routers, automatically summarizing the entries as a single natural mask route. This configuration command is supported in both RIP-1 and RIP-2.

The following table describes the types of addresses for which automatic route summarization is supported.

Table 9-9: Address types for which automatic route summarization is supported

Address types	Summarization	
	RIP-1	RIP-2
Default routing information	N	N
Natural mask routing information	N	N
Subnet routing information	Y[#1]	Y[#2]
Supernet routing information	N	N
Host routing information	N	N

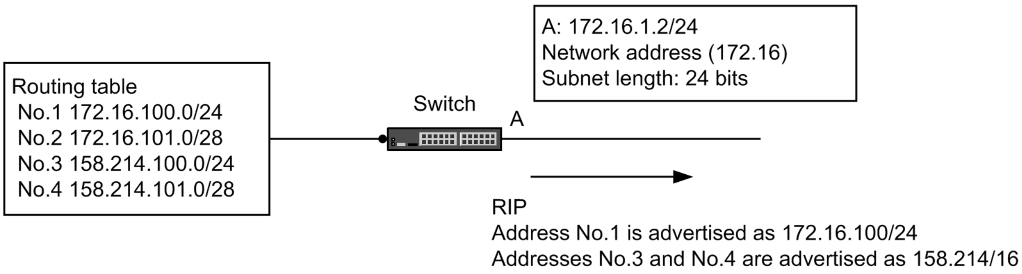
Legend: Y: Can be summarized, N: Cannot be summarized

#1: In RIP-1, when the advertised routing information and the interface of the advertisement destination are in the same natural network and have the same mask length, the information is advertised to neighboring routers as subnet routing information without automatic summarization. For details, see *Figure 9-6: Automatic route summarization when using RIP-1*.

#2: In RIP-2, when the advertised routing information and the interface of the advertisement destination are in the same natural network, the information is advertised to neighboring routers as subnet routing information without automatic summarization. For details, see *Figure 9-7: Automatic route summarization when using RIP-2*.

The following figure shows automatic summarization of subnet routes when RIP-1 is used.

Figure 9-6: Automatic route summarization when using RIP-1



● Handling of the routing information in the routing table

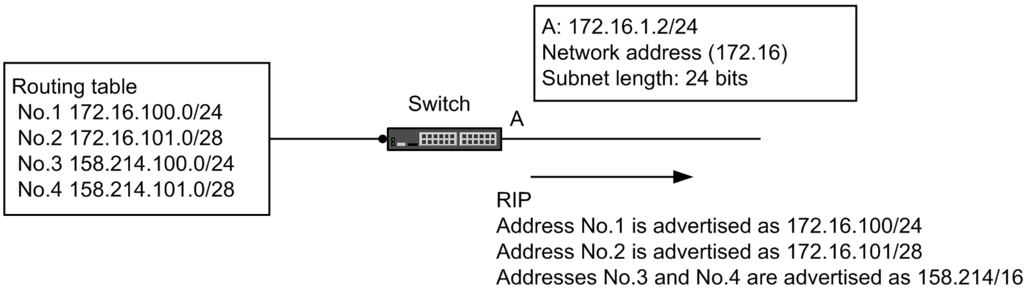
No.1: Advertised without summarization because it has the same network address and subnet length as interface A.

No.2: Not advertised because it has the same network address but not the same subnet length as interface A.

No.3/No.4:
Summarized and advertised as a natural mask route because they are in a different network from interface A.

The following figure shows automatic summarization of subnet routes when RIP-2 is used.

Figure 9-7: Automatic route summarization when using RIP-2



● Handling of the routing information in the routing table

No.1: Advertised without summarization because it has the same network address as interface A.

No.2: Advertised without summarization because it has the same network address as interface A.

No.3/No.4:
Summarized and advertised as a natural mask route because they are in a different network from interface A.

(a) Metric used with automatic route summarization

The summarized information is advertised with the smallest metric in the original subnet routing information.

(b) Route tagging with automatic route summarization (only with RIP-2)

The route tag in a summarized advertisement is 0.

(c) Next-hop advertisements with automatic summarization (only with RIP-2)

The next-hop field in a summarized advertisement is 0.

9.1.4 Learning routing information

(1) Origin of learned routing information

In RIP, routing information can be learned from all neighboring routers on the network specified by the `network` configuration command. This includes routers on a network to which the secondary address of the interface belongs.

(2) Timing of route learning and updating

The following table describes the functionality related to the timing of route updates learned via RIP.

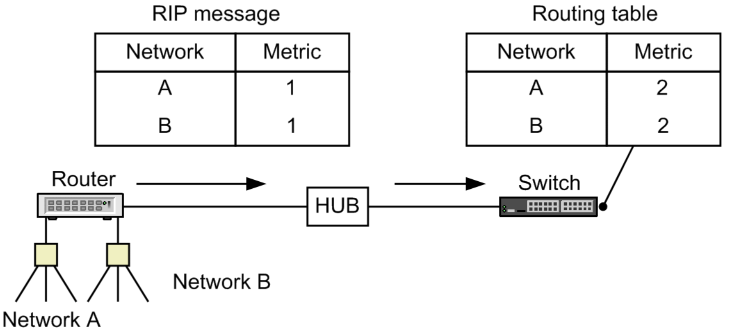
Table 9-10: Timing of route learning and updating

Functionality	Description
Response packets received from neighboring routers	Adds, changes, or deletes routing information as reported by neighboring routers.
Aging timeout	Deletes routing information if there is no periodic notification from a neighboring router for a set time.
Recognizing interface failures	Deletes routing information learned from a RIP-enabled interface on which a failure has been discovered.

(a) Receiving response packets

Under RIP, the routing information in the response packets received from neighboring routers is written to the Switch's routing table. The following figure shows the generation of routing information from a received response packet.

Figure 9-8: Generation of routing information from a received response packet



RIP message

Network	Metric
A	1
B	1

Routing table

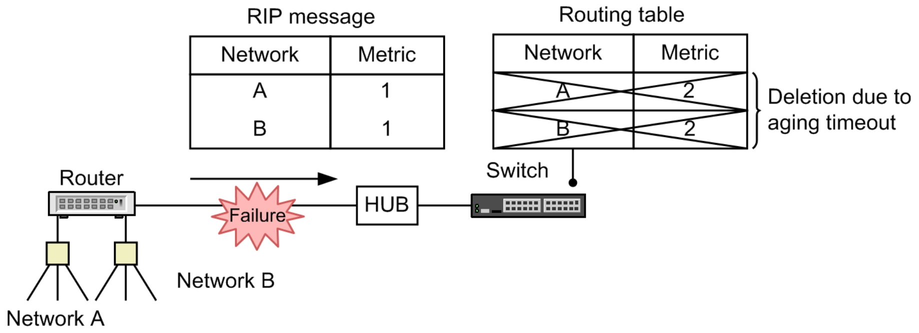
Network	Metric
A	2
B	2

The Switch adds the routing information for networks A and B, which were learned from the neighboring router, to its routing table.

(b) Aging timeout

Routing information generated from a received response packet is monitored by an aging timer. The aging timer is reset (cleared) by a periodic advertisement from the neighboring router. If the router fails to generate an advertisement for the route being monitored for 180 seconds (aging timeout value) due to a hardware failure or a line error between the router and switch, the affected entries are deleted from the switch's routing table. The following figure shows deletion of routing information by aging timeout.

Figure 9-9: Deletion of routing information by aging timeout



RIP message

Network	Metric
A	1
B	1

Routing table

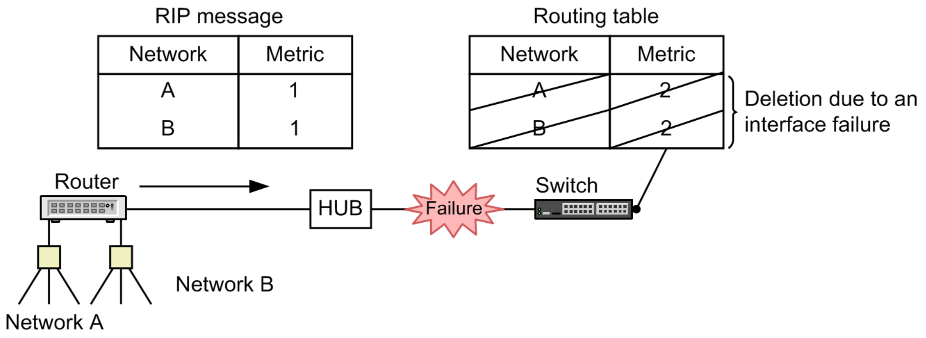
Network	Metric
~~A~~	~~2~~
~~B~~	~~2~~

If a failure occurs between the router and hub, routing information for networks A and B is not advertised to the Switch.
The Switch deletes the affected routing information from its routing table if no advertisement is received for 180 seconds (aging timeout).

(c) Recognizing interface failures

On recognizing that the interface that connects the switch to a neighboring router has failed, the Switch immediately deletes all routing information learned from that interface. The following figure shows the deletion of routing information due to an interface failure.

Figure 9-10: Deletion of routing information due to an interface failure



RIP message

Network	Metric
A	1
B	1

Routing table

Network	Metric
~~A~~	~~2~~
~~B~~	~~2~~

If the Switch detects a failure in the interface that connects it to neighboring router, the Switch deletes all routing information learned from that interface from its routing table.

9.1.5 RIP-1

(1) Route advertisements in RIP-1

In RIP-1, depending on the value of the subnet mask of the port that sent the RIP message, restrictions apply to the routing entries that can be advertised. There is no problem when all entries have the same network address and use the same subnet mask. Difficulties arise, however, when two or more subnet masks, such as Variable Length Subnet Mask (VLSM) are in use. In a VLSM network, RIP-2 (compliant with RFC 2453) must be used as the routing protocol. If RIP-1 is also used, note that advertising of routing information in RIP-1 is subject to the following conditions:

Table 9-11: Conditions governing route advertisements in RIP1

Routing information to be advertised	Conditions
Default routing information	Advertised unconditionally. However, advertised route filtering must be enabled to advertise default routing information learned via a non-RIP protocol.

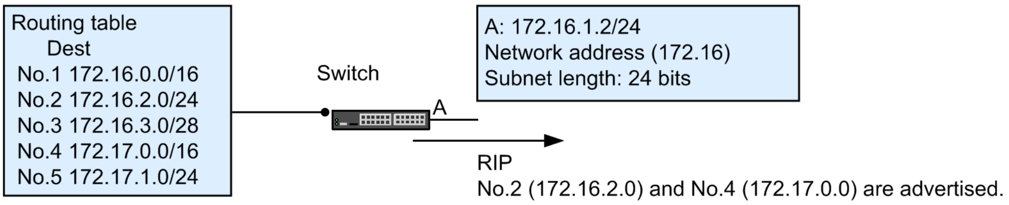
Routing information to be advertised	Conditions
Natural mask routing information	Advertised when the natural mask routing information held by the Switch has a different network address (address class) from the interface.
Subnet routing information#	Advertised when the subnet routing information held by the Switch has the same network address (address class) and the same subnet length as the interface.
Host routing information	Advertised unconditionally.

#: When the `auto-summary` configuration command is set, subnet routing information is automatically summarized and advertised as natural mask routing information.

(a) Natural mask route and subnet mask route advertisements

The following figure shows the natural mask and subnet mask routing information that is advertised in RIP.

Figure 9-11: Natural mask and subnet mask routing information advertised in RIP



- Handling of the routing information in the routing table

No.1: Not advertised because the natural mask route information has the same network address as interface A.

No.2: Advertised because the subnet route information has the same network address and subnet length as interface A.

No.3: Not advertised because the subnet route information has the same network address as interface A but a different subnet length.

No.4: Advertised because the natural mask route information has a different network address from interface A.

No.5: Not advertised because the subnet route information has a different network address from interface A.

The advertising conditions in the above figure are described in the following table.

Table 9-12: Conditions governing advertising of natural mask routes and subnet mask routes

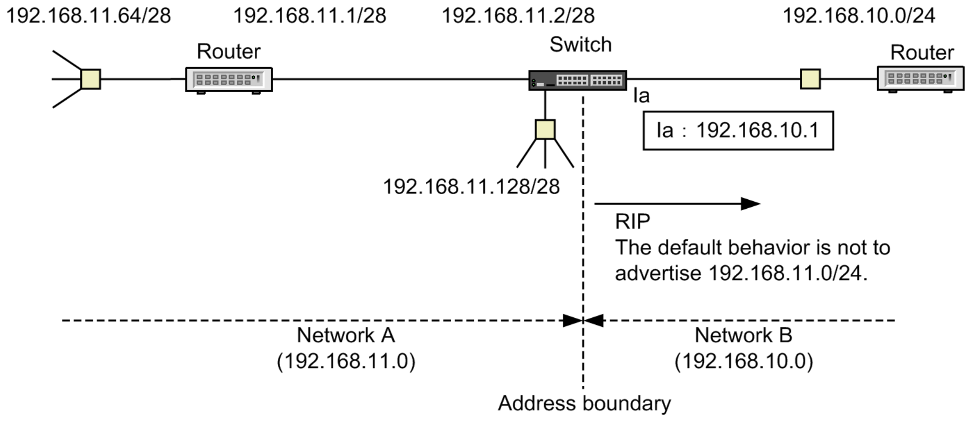
Type of routing information	Routing information in the routing table	Conditions		Advertised
		Comparison with the network address of interface A	Comparison with the subnet length of interface A	
Natural mask route	172.16.0.0/16 (No.1)	Match	--	N
	172.17.0.0/16 (No.4)	Mismatch	--	Y
Subnet route	172.17.1.0/24 (No.5)	Mismatch	Match	N
	172.16.2.0/24 (No.2)	Match	Match	Y
	172.16.3.0/28 (No.3)	Match	Mismatch	N

Legend: Y: Advertised, N: Not advertised, --: Not applicable

(b) Notes on subnet route advertisements

In the Switch, if you have not set automatic route summarization using the `auto-summary` configuration command, only subnet routing information will be generated for the IP address of each interface, and the routes will not be automatically summarized as natural mask routing information. Note that when the route extends beyond the address boundary, subnet routing information is not advertised in RIP-1. The following figure shows the configuration example.

Figure 9-12: Configuration example in which directly connected routes are not advertised



Key points in the above configuration

- The routing protocol is RIP-1.
- Automatic route summarization has not been set by the `auto-summary` configuration command.
- Address boundaries are aligned on the Switch.
- The interface's subnet mask is not a natural mask.

Solution 1

- Set the configuration command to enable automatic route summarization.

Solution 2

- Set the configuration command to enable automatic route summarization (summarize subnet routing information and host routing information into a natural mask route).
- Set the configuration command to enable advertised route filtering (redistribute summarized routes via RIP).

Solution 3

- Enable generation of directly connected routes that have a natural mask for the subnetworked interface (`ip auto-class-route` configuration command).
- Because the above routes are handled as directly connected routes, they are advertised by default (the redistribution filter is disabled).

(2) Differences in implementation from the RFCs

With some exceptions, RIP-1 as implemented in the Switch complies with RFC 1058. Differences are due to the functional limitations of the software, and are described in the table below.

Table 9-13: Differences with the RFC

RFC			Switch
RFC 1058	Subnet advertisements	A border gateway connected to a subnetted network advertises only a single entry for the entire network to neighboring gateways.	RIP automatic route summarization must be used to generate a network route from subnet routes.
		The metric for the network as a whole is normally the smallest metric among the subnets.	RIP automatic route summarization must be used to generate a network route from subnet routes.
		Border gateways must not advertise host routes within a directly connected network to other networks.	Host routes within directly connected networks are added to the routing table and are advertised.
	Receipt of response messages	Host routes included in an existing network route or subnet route should not be added to the routing tables.	Host routes received in responses are added to the routing tables.

9.1.6 RIP-2

(1) RIP-2 functionality

RIP-2 sets the subnet masks of advertised routes in the routing information, allowing variable-length subnets to be handled without any of the advertising limitations imposed by RIP-1. The functionality unique to RIP-2 is described below.

(a) Route tagging

In the Switch, route tags are written to routing tables if set in the routing information reported in a response message. The route tag of the corresponding entry in the routing table is set as the route tag in the routing information in the response message sent by the Switch. The valid range is 1 to 65535 (in decimal).

(b) Subnet mask

In the Switch, subnet mask information is written to the routing tables if set in the routing information reported in a response message. If no subnet mask information is set, the routing information in a response message is handled in the same manner as routing information received in RIP-1.

The subnet mask of the corresponding entry in the routing table is set as the subnet mask in the routing information in the response message sent by the Switch.

(c) Next hop

In the Switch, next-hop information is written to the routing tables if set in the routing information reported in a response message. If no next-hop information is set, the originating gateway is regarded as the next hop.

When the next hop in the reported routing information is on the same network as the destination gateway, the next hop of the corresponding entry in the routing table is set as the next hop in the routing information in the response message sent by the Switch. If the next hop is not on the same network, the source interface address is set.

(d) Multicasting

The Switch supports multicasting to reduce the unnecessary load on hosts that do not receive RIP-2 messages. The multicast address used for transmitting RIP-2 messages is 224.0.0.9.

(e) Authentication functionality

In RIP, authentication can be used during message exchange between routers to verify that the router that sent a message is in the same management domain. By using authentication between neighboring routers, you can protect routers in the same authentication and management domain from attacks related to routing control that are triggered by sending invalid routing information.

Plain-text password authentication and cryptographic authentication can be used as authentication methods. The Switch supports Keyed-MD5 as the authentication algorithm for cryptographic authentication.

Using a configuration command, you can set an authentication method and authentication key on a per-interface basis. If you do not specify any settings, authentication will not be performed.

■ Plain-text password authentication

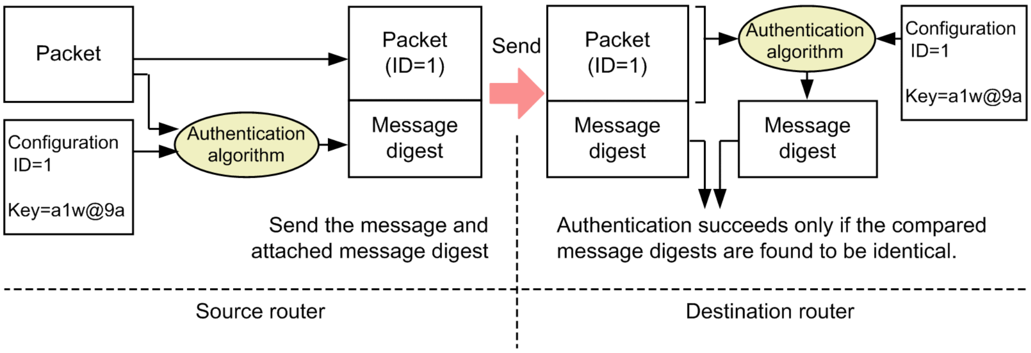
In plain-text password authentication, the authentication key set by a configuration command is embedded as the password in the sent message. If more than one authentication key has been set by a configuration command, the message is replicated for each authentication key and is sent multiple times.

When a message is received, authentication is considered successful if the password in the message matches one of the set authentication keys. If authentication fails, the message is discarded.

■ Cryptographic authentication

In cryptographic authentication, messages can be authenticated by comparing message digests. The following figure shows the data flow.

Figure 9-13: Data flow in cryptographic authentication



When a message is sent, it is accompanied by a message digest, which is generated from the authentication key and the message itself based on an authentication algorithm (Keyed-MD5). If more than one authentication key has been set by a configuration command, the message is replicated for each authentication key and is sent multiple times.

When a message is received, it is authenticated by using the authentication key that has the same key identifier as the key identifier contained in the message. A message digest is generated from this authentication key in the same manner as when a message is sent. If the generated message digest matches the received message digest, authentication is considered successful. If authentication fails, the message is discarded.

■ Changing the authentication key

In a RIP-2 network, normally each router uses one authentication key. When you change an authentication key, however, the router will temporarily have multiple authentication keys.

To change an authentication key:

1. Enable both the old and new authentication keys at each router that uses authentication. In the Switch, all keys that have been set by a configuration command will be enabled.
2. Delete or disable the old authentication key at each router in the network that uses authentication.

■ Notes on using cryptographic authentication

To prevent replay attacks, a sequence number is appended to messages that use cryptographic authentication. Each sequence number must be larger than the previously sent number. In the Switch, the elapsed time in seconds from 1970/1/1 0:00 is set as the sequence number.

Sequence numbers are incremented so that authentication will not fail at any neighboring devices even if the current system time has been turned back by the `set clock` operation command. However, when the switch is restarted, because the sequence numbers cannot be adjusted, the next message might be sent with a smaller sequence number than the message sent before the restart. In this case, authentication will fail at the next router in the path. The risk of authentication failure at a neighboring router is particularly high if the switch is restarted after a major backward adjustment of the system clock while cryptographic authentication is in use.

If authentication fails continuously, change the authentication keys at all routers in the network.

(2) Differences in implementation from the RFCs

With some exceptions, RIP-2 as implemented in the Switch complies with RFC 2453 and RFC 4822. Differences are due to the functional limitations of the software, and are described in the table below.

Table 9-14: Differences with the RFC

RFC		Switch
RFC 2453	If a RIP-2 router receives a RIP-1 request, it should respond with a RIP-1 response. If the router is configured to send only RIP-2 messages, it should not respond to a RIP-1 request.	The Switch sends only RIP-2 responses via a RIP-2 interface. Therefore, no response is sent to a RIP-1 request.
	Routers should implement a receive control switch that determines whether to accept RIP-1 only, RIP-2 only, both, or none. These options should be configurable on a per-interface basis.	The Switch can control reception of RIP messages on a per-interface basis, but cannot implement reception control that distinguishes between RIP-1 and RIP-2.
RFC 4822	The set of authentication configuration parameters including the authentication key and key identifier should have a key lifetime and other configuration parameters associated with it.	The Switch does not support key lifetime settings.
	The Keyed-MD5 authentication algorithm and the HMAC-SHA1 authentication algorithm must be implemented by all conforming implementations.	The Switch supports only the Keyed-MD5 authentication algorithm.

(3) Notes on designing multihomed networks

Note the following when using RIP-2 in an interface that has secondary addresses.

RIP-2 forwards packets using multicasting. As packets for which multicasting is specified are delivered to all routers belonging to the primary or secondary network, those routers that do not need to receive the RIP packets are burdened with an unnecessary load.

9.2 Configuration

9.2.1 List of configuration commands

The following table describes the configuration commands for the RIP.

Table 9-15: List of configuration commands

Command name	Description
address-family ipv4 (RIP)	Configure settings for each VRF. Switch to `config-router-af` mode.
auto-summary	Specifies that subnet routing information advertised by RIP is to be automatically summarized and advertised as natural mask routing information.
default-metric	Specifies the metric to be used when routing information learned by another protocol is advertised by RIP.
disable	Disables RIP.
distance	Specifies the distance for routing information learned by RIP.
generate-secondary-route	Registers a secondary route in the routing table.
inherit-metric	Specifies that the metric is to be inherited when routing information learned by another routing protocol is advertised by RIP.
ip rip authentication key	Specifies the authentication method and authentication key for RIP-2 packets.
ip rip v2-broadcast	Specifies that the broadcast address is to be used as the destination address for packets forwarded from a specified interface.
ip rip version	Specifies the RIP version to be used by a specified interface.
metric-offset	Specifies the metric increment when RIP packets are forwarded or received via a specific interface.
neighbor	Specifies the neighboring router to which RIP packets are forwarded.
network	Specifies the destination network for RIP transmission.
passive-interface	Disables a specified interface from sending routing information in RIP packets.
router rip	Configures RIP-related operation information.
timers basic	Specifies the values of the various RIP timers.
version	Specifies the RIP version.
distribute-list in (RIP)#	Filters which RIP-learned routes are added to the routing table.
distribute-list out (RIP)#	Filters which RIP routes are advertised.
ip prefix-list#	Configures an IPv4 prefix list.
redistribute (RIP)#	Specifies the protocol for routes advertised by RIP.
route-map#	Configures a route map.

#

See *14. Route Filters (IPv4 and IPv6)* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

9.2.2 Applying RIP

Set the network and RIP version for forwarding and receiving RIP packets.

Points to note

The example below shows how to specify the network on which RIP is to be enabled by using the `network` command, and the RIP version by using the `version` command.

Command examples

1.
```
(config)# router rip
(config-router)# network 192.168.1.0 0.0.0.255
```
Enables RIP packet transmission on network 192.168.1.0/24.

2.
```
(config-router)# network 192.168.2.0 0.0.0.255
```
Enables RIP packet transmission on network 192.168.2.0/24.

3.
```
(config-router)# version 2
```
Sets the RIP version to RIP-2.

9.2.3 Configuring metrics

(1) Setting the metric for advertising non-RIP routing information

Set the metric to be used when routing information learned by another protocol is advertised by RIP.

Points to note

To advertise OSPF routes or BGP4 routes via RIP, you must set the metric by using the `default-metric` command.

Command examples

1.
```
(config)# router rip
(config-router)# network 192.168.1.0 0.0.0.255
(config-router)# network 192.168.2.0 0.0.0.255
(config-router)# default-metric 3
```
Sets 3 as the metric to be used when routing information learned by another protocol is advertised by RIP.

2.
```
(config-router)# redistribute static
```
Specifies that static routes are to be advertised by RIP.

3.
```
(config-router)# redistribute ospf
```
Specifies that OSPF routes are to be advertised by RIP.

(2) Setting the metric increment for packet transmission

Set the value by which to increment the metric for sending and receiving RIP packets.

Points to note

The example below shows how to use the `metric-offset` command to set the value to be added to the metric of an incoming or outgoing route via a specific interface.

Command examples

1.
```
(config)# router rip
(config-router)# network 192.168.1.0 0.0.0.255
(config-router)# network 192.168.2.0 0.0.0.255
(config-router)# metric-offset 2 vlan 10 out
```
Adds 2 to the metric of RIP packets sent from interface `vlan 10`.

2.
```
(config-router)# metric-offset 2 vlan 20 in
```
Adds 2 to the metric of RIP packets received from interface `vlan 20`.

9.2.4 Adjusting the timers

Adjust the value of the RIP periodic advertisement timer, the value of the aging timer, and the wait time before entries are deleted from the routing tables.

To reduce the convergence time when a route is changed, set a value smaller than the default for the periodic advertisement timer and aging timer. To reduce RIP advertisement traffic, set a value greater than the default for the periodic advertisement timer.

If you change a RIP timer value, apply the same timer value to all routers on the RIP network.

Points to note

The example below shows how to use the `timers basic` command to change the value of the RIP timers.

Command examples

1.
```
(config)# router rip
(config-router)# network 192.168.1.0 0.0.0.255
(config-router)# network 192.168.2.0 0.0.0.255
(config-router)# timers basic 40 200 100
```
Sets 40 seconds for the RIP periodic advertisement timer, 200 seconds for the aging timer, and 100 seconds before entries are deleted from the routing tables.

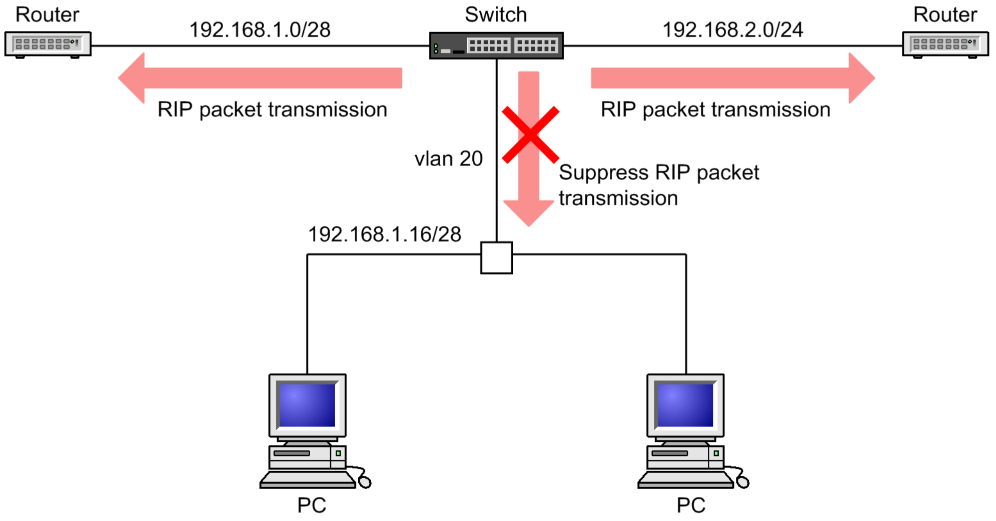
9.2.5 Suppressing RIP packet transmission

Set suppression of RIP packet transmission on a per-interface basis.

Points to note

The example below shows how to use the `passive-interface` command to set suppression of RIP packet transmission on a per-interface basis.

Figure 9-14: Suppressing RIP packet transmission



These PCs do not need to learn the route by RIP because a default route via the Switch has been set.

Command examples

1.
```
(config)# router rip
(config-router)# network 192.168.1.0 0.0.0.255
(config-router)# network 192.168.2.0 0.0.0.255
(config-router)# passive-interface vlan 20
```

   Suppresses transmission of RIP packets via interface `vlan 20`.

9.2.6 Restricting RIP packet destinations

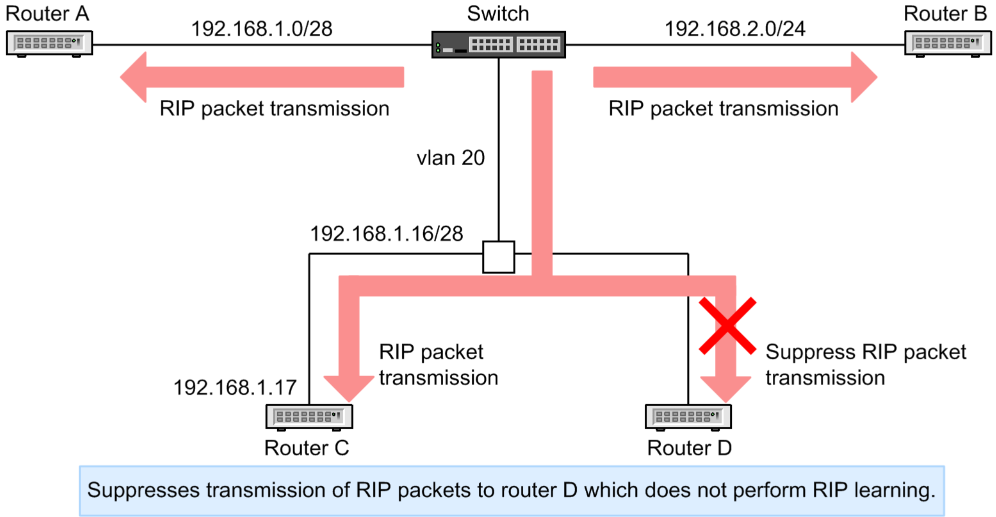
Set up the Switch to send route advertisements to specific neighboring routers by using unicast routing.

Points to note

The example below shows how to use the `neighbor` command to send route advertisements to a specific neighboring router.

Before using the `neighbor` command, use the `passive-interface` command to suppress the broadcast (or multicast) of advertisements via the interface.

Figure 9-15: Restricting RIP packet destinations



Suppresses transmission of RIP packets to router D which does not perform RIP learning.

Command examples

1.
```
(config)# router rip
(config-router)# network 192.168.1.0 0.0.0.255
(config-router)# network 192.168.2.0 0.0.0.255
(config-router)# passive-interface vlan 20
```
Suppresses transmission of RIP packets via interface `vlan 20`.

2.
```
(config-router)# neighbor 192.168.1.17
```
Specifies that route advertisements are to be sent to neighboring router 192.168.1.17 by unicast routing.

9.2.7 Applying authentication

Apply the authentication functionality to RIP-2 packets transmitted through a specific interface.

Points to note

The example below shows how to, using the `ip rip authentication key` command, set the key identifier, authentication method, and authentication key for RIP-2 packets. Use the same authentication key for all routers in the same network.

Command examples

1.
```
(config)# interface vlan 1
(config-if)# ip rip authentication key 1 md5 a1w@9a
(config-if)# ip rip version 2
```
Applies RIP-2 authentication to the interface `vlan 1`.

Sets 1 as the key identifier, cryptographic authentication (Keyed-MD5) as the authentication method, and `a1w@9a` as the authentication key.

9.2.8 Applying RIP for a VRF [OS-L3SA]

Enable RIP for a VRF.

Points to note

The example below shows how to use the `address-family ipv4 vrf` command to switch to `config-router-af` mode, and then specify the necessary information.

Command examples

1.
```
(config)# router rip
(config-router)# address-family ipv4 vrf 2
```
Switches to `config-router-af` mode, and then specifies information about the RIP that runs in VRF 2.

2.
```
(config-router-af)# network 172.16.2.0 0.0.0.255
```
Enables RIP packet transmission on network 172.16.2.0/24.

3.
```
(config-router-af)# version 2
```
Sets the RIP version to RIP-2.

4.
```
(config-router-af)# exit
```
Ends `config-router-af` mode.

9.3 Operation

9.3.1 List of operation commands

The following table describes the operation commands for the RIP.

Table 9-16: List of operation commands

Command name	Description
show ip route	Shows routing information stored in the routing table.
clear ip route	Clears the IPv4 forwarding entries stored in the Switch and re-registers the routing entries.
show ip rip	Shows information about the RIP protocol.
clear counters rip ipv4-unicast	Clears information about the RIP protocol.
show ip vrf	Shows the IPv4 information of a VRF.
show ip interface ipv4-unicast	Shows information about the IPv4 interfaces on the Switch recognized by the unicast routing program.
debug ip	Shows the packets being routed by IPv4 routing protocols in real time.
show processes cpu unicast#	Shows the CPU usage of a unicast routing program.
debug protocols unicast#	Starts the operation message display for event log information output by a unicast routing program.
no debug protocols unicast#	Stops the operation message display for event log information output by a unicast routing program.
dump protocols unicast#	Outputs the trace information and control table information collected by the unicast routing program to a file.
erase protocol-dump unicast#	Deletes the file of trace information and control table information generated by the unicast routing program.

#

See *8. Routing Protocol Common to IPv4 and IPv6* in the manual *Operation Command Reference Vol. 2 For Version 11.10*.

9.3.2 Checking the RIP operating status

Show information about the RIP protocol.

Figure 9-16: Results of executing the show ip rip command

```
> show ip rip
Date 20XX/07/14 12:00:00 UTC
RIP Flags: <ON>
Default Metric: 1, Distance: 120
Timers (seconds)
  Update             : 30
  Aging              : 180
  Garbage-Collection : 60
```

9.3.3 Checking destination information

Show the information at the RIP transmission destination.

Figure 9-17: Results of executing the show ip rip target command

```
> show ip rip target
```

```
Date 20XX/07/14 12:00:00 UTC
Source Address  Destination     Flags
192.168.1.1     192.168.1.100   <V1 Unicast>
192.168.1.1     192.168.1.200   <V1 Unicast>
192.168.1.1     192.168.1.255   <V1 Passive>
192.168.2.1     192.168.2.255   <V2 Multicast>
```

9.3.4 Checking learned routing information

(1) Per-network check

Show all routing information for a specific network that has been learned by RIP and stored in the routing tables.

Figure 9-18: Results of executing the show ip rip route command

```
> show ip rip route 172.0.0.0/8
Date 20XX/07/14 12:00:00 UTC
Status Codes: * valid, > active, r RIB failure
   Destination        Next Hop        Interface       Metric Tag   Timer
*> 172.16/16          192.168.1.100   VLAN0010        3      0      4s
*> 172.17/16          192.168.2.2     VLAN0020        4      0     10s
*> 172.18/16          192.168.2.2     VLAN0020        3      0     10s
*> 172.19/16          192.168.1.200   VLAN0010        5      0     17s
```

(2) Per-gateway check

Show routing information learned from a specific gateway and stored in the routing tables.

Figure 9-19: Results of executing the show ip rip received-routes command

```
> show ip rip received-routes 192.168.2.2
Date 20XX/07/14 12:00:00 UTC
Status Codes: * valid, > active, r RIB failure

Neighbor Address: 192.168.2.2
   Destination        Next Hop        Interface       Metric Tag   Timer
*> 172.17/16          192.168.2.2     VLAN0020        4      0     15s
*> 172.18/16          192.168.2.2     VLAN0020        3      0     15s
*> 192.168.3/24       192.168.2.2     VLAN0020        2      0     15s
*> 192.168.5/24       192.168.2.2     VLAN0020        4      0     15s
```

9.3.5 Checking advertised routing information

(1) Checking per destination

Show the routing information sent to a specific target.

Figure 9-20: Results of executing the show ip rip advertised-routes command (1)

```
> show ip rip advertised-routes 192.168.2.255
Date 20XX/07/14 12:00:00 UTC
Target Address: 192.168.2.255
Destination        Next Hop         Interface       Metric Tag   Age
172.16/16          192.168.1.100    VLAN0010        4      0     19s
172.19/16          192.168.1.200    VLAN0010        6      0      2s
192.168.4/24       192.168.1.200    VLAN0010        3      0      2s
192.168.6/24       192.168.1.100    VLAN0010        5      0     19s
```

(2) Per-network check

Show all routing information for a specific network sent by RIP, grouped by target.

Figure 9-21: Results of executing the show ip rip advertised-routes command (2)

```
> show ip rip advertised-routes 172.0.0.0/8
Date 20XX/07/14 12:00:00 UTC
Target Address: 192.168.1.100
```

```
Destination        Next Hop         Interface       Metric Tag   Age
172.17/16          192.168.2.2      VLAN0020        5      0      1s
172.18/16          192.168.2.2      VLAN0020        4      0      1s
172.19/16          192.168.1.200    VLAN0010        6      0      7s
Target Address: 192.168.1.200
Destination        Next Hop         Interface       Metric Tag   Age
172.16/16          192.168.1.100    VLAN0010        4      0     24s
172.17/16          192.168.2.2      VLAN0020        5      0      1s
172.18/16          192.168.2.2      VLAN0020        4      0      1s
Target Address: 192.168.2.255
Destination        Next Hop         Interface       Metric Tag   Age
172.16/16          192.168.1.100    VLAN0010        4      0     24s
172.19/16          192.168.1.200    VLAN0010        6      0      7s
```

Chapter
10. OSPF [OS-L3SA]

This chapter describes the OSPF routing protocol for IPv4.

10.1 Description of basic OSPF functionality

OSPF (Open Shortest Path First) is a routing protocol that uses Dijkstra's algorithm to calculate the shortest path to known destinations, based on a topology map constructed from information about the state of links between routers.

10.1.1 Features of OSPF

OSPF is typically used to route packets within a single autonomous system (AS). OSPF maintains a network topology constructed from information about link states within the AS in a database on each router, and uses this database to calculate shortest routes. OSPF has the following advantages over RIP:

- Less routing traffic

  OSPF sends updates to other routers only when there is a change to the link state between routers. This generates far less traffic than routing protocols such as RIP, which exchange the entire routing table at fixed time intervals. In OSPF, each router distributes information about its own link state every 30 minutes.

- Elimination of routing loops

  Each router using OSPF maintains an identical database, which it uses to select suitable routes. Therefore, unlike RIP, OSPF does not produce routing loops.

- Cost-based route selection

  When there is more than one route to a given destination, OSPF selects the route with the lowest overall cost. Unlike RIP, OSPF allows route costs to be set in a flexible manner. This guarantees that the most desirable route is selected irrespective of the number of hops.

- Operation of large-scale networks

  OSPF can process routes whose total cost is 16777214 or less. Therefore, in contrast to RIP, which offers metrics in a range from 1 to 15, OSPF is well-suited for use in larger-scale networks where each route might traverse a large number of routers.

- Variable-length subnets

  Unlike RIP-1, OSPF can route traffic to subnetted networks because its routing information includes a subnet mask.

  Note on selecting a protocol

  > Unlike RIP-1, RIP-2 includes a subnet mask in its routing information and can route traffic to subnetted networks. We recommend that you use RIP-2 if your objective is simply to provide routing to subnets and all routers can use RIP-2.

10.1.2 OSPF functionality

The table below lists the functionality provided by OSPF. The Switch allows the division of an AS into up to four OSPF networks, and can exchange, calculate, and generate routing information for OSPF networks individually. This functionality is called OSPF multi-backbone. Each independent OSPF network is called an OSPF domain.

OSPF must be configured for each OSPF domain.

When OSPF is used with VRF, each VRF can be partitioned into a maximum of four domains.

Table 10-1: OSPF functionality

Functionality	OSPF
Forwarding addresses for external AS routes	Y

Functionality	OSPF
Not-so-stubby areas (NSSA)	Y
Per	Y
Non-broadcast (NBMA) networks	Y
Equal-cost multipath	Y
Virtual links	Y
Multi-backbone	Y
Helper functionality for graceful restarts	Y
Restart functionality for graceful restarts	N
Stub router	Y

Legend: Y: Supported, N: Not supported

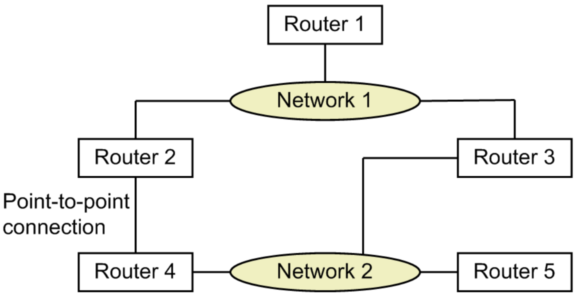
10.1.3 Route selection algorithm

OSPF uses the SPF (Shortest Path First) algorithm for route selection.

Each router that runs OSPF maintains a database containing information on all OSPF-enabled routers and the connections between them, as well as the connections between routers and networks. From this database, the protocol constructs a network topology that has routers and networks at its vertices, and connections between routers and between routers and networks at its edges. By applying the SPF algorithm to this topology, the OSPF protocol computes a shortest-path tree that it uses to determine the routes to each vertex and address.

The following figure shows an example network configuration.

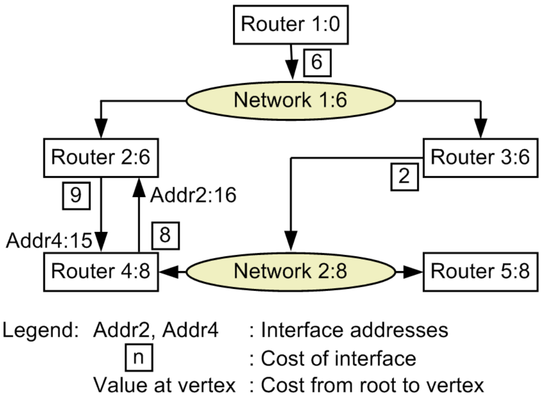
Figure 10-1: Example network configuration



The next figure shows a shortest-path tree generated with Router 1 as its root node. It shows the OSPF topology and an example of assigning costs to the routes between vertices. For connections between routers and networks, you can only assign a cost to the connection from the router to the network. Paths leading from networks to routers always have a cost of zero.

The cost to a given destination reflects the total transmission cost of the interfaces that the route traverses. For example, the cost of the route from Router 1 to Network 2 is 6 (Router 1 to Network 1) + 0 (Network 1 to Router 3) + 2 (Router 3 to Network 2) = 8.

Figure 10-2: Shortest-path tree with Router 1 as the root node



In OSPF, optimum paths are selected based on cost. If the protocol selects a suboptimal path for a given configuration, you can direct it to select a more suitable path by increasing the cost of the unfavorable network interface, or lowering the cost of more suitable network interfaces. Given that 1 is the minimum cost you can assign, you may need to increase the cost of the interfaces in all other routers if the cost of an interface is already too low. Be careful not to assign too low a cost to the interfaces in a large-scale network in case future optimization requires that the cost of a given interface be reduced.

10.1.4 Advertising link states

(1) LSA types

Routing updates for OSPF are called link state advertisements (LSA).

There are three main LSA types:

(a) Intra-area routing information

Announces the status of links to routers and networks for use by the SPF algorithm.

(b) Inter-area routing information

Announces the status of routes to other areas.

(c) External AS routing information

When an OSPF router learns routing information for an external AS, the router can use OSPF to distribute these routes to all other OSPF routers. Any router that injects external AS routing information into an OSPF domain is called an AS boundary router.

(2) External AS route

If you configure route redistribution filtering in the switch configuration, the switch will advertise external AS routes. An AS boundary router adds the following information to LSAs that advertise external routes:

- Metric

  The metric is used by the learning router for path selection among LSAs. Use the `default-metric` command to set the default metric.

- Metric types

  There are two types of metric, type 1 and type 2. The route priority and the calculation method when using the metric for route selection differ between the types. The default metric type is 2.

- Forwarding address (destination)

  An address used as the destination that is reachable by OSPF. If the address is unreachable by

OSPF, the router sets 0.0.0.0 as the forwarding address.

- Tag

  Additional information can be advertised as a tag.

(3) Advertising external AS routes between domains

The multiple OSPF domains connected via a single router operate as independent OSPF networks. For this reason, unless there are configuration settings that specify route redistribution, the router does not distribute the routes of one OSPF domain to the others. If you configure redistribution filtering in the configuration for OSPF routes learned from other domains, routes from other domains are advertised as external AS routes. The following table describes the default filter attributes.

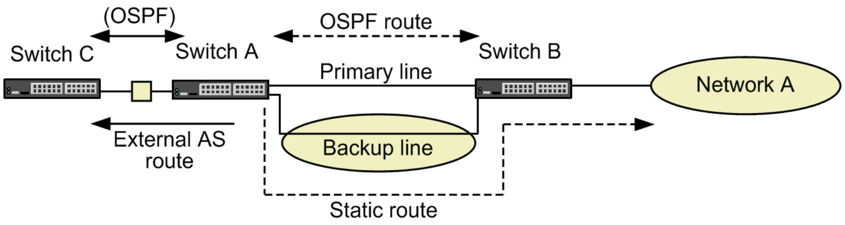
Table 10-2: Filter attributes when redistributing routes to other domains

Attribute	Default value	
	External AS route	Intra-area/inter-area route
Metric	The value specified by the `default-metric` command. If no value is specified, 20 is used.	The value specified by the `default-metric` command. If no value is specified, 20 is used.
Metric type	Type 2 for external AS or NSSA routes.	
Tag value	Inherits the tag value of the route.	0

10.1.5 Example of injecting external AS routes

The following figure shows an example of injecting external AS routes in a configuration that includes a backup line.

Figure 10-3: External AS route injection in configuration using backup line



In OSPF, packets are exchanged at regular intervals to detect neighboring routers. If a backup line is incorporated into the OSPF topology, the backup line stays active because it is called on consistently to carry these packets. However, if you want the backup line to enter an idle state when it is not required for communication, perform the configuration described below.

On Switch A, enable OSPF on the primary line. On the backup line, configure a static route to Network A. As the distance of the static route, assign a larger value (lower priority) than the distance for an internal OSPF route. This results in the internal AS route learned by the OSPF protocol being selected as the route to Network A. In the event of a primary line failure, the relevant internal AS route is deleted from Switch A, and the static route is re-selected. However, the routing information to Network A will no longer exist in Switch C. To advertise an external AS route to Switch C that describes the static route to Network A, you need to configure route redistribution at Switch A. By doing so, information about a valid route to Network A can be injected into OSPF should the primary line fail, without hello packets being exchanged over the backup line.

10.1.6 Criteria for route selection

The OSPF protocol performs SPF calculation whenever LSAs are updated by LSA generation or learning. In SPF calculation, routes are selected based on the SPF algorithm. When a destination becomes unreachable, the protocol deletes the route to that destination.

An area border router performs route selection separately via the SPF algorithm for each area it serves.

The following table describes the priority of route selection in OSPF. You cannot change this priority.

Table 10-3: Route selection priority

Priorit y	Selected item	Description
High	Type of routing information	An OSPF internal AS route (intra-area or inter-area route) has priority over an external AS route.
↑	Learning source domain	If routes exist in more than one domain, the protocol selects the route with the smallest distance. When the distances are equal, the protocol selects the route with the smallest OSPF domain number.
	Route destination type	• Internal AS route: An intra-area route has priority over an inter-area route. • External AS route: A route advertised by an AS boundary router within the same area has priority over a route advertised by a router in another area.
	External AS route type	An external AS route with metric type 1 has priority over an external AS route with metric type 2.
	Areas traversed by external AS route	For a router located at an area border, when a destination AS boundary router connects to more than one area, the area with the smallest cost to the AS boundary router is selected. When the cost values are equal, the area with the largest area ID is selected.
	Cost	• Internal AS route: The route with the smallest cost to the destination has priority. • Type 1 external AS route: The route for which the sum of the metric of the external AS route and the cost to the AS boundary router is smallest is selected. • Type 2 external AS route: The route with the smallest external AS route metric is selected. If the metrics are equal, the route with the smallest cost to the AS boundary router is selected.
↓ Low	Next hop address	The protocol selects the address with the smallest next-hop address.

(1) Distance

When the Switch learns more than one route to the same destination via a number of protocols, it compares the distance of each route and applies the route with the highest priority.

In OSPF, you can set default distances on a per-domain basis. You can assign different distances to external AS routes, intra-area routes, and inter-area routes. To change distances, use the `distance` command.

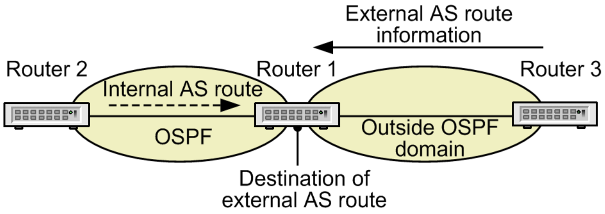
(2) Next hop selection for external AS routes

The destination (next-hop address) of an external AS path is either the address of a neighboring OSPF router, or the forwarding address advertised by LSA. Details are given below.

(a) When targeting an AS boundary router

The figure below shows an example of a system configuration in which an AS boundary router is a target for packet forwarding. In this example, when Router 1 injects a route that Router 1 learns from Router 3 as an external AS route, Router 1 is designated as the forwarding destination. As the route to Router 1, a route selected by the internal AS route selection process is used.

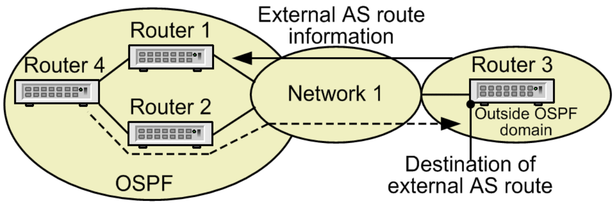
Figure 10-4: Example system configuration (when targeting an AS boundary router)



(b) When targeting a forwarding address

The figure below shows an example of a system configuration in which a forwarding address is a target for packet forwarding. In this example, when Router 1 injects a route that Router 1 (an AS boundary router) learns from Router 3 as an external AS route, the destination (forwarding address) is the address of the Network 1 interface of Router 3. When Router 4 forwards packets to Network 1, it selects the route that passes through Router 2 to forward packets to the injected external route if this route is less costly.

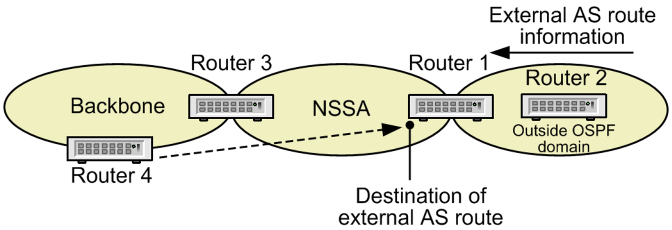
Figure 10-5: Example system configuration (when targeting a forwarding address)



(3) Packet forwarding destinations for external AS routes in NSSAs

When the routing protocol injects routing information as external AS routes, it must designate a forwarding destination address. If the routing information is injected by a broadcast OSPF interface, the forwarding destination is the address of the injection source. Otherwise, the forwarding destination is the address of an arbitrary interface in the NSSA. The figure below shows an example of a system configuration in which packet forwarding targets an arbitrary interface. In this example, when Router 1 injects a route that Router 1 learns from Router 2 as an external AS route, an arbitrary interface in the NSSA is designated as the forwarding destination. Router 4 selects the route to the forwarding destination of the external AS route by an inter-area path selection process.

Figure 10-6: Example system configuration (when targeting an arbitrary interface)



(4) Note regarding NSSAs

The routing protocol selects the forwarding destination address for an external AS route from among the OSPF-enabled interfaces in the NSSA. The forwarding destination is changed if the interface goes down. In the period after the forwarding address changes but before the new external AS route is advertised, the route may be temporarily deleted. To ensure that the forwarding destination remains constant, we recommend that you configure as an OSPF interface the broadcast interface that is the source of the routing information.

10.1.7 Equal-cost multipath

When an equal-cost multipath exists from the local router to a given destination in a topology that includes more than one forwarding destination router, the OSPF protocol can balance the load to that destination by distributing packets among multiple next hops.

For internal AS routes, the Switch selects multiple paths which share the same learning source domain, destination type (inter-area or intra-area route), and cost. Similarly, for external AS routes, the switch selects multiple paths that share the same learning source domain, external AS route type, cost, and metric.

You can use the `maximum-paths` command to change the maximum number of paths. The default is 4.

10.1.8 Notes

(1) Notes on router IDs and network addresses

The OSPF protocol uses a router ID to identify routers when constructing the network topology.

However, the protocol cannot construct an accurate topology when the network design contains either of the following irregularities:

- The same router ID is set for more than one router in a domain
- The same network address is assigned to different networks

If such an irregularity exists, the OSPF protocol cannot select paths accurately because the network design is based on inaccurate topology. We recommend using the following method to determine router IDs:

Determining router IDs

As the router ID of each router, choose one of the IP addresses assigned to an interface running OSPF on that router. The router ID can be any 32-bit numerical value, and this method can prevent a situation in which errors introduced during the design of the OSPF network cause router IDs to be duplicated.

When one router connects to more than one OSPF domain, using the same router ID in every domain does not present a problem.

(2) Notes on route redistribution filter and learning filter

The OSPF protocol advertises every LSA it learns from neighboring routers to its other neighbors. You cannot use redistribution filtering to prevent the advertisement of routes learned by OSPF within the same domain. Also, when using the route summarization functionality (the `ip summary-address` command) to summarize OSPF routes, you cannot prevent LSA advertisement within the same domain even if you configure route filtering to exclude summarization source routes from advertisement.

You can use the `distribute-list in` command to suppress learning of external AS routes that match the filter conditions. However, you cannot prevent OSPF from learning and advertising LSAs. Therefore, OSPF will also advertise the routes it has not itself learned.

(3) Notes on using the multi-backbone functionality

(a) Notes on using multi-backbone

In an environment that divides the network into more than one OSPF domain, the advantages of OSPF, such as cost-based route selection and the avoidance of routing loops, are lost when the selection and distribution of routes crosses OSPF domain boundaries. When building a new network, if there is no need to divide the network into more than one OSPF domain, we recommend that you design it to operate as a single OSPF network.

(b) Notes on configuring multiple domains

If you need to advertise the switch address to more than one OSPF domain, advertise it as an OSPF external AS route. You cannot assign an interface to more than one OSPF domain in the configuration.

10.2 Configuration of basic OSPF functionality

10.2.1 List of configuration commands

The following tables describe the configuration commands for the basic OSPF functionality.

Table 10-4: List of configuration commands related to enabling OSPF

Command name	Description
disable	Disables OSPF.
ip ospf area	Controls OSPF operation at the interface level.
network	Defines the network address range in which OSPF operates (by a combination of IP address and wildcard mask), and the associated area ID.
router-id	Assigns a router ID (to identify a specific router).

Table 10-5: List of configuration command relating to external AS route advertisement

Command name	Description
default-metric	Sets a fixed value as the metric to a destination.
suppress-fa	Suppresses the advertisement of forwarding addresses.
distribute-list out (OSPF)#	Configures redistribution filtering to control which routes are advertised.
redistribute (OSPF)#	Configures redistribution filtering to advertise external AS routes.

#

See *14. Route Filters (IPv4 and IPv6)* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

Table 10-6: List of configuration commands related to route selection and learning

Command name	Description
distance ospf	Sets the distance for OSPF routes.
ip ospf cost	Sets the cost value for OSPF routes.
maximum-paths	Sets the maximum number of equal-cost paths to a given destination.
timers spf	Configures the delay time between when OSPF generates or learns an LSA and when it starts an SPF calculation, and the minimum time between consecutive SPF calculations.
distribute-list in (OSPF)#	Suppresses learning of external AS routes.

#

See *14. Route Filters (IPv4 and IPv6)* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

10.2.2 Overview of configuration

(1) Setting basic OSPF functionality

1. Set up the IP interfaces in advance.
2. Enable OSPF.

Assign a unique router ID to each router.

Router IDs can be selected automatically.

3. Configure advertisement of external AS routes.

   This step is required to advertise the routes of other protocols in OSPF.

   It is also required if you intend to redistribute routes between domains using multi-backbone functionality.

4. Configure route selection.

   If you need to assign a weighting to routes that travel over specific interfaces, use the `ip ospf cost` command to assign a cost.

10.2.3 Configuring OSPF

Points to note

- Exchange of LSAs with neighboring routers is enabled on an interface when the network address for the interface matches the range of addresses specified by the `network` command.
- If you are not using multiple areas, make sure that the same area ID is set on all OSPF routers.

Command examples

1. `(config)# router ospf 1`

   Places the router in OSPF mode. Assigns 1 as the domain number.

2. `(config-router)# router-id 100.1.1.1`

   Assigns 100.1.1.1 as the router ID.

3. `(config-router)# network 10.0.0.0 0.255.255.255 area 0`

   Associates interfaces in the 10.0.0.0/8 network range with area 0.

10.2.4 Configuring advertisement of external AS routes

Points to note

- The `redistribute` command can be used to specify what information (such as the metric, tags, and metric type) the protocol adds to redistributed routes. If you omit the metric in the `redistribute` command, the value specified by the `default-metric` command applies.
- You cannot control redistribution in the same domain for routes learned by OSPF.
- If you specify the `suppress-fa` command, the forwarding address is fixed at 0.0.0.0.

Command examples

1. `(config)# router ospf 1`

   Places the router in OSPF mode.

2. `(config-router)# default-metric 10`

   Sets 10 as the default metric.

3.
```
(config-router)# redistribute static
```
Advertises static routes using the default metric mentioned above.

10.2.5 Configuring route selection

Points to note

The example below shows how to use the `ip ospf cost` command to specify the cost of sending a packet on a specific interface.

If you specify 1 in the `maximum-paths` command, the protocol does not establish an equal-cost multipath even when routes have an equal-cost value.

Command examples

The following is an example of configuring route selection when using single-path routes.

1.
```
(config)# router ospf 1
(config-router)# maximum-paths 1
```
Sets 1 as the maximum number of OSPF paths.

2.
```
(config-router)# network 10.0.0.0 0.255.255.255 area 0
(config-router)# exit
```
Associates interfaces in the 10.0.0.0/8 network range with area 0.

3.
```
(config)# interface vlan 1
(config-if)# ip ospf cost 10
(config-if)# exit
```
Sets the cost to 10.

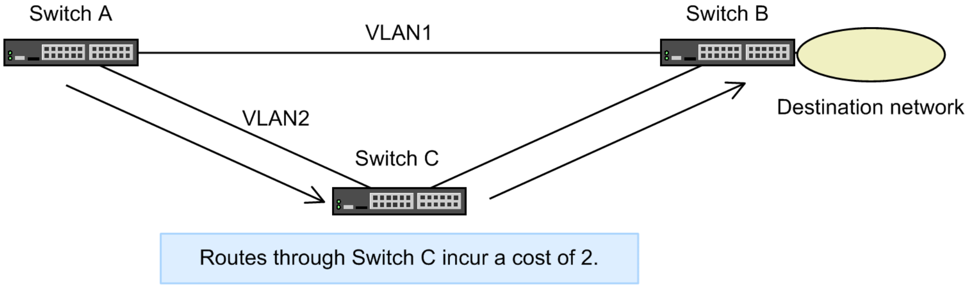
4.
```
(config)# interface vlan 2
(config-if)# ip ospf cost 2
```
Sets the cost to 2. The route through VLAN 2 has priority because it has a smaller cost value than VLAN 1.

10.2.6 Configuring multipath

Points to note

As shown in the example below, by adjusting cost values, you can establish an equal-cost multipath to a destination regardless of the number of routers through which the route passes.

Figure 10-7: Multipath configuration



Command examples

This procedure establishes an equal-cost multipath on Switch A.

1.
```
(config)# router ospf 1
(config-router)# network 10.0.0.0 0.255.255.255 area 0
(config-router)# exit
```
Associates interfaces in the 10.0.0.0/8 network range with area 0.

2.
```
(config)# interface vlan 1
(config-if)# ip ospf cost 2
```
Sets the cost value of VLAN 1 to 2. This gives routes through VLAN 1 the same cost as routes through VLAN 2.

10.2.7 Applying OSPF for a VRF

Points to note

The example below shows how to specify the `vrf` parameter in the `router ospf` command.

Command examples

This procedure uses the loopback address as the router ID and enables OSPF for VRF 2.

1.
```
(config)# interface loopback 2
```
Switches to interface mode and specifies information about loopback 2.

2.
```
(config-if)# vrf forwarding 2
(config-if)# ip address 100.1.1.1
```
Specifies VRF 2 and sets the IP address to 100.1.1.1.

3.
```
(config-if)# ip ospf 1 area 0
(config-if)# exit
```
Specifies that OSPF runs in area 0 of domain 1.

4.
```
(config)# router ospf 1 vrf 2
```
Switches to ospf mode, and then specifies information about the OSPF that runs in VRF 2. Assigns 1 as the domain number.

5. `(config-router)# network 10.0.0.0 0.255.255.255 area 0`

   Associates interfaces in the 10.0.0.0/8 network range with area 0.

10.3 Description of interfaces

10.3.1 OSPF interface types

OSPF classifies the interfaces that connect routers into the following three types, in terms of how packets are sent and received.

- Broadcast

  A broadcast type network in which multiple neighboring routers are managed in an integrated manner and addressed by multicast packets

- Non-broadcast (NBMA)

  A broadcast type network in which multiple neighboring routers are managed in an integrated manner, but which has no broadcast or multicast capability

- Point-to-point

  An interface that directly connects to a single neighboring router. On a virtual link, this interface type operates as a virtual point-to-point interface.

(1) Multihomed network

The Switch can also run OSPF on the secondary address assigned to an interface. Note the following when using OSPF on more than one IP network between multihomed routers in such a configuration:

- On non-NBMA interfaces, routing packets that specify the multicast address are sent to all multihomed routers, which places an undue load on the networks and routers. To reduce the amount of unnecessary network traffic, configure the interfaces as NBMA interfaces.

(2) Notes on configuring OSPF interfaces

OSPF sometimes transmits packets that are equal in length to the Maximum Transfer Unit (MTU) value assigned to the interface. If the packet is longer than the Maximum Receive Unit (MRU, typically equal to the MTU) set on the interface that receives it, a situation might arise in which routers are unable to communicate with each other. This would not occur for normal traffic. For this reason, we recommend that you make sure that the MTU of every network and every router that connects them is set to a value equal to or less than the MRU of all other interfaces when using OSPF.

10.3.2 Connecting with neighboring routers

(1) Hello packets

Routers running OSPF send hello packets on each interface to verify that the links between routers are active. A hello packet is the means by which a router recognizes other routers running OSPF.

(2) Conditions for connecting with neighboring routers

For each network that provides a direct connection between routers, the interface parameters below must be consistent among all connected routers. Routers that do not share the same parameters are not considered to be connected.

(a) Interface address

The interfaces of all routers connected to the same network must have the same IP network address and mask.

(b) Authentication method and authentication key

OSPF can use authentication to verify whether routing information received from a connected router was actually sent by that router. When using authentication, the authentication method and key must match on every interface of every router that connects to a given network.

(c) Area ID

To establish a direct connection between routers, the interfaces on both routers must be assigned the same area ID.

(d) Hello interval and dead interval

The hello interval is the sending interval for hello packets. The dead interval is the number of seconds that a router's hello packets must not have been seen before its neighbors declare the connection down. For a router to properly judge when a connection is lost, these two parameters must have the same values on the interfaces of the directly connected routers.

(e) Area configuration

The information reported to an area differs for stub areas and NSSAs and other area types. To allow OSPF to determine that two routers are directly connected, the areas to which the interfaces belong must share the same stub configuration.

10.3.3 Designated routers in broadcast networks

On a broadcast network, OSPF selects a designated router and backup designated router to manage the connections between the network at the vertex of the topology and the routers directly connected to the network. To limit disruption to routing behavior, the backup designated router takes over the role of the designated router immediately if the designated router fails.

(1) Selecting the designated router and backup designated router

Each router advertises its priority to become a designated router on an interface in its hello packets.

If there is no designated router or backup designated router on an interface, OSPF selects the router with the highest priority as the designated router. If the interface has a designated router but lacks a backup designated router, OSPF selects the router with the next highest priority as the backup designated router. The designated router and backup designated router do not give up their roles if a router comes online with a higher priority.

When an interface of a given router has a priority of 0, that router will never be selected as the designated router or backup designated router in an area where the interface is connected.

When multiple routers exist in a broadcast network that is used to forward traffic, an interface of at least one router that connects to the network must have a priority of 1 or higher.

10.3.4 Transmitting LSAs

In OSPF, neighboring routers exchange link state advertisements (LSAs) to fill the gaps in their routing tables. When a router generates or receives a new LSA, it sends the LSA to all its neighboring routers. This ensures that an identical database is maintained by the Switch and its neighbors. The relationships maintained to create a synchronized database by exchanging LSAs with other routers are called adjacencies.

The Switch uses an LSA synchronization process to send its LSAs to all of its neighboring routers. In turn, the neighboring router sends the LSAs of the Switch to all of its neighbors. Those neighbors then send the LSAs to all of their neighbors. By this process, the LSAs of the Switch are distributed to every router in the area.

(1) LSA age

The age of an LSA is the length of time since it was generated. An LSA remains valid until its age reaches 3600 seconds or it is deleted by the originating router. The switch adds a delay time (set by the `ip ospf transmit-delay` command) to the value of the `Age` field in valid LSAs it transmits.

10.3.5 Passive interfaces

An interface that has no neighboring OSPF routers can be configured as a passive interface. When you apply OSPF to a loopback interface, it becomes a passive interface.

A passive interface does not transmit or receive OSPF packets.

Routes that directly connect to a passive interface are advertised as inter-area or intra-area routes.

10.4 Interface configuration

10.4.1 List of configuration commands

The following table describes the configuration commands for OSPF packets and NBMA.

Table 10-7: List of configuration commands

Command name	Description
ip ospf dead-interval	Sets the length of time that the router maintains adjacency after receiving no hello packets from a neighboring router.
ip ospf hello-interval	Sets the sending interval for hello packets.
ip ospf network	Sets the interface type (broadcast, NBMA, or point-to-point).
ip ospf priority	Sets the router priority that determines the designated router for the network.
ip ospf retransmit-interval	Sets the time between link-state advertisement (LSA) retransmissions.
ip ospf transmit-delay	Sets the time required to send an OSPF packet.
neighbor (OSPF)	Specifies the addresses of the neighboring routers.
passive-interface (OSPF)	Sets the interface to passive mode.

The table below describes the configuration commands related to OSPF operation.

OSPF can receive error packets and send OSPF state change traps.

Table 10-8: List of configuration commands (commands related to OSPF operation)

Command name	Description
system mtu[#1]	Sets the MTU of the switch.
snmp-server host[#2]	Sets a network management device to which traps are sent.
ip mtu[#3]	Specifies the MTU length of IP packets sent on the interface.
interface loopback[#4]	Sets the loopback interface (used as a passive interface in OSPF).

#1

See *10. Ethernet* in the manual *Configuration Command Reference Vol. 1 For Version 11.10*.

#2

See *35. SNMP* in the manual *Configuration Command Reference Vol. 1 For Version 11.10*.

#3

See *2. IPv4, ARP, and ICMP* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

#4

See *3. Loopback Interface (IPv4)* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

10.4.2 Overview of configuration

(1) Configuring an NBMA interface

1. Set up the IP interfaces in advance.
2. Configure the basic OSPF functionality.

   Perform the required setup including enabling OSPF.

   For details, see *10.2 Configuration of basic OSPF functionality*.
3. Configure the interface.

   Use the `ip ospf network` command to set NBMA as the network type.

   You can change parameters, such as the sending interval for hello packets, as required.
4. Use the `neighbor` command to specify the neighboring routers.

(2) Configuring a broadcast interface

1. Set up the IP interfaces in advance.
2. Configure the basic OSPF functionality.

   Perform the required setup including enabling OSPF.

   For details, see *10.2 Configuration of basic OSPF functionality*.
3. Configure the interface.

   You can change parameters, such as the sending interval for hello packets, as required.

10.4.3 Configuring neighboring routers for NBMA interfaces

Points to note

The `neighbor` command applies to NBMA interfaces only.

As shown in the example below, you can use the `priority` parameter of the `neighbor` command to specify the neighboring router's eligibility to become a designated router. A priority of 0 makes the router ineligible to become a designated router. You must assign a priority value to any neighboring router that is eligible to become the designated router.

Command examples

1.
```
(config)# interface vlan 1
(config-if)# ip ospf 1 area 0
```
Enables OSPF on the interface.

2.
```
(config-if)# ip ospf network non-broadcast
(config-if)# exit
```
Sets the interface type to NBMA.

3.
```
(config)# router ospf 1
(config-router)# neighbor 192.168.1.1 priority 2
(config-router)# neighbor 192.168.1.2 priority 2
```
Specifies the interface addresses of the neighboring routers in the domain and sets the `priority` of the neighboring routers to 2.

10.4.4 Changing interface parameters

On interfaces with OSPF enabled, behavior such as hello packet sending conforms to the default values specified in the configuration. You can change how the interface behaves by using the `priority` parameter and `passive-interface` command, among others.

(1) Priority for designated router selection

Networks that contain a large number of routers place a heavy load on the designated router. When a router connects to multiple such networks, we recommend that you set its priority such that it does not become the designated router for more than one network.

Points to note

The greater the value of the `priority` parameter, the higher the priority of the router.

Command examples

1.
```
(config)# interface vlan 1
(config-if)# ip ospf 1 area 0
(config-if)# ip ospf priority 10
```
Sets the interface priority to 10.

(2) Passive interfaces

Points to note

The example below shows how to use the `passive-interface` command to configure a passive interface. If you specify the `ip ospf cost` command, the interface assigns the specified cost to the directly connected routes it advertises.

Command examples

1.
```
(config)# interface vlan 2
(config-if)# ip ospf 1 area 0
(config-if)# ip ospf cost 10
(config-if)# exit
```
Enables OSPF on the interface.

2.
```
(config)# router ospf 1
(config-router)# passive-interface vlan 2
```
Sets VLAN 2 as a passive interface.

10.5 OSPF operation

10.5.1 List of operation commands

The following table describes the operation commands for OSPF.

Table 10-9: List of operation commands

Command name	Description
show ip route	Shows the entries in the routing table.
clear ip route	Clears the IPv4 forwarding entries and re-registers the routing entries.
show ip ospf	Shows OSPF information such as the domain number, information about neighboring routers, interface information, and LSAs.
clear ip ospf	Clears information about the OSPF protocol.
show ip vrf	Shows the IPv4 information of a VRF.
show ip interface ipv4-unicast	Shows information about the IPv4 interfaces on the Switch recognized by the unicast routing program.
debug ip	Shows the packets being routed by IPv4 routing protocols in real time.
show processes cpu unicast#	Shows the CPU usage of a unicast routing program.
restart unicast#	Restarts the unicast routing program.
debug protocols unicast#	Starts the operation message display for event log information output by a unicast routing program.
no debug protocols unicast#	Stops the operation message display for event log information output by a unicast routing program.
dump protocols unicast#	Outputs the trace information and control table information collected by the unicast routing program to a file.
erase protocol-dump unicast#	Deletes the file of trace information and control table information generated by the unicast routing program.

#

See *8. Routing Protocol Common to IPv4 and IPv6* in the manual *Operation Command Reference Vol. 2 For Version 11.10.*

10.5.2 Checking the domain

While OSPF is active, you can use the `show ip ospf` operation command to check relevant settings such as router IDs and distances.

Figure 10-8: Results of executing the show ip ospf command

```
>show ip ospf
Date 20XX/07/14 12:00:00 UTC
OSPF protocol: ON

Domain: 1
Router ID: 172.16.1.1
Distance:
  Intra Area: 10, Inter Area: 10, External: 150
Flags: <AreaBorder ASBoundary>
SPF Interval: 7s, SPF Delay: 3s
Area            Interfaces  Network Range       State
0               1           -                   -
```

```
10             1             192.168.1/24       Advertise
                             172.19/18          DoNotAdvertise
```

10.5.3 Checking information about neighboring routers

You can use the `show ip ospf neighbor` operation command to check the IP address, neighbor state, router ID, and priority of neighboring routers.

On an OSPF interface, the designated router (DR) forms adjacencies with the other routers. You can monitor its progress by checking the `State` field in the command output.

`Full` appears as the status of routers that have formed an adjacency. Otherwise, the router is in the process of forming an adjacency with the designated router and cannot learn OSPF routes at its interfaces.

You can examine the neighboring routers in more detail by executing the `show ip ospf interface` and `show ip ospf neighbor detail` operation commands. The command output includes the interface state, network type, and connectivity with neighboring routers.

- Make sure that the OSPF network type of the interface is the same as that of its neighboring routers.
- When `DR` or `P to P` appears as the interface state, make sure that the status of each neighboring router in the neighbor list is `Full`.
  - A value other than `Full` indicates that the interface has not formed an adjacency with the neighboring router. Check the neighboring routers.
- When `BackupDR` or `DR Other` appears as the interface state, check whether the neighbor list contains a router that is eligible to be selected as the DR.
  - If a DR exists in the neighbor list but its neighboring routers have a status other than `Full`, this indicates that the interface has not formed an adjacency with the neighboring router. Check the neighboring routers.
  - If no DR exists, this might indicate that no priority has been assigned to the switch or its neighbors. Check the priority setting for the switch and neighboring routers.

Figure 10-9: Results of executing the show ip ospf neighbor command

```
>show ip ospf neighbor
Date 20XX/07/14 12:00:00 UTC
Domain: 1
Area: 0
Address         State                RouterID       Priority Interface
172.16.10.11    Full/BackupDR        172.16.1.1            1  172.16.10.10
172.16.10.12    Full/DR Other        172.16.1.2            1  172.16.10.10
172.126.110.111 Exch Start/BackupDR  172.126.123.111       1  172.126.120.130
```

Figure 10-10: Results of executing the show ip ospf interface command (with IP address specified)

```
>show ip ospf interface 192.168.50.1
Date 20XX/05/30 12:00:00 UTC
Domain: 1
Index: 2, Name: VLAN0010, Address: 192.168.50.1, State: P to P
Auth Type: Simple
MTU: 1436, DDinPacket: 70, LSRinPacket: 117, ACKinPacket: 70
Router ID: 192.168.50.1, Network Type: P to P
Area: 0, DR: none, Backup DR: none
Priority: 0, Cost: 1
Transmit Delay: 1s
Intervals:
  Hello: 10s, Dead: 40s, Retransmit: 5s
```

```
Neighbor List (1):
Address         State      RouterID     Priority DR             Backup DR
192.168.50.2    Full       192.168.50.2    0     none           none
>
```

Figure 10-11: Results of executing the show ip ospf neighbor command (detail)

```
>show ip ospf neighbor detail
Date 20XX/05/30 12:00:00 UTC
Domain: 1
Area: 0
Interface Address: 172.16.10.10, Interface State: BackupDR
    Interface Name: VLAN0020
    Neighbor Router ID: 172.16.1.1, Neighbor State: Full/DR
    Neighbor Address: 172.16.10.11, Priority: 1, Poll Interval: 0s
    Last Hello: 6s, Last Exchange: 45d 12h
    DR: 172.16.10.11, Backup DR: 172.16.10.10
    DS: 0, LSR: 0, Retrans: 0, <Master>

    Neighbor Router ID: 172.16.1.2, Neighbor State: Full/DR Other
    Neighbor Address: 172.16.10.12, Priority: 1, Poll Interval: 0s
    Last Hello:  3s, Last Exchange:  1s
    DR: 172.16.10.11, Backup DR: 172.16.10.10
    DS: 0, LSR: 0, Retrans: 0, <>
>
```

10.5.4 Checking interface information

You can use the `show ip ospf interface` operation command to display information about interfaces running OSPF, including the address, state, priority, and cost.

The command does not display information for IP interfaces that are down.

Figure 10-12: Results of executing the show ip ospf interface command

```
>show ip ospf interface
Date 20XX/07/14 12:00:00 UTC
Domain: 1
Area 0
Address         State    Priority Cost  Neighbor DR              Backup DR
172.16.10.10    DR       1        1     1        172.17.1.1      172.16.1.1
Area 1
Address         State    Priority Cost  Neighbor DR              Backup DR
172.18.10.11    DR       1        1     1        172.18.1.1      172.16.1.1
```

10.5.5 Checking LSA information

(1) Checking the number of LSAs

You can use the `show ip ospf database database-summary` operation command to check how many LSAs are in the OSPF database.

Figure 10-13: Results of executing the show ip ospf database database-summary command

```
>show ip ospf database database-summary
Date 20XX/07/14 12:00:00 UTC

Domain: 1
Local Router ID: 172.16.1.1
Area            Router Network Summary Asb-     NSSA   Area   External Opaque-
                                       summary         Total           link
0               4      2       1       2        0      9      2        1
```

(2) Checking information about LSA advertisement

You can use the `show ip ospf database` operation command to check information about LSA advertisement and LSA ages for each LSA type.

Common LSA types include router link and network link LSAs. Use the `show ip ospf database` command to confirm that LSA advertisement is configured as follows on the Switch:

(a) Router link LSAs are advertised

The LSID in the command output is the router ID.

(b) The Switch advertises the interface of the designated router as a network link LSA

The LSID in the command output is the interface address.

(c) If the Switch is an AS border router, it advertises its routes as `AS External Link`.

Figure 10-14: Results of executing the show ip ospf database command

```
>show ip ospf database
Date 20XX/07/14 12:00:00 UTC

Domain: 1
Local Router ID: 10.1.2.8
Area : 1
  LS Database: Router Link
   Router ID          LSID            ADV Router      Age  Sequence Link Count
   10.1.2.8           10.1.2.8        10.1.2.8        3    80000021 1
   10.1.10.11         10.1.10.11      10.1.10.11      2    80000002 1
  LS Database: Network Link
   DR Interface       LSID            ADV Router         Age  Sequence
   100.1.2.2/24       100.1.2.2       10.1.2.8           3    80000001

  LS Database: AS External Link
   Network Address    LSID            AS Boundary Router Age  Sequence
   10.1.1.0/24        10.1.1.0        10.1.2.8           778  80000005
```

Chapter

11. Extended OSPF Functionality [OS-L3SA]

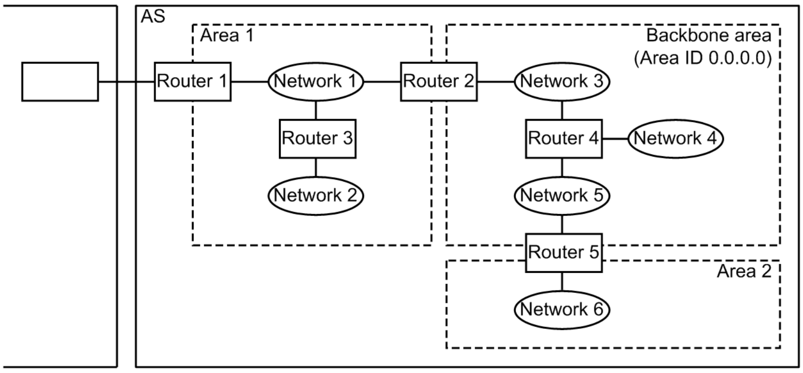
This chapter describes the extended functionality of the OSPF protocol.

11.1 Description of areas and area division functionality

11.1.1 Area border

The OSPF protocol allows you to divide an autonomous system (AS) into a number of areas to minimize routing traffic and cut down the processing time required by the SPF algorithm. The following figure shows an example of an OSPF network topology in which an AS is divided into several areas.

Figure 11-1: Example OSPF network topology using area division



A router that attaches to multiple areas is called an area border router. Examples are Router 2 and Router 5 in the figure above.

Information about the connection status of a given area is not reported outside that area. A router contains no information about the connection status of areas to which it is not connected.

(1) Backbone area

An area with an area ID of 0 is called a backbone. The backbone has a special role in a split-area topology. If an AS is divided into areas, one of the areas must be designated as the backbone area. Be careful to avoid creating a configuration in which an AS has multiple backbones. In such a configuration, information about inter-area routes is spread across multiple backbones. This can result in unreachable routes being generated or the protocol failing to select the best route.

An area border router uses the backbone to disseminate routing information to other areas. For this reason, an area border router must connect to the backbone area.

(2) Notes on area division

Although dividing an AS into areas reduces the load on the routers and minimizes routing traffic, it also adds a level of complexity to the OSPF algorithm. In particular, you can run into difficulties putting the appropriate fault handling in place. We recommend that you do not use area division unless you have a specific need to reduce the load on routers or networks.

(3) Notes on area border routers

- An area border router runs an iteration of the SPF algorithm for each area it serves. It provides functionality that summarizes the topology information for the areas to which it is connected for distribution to other areas in the AS. Thus, an area border router that is associated with a large number of areas is subject to heavy loads. We recommend a network configuration that limits the number of areas each area border router serves.
- When an area has only one area border router, a fault in that router can partition the area from the backbone, resulting in a loss of connectivity with other areas. When building a network, we recommend that you place more than one area border router in areas with servers and AS boundary routers that support mission-critical functions and connections, so that there are sufficient alternate routes available if a router fails.

11.1.2 Route control for divided areas

Each area border router summarizes the routing information for each non-backbone area to which it belongs, and then sends it to all other routers in the backbone area. The summarized routing information for the backbone area and the summarized routing information sent from other areas to the backbone area are reported to the routers in non-backbone areas.

When a router uses this summarized routing information to determine the route to an address, the route will traverse the area border router that is the source of the summarized routing information. Therefore, a route between different areas always passes through the backbone area.

When advertising routing information to other areas, the area border router summarizes topology information derived from link states between routers and networks and the costs of those links. This information describes the cost between the area border router and specific routers and networks. The summarized information is called *inter-area routing information*. The information that describes routes to networks is advertised in Type 3 LSAs, and the information that describes routes to AS boundary routers is advertised in Type 4 LSAs.

(1) Route summarization in area border routers

The following table describes the summarization and suppression of routing information, and the summary information that the area border router reports to other areas.

Table 11-1: Route summarization and suppression, and summaries sent to external areas

Network addresses in area	Summarization and suppression	Summary reported to other areas
10.0.1.0/24 10.0.2.0/25 10.0.2.128/25 10.0.3.0/24	None	10.0.1.0/24 10.0.2.0/25 10.0.2.128/25 10.0.3.0/24
10.0.1.0/24 10.0.2.0/25 10.0.2.128/25 10.0.3.0/24	10.0.0.0/23 10.0.2.0/24	10.0.0.0/23 10.0.2.0/24 10.0.3.0/24
10.0.1.0/24 10.0.2.0/25 10.0.2.128/25 10.0.3.0/24 192.168.3.0/26 192.168.3.64/26 192.168.3.128/26	10.0.0.0/8 (suppressed) 192.168.3.0/24	192.168.3.0/24

You can use configuration commands to specify an address range that the area border router condenses into a single summarized route when summarizing the topology information for a given area. To do so, use the `area range` command specifying an address with a network mask. You can also specify a parameter that suppresses advertisement of the address range.

If the area contains even one network that matches the address range specified in the configuration command, the area border router creates a summary LSA for all networks within the range, with the masked address as its destination. The router does not report information for each network in the range outside area boundaries. The summarized routing information will adopt the highest cost among its constituent routes.

If you suppress advertisement, no summary LSAs for networks in the address range are advertised outside area boundaries, nor are the routes summarized under the masked address advertised to other areas. As a result, routes to addresses within the specified address range remain hidden from other areas.

11.1.3 Stub areas

You can configure an area as a stub area if it is not a backbone area and does not contain an AS boundary router. To do so, use the `area stub` configuration command.

Stub areas do not accept information about routes external to the AS. This reduces the amount of routing traffic in stub areas, and minimizes the demand that route updates and route selection place on router resources. The area border router employs a default route in place of the external routes in the stub area.

If you execute the `area stub` command with the `no-summary` parameter specified, the area border router does not advertise routes from other areas (inter-area routes) into the stub. Routers in the stub area must use the default route to send any traffic outside the area.

11.1.4 NSSA

You can configure a non-backbone area as a not-so-stubby area (NSSA). To do so, use the `area nssa` configuration command.

Like a stub area, an NSSA does not receive information about external AS routes from other areas.

Even with advertised route filtering configured (using the `redistribute` configuration command), if you specify the `no-redistribution` parameter of the `area nssa` command, the area border router will not inject external routes into the NSSA. This reduces the amount of routing traffic in NSSAs, and minimizes the demand that route updates and route selection place on router resources.

If you execute the `area nssa` command with the `no-summary` parameter specified, the area border router does not advertise routes from other areas (inter-area routes) to the NSSA, replacing them instead with a single default route. This default route is advertised to the NSSA as inter-area routing information (a Type 3 LSA).

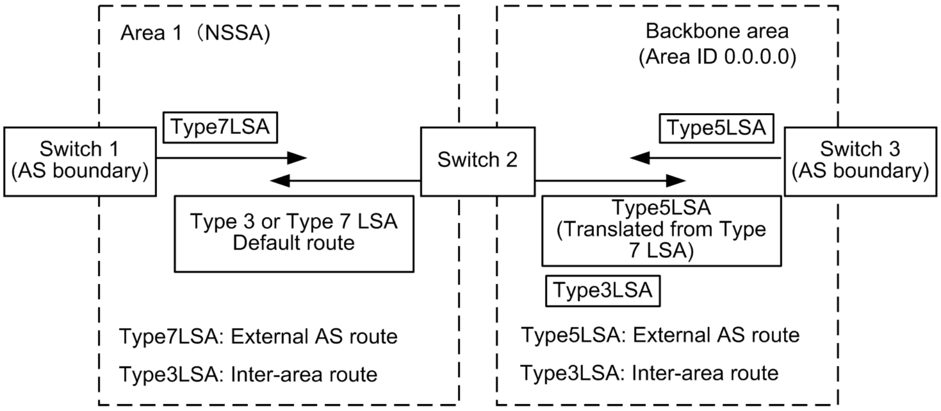
(1) Advertising external AS routes

An AS boundary router in an NSSA uses Type 7 LSAs (NSSA external) to advertise external AS routes. Type 7 LSAs are propagated only within the originating area.

If you execute the `area nssa` command with the `default-information-originate` parameter specified, the area border router injects a Type 7 default route into the NSSA area. If more than one router in the NSSA advertises a Type 7 LSA for the default route, these routers select the default route with the highest priority as an external AS route.

The area border router translates the external AS routes it learns in the NSSA to Type 5 LSAs and advertises them to other areas. The Type 5 LSA inherits the tags and forwarding address of the Type 7 LSA. If you specify the `area nssa translate type7 suppress-fa` configuration command in the NSSA that generates the Type 7 LSA, the router will use 0.0.0.0 as the forwarding address of the Type 5 LSA. The following figure illustrates how routes are exchanged between an NSSA and the backbone.

Figure 11-2: Exchanging routes between an NSSA and the backbone



(2) Restrictions

The Switch conforms to RFC 3101 (The OSPF Not-So-Stubby Area (NSSA) Option), with the exception that the following functionality is not implemented due to software limitations:

- Type-7 address ranges
- Type-7 translator election

As a consequence, the external AS routes learned from an NSSA are always advertised outside its boundaries.

11.1.5 Virtual links

In OSPF, you can use the concept of a virtual link to simulate a point-to-point connection between two area border routers. Both routers must be located in a non-backbone area that is not configured as a stub area or NSSA. The virtual link can be used as an interface to the backbone area. This virtual line is called a *virtual link*. The area that carries its routes is called the transit area.

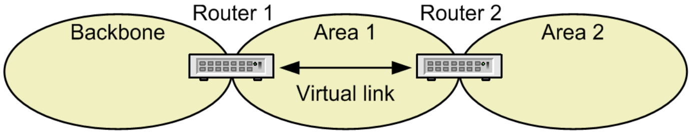
The use of virtual links is described based on the following three examples:

- Providing virtual connections for areas not physically connected to the backbone
- Joining multiple backbones
- Adding redundant connectivity to prevent faults from partitioning the backbone

(a) Providing virtual connections for areas not physically connected to the backbone

In the figure below, Area 2 is not connected to the backbone. If you configure a virtual link between Router 1 and Router 2 that uses Area 1 as its transit area, Router 2 considers itself an area border router because it has a connection to the backbone. Area 2 can now connect to the backbone through Router 2.

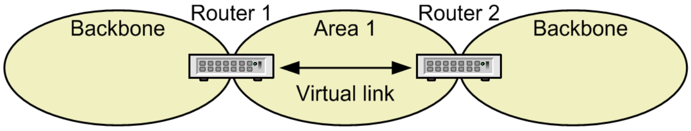
Figure 11-3: Connecting an area to the backbone



(b) Joining multiple backbones

The figure below shows an AS containing two backbone areas. This partition in the backbone can cause some destinations to become unreachable. You can avoid this problem by configuring a virtual link between Router 1 and Router 2 that uses Area 1 as its transit area to join the backbones.

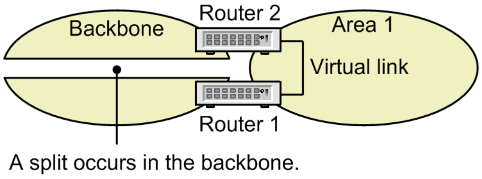
Figure 11-4: Providing a connection between backbones



(c) Adding redundant connectivity to prevent faults from partitioning the backbone

In the figure below, the connection between Router 1 and Router 2 is lost when a network fault occurs in the backbone, causing the backbone to be partitioned. By configuring a virtual link between Router 1 and Router 2 that uses Area 1 as its transit area, you can provide a secondary route (or a primary route if the cost of the virtual link is sufficiently lower than that between Router 1 and Router 2 on the backbone) that acts as a contingency against partitioning of the backbone.

Figure 11-5: Providing an alternate route in case the backbone is partitioned



11.1.6 Operation of virtual links

A virtual link must be configured in both routers that serve as the endpoints.

The routers at each end of the virtual link exchange OSPF packets over the virtual link and learn routing information for the backbone area.

Note the following when operating a virtual link:

- The cost of a virtual link is the cost of the route between the routers at the two endpoints over the transit area.
- The route for normal traffic might differ from that of routing information traffic over the virtual link if the route between the two endpoints over the transit area is an equal-cost multipath.

(1) Connections with neighboring routers

When the virtual link is active, hello packets are sent to the neighboring routers on the virtual link to detect connectivity. A virtual link becomes active when the transit area contains a path to the router at the other end of the virtual link.

A hello packet is the means by which a router recognizes other routers running OSPF.

Use the `area virtual-link` command to configure hello packets in the context of the virtual link. The `dead-interval` parameter must be longer than any interval value (as set by the `ip ospf dead-interval` command) set for an interface in any network that forms the route between the end points of the virtual link in the transit area. If you specify a shorter value and a network failure occurs that affects routes in the transit area, the virtual link might be shut down before the routers have time to switch to alternate routes.

The resending interval for LSAs (as set by the `retransmit-interval` parameter of the `area virtual-link` command) must comfortably exceed the expected round-trip time for packets between the endpoint routers of the virtual link.

11.2 Area configuration

11.2.1 List of configuration commands

The tables below list the configuration commands used when a stub area or NSSA is used and the Switch operates as an area border router.

For details about the commands for the functionality described in *10. OSPF [OS-L3SA]*, see *Table 10-5: List of configuration command relating to external AS route advertisement*, *Table 10-6: List of configuration commands related to route selection and learning*, and *Table 10-7: List of configuration commands*.

Table 11-2: List of configuration commands related to areas

Command name	Description
area default-cost	Sets the cost of the default route advertised to stub areas.
area nssa	Sets an area as an NSSA.
area range	Summarizes inter-area routes on an area border router into a single masked address for advertisement to other areas.
area stub	Sets the area to operate as a stub area.
area virtual-link	Establishes a virtual link.

Table 11-3: List of configuration commands related to enabling OSPF

Command name	Description
disable	Disables OSPF.
ip ospf area	Controls OSPF operation at the interface level.
network	Defines the network address range in which OSPF operates (by a combination of IP address and wildcard mask), and the associated area ID.
router-id	Assigns a router ID (to identify a specific router).

11.2.2 Overview of configuration

(1) Configuring a stub area or NSSA when not an area border router

1. Set up the IP interfaces in advance.
2. Configure the area to operate as a stub area or NSSA.
3. Enable OSPF.

(2) Configuring an area border router

1. Set up the IP interfaces in advance.
2. Configure the relevant areas to operate as stub areas or NSSAs.

   For a stub area, specify the cost of the advertised default route.

   In an NSSA, the default route can be advertised as an external AS route.
3. Configure route summarization.
4. Enable OSPF.

   Configure more than one area. At this time, you must configure an interface that connects to area 0 (the backbone) or establish a virtual link.

5. Establish virtual links.

11.2.3 Configuring a stub area

Points to note

An area border router advertises a default route into areas specified by the `area stub` command.

A stub area or NSSA must be configured on every router in the same area.

Command examples

1. `(config)# router ospf 1`

   Places the router in OSPF mode. Assigns 1 as the domain number.

2. `(config-router)# area 1 stub`

   Designates area 1 as a stub area.

3. `(config-router)# router-id 100.1.1.1`

   Assigns 100.1.1.1 as the router ID.

4. `(config-router)# network 10.0.0.0 0.255.255.255 area 1`

   Associates interfaces in the 10.0.0.0/8 network range with area 1.

11.2.4 Configuring an area border router

Points to note

You can use the `area range` command with the `not-advertise` parameter specified to prevent the networks in a masked address range from being advertised outside the area.

For a given area, you can specify more than one address range for route summarization and for route advertisement suppression. A router or network in the area can be assigned an address not included in any of the address ranges you specify. However, when building a network, by assigning addresses to suit the topology first and then configuring summarization for address ranges in a manner appropriate for the topology, you can efficiently reduce the amount of OSPF routing traffic without impairing the ability of the protocol to select the best routes.

Command examples

The following is an example of configuring route summarization for an area border router associated with areas 0 and 1.

1. `(config)# router ospf 1`

   `(config-router)# area 0 range 10.0.0.0 255.255.254.0`

   Having learned routes in the range of network 10.0.0.0 masked by 255.255.254.0 in area 0, the router advertises a summarized route to area 1.

2. `(config-router)# area 1 range 10.0.2.0 255.255.255.0`

   Having learned routes in the range of network 10.0.2.0 masked by 255.255.255.0 in area 1, the router advertises a summarized route to area 0.

3.
```
(config-router)# network 10.0.0.0 0.0.0.255 area 0
```
Associates interfaces in the 10.0.0.0/24 network range with area 0.

4.
```
(config-router)# network 10.0.2.0 0.0.0.255 area 1
```
Associates interfaces in the 10.0.2.0/24 network range with area 1.

11.2.5 Configuring virtual links

Points to note

The example below shows how to use the `area virtual-link` command to specify the router ID of the virtual link neighbor.

Command examples

1.
```
(config)# router ospf 1
(config-router)# network 10.0.0.0 0.0.0.255 area 0
```
Associates interfaces in the 10.0.0.0/24 network range with area 0.

2.
```
(config-router)# network 10.0.2.0 0.0.0.255 area 1
```
Associates interfaces in the 10.0.2.0/24 network range with area 1.

3.
```
(config-router)# area 1 virtual-link 10.0.0.1
(config-router)# area 1 virtual-link 10.0.0.2
```
Sets the virtual link neighbor in transit area 1.

11.3 Description of neighboring router authentication

In OSPF, you can use authentication to verify that the sender and recipient of routing updates are managed by the same entity. By using authentication between neighboring routers, you can defend an authenticating router from attacks involving false route updates.

- Authentication method

  You can perform authentication using either a plain-text password or an MD5 digest.

  In the switch configuration, you can specify an authentication method at the area level or the interface level. If you specify neither, the switch does not participate in authentication. Moreover, an interface for which an authentication method is specified but not an authentication key does not participate in authentication. A virtual link will use the authentication method specified for area 0.

11.3.1 Authentication procedure

You can perform authentication using either a plain-text password or an MD5 digest.

(1) Plain-text password authentication

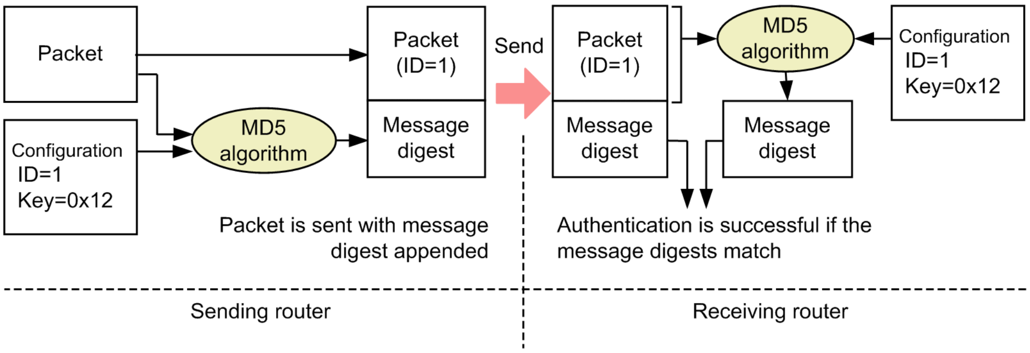
In plain-text authentication, the authenticating key specified in the switch configuration is sent as a password along with each piece of routing information.

Authentication succeeds if the password in the routing information matches the authenticating key specified in the configuration. The router discards information that fails authentication.

(2) MD5 authentication

In MD5 authentication, the receiving router authenticates the source of routing information based on a message digest produced by the MD5 algorithm. The following figure shows the flow of data in MD5 authentication.

Figure 11-6: Flow of MD5 authentication



A router uses the MD5 hashing algorithm to produce a message digest from the authenticating key, the key ID, and the routing information itself. The sending router sends this message digest along with the routing information.

The receiving router tries all of the authenticating keys configured in the router that have the key ID contained in the routing information. It uses the authenticating keys to compute its own versions of the message digest. Authentication is considered successful if any of the message digests matches the message digest received with the routing information. The routing information is considered to have come from an untrustworthy source if authentication fails despite trying all valid keys. The router disregards information that fails authentication.

11.4 Configuration of neighboring router authentication

11.4.1 List of configuration commands

The table below lists the configuration commands for neighboring router authentication functionality. You can also configure the switch to issue an SNMP trap when it receives an error packet indicating an authentication failure or other error.

Table 11-4: List of configuration commands

Command name	Description
area authentication	Sets the authentication method (plain-text password or MD5 authentication).
area virtual-link	Sets the authentication key via the `authentication-key` and `message digest-key md5` parameters.
ip ospf authentication	Sets the authentication method (plain-text password or MD5 authentication).
ip ospf authentication-key	Sets the authentication key.
ip ospf message-digest-key	Sets the MD5 authentication key.
snmp-server host#	Sets a network management device to send SNMP traps.

#

See *35. SNMP* in the manual *Configuration Command Reference Vol. 1 For Version 11.10*.

11.4.2 Changing the MD5 authentication key

You can configure multiple MD5 keys to transition from one key to another.

To transition to a new MD5 key:

1. Create a new key that has a different ID number from the key currently in use.
2. Configure the new key on each neighboring router.
3. Delete the old authentication key.

11.4.3 Configuring plain-text password authentication

Points to note

The example below shows how to use the `area authentication` command to set the authentication method for an OSPF area.

Command examples

1.
```
(config)# router ospf 1
(config-router)# area 1 authentication
(config-router)# exit
```
Configures area 1 to use plain-text password authentication.

2.
```
(config)# interface vlan 1
(config-if)# ip ospf authentication-key a1w@9a
```
Sets `a1w@9a` as the authentication key.

If VLAN 1 is assigned to area 1, OSPF packets conveyed over VLAN 1 are subject to plain-text password authentication.

11.4.4 Configuring MD5 authentication

Points to note

When configuring an authentication key, you must specify a key ID in addition to the key itself.

Command examples

1.
```
(config)# router ospf 1
(config-router)# area 1 authentication message-digest
(config-router)# exit
```

Configures area 1 to use MD5 authentication.

2.
```
(config)# interface vlan 1
(config-if)# ip ospf message-digest-key 1 md5 a1w@9a
```

Sets `a1w@9a` as the authentication key associated with key ID 1. If VLAN 1 is assigned to area 1, OSPF packets conveyed over VLAN 1 are subject to authentication using message digests.

11.5 Description of graceful restart

11.5.1 Overview

In OSPF, a device that uses a graceful restart to restart the OSPF protocol is called a restarting router. The functionality that implements this graceful restart is called the graceful restart functionality. A neighboring router that assists the restarting router in this process is called a helper router. The functionality by which the helper router assists the graceful-restart process is called the helper functionality.

The Switch supports an implementation of the helper functionality.

11.5.2 Helper functionality

When acting as a helper router, the Switch preserves the routes associated with the restarting router for the duration of the graceful-restart process.

(1) Operating conditions for the helper functionality

The switch can operate as a helper router as long as the following conditions are met:

- It is not already acting as a helper for another restarting router in the same domain. In a given domain, the Switch can only act as a helper for one router at a time. If only one router is restarting, the switch provides the helper functionality on every interface that connects to the restarting router.
- It is not waiting to receive an ACK for an OSPF update packet sent to the restarting router.

(2) Scenarios where the helper functionality fails

The switch continues to operate as a helper router until the restarting router re-establishes adjacency or a notification is received indicating that the restarting router has exited the graceful-restart process.

To avoid potential conflicts with the routes maintained by the restarting router, the switch stops its helper functionality and recalculates its OSPF routes if any of the following events occurs:

- A new LSA (excluding periodic updates) is learned from a neighbor and advertised to the restarting router.
- The OSPF interface goes down.
- The switch loses or establishes adjacency with a router other than the restarting router, thus generating an LSA update.
- More than one router is restarting in the same OSPF domain.
- You use the `graceful-restart mode` command to disable helper mode in the switch configuration.

11.5.3 Opaque LSA

The switch learns and advertises a Type 9 opaque LSA at the beginning and end of the graceful restart.

The following limitations apply to the use of opaque LSAs on the Switch:

- In terms of Type 9 opaque LSAs, functionality other than the grace-LSAs used for graceful restarts is not supported.
- Learning and advertisement of Type 10 and Type 11 opaque LSAs is not supported.

11.6 Graceful restart configuration

11.6.1 List of configuration commands

Configure the OSPF helper functionality of the Switch if you intend to use the OSPF restart functionality on any of its neighboring routers.

The following table describes the configuration commands for the graceful restart functionality.

Table 11-5: List of configuration commands

Command name	Description
graceful-restart mode	Enables the helper functionality.
graceful-restart strict-lsa-checking	Allows the helper router to terminate the graceful-restart process if it detects a discrepancy between its own LSA database and that of the restarting router.

11.6.2 Configuring the helper functionality

Points to note

The example below shows how to enable the helper functionality on the switch. If you omit this configuration, the switch will not act as a helper router.

Command examples

1.
```
(config)# router ospf 1
(config-router)# graceful-restart mode helper
```
Enables the helper functionality on the switch.

11.7 Description of stub routers

11.7.1 Overview

If the connection with neighboring router is incomplete or unstable, the entire routing might become unstable. This situation might occur when starting, restarting or adding a router to the network. In such a situation, the OSPF protocol allows a router to disseminate routing information that persuades nearby devices to use alternate routes. A router that uses this behavior is called a stub router in OSPF. This functionality allows you to prevent instability in a specific device from destabilizing the rest of the network.

(1) Max metric

A stub router advertises the maximum cost of 65535 for all of its connected OSPF interfaces. This action gives OSPF routes through the stub router a higher cost than their alternatives.

However, if the interface is a stub network (having no neighboring routers), the stub router will advertise the cost specified in the configuration. The routes advertised by the stub router might take priority over the stub network or external AS routes.

The maximum metric advertised for routes through the stub router causes nearby routers to give priority to alternate routes. You can also use the address of the stub router for Telnet or SNMP management or to exchange BGP4 routing information.

11.7.2 Stub router operation

You can use the `max-metric router-lsa` configuration command to enable stub router functionality at the domain level. You can also specify whether the router functions as a stub router on a permanent basis, or only at startup.

(1) Permanently operating as a stub router

The router advertises maximum cost for its routes on a permanent basis. It continues to operate as a stub router until the user deletes the setting.

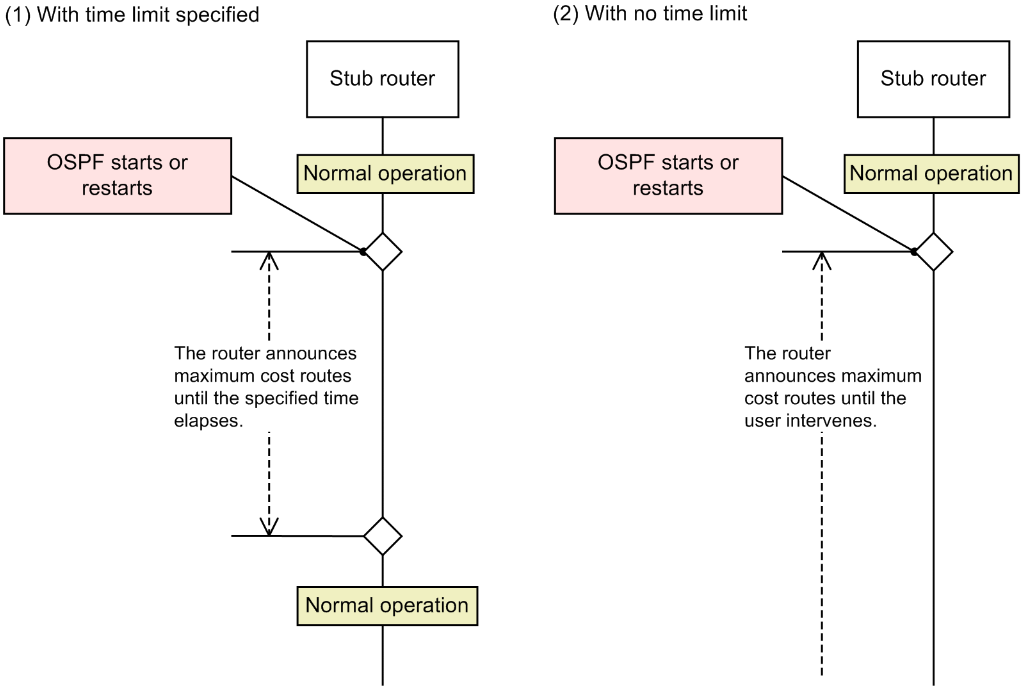
(2) Operating as a stub router at startup

The router advertises maximum cost for its routes when either of the events below occurs. It continues to operate as a stub router until the timer specified in the configuration expires.

- The routing program restarts.
- The switch starts.

You can stop the router from operating as a stub router by executing the `clear ip ospf stub-router` operation command or removing the setting from the configuration. The following figure shows the operation of a stub router:

Figure 11-7: Stub router operation


(1) With time limit specified
Stub router
OSPF starts or restarts
Normal operation
The router announces maximum cost routes until the specified time elapses.
Normal operation
(2) With no time limit
Stub router
OSPF starts or restarts
Normal operation
The router announces maximum cost routes until the user intervenes.

(3) Notes

1. Do not change the configuration of a stub router that is participating as a helper router in a graceful restart. The stub router might begin or cease operation causing disruption to the helper process.
2. If a router is set up to always operate as a stub router and you then change the settings so that it operates as a stub router after startup, it will immediately cease to operate as a stub router.
3. You cannot configure a virtual link through a stub router.

   If the cost in the transit area is higher than 65535, the virtual link will mark itself as unreachable.
4. In the older OSPF standard described in RFC 1247, routing information subject to a maximum metric is not used for SPF computation. For this reason, devices that do not conform to the newer OSPF standard will not register routes through the stub network.

11.8 Stub router configuration

11.8.1 List of configuration commands

You can configure the Switch as a stub router to de-prioritize the routes that pass through it.

This process increases the metric of routes that traverse the stub router.

The following table describes the configuration command for a stub router.

Table 11-6: List of configuration commands

Command name	Description
max-metric router-lsa	Sets the switch to operate as a stub router.

11.8.2 Stub router functionality

Points to note

The example below shows how to specify the switch to operate as a stub router. If you omit the `on-startup` parameter, the switch operates as a stub router permanently.

Command examples

1.
```
(config)# router ospf 1
(config-router)# max-metric router-lsa
```
Enables stub router functionality.

11.9 Operation for extended OSPF functionality

11.9.1 List of operation commands

The following table describes the operation commands for the extended functionality of the OSPF protocol.

Table 11-7: List of operation commands

Command name	Description
show ip ospf	Shows domain information (including the status of area borders and graceful restarts) and the area configuration.
clear ip ospf	Clears information about the OSPF protocol. You can stop the switch from acting as a stub router by executing this command with the `stub-router` parameter.

11.9.2 Checking area borders

On an area border router, you can use the `show ip ospf` operation command to make sure that the `AreaBorder` appears in the `Flags` section of the command output.

You can also check whether inter-area route summarization has been applied correctly.

Figure 11-8: Results of executing the show ip ospf command

```
>show ip ospf
Date 20XX/07/14 12:00:00 UTC
OSPF protocol: ON

Domain: 1
Router ID: 172.16.1.1
Distance:
  Intra Area: 10, Inter Area: 10, External: 150
Flags: <AreaBorder ASBoundary>
SPF Interval: 7s, SPF Delay: 3s
Graceful Restart: Helper
  Helper Status : Finished  20XX/02/15 14:12:22
Area            Interfaces  Network Range      State
0               1           -                  -
10              1           192.168.1/24       Advertise
                            172.19/18          DoNotAdvertise
```

11.9.3 Checking areas

You can check whether the areas you set in the switch configuration have been applied correctly. To display a list of areas, execute the `show ip ospf` operation command with the `area` parameter specified.

Figure 11-9: Results of executing the show ip ospf area command

```
>show ip ospf area
Date 20XX/07/14 12:00:00 UTC
Domain: 1
ID              Neighbor   SPFcount   Flags
0               2          14         <ASBoundary>
1               2          8          <NSSA>
>
```

11.9.4 Checking graceful restart

You can check the status of the graceful restart functionality by executing the `show ip ospf` command.

Figure 11-10: Results of executing the show ip ospf command

```
>show ip ospf
Date 20XX/07/14 12:00:00 UTC
OSPF protocol: ON

Domain: 1
Router ID: 172.16.1.1
Distance:
  Intra Area: 10, Inter Area: 10, External: 150
Flags: <AreaBorder ASBoundary>
SPF Interval: 7s, SPF Delay: 3s
Graceful Restart: Helper
  Helper Status : Finished  20XX/02/15 14:12:22
Area            Interfaces  Network Range      State
0               1           -                  -
10              1           192.168.1/24       Advertise
```

Chapter

12. BGP4 [OS-L3SA]

This chapter describes the IPv4 routing protocol Border Gateway Protocol 4 (BGP4) and explains how to use it.

12.1 Description of basic functionality

12.1.1 Overview

BGP4 is a routing protocol used for Internet connections over which vast amounts of routing information need to be exchanged between Internet service providers (ISPs). BGP4 is based on hierarchical network concepts, and is used for the exchange of routing tables between ISPs at the Internet backbone level. It is also used for connecting an intranet to two or more ISPs.

Within an AS, routing information is exchanged using an Interior Gateway Protocol (IGP) such as RIP or OSPF. BGP4 is an Exterior Gateway Protocol (EGP) that operates between ASs. It can handle all the routing information used on the Internet.

The following table summarizes BGP4 functionality.

Table 12-1: BGP4 (IPv4) functionality

Functionality	BGP4
EBGP, IBGP peering, and route distribution	Y
Route filtering and BGP attribute manipulation	Y
Community	Y
Route reflection	Y
Confederations	Y
Capability negotiation	Y
Route refresh capability	Y
Multipathing	Y
Peer groups#1	Y
Route flap dampening#2	Y
BGP4 MIB#2	Y
TCP MD5 authentication	Y
Graceful restart	Y#3
Maximum number of learned routes	Y

Legend: Y: Supported

#1: A peer group is a grouping of external peers and member AS peers, or a grouping of internal peers.

#2: Not supported in VRFs

#3: Only the receiving router functionality is supported.

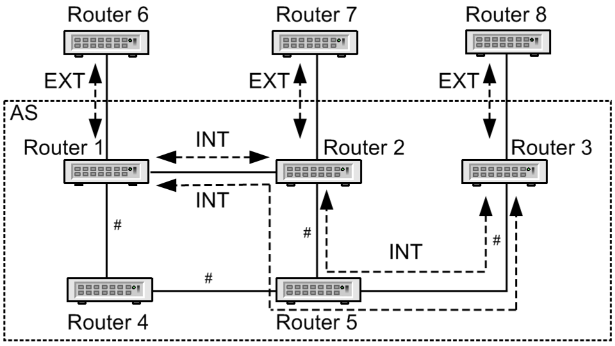
12.1.2 Peer types and connectivity

Because BGP4 operates between ASs, the routing information it handles consists of AS path information to a destination network (the series of ASs that a packet traverses to reach the destination network). A router running BGP4 is known as a BGP speaker. This BGP speaker forms a peering relationship with another BGP speaker in order to exchange routing information.

The Switch uses two types of peers: external peers and internal peers. In addition to these two types, a third category called a member AS peer is used when configuring a confederation. For details about member AS peers, see *12.4.10 Confederations*.

Use the appropriate peer type for your network configuration. The following figure shows internal and external peers.

Figure 12-1: Internal and external peers



Legend: Routers 1, 2, and 3: Internal BGP4 speaker
Routers 6, 7, and 8: External BGP4 speaker
Routers 4 and 5: Internal non-BGP4 speaker
INT: Internal peer
EXT: External peer
#: IGP is enabled.

(1) External peer

External peer relations are formed between BGP speakers that belong to different ASs. The interface address of the directly connected interface is used as the IP address for peering. The peers can be connected by a different address (for example, the device address) by using the `neighbor ebgp-multihop` configuration command.

In *Figure 12-1: Internal and external peers*, Routers 1 and 6, Routers 2 and 7, and Routers 3 and 8 are in external peer relationships.

(2) Internal peer

Internal peer relations are formed between BGP speakers in the same AS. BGP4 uses TCP (port 179) to establish connections between peers. Therefore, although there is no requirement for all BGP speakers to be physically fully meshed, internal peers must be logically fully meshed with each BGP speaker within the AS. This is because an internal peer does not announce received routing information to other internal peers. When route reflection or confederation is used, this condition is relaxed.

In *Figure 12-1: Internal and external peers*, Routers 1 and 2, Routers 1 and 3, and Routers 2 and 3 are in internal peer relationships.

(3) Peering using the device address

In the Switch, the state of a particular physical interface (TCP connection) can affect peering. Such effects can be eliminated by using the IP address of the loopback interface (the device address) as the IP address of an external or internal peer.

For example, suppose that the interface IP address is used for connectivity between the internal peers Router 1 and Router 2 in *Figure 12-1: Internal and external peers*. If the interface between these two routers fails, a peering relationship cannot be established. In contrast, by using the device address as an internal peer IP address, internal peering relations can be established between Routers 1 and 2 via Routers 4 and 5 even though the interface between Routers 1 and 2 is down.

Notes on using the device address

Before peers can use a device address for interconnectivity, they must learn information about routes to that address via static routing or by an IGP (such as RIP or OSPF). The Switch handles the device address as directly connected routing information.

Notes on internal peering via a router that is not a BGP speaker

When routing information is reported via an internal peer that is not a BGP speaker (for example, via the route from Router 2 to Router 3), the non-BGP speaker must have already learned the routing information via an IGP. This is necessary to prevent IP packets sent from the notified BGP speaker from being discarded by a non-BGP router that has not learned the route to the originating router. For example, in *Figure 12-1: Internal and external peers*, this would prevent an IP packet from Router 3 from being discarded by Router 5.

12.1.3 Route selection

From the routing information to a given destination learned by the various protocols, the Switch selects the most appropriate route in accordance with each protocol's independent route selection criteria. If the various protocols generate more than one route to the same destination, the distances are compared and the route with the highest priority is selected.

In BGP4, the best route is selected from the multiple routes to the same destination learned using BGP4, following the steps in the table below. If multiple routes to a given destination are found to exist after route selection by each protocol (RIP, OSPF, and static), their distances are compared and the route with the highest priority is entered in the routing table.

For route selection in a confederation, see *12.4.10 Confederations*.

Table 12-2: Route selection priority

Priority	Description
High	Selects the route with the greatest weight.
↑	Selects the route with the largest `LOCAL_PREF` attribute value.
	Selects the route whose `AS_PATH` attribute has the smallest number of ASs.#1
	Selects by the `ORIGIN` attribute value, preferring `IGP`, `EGP`, and `Incomplete`, in that order.
	Selects the route with the smallest `MED` attribute value.#2
	Selects a route learned by an external peer in preference to a route learned by an internal peer.
	Selects the route with the closest next hop (the smallest metric of the IGP routes used when resolving the next-hop address).
	Selects the route whose peer has the smallest BGP identifier (router ID). For a route having the `ORIGINATOR_ID` attribute, however, the `ORIGINATOR_ID` attribute values are compared instead of the peers' BGP identifiers.#3
↓	Selects the route with the smallest `CLUSTER_LIST` attribute length.#4
Low	Selects the route whose learning source peer has the smallest address.#3

#1

The `AS_SET` path type of the `AS_PATH` attribute is counted as one AS.

#2

Route selection by `MED` attribute value applies only to redundant routes learned from the same neighboring AS. However, redundant routes learned from different neighboring ASs can also be compared by specifying the `bgp always-compare-med` configuration command.

#3

If the routes received from an external peer have different peer BGP identifiers (router IDs), the route selected at the previous step is used, and route selection based on peer BGP identifier and learning source peer address is skipped. However, routes that have different peer BGP identifiers can also be considered by specifying the `bgp bestpath compare-routerid` configuration command.

#4

When a route does not have the `CLUSTER_LIST` attribute, it is compared assuming a `CLUSTER_LIST` attribute length of 0.

Weights and the BGP attributes involved in route selection (the `LOCAL_PREF`, `AS_PATH`, `ORIGIN`, `MED`, and `NEXT_HOP` attributes) are explained below.

(1) Weight

A weight is a valuation applied to routes per learning source peer. It is set using the `neighbor weight` configuration command. Routes with higher weights are preferred.

Weights in the range from 0 to 255 can be used in the Switch. The default is 0.

(a) Changing a weight

Using the `neighbor weight` configuration command, you can change the weight of routes learned from a peer.

(2) LOCAL_PREF attribute

The `LOCAL_PREF` attribute is reported among routers in the same AS. When there is more than one route to a destination network, the `LOCAL_PREF` attribute indicates the preferred route. Routes with higher `LOCAL_PREF` values are preferred.

`LOCAL_PREF` attribute values in the range from 0 to 65535 can be used in the Switch. The default is 100.

(a) Changing the LOCAL_PREF attribute default

Using the `bgp default local-preference` configuration command, you can change the `LOCAL_PREF` attribute value of routing information imported to the Switch from an external peer.

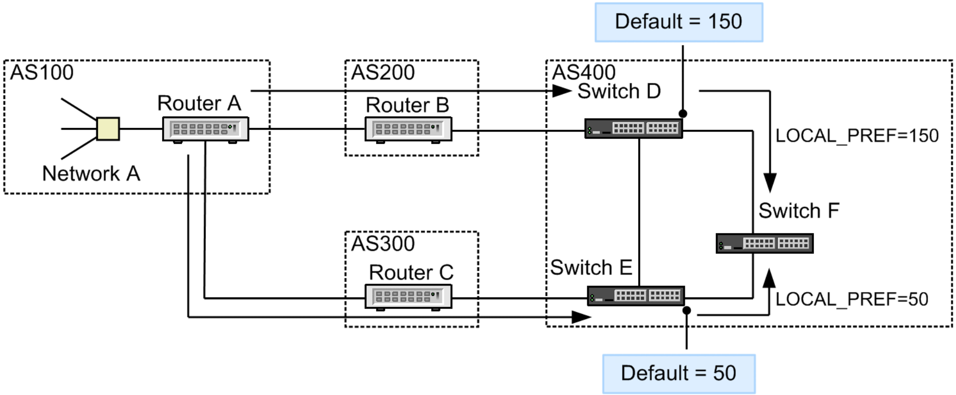
(b) Changing the LOCAL_PREF attribute on a per-filter basis

Using the `set local-preference` configuration command in combination with a learned route filter and advertised route filter, you can change the `LOCAL_PREF` attribute value of routing information imported to the Switch or reported to other switches.

(c) Example of route selection by LOCAL_PREF attribute

The following figure shows route selection using the `LOCAL_PREF` attribute.

Figure 12-2: Route selection by LOCAL_PREF attribute



In the above figure, AS400 receives routing information for network A from AS200 and AS300.

Assume that the LOCAL_PREF attribute value of Switch D is set to 150 and that the LOCAL_PREF attribute value of Switch E is set to 50. Thus, a LOCAL_PREF value of 150 would apply when Switch D notifies Switch F of the routing information received from AS200, and a LOCAL_PREF value of 50 would apply when Switch E notifies Switch F of the routing information received from AS300. Of the two sets of network-A routing information destined for Switch F, the routing information from Switch D has a higher LOCAL_PREF value than the routing information from Switch E. Therefore, the routing information from Switch D via AS200 is selected.

(3) ORIGIN attribute

The ORIGIN attribute indicates the source of the routing information. The following table describes the attribute values.

Table 12-3: ORIGIN attribute

ORIGIN attribute	Description
IGP	The route was generated within the AS.
EGP	The route was learned via EGP.
Incomplete	The route was learned by some other means.

When there are multiple paths to a given destination, the ORIGIN attribute is selected in the order IGP, EGP, and then Incomplete.

(a) Changing the ORIGIN attribute value

Using the set origin configuration command in combination with a route filter, you can change the ORIGIN attribute value of routing information imported to the Switch or reported to other switches.

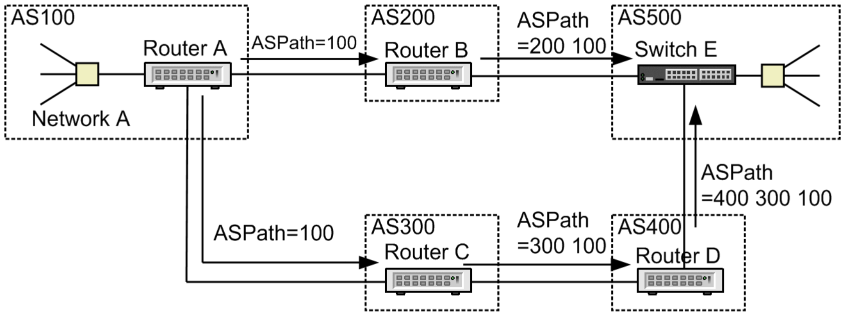
(4) AS_PATH attribute

The AS_PATH attribute is a list of AS numbers that a route traverses to reach the destination network. When routing information is advertised to another AS, the local AS number is added to the AS_PATH attribute in that routing information. Using learning filter information and advertising filter information in combination with the set as-path prepend count configuration command, you can add more than one local AS number to the AS_PATH attribute. This is useful for selecting a specific path when there are a number of routes to a given destination network.

(a) Example of route selection by AS_PATH attribute

The following figure shows route selection using the AS_PATH attribute.

Figure 12-3: Route selection by the AS_PATH attribute

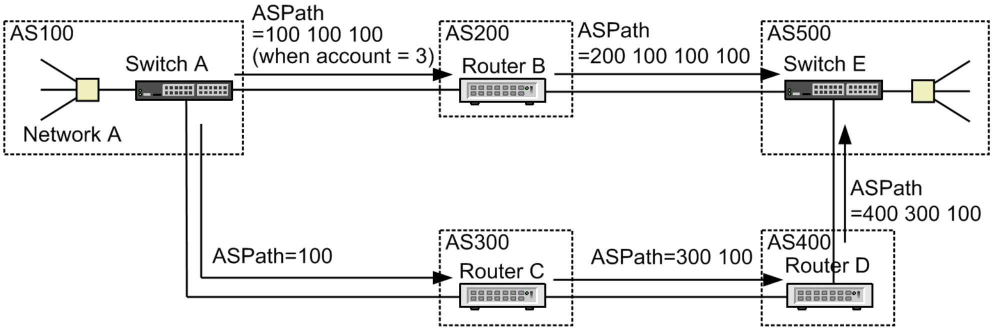


Suppose that Router A reports information about network A in its local AS to Switch E. When the routing information reaches AS500, it will have an AS_PATH attribute of "200 100". However, if the routing information is instead directed through AS300 and AS400, the AS_PATH attribute when the routing information reaches AS500 will be "400 300 100". Therefore, Switch E determines that the route via AS200 has the smaller AS_PATH attribute and selects that route.

(b) Route selection when using the set as-path prepend count command

The following figure shows an example of the `set as-path prepend count` configuration command.

Figure 12-4: Example of using the set as-path prepend count command



In the above figure, to force Switch E to select the route from Router A via AS300 and AS400, you need to add multiple local AS numbers to the `AS_PATH` attribute of the routing information sent from Router A to AS200. For example, if you add three local AS numbers, the `AS_PATH` attribute when the routing information reaches AS500 via AS200 is "200 100 100 100". Therefore, Switch E determines that the route through AS300 and AS400 has the smaller `AS_PATH` attribute and selects that route.

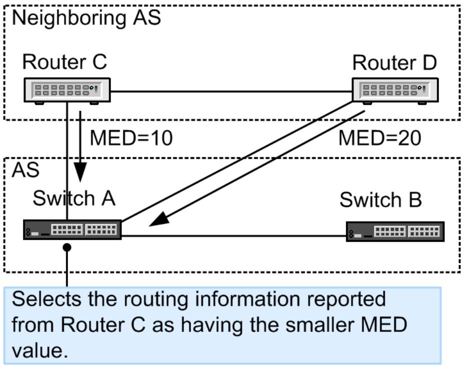
(5) MED attribute

The `MED` attribute determines the priority of multiple BGP4 routes to a given destination learned from the same neighboring AS. Routes with lower `MED` attribute values are preferred. Using the `bgp always-compare-med` configuration command, you can use `MED` attribute values for priority selection among BGP4 routes learned from different neighboring ASs.

(a) Example of route selection by MED attribute

The following figure shows route selection using the `MED` attribute.

Figure 12-5: Route selection by MED attribute


Neighboring AS
Router C
Router D
MED=10
MED=20
AS
Switch A
Switch B
Selects the routing information reported from Router C as having the smaller MED value.

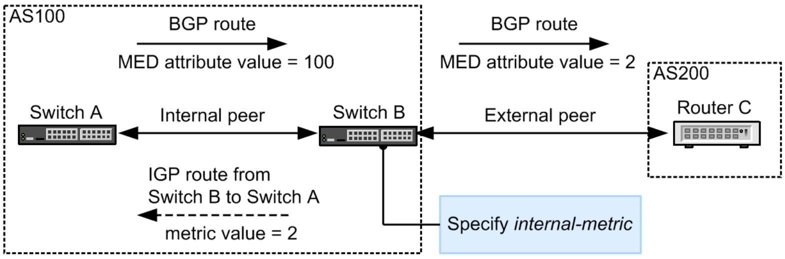
Suppose that Router C advertises routing information for a given destination network with a `MED` attribute value of 10, and Router D advertises routing information to the same network with a `MED` attribute value of 20. Switch A will therefore select the routing information reported from Router C as the preferred route to the destination network.

(b) Changing the MED attribute value

Using learned filter information and advertised filter information in combination with the `set metric` configuration command, you can change the `MED` attribute value of routing information imported to the Switch or reported to other switches.

By specifying `internal` in `set metric-type`, you can set the metric of the IGP route used for next-hop resolution as the `MED` attribute value of the BGP4 route to be advertised. The following figure shows an example of using `set metric-type internal`.

Figure 12-6: Example of using the set metric-type internal command



In the above figure, Switches A and B are internal peers. Suppose that you want to set the metric 2 of the IGP route from Switch B to Switch A as the `MED` attribute value when the BGP4 routing information that was reported from Switch A with a `MED` attribute value of 100 is advertised by Switch B to Router C. You can do so by specifying the `set metric-type internal` configuration command on Switch B.

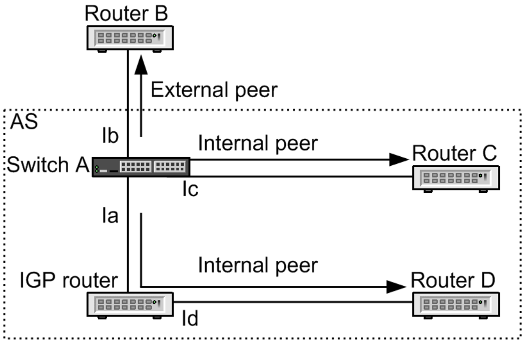
(6) NEXT_HOP attribute

The `NEXT_HOP` attribute is the IP address of the next hop used to reach a given destination network. In the Switch, the local IP address used for peering is set in the `NEXT_HOP` attribute when reporting routing information to an external peer. When reporting routing information to an internal peer and member AS peer, the `NEXT_HOP` attribute is not modified.

(a) Example of setting the NEXT_HOP attribute

The following figure shows examples of setting the `NEXT_HOP` attribute of the reported routing information when advertising a route learned from a BGP4 peer.

Figure 12-7: Examples of setting the NEXT_HOP attribute of the reported routing information when advertising a route learned from a BGP4 peer



- Routing information destined for external peer Router B

  The `NEXT_HOP` attribute is the interface address Ib on the A side of the interface between Switch A and Router B.

- Routing information destined for internal peer Router C

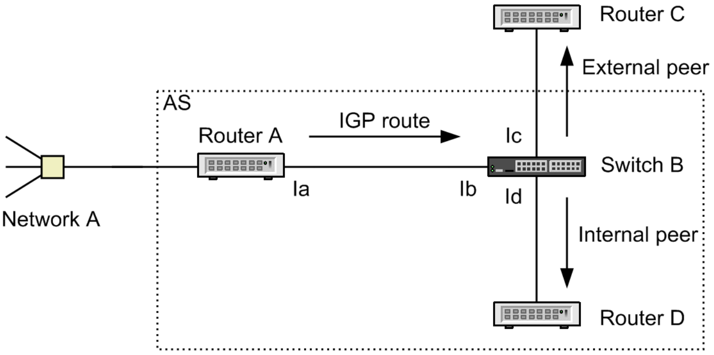
  The `NEXT_HOP` attribute is the attribute set in the routing information received from Router B.

- Routing information destined for internal peer Router D

  The `NEXT_HOP` attribute is the attribute set in the routing information received from Router B.

The following figure shows examples of setting the `NEXT_HOP` attribute of the reported routing information when advertising an IGP route via BGP4.

Figure 12-8: Examples of setting the NEXT_HOP attribute of the reported routing information when advertising an IGP route via BGP4



- Routing information destined for external peer Router C

  The `NEXT_HOP` attribute is the interface address Ic on the B side of the interface between Switch B and Router C.

- Routing information destined for internal peer Router D

  The `NEXT_HOP` attribute is the interface address Ia of Router A, which is the next-hop address to network A resolved to the IGP route.

(b) Modifying the NEXT_HOP attribute

Use the following configuration commands to modify the `NEXT_HOP` attribute in the Switch:

- `neighbor next-hop-self` command

  Changes the `NEXT_HOP` attribute that is used when routing information received from a BGP4 peer is advertised to another BGP4 peer; the attribute is changed to the peering address of the local router. This command does not apply when using route reflection or when advertising an IGP route to an internal peer by BGP4.

- `neighbor always-nexthop-self` command

  Changes the `NEXT_HOP` attribute that is used when routing information is advertised to an internal peer; the attribute is changed to the peering address of the local router. This command also applies when using route reflection or when advertising an IGP route by BGP4.

- `neighbor set-nexthop-peer` command

  Changes the `NEXT_HOP` attribute of learned routing information to the remote peer address used for peering.

(c) Resolving the NEXT_HOP attribute

When BGP4 routing information has been learned from an internal peer, the path to reach an address indicated by the `NEXT_HOP` attribute resolves to an IGP route, a static route, or a directly connected route. Among the routes that can reach the next hop of the BGP4 route, the route having the longest destination mask length is selected and the path of that route is used as the BGP4 route. By using the `bgp nexthop` configuration command, you can specify the protocol type and prefix of the route to be used in resolving the `NEXT_HOP` attribute.

If the route to which the next hop resolves is a static route and the `noinstall` parameter is specified, the BGP4 route is suppressed.

12.1.4 BGP4 functionality for VRFs

(1) Overview

BGP4 operates independently within networks logically divided by the VRF functionality. Note that a peer connection between different VRFs is not possible.

(2) Notes on using the BGP4 functionality in a VRF

On the Switch, a route imported from a different VRF or the global network inherits the `PATH` attribute from the source route. Therefore, if a target route is advertised from the Switch, a route loop might be detected on neighboring devices.

1. Notes on when different VRFs or global networks use the same AS number

   Be careful when you import a route between VRFs or global networks that use the same AS number. If the target route is advertised to the import-source VRF or global network, the route is not treated as a valid route because an AS loop is detected on neighboring devices. The Switch provides the `neighbor as-override` configuration command that overwrites the first AS number set for the `AS_PATH` attribute of the VRF or the global network with the AS number of the Switch. If you use BGP4 for a connection between VRFs or global networks that have the same AS number, be sure to use this command.

   Note that the `neighbor as-override` configuration command cannot resolve AS loops in neighboring devices when the VRFs or global networks are not directly connected to the Switch and using the same AS number. If the Switch is used as a neighboring device, you can use the `neighbor permit-asloop` configuration command, which treats an AS loop route as a valid route. Use this command when using the same AS number in VRFs or global networks. Use this command in carefully designed networks because its use can increase the risk of a route loop.

2. Notes on when different VRFs or global networks use the same router ID or cluster ID (route reflection)

   When different VRFs or global networks use the same router ID (originator ID) or when the route reflectors in different VRFs or global networks use the same cluster ID, the routes are not treated as valid routes because the route reflectors detect a loop. Keep this in mind when designing networks.

12.1.5 Notes on using BGP4

Note the following restrictions when configuring a network using BGP4.

(1) BGP4 restrictions

With some exceptions, the Switch is compliant with RFC 4271 (Border Gateway Protocol 4), RFC 1997 (Communities), RFC 5492 (Capabilities Advertisement), RFC 2918 (Route Refresh Capability), RFC 4456 (Route Reflection), and RFC 5065 (Confederations). Differences are due to the functional limitations of the software, and are described in the table below. The Switch supports only BGP version 4.

Table 12-4: Differences with the RFC

RFC No.	RFC		The Switch
RFC 4271	Path attribute: NEXT_HOP	If the external peer to which the route is being advertised shares a subnet with one of the interfaces of the announcing BGP speaker, the speaker may use the IP address associated with such an interface in the `NEXT_HOP` attribute. This is known as a "first party" `NEXT_HOP` attribute.	The "first party" `NEXT_HOP` attribute is not supported.
		When sending a message to an external peer, and the peer is multiple IP hops away from the speaker (also known as a "multihop EBGP"), the BGP speaker may be configured to propagate the `NEXT_HOP` attribute without modification.	The Switch changes the `NEXT_HOP` attribute to the address of the local router when advertising a route to an external peer.

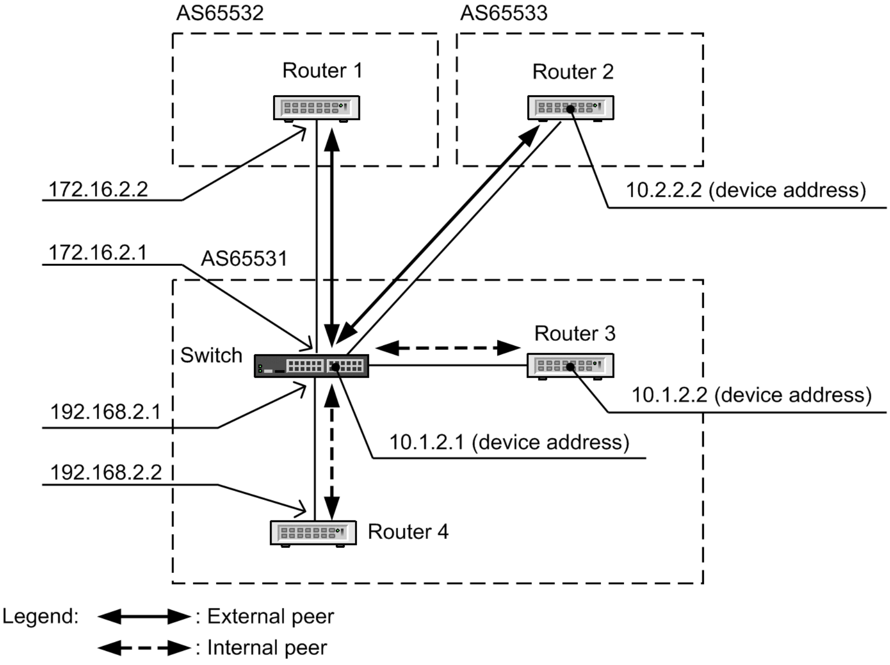
RFC No.	RFC		The Switch
	Path attribute: MULTI_EXIT_DISC	A BGP speaker must implement a mechanism (based on local configuration) that allows the MULTI_EXIT_DISC attribute to be removed from a route. If a BGP speaker is configured to remove the MULTI_EXIT_DISC attribute from a route, then this removal must be done prior to determining the degree of preference of the route and prior to performing route selection.	A mechanism for removing the MULTI_EXIT_DISC attribute from a route is not implemented.
	Connection collision detection	Upon receipt of an OPEN message, the local system must examine all of its connections that are in the OpenConfirm state. The local system also examines connections in an OpenSent state if it knows the BGP identifier of the peer by some means outside the protocol.	Upon receipt of an OPEN message, the Switch examines all connections that are in the OpenSent state or Connect state.
	BGP FSM: Idle state	For a peer that has previously transitioned to the Idle state because of an error, the time between consecutive generations of Start events (if such events are generated automatically) must increase exponentially. The value of the initial timer is 60 seconds. The time is doubled for each consecutive retry.	The initial value of the timer until the peer changes from the Idle to the Start state is from 16 to 36 seconds.
	BGP FSM: Active state	If the transport protocol connection succeeds, the local system clears the Connect Retry timer, and completes initialization. It then sends an OPEN message to its peer, sets its hold timer, and changes its state to Open Sent. A hold timer value of 4 minutes is suggested.	The hold timer is 180 seconds (3 minutes) by default, or, if specified, the value in the configuration.
	Frequency of route advertisement	Min Route Advertisement Interval determines the minimum interval between route advertisements to a particular destination by a BGP speaker. This rate limit applies on a per-destination basis. However, the value of Min Route Advertisement Interval is set on a per-BGP4 peer basis.	Min Route Advertisement Interval is not supported.
		Min AS Origination Interval determines the minimum time that must elapse between successive advertisements of UPDATE messages that report changes within the advertising BGP speaker's own ASs.	Min AS Origination Interval is not supported.
	Jitter	To minimize the likelihood that the distribution of BGP messages by a given BGP speaker will contain peaks, jitter should be applied to the timers associated with Min AS Origination Interval, Keep Alive, and Min Route Advertisement Interval.	Jitter is not applied.
	Route summarization	Routes that have different MULTI_EXIT_DISC attributes must not be aggregated.	Routes that have different MULTI_EXIT_DISC attributes are aggregated.

RFC No.	RFC		The Switch
		When aggregating routes that have different NEXT_HOP attributes, the NEXT_HOP attribute of the aggregated route must identify an interface on the BGP speaker that performs the aggregation.	The NEXT_HOP attribute is not set for an aggregated route.
	BGP timers	The suggested value for the Connect Retry timer is 120 seconds.	The Connect Retry timer value is a variable (from 16 to 148 seconds) that changes according to the Connect Retry count.
		The suggested value for the Hold Time attribute is 90 seconds.	The default Hold Time value is 180 seconds. The value of Hold Time in the configuration is used if specified.
		The suggested value for the Keep Alive timer is 30 seconds.	The default value of the Keep Alive timer is one-third of the Hold Time value. The value of Keep Alive in the configuration is used if specified.
		Two optional timers (DelayOpenTimer and IdleHoldTimer) may be supported by BGP.	DelayOpenTimer and IdleHoldTimer are not supported.
RFC 5065	All BGP speakers participating as members of a confederation must recognize the AS_CONFED_SET and AS_CONFED_SEQUENCE path types.		AS_CONFED_SET is not supported. Upon receipt of a route containing AS_CONFED_SET, the Switch ignores that path type.

12.2 Configuration of basic functionality

This section describes how to configure the basic BGP4 functionality, based on the following configuration example.

Figure 12-9: Connectivity example



- Between the Switch and Router 1: Interface address
- Between the Switch and Router 2: Device address
- Between the Switch and Router 3: Device address
- Between the Switch and Router 4: Interface address

12.2.1 List of configuration commands

The following tables describe the configuration commands for BGP4 basic functionality, and operation commands.

Table 12-5: List of configuration commands

Command name	Description
address-family ipv4	Configures settings for each VRF. Switches to `config-router-af (ipv4 vrf)` mode.
bgp always-compare-med	Enables comparison of the `MED` attribute of routes learned from different ASs.
bgp bestpath compare-routerid#1	Enables selection among routes learned from external peers based on the peer's BGP identifier (router ID).
bgp default local-preference	Sets the default of the `LOCAL_PREF` attribute of routes advertised by BGP4.
bgp nexthop	Specifies the route map (`route-map`) to be used in resolving the next hop of a BGP4 route.
bgp router-id#1	Specifies the identifier of the local router.
default-information originate	Advertises a default route to all peers.

Command name	Description
default-metric	Sets the default of the MED attribute of routes advertised by BGP4.
disable#1	Disables BGP4/BGP4+.
distance bgp	Sets the distance of a route learned via BGP4.
neighbor description	Sets a supplementary description of a peer.
neighbor ebgp-multihop	Allows connections to external peers that are not directly connected on the interface and to member AS peers.
neighbor next-hop-self	Allows the NEXT_HOP attribute when advertising a route learned by a BGP4 peer to BGP4 peers to be changed to the address used for peering on the local side.
neighbor remote-as	Configures a BGP4/BGP4+ peer.
neighbor remove-private-as	Specifies the removal of private AS numbers when advertising to BGP4 peers.
neighbor shutdown	Disables a peer connection.
neighbor soft-reconfiguration	Stores routes that have been suppressed by the input policy.
neighbor timers	Sets the KEEPALIVE message sending interval and hold timer value used in a peer connection.
neighbor update-source	Sets the device address as the local address to be used for peering.
neighbor weight	Sets the weighting of routes learned from a peer.
router bgp#1	Configures information about the behavior of the BGP4/BGP4+ routing protocol.
timers bgp#1	Sets the KEEPALIVE message sending interval and hold timer value to be applied to all peers.
distribute-list in (BGP4)#2	Specifies the route filter to be used as BGP4 learned route filter conditions.
distribute-list out (BGP4)#2	Specifies the route filter to be used as BGP4 advertised route filter conditions.
neighbor in (BGP4)#2	Specifies the route filter to be used as the learned route filter conditions for a specific BGP4 peer.
neighbor out (BGP4)#2	Specifies the route filter to be used as the advertised route filter conditions for a specific BGP4 peer.
redistribute (BGP4)#2	Specifies the protocol of learned routes advertised by BGP4.

#1

Common command for BGP4 and BGP4+ (IPv6) peers.

#2

See *14. Route Filters (IPv4 and IPv6)* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

Table 12-6: List of operation commands used in BGP4+ configuration

Command name	Description
clear ip bgp	1. When `* in` is specified in the parameter: Applies the latest route filtering settings to BGP4 learned route filtering. Requests re-advertisement of BGP4 routes to all BGP4 peers. 2. When `* out` is specified in the parameter Applies the latest route filtering settings to BGP4 advertised route filtering. Applies the `neighbor remove-private-as` setting to the operation. Re-advertises BGP4 routes to all BGP4 peers. 3. When `* both` is specified in the parameter Applies the latest route filtering settings to BGP4 learned route filtering and advertised route filtering. Applies the `neighbor remove-private-as` setting to the operation. Requests re-advertisement of BGP4 routes to all BGP4 peers and redistributes BGP4 routes to all BGP4 peers. 4. When `*` is specified in the parameter Disconnects all BGP4 peers.

12.2.2 Overview of configuration

1. Configure the IPv4 interfaces in advance.
2. Set the address of the local device in the loopback interface in advance.
3. Configure the BGP4 peers.
4. Set the BGP4 route learning policy.
5. Set the BGP4 route advertising policy.
6. Configure learned route filtering.
7. Configure advertised route filtering.
8. Set the learned route filter conditions.
9. Set the advertised route filter conditions.
10. Apply the filters to BGP4+ operation.

Notes

If you set up BGP4 peering without first configuring the route filters, the Switch will begin performing route learning and advertising as soon as the peer relations are established. To prevent unintended route learning and advertising, first set the `disable` configuration command to disable BGP4 operation, and then set the `neighbor remote-as` configuration command. To start BGP4 after configuring the route filters, use the `no` form of the `disable` configuration command.

12.2.3 Configuring BGP4 peers

Points to note

To configure a peer, first set the address and AS number on the remote side using the `neighbor remote-as` command, and then set other information about the peer.

Command examples

1. `(config)# router bgp 65531`

   Applies BGP4/BGP4+ to the routing protocol. Specify the AS number (65531) of the autonomous system to which the local router belongs for the parameter.

2.
```
(config-router)# bgp router-id 192.168.1.100
```
Sets the identifier of the local router (192.168.1.100).

3.
```
(config-router)# neighbor 172.16.2.2 remote-as 65532
```
Configures an external peer (remote peer address: 172.16.2.2; AS number: 65532).

4.
```
(config-router)# neighbor 10.2.2.2 remote-as 65533
```
Configures an external peer (remote peer address:10.2.2.2; AS number: 65533).

5.
```
(config-router)# neighbor 10.2.2.2 ebgp-multihop
```
Disallows use of the interface address of the directly connected interface as the peering address.

6.
```
(config-router)# neighbor 10.2.2.2 update-source loopback 0
```
Sets the device address as the peering address of the local router.

7.
```
(config-router)# neighbor 192.168.2.2 remote-as 65531
```
Configures an internal peer (remote peer address: 192.168.2.2).

8.
```
(config-router)# neighbor 10.1.2.2 remote-as 65531
```
Configures an internal peer (remote peer address: 10.1.2.2).

9.
```
(config-router)# neighbor 10.1.2.2 update-source loopback 0
```
Sets the device address as the peering address of the local router.

12.2.4 Configuring the BGP4 route learning policy

Points to note

The example below shows how to set learned route preferences on a per-peer basis by setting the weight of each peer.

Command examples

1.
```
(config-router)# bgp always-compare-med
```
For the purpose of route selection, allows comparison of MED attribute values in routing information received from different ASs.

2.
```
(config-router)# neighbor 172.16.2.2 weight 20
(config-router)# neighbor 10.2.2.2 weight 20
```

```
(config-router)# neighbor 10.1.2.2 weight 10
(config-router)# neighbor 192.168.2.2 weight 10
```

Specifies a weight for the routes learned from each peer.

Gives priority to routes learned from an external peer over routes learned from an internal peer.

12.2.5 Configuring the BGP4 route advertising policy

Points to note

The example below shows how to set the BGP4 path attributes to be used in route selection at the advertisement destination router.

Command examples

1. `(config-router)# default-metric 100`

   Sets 100 as the `MED` attribute value of advertised routes.

2.
```
(config-router)# bgp default local-preference 80
(config-router)# exit
```

   Sets 80 as the `LOCAL_PREF` attribute value advertised to internal peers.

12.2.6 Configuring learned route filtering

Points to note

The example below shows how to set the priority of learned BGP4 routes by using `route-map` and specifying the conditions and settings.

Command examples

1.
```
(config)# ip prefix-list EXT_IN seq 10 permit 10.10.0.0/16
(config)# route-map SET_LOCPREF_IN permit 10
(config-route-map)# match ip address prefix-list EXT_IN
(config-route-map)# set local-preference 120
(config-route-map)# exit
(config)# route-map SET_LOCPREF_IN permit 20
(config-route-map)# exit
```

   Sets 120 in the `LOCAL_PREF` attribute when the destination network is 10.10.0.0/16.

2.
```
(config)# ip as-path access-list 10 permit "_65529$"
(config)# route-map SET_ASPREPEND_IN permit 10
(config-route-map)# match as-path 10
(config-route-map)# set as-path prepend count 1
(config-route-map)# exit
(config)# route-map SET_ASPREPEND_IN permit 20
```

```
(config-route-map)# exit
```

Prepends one AS to the AS array when the end of the AS array of the `AS_PATH` attribute is 65529.

3.
```
(config)# ip prefix-list INT_IN_1 seq 10 permit 172.20.0.0/16
(config)# route-map SET_ORIGIN_IN permit 10
(config-route-map)# match ip address prefix-list INT_IN_1
(config-route-map)# set origin incomplete
(config-route-map)# exit
(config)# route-map SET_ORIGIN_IN permit 20
(config-route-map)# exit
```

Sets `INCOMPLETE` in the `ORIGIN` attribute when the destination network is 172.20.0.0/16.

4.
```
(config)# ip prefix-list INT_IN_2 seq 10 permit 172.30.0.0/16
(config)# route-map SET_MED_IN permit 10
(config-route-map)# match ip address prefix-list INT_IN_2
(config-route-map)# set metric 100
(config-route-map)# exit
(config)# route-map SET_MED_IN permit 20
(config-route-map)# exit
```

Sets 100 in the `MED` attribute when the destination network is 172.30.0.0/16.

12.2.7 Configuring advertised route filtering

Points to note

The example below shows how to set the priority of advertised BGP4 routes by using `route-map` and specifying the conditions and settings.

Command examples

1.
```
(config)# ip prefix-list MY_NET_1 seq 10 permit 192.169.10.0/24
(config)# ip prefix-list MY_NET_2 seq 10 permit 192.169.20.0/24
(config)# route-map SET_EXT_OUT permit 10
(config-route-map)# match ip address prefix-list MY_NET_1
(config-route-map)# set metric 120
(config-route-map)# exit
(config)# route-map SET_EXT_OUT permit 20
(config-route-map)# match ip address prefix-list MY_NET_2
(config-route-map)# exit
```

Sets 120 in the `MED` attribute when the destination network is 192.169.10.0/24.

Allows route advertisements to be sent also to destination network 192.169.20.0/24.

12.2.8 Configuring the learned route filter conditions

Points to note

The example below shows how to apply learned route filtering on a per-peer basis by specifying the filters to be applied in the `neighbor in` command.

Command examples

1.
```
(config)# router bgp 65531
(config-router)# neighbor 172.16.2.2 route-map SET_LOCPREF_IN
in
```

Sets 120 in the `LOCAL_PREF` attribute of routes to the destination network 10.10.0.0/16 that were learned from the peer (remote peer address: 172.16.2.2) and gives such routes higher priority than routes learned from other peers.

2.
```
(config-router)# neighbor 10.2.2.2 route-map SET_ASPREPEND_IN
in
```

Prepends one AS in the AS array when the end of the AS array of the `AS_PATH` attribute learned from the peer (remote peer address: 10.2.2.2) is 65529, and gives such routes lower priority than routes learned from other peers.

3.
```
(config-router)# neighbor 10.1.2.2 route-map SET_ORIGIN_IN in
```

Sets `INCOMPLETE` in the `ORIGIN` attribute of routes to the destination network 172.20.0.0/16 that were learned from the peer (remote peer address: 10.1.2.2) and gives such routes lower priority than routes learned from other peers.

4.
```
(config-router)# neighbor 192.168.2.2 route-map SET_MED_IN in
```

Sets 100 for the `MED` attribute of routes to the destination network 172.30.0.0/16 that were learned from the peer (remote peer address: 192.168.2.2).

12.2.9 Configuring the advertised route filter conditions

Points to note

The example below shows how to apply the same advertised route filter to all peers by specifying the filter in the `distribute-list out` command.

Command examples

1.
```
(config-router)# distribute-list route-map SET_EXT_OUT out
(config-router)# exit
(config)# exit
```

Advertises routes to the destination networks 192.169.10.0/24 and 192.169.20.0/24 to all external peers.

12.2.10 Applying filters

Points to note

The example below shows how to apply the route filters set as the learned route and advertised route filter conditions, by using the `clear ip bgp` operation command.

Command examples

1. `# clear ip bgp * both`

   Applies the learned route filter and advertised route filter to BGP4+ operation.

Notes

The `clear ip bgp` operation command (with `* in`, `* out`, or `* both` specified) applies the new route filtering settings and implements the route refresh capability (see *12.4.5 Route refresh capability*). If route refresh capability has not been negotiated, no route refresh requests are made in order to relearn changed routes, although the route filter changes are applied.

12.2.11 Configuring BGP4 for a VRF

Points to note

The example below shows how to configure the BGP4 functionality for a VRF in `config-router-af` (`ipv4 vrf`) mode.

Command examples

1. `(config)# router bgp 65531`

   Specifies the local AS number (65531).

2. `(config-router)# address-family ipv4 vrf 10`

   Switches to `config-router-af` (`ipv4 vrf`) mode for VRF 10.

3. `(config-router-af)# bgp router-id 192.168.1.100`

   Specifies the identifier of the local router (192.168.1.100).

4. `(config-router-af)# neighbor 10.1.2.2 remote-as 65531`

   Specifies an internal peer (remote peer address: 10.1.2.2).

5. `(config-router-af)# neighbor 172.16.2.2 remote-as 65532`

   Specifies an internal peer (remote peer address: 172.16.2.2).

12.3 Operation for basic functionality

12.3.1 List of operation commands

The following table describes the operation commands for basic BGP4+ functionality.

Table 12-7: List of operation commands

Command name	Description
show ip route	Shows routing information stored in the routing table.
clear ip route	Clears the IPv4 forwarding entries stored in the Switch and re-registers the routing entries.
show ip bgp	Shows information related to the BGP4 protocol.
clear ip bgp	Clears BGP4 sessions or BGP4-related information, or filters inbound or outbound routes using new BGP filter information. Also reconnects a BGP4 session that was terminated because a peer exceeded the maximum number of BGP4 routes that can be learned from a particular device.
show ip vrf	Shows the IPv4 information of a VRF.
show processes cpu unicast#	Shows the CPU usage of a unicast routing program.
restart unicast#	Restarts the unicast routing program.
dump protocols unicast#	Outputs the trace information and control table information collected by the unicast routing program to a file.
erase protocol-dump unicast#	Deletes the file of trace information and control table information generated by the unicast routing program.

#

See *8. Routing Protocol Common to IPv4 and IPv6* in the manual *Operation Command Reference Vol. 2 For Version 11.10.*

12.3.2 Checking peer type and connectivity

The figures below show the execution results for the connectivity example in *Figure 12-9: Connectivity example*. To view peer connection information, use the `show ip bgp` operation command with the `neighbors` parameter specified. To view detailed information, specify the `detail` parameter.

Figure 12-10: Results of executing the show ip bgp command (with the neighbors parameter specified)

```
> show ip bgp neighbors
Date 20XX/10/18 22:45:55 UTC
Peer Address      Peer AS  Local Address     Local AS  Type      Status
10.1.2.2          65531    10.1.2.1          65531     Internal  Established
192.168.2.2       65531    192.168.2.1       65531     Internal  Established
10.2.2.2          65533    10.1.2.1          65531     External  Established
172.16.2.2        65532    172.16.2.1        65531     External  Established
```

Figure 12-11: Results of executing the show ip bgp command (with the detail parameter specified)

```
> show ip bgp neighbors detail
Date 20XX/10/17 15:52:14 UTC
BGP Peer: 10.1.2.2        , Remote AS: 65531
Remote Router ID: 10.1.2.102
    BGP Status: Established         HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/10/17 15:51:00
    BGP Version: 4                  Type: Internal
    Local Address: 10.1.2.1         Local AS: 65531
    Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:52:00  Last Keep Alive Received: 15:52:00
    BGP Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                 0          0          2          4
    BGP Capability Negotiation: <IPv4-Uni Refresh Refresh(v)>
      Send   : <IPv4-Uni Refresh Refresh(v)>
      Receive: <IPv4-Uni Refresh Refresh(v)>
    Password: UnConfigured
BGP Peer: 192.168.2.2    , Remote AS: 65531
Remote Router ID: 192.168.1.102
    BGP Status: Established         HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/10/17 15:50:43
    BGP Version: 4                  Type: Internal
    Local Address: 192.168.2.1      Local AS: 65531
    Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:43  Last Keep Alive Received: 15:51:43
    BGP Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                 0          0          2          4
    BGP Capability Negotiation: <IPv4-Uni Refresh Refresh(v)>
      Send   : <IPv4-Uni Refresh Refresh(v)>
      Receive: <IPv4-Uni Refresh Refresh(v)>
    Password: UnConfigured
BGP Peer: 10.2.2.2       , Remote AS: 65533
Remote Router ID: 10.2.2.102
    BGP Status: Established         HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/10/17 15:50:30
    BGP Version: 4                  Type: External
    Local Address: 10.1.2.1         Local AS: 65531
    Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:30  Last Keep Alive Received: 15:51:30
    BGP Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                 0          0          2          4
    BGP Capability Negotiation: <IPv4-Uni Refresh>
      Send   : <IPv4-Uni Refresh Refresh(v)>
      Receive: <IPv4-Uni Refresh Refresh(v)>
    Password: UnConfigured
BGP Peer: 172.16.2.2     , Remote AS: 65532
Remote Router ID: 172.16.1.102
    BGP Status: Established         HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/10/17 15:49:35
    BGP Version: 4                  Type: External
    Local Address: 172.16.2.1       Local AS: 65531
    Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:35  Last Keep Alive Received: 15:51:35
    BGP Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                 0          0          3          5
    BGP Capability Negotiation: <IPv4-Uni Refresh Refresh(v)>
      Send   : <IPv4-Uni Refresh Refresh(v)>
      Receive: <IPv4-Uni Refresh Refresh(v)>
    Password: UnConfigured
>
```

12.3.3 Checking the BGP4 route selection result

To check the BGP4 route selection result, use the `show ip bgp` operation command.

Figure 12-12: Results of executing the show ip bgp command

```
> show ip bgp
Date 20XX/10/18 22:44:23 UTC
Local AS: 65531, Local Router ID: 192.168.1.100
Status Codes: d dampened, * valid, > active, S Stale, r RIB failure
```

```
Origin Codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop      MED   LocalPref Weight Path
*> 10.10/16         172.16.2.2    -     120       20     65532 65528 i       ...1
*  10.10/16         10.2.2.2      -     80        20     65533 65533 65529 i...2
*  10.10/16         10.1.2.2      -     80        10     65534 i             ...3
*> 10.20/16         172.16.2.2    -     80        20     65532 65528 i       ...4
*  10.20/16         10.2.2.2      -     80        20     65533 65533 65529 i...5
*> 172.20/16        192.168.2.2   -     100       10     65530    i          ...6
*  172.20/16        10.1.2.2      -     100       10     65534 65530 ?       ...7
*> 172.30/16        10.1.2.2      -     100       10     65534 i             ...8
*  172.30/16        192.168.2.2  100    100       10     65530 i             ...9
*> 192.168.10/24    10.1.2.2      -     100       10     65534 i            ...10
*  192.168.10/24    192.168.2.2   -     100       10     65530 i            ...11
*> 192.169.10/24    192.168.2.2   -     100       10           i            ...12
*> 192.169.20/24    192.168.2.2   -     100       10           i            ...13
```

1 to 3: Route selection for destination network 10.10/16

Routes 1 and 2 are preferred by comparing the weights of the three routes, and then 1 is selected by comparing the LOCAL_PREF values of 1 and 2.

4 to 5: Route selection for destination network 10.20/16

Route 4 is selected by comparing the AS_PATH length of each route.

6 to 7: Route selection for destination network 172.20/16

Route 6 is selected by comparing the ORIGIN attribute of each route.

8 to 9: Route selection for destination network 172.30/16

Route 8 is selected by comparing the MED attribute of each route.

10 to 11: Route selection for destination network 192.168.10/24

Route 10 is selected by comparing the peer BGP identifier of each route.

12 to 13: Route selection for destination networks 192.169.10/24 and 192.169.20/24

Route 12 and route 13 are selected because there are no other routes to the destinations.

12.3.4 Checking the content of BGP4 route advertisements

To check the path attributes of advertised BGP4 routes, use the show ip bgp command with the advertised-routes parameter specified.

Figure 12-13: Results of executing the show ip bgp command (with the advertised-routes parameter specified)

```
> show ip bgp advertised-routes
Date 20XX/10/18 22:44:54 UTC
BGP Peer: 10.2.2.2       , Remote AS: 65533
Local AS: 65531, Local Router ID: 192.168.1.100
Origin Codes: i - IGP, e - EGP, ? - incomplete
Network             Next Hop        MED    LocalPref Path
192.169.10/24       192.168.2.2     120    -         65531 i           ...1
192.169.20/24       192.168.2.2     100    -         65531 i
BGP Peer: 172.16.2.2     , Remote AS: 65532
Local AS: 65531, Local Router ID: 192.168.1.100
Origin Codes: i - IGP, e - EGP, ? - incomplete
Network             Next Hop        MED    LocalPref Path
192.169.10/24       192.168.2.2     120    -         65531 i           ...2
192.169.20/24       192.168.2.2     100    -         65531 i
```

1 and 2: The MED attribute (value: 120) is set for the advertised routes.

12.4 Description of extended functionality

12.4.1 BGP4 peer groups

A BGP4 peer group is a way of simplifying settings by grouping peers and performing settings for the group by using the `neighbor` configuration command. Settings performed with the `neighbor` command apply to all peers that belong to the specified peer group. You can also use the `neighbor` command for a specific peer in a peer group. In this case, the individual peer settings override the peer group settings. You can create a peer group consisting of external peers and member AS peers, or internal peers alone, in either BGP4 or BGP4+. Multiple peer groups can be created. A peer can belong to only one peer group. If you assign peers to a different peer group, you must execute the `clear ip bgp * {both| in | out}` operation command to apply the route filtering configured for the new peer group.

12.4.2 Community

By adding the `COMMUNITIES` attribute to routing information, you can restrict the range of route advertisements handled by the Switch.

(1) Community types

The community values supported by the Switch can be divided into two types:

- Values (codes) predefined in RFC 1997

  When a community whose values are predefined in RFC 1997 is added to the reported routing information, the route is advertised in accordance with those values. For the communities defined in RFC 1997 and supported by the Switch, see *Table 12-8: Communities supported by the Switch*.

- Values freely specified by the user (that is, not predefined in RFC 1997) in a learned route filter or advertised route filter in the configuration settings

  When a community whose values are specified in a learned route filter or advertised route filter in the configuration settings is added to the reported routing information, the configuration settings govern whether that routing information is imported (if a learned route filter) or advertised (if an advertised route filter).

Communities can also be added to the routing information reported by the Switch by using learned route filters and advertised route filters.

The following table describes the communities defined in RFC 1997 and supported by the Switch.

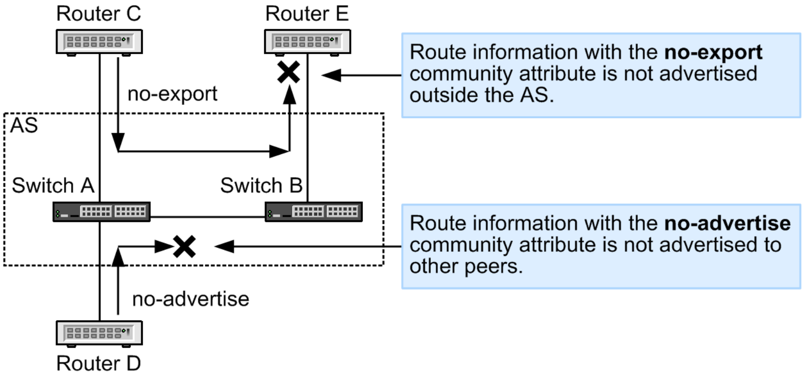
Table 12-8: Communities supported by the Switch

Community	Description
no-export	Do not advertise this routing information outside the AS.
no-advertise	Do not advertise this routing information to other peers.
local-AS	Do not advertise this routing information outside the local member AS, including to any other AS.

Note: `no-export` and `local-AS` have the same meaning in most configurations.

The following figure shows the range of routes with the `COMMUNITIES` attribute that can be propagated in a network.

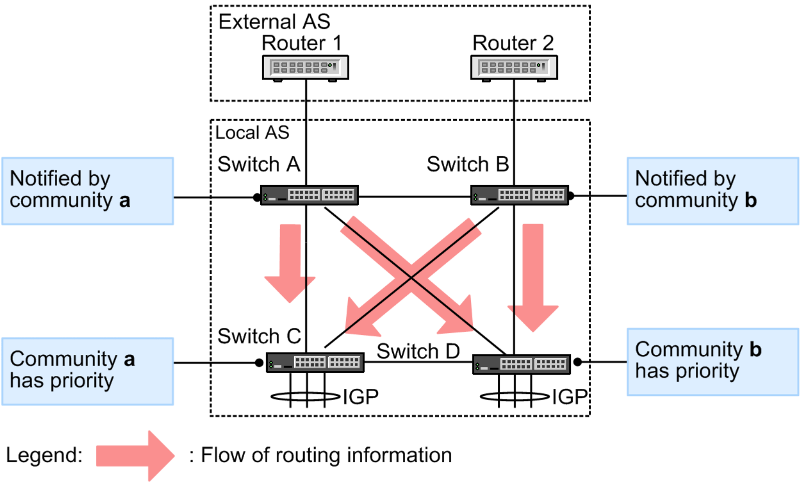
Figure 12-14: Propagation of routes with the COMMUNITIES attribute specified


Router C
Router E
no-export
AS
Switch A
Switch B
no-advertise
Router D
Route information with the **no-export** community attribute is not advertised outside the AS.
Route information with the **no-advertise** community attribute is not advertised to other peers.

(2) Example of using learned route filtering and the COMMUNITIES attribute

The following figure shows an example of using learned route filtering and the `COMMUNITIES` attribute.

Figure 12-15: Example of using learned route filtering and the COMMUNITIES attribute


External AS
Router 1
Router 2
Local AS
Switch A
Switch B
Switch C
Switch D
IGP
IGP
Notified by community **a**
Notified by community **b**
Community **a** has priority
Community **b** has priority
Legend:
: Flow of routing information

In this example, two Switches (A and B) are connected to an external AS. Considering the need to equalize traffic distribution, outbound traffic from Switch C should preferentially be routed through Switch A, and outbound traffic from Switch D should preferentially be routed through Switch B. In this scenario, load balancing can be achieved by setting up the routers as follows:

1. Add community a to the routing information propagated from Switch A to internal peers.

   (You can set an advertised route filter for this purpose.)

2. Add community b to the routing information propagated from Switch B to internal peers.

   (You can set an advertised route filter for this purpose.)

3. At Switch C, the `LOCAL-PREF` value is set to x ($x > y$) if the received routing information is tagged with community a, or to y ($x > y$) if the received routing information is tagged with community b. That is, routing information with greater `LOCAL-PREF` value which was reported from Switch A takes priority.

   (You can set a learned route filter for this purpose.)

4. At Switch D, the `LOCAL-PREF` value is set to y ($x > y$) if the received routing information is tagged with community a, or to x ($x > y$) if the received routing information is tagged with community b. That is, routing information with a greater `LOCAL-PREF` value, which was reported from Switch B, takes priority.

(You can set a learned route filter for this purpose.)

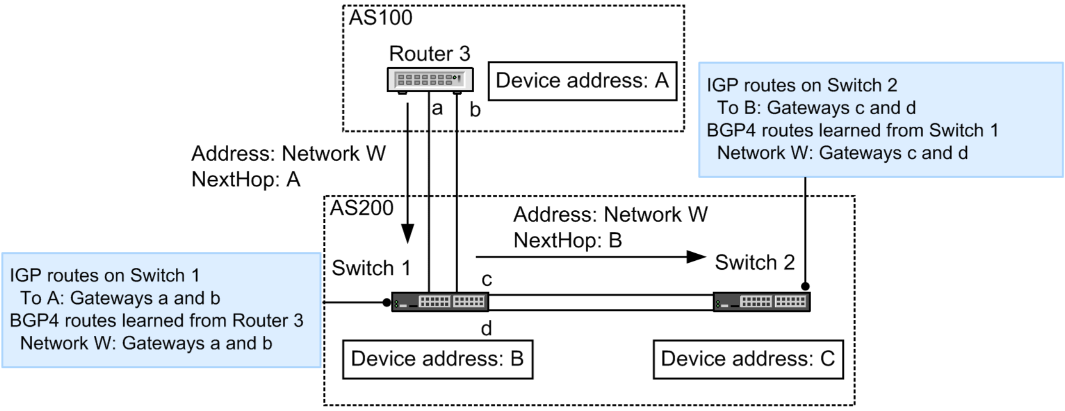
12.4.3 BGP4 multipath

BGP4 multipath routing enables traffic to be distributed equally among multiple paths (routes) to one destination network. The following describes how BGP4 multipathing operates in the Switch.

(1) BGP4 multipath via multiple IGP routes

In the Switch, the next hop of a BGP4 route is resolved through an IGP route. If there are multiple IGP routes corresponding to the `NEXT_HOP` attribute value of the BGP4 route being resolved, the BGP4 route will also take multiple paths. The following figure illustrates the concept of multipath generation.

Figure 12-16: Concept of BGP4 multipath via multiple IGP routes



In this figure, the routers are connected by two physical interfaces. For peering, they are configured to use the address assigned to the peer router itself. You can assign a peer address to each router using the `ip address` configuration command with the loopback interface specified. Also, you can specify use of the device address as the peering address on the local side by setting the `neighbor update-source` configuration command. If the local side connects to an external peer or member AS peer in this command, you must also set the `neighbor ebgp-multihop` configuration command.

The BGP4 route (destination network: W, next hop: A) that is reported to Switch 1 by AS100 references an IGP route to resolve the next hop. Next hop: Because the gateways of the IGP route to next hop A are a and b, the gateways of the BGP4 route are also a and b. Similarly, for the BGP4 route (destination network: W, next hop: B) that is reported to Switch 2 by Switch 1, the gateways of the IGP route to next hop B are c and d, so the gateways of the BGP4 route are also c and d.

Notes on BGP4 multipath routing via multiple IGP routes

Only static routes and OSPF routes can be configured as a multipath IGP route in the Switch. For a description of static multipath routing, see *8.1 Description*, and for a description of OSPF multipath routing, see *10.1.7 Equal-cost multipath*.

(2) BGP4 multipaths learned from multiple peers

In the Switch, you can generate a BGP4 multipath based on equal-cost routes to the same destination, learned from different peers connected to the same neighboring AS. To generate a BGP4 multipath, use the `maximum-paths` configuration command. By specifying the `all-as` parameter in this command, you can generate a BGP4 multipath from routes learned from different neighboring ASs. The following table describes the tie-breaker conditions.

Table 12-9: Tie-breaker conditions

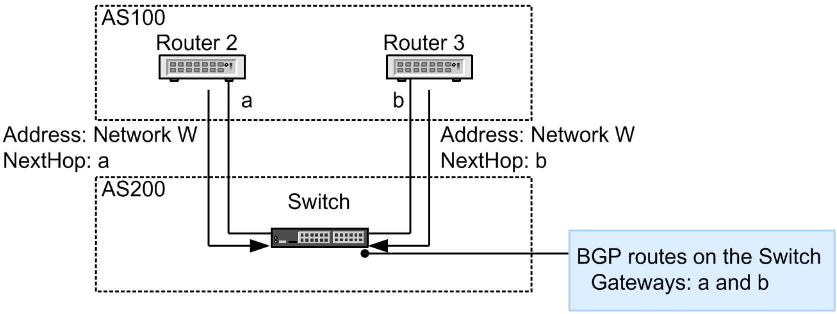
Conditions	Remarks
Equal weights	--

Conditions	Remarks
Equal LOCAL_PREF attribute values	--
Equal number of ASs in the AS_PATH attribute	The AS_SET path type in the AS_PATH attribute is counted as a single AS.
Equal ORIGIN attribute values	--
Equal MED attribute values	The tie-breaker based on MED attribute values applies only to redundant routes learned from the same neighboring AS. To include redundant routes learned from different neighboring ASs, specify the bgp always-compare-med configuration command.
Learned from the same peer type (external peer, member AS peer, or internal peer)	--
Equal next hop (equal IGP metric used at next-hop resolution)	--

Legend: --: Not applicable

The following figure illustrates the concept of BGP4 multipaths learned from multiple peers.

Figure 12-17: Concept of BGP4 multipaths learned from multiple peers



In this figure, Routers 2 and 3 in AS100 notify the Switch of two BGP4 routes to network W, via next hops *a* and *b*, respectively. If the routes are in a tie-break state, the Switch will generate gateways based on each route's NEXT_HOP attribute (gateways a and b). When propagating these routes to other BGP4 peers, the Switch advertises the route that has higher priority.

12.4.4 Capability negotiation

Capability negotiation is a means of negotiating which capabilities are supported by the peers. This is done by adding capability information to the OPEN message at the establishment of a BGP4 negotiation session. Functionality that is matched (supported) by the respective peers' advertised capability information can be used by those peers.

In the Switch, the following capability information can be added to the OPEN message: IPv4 unicast routing, route refresh capability (Capability Code:2), route refresh capability (Capability Code:128), and the graceful restart functionality (Capability Code: 64). When an OPEN message without any capability information is received from a peer, only IPv4 unicast route advertisements will be sent over the established BGP4 connection.

The following table describes the capabilities that can be negotiated.

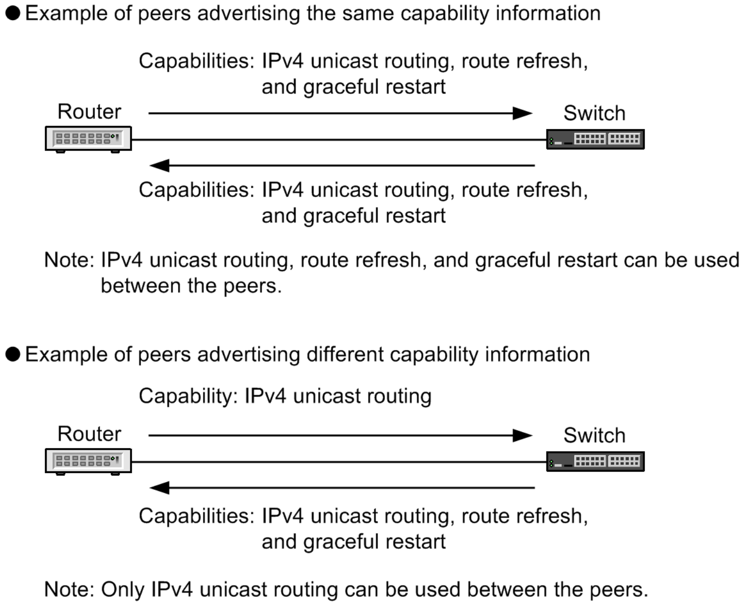
Table 12-10: Negotiable capabilities

Capability	Capability information in OPEN message	Description
IPv4 routing	Capability Code: 1 AFI field of the Capability Value: 1 SAFI field of the Capability Value: 1	IPv4 unicast routes can be sent and received between the peers.
Route refresh capability	Capability Code: 2 AFI field of the Capability Value: 1#	Route refresh is supported on IPv4 routes.
	Capability Code: 128 AFI field of the Capability Value: 1#	
Graceful restart	Capability Code: 64 AFI field of the Capability Value: 1 SAFI field of the Capability Value: 1	Graceful restart is supported.

#: IPv4 route refresh is supported if negotiation of either of these two capabilities is successful.

The following figure illustrates how negotiation operates.

Figure 12-18: Operation of capability negotiation


● Example of peers advertising the same capability information
Capabilities: IPv4 unicast routing, route refresh, and graceful restart
Router
Switch
Capabilities: IPv4 unicast routing, route refresh, and graceful restart
Note: IPv4 unicast routing, route refresh, and graceful restart can be used between the peers.
● Example of peers advertising different capability information
Capability: IPv4 unicast routing
Router
Switch
Capabilities: IPv4 unicast routing, route refresh, and graceful restart
Note: Only IPv4 unicast routing can be used between the peers.

12.4.5 Route refresh capability

A fundamental aspect of BGP4 is that only route updates are advertised. In contrast, the route refresh capability forcibly re-advertises previously propagated routes, which otherwise would not be advertised in BGP4.

Route refresh covers the re-advertisement of routes from the local device and from the remote BGP4 peer. You can select which types of routes to re-advertise. This can be executed by using the `clear ip bgp` command.

The following table describes the route refresh capability.

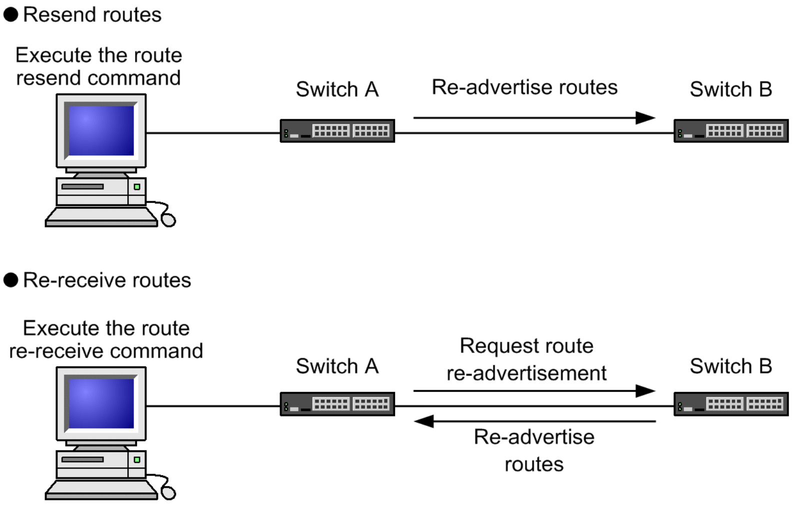
Table 12-11: Route refresh capability

Capability	Route type	Direction of re-advertisement
Resend IPv4 unicast routes	IPv4 unicast route	Re-advertised from local peer to remote peer.

Capability	Route type	Direction of re-advertisement
Re-receive IPv4 unicast routes		Re-advertised from remote peer to local peer.

The following figure illustrates how the route refresh capability operates.

Figure 12-19: Operation of the route refresh capability



(1) Notes on using the route refresh capability

For a route to be resent from the remote device, both routers in a peering relationship must support the route refresh capability. To use the route refresh capability, its use must be negotiated between the routers at the time of BGP4 peer establishment.

When the `inbound` parameter is specified in the `neighbor soft-reconfiguration` configuration command, because the routes suppressed by the learned route filter are retained as invalid routes, the remote peer does not send any route refresh requests for route re-advertisement to the local peer.

The route refresh capability of the Switch complies with RFC 2918. The capability codes used in negotiation are the RFC-2918-compliant code (value:2) and a private code (value: 128). In addition, a private capability code (value:128 to 255) defined in RFC 2434 may be used by other vendors.

Take care when using the route refresh capability between the Switch and devices from other vendors.

12.4.6 TCP MD5 authentication

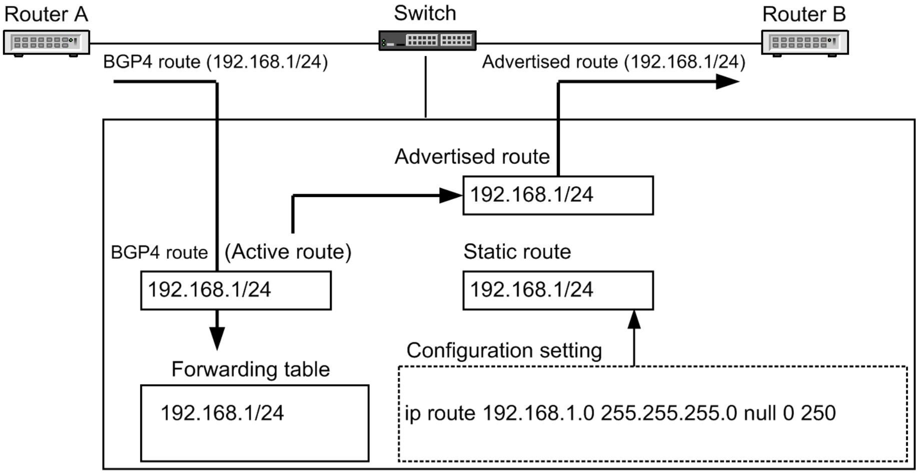
The Switch complies with RFC 2385 (Protection of BGP Sessions via the TCP MD5 Signature Option). TCP MD5 authentication guarantees that a TCP segment received over a BGP4 connection originated from a trusted source (peer). TCP MD5 authentication can be specified on a per-peer basis. To apply TCP MD5 authentication to a BGP4 connection with a remote peer, specify the authentication key in the `neighbor password` configuration command. The same authentication key must be used by both devices in a peering relationship. If the authentication keys do not match, a BGP4 connection cannot be established between the peers.

12.4.7 BGP4 advertised route generation

From the routing information stored in the local switch, you can generate a route to a given destination and advertise it via BGP4. By using the actual BGP4 route for packet forwarding, and the generated route for advertising the same routing information to other devices, you can ensure reliable forwarding to a BGP4 route destination as well as stable route advertising. The figure below shows an example of using this functionality.

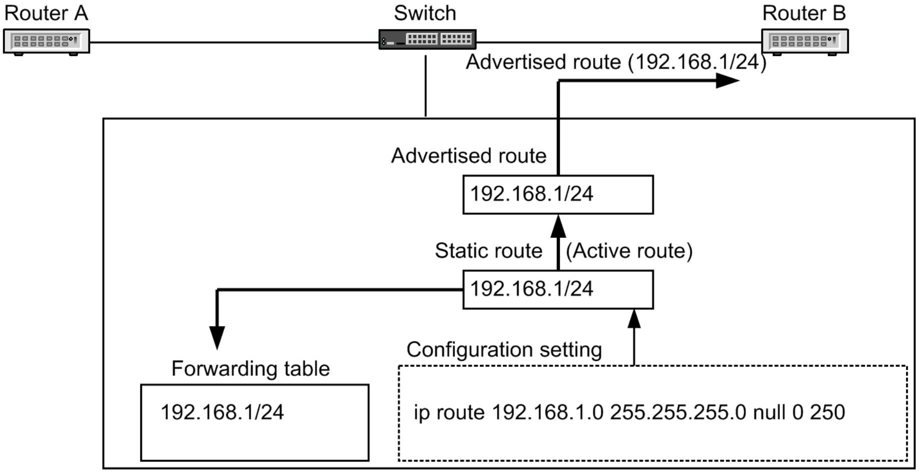
Normally, set the BGP4 route received from router A as the forwarding table, and then advertise an advertised route generated from the route to router B.

Figure 12-20: Advertised route generation and propagation (common case)



If the BGP4 route learned from router A was deleted, a static route is activated and an advertised route generated from this static route is advertised to router B.

Figure 12-21: Advertised route generation and propagation (when a BGP4 route is deleted)



By setting up the system in this way, the BGP4 route will be used for usual packet forwarding. If the BGP4 route received from Router A flaps, the BGP4 advertised route to Router B will not be affected.

Use the `network` configuration command to generate advertised routes.

Advertised routes are advertised to all peers unless route filtering is explicitly set. If an advertised route with the same destination that was generated from the BGP4 route was advertised to the learning source of the BGP4 route (router A in this example), a route loop may occur. In that case, use route filtering to suppress advertisement.

12.4.8 Route flap dampening

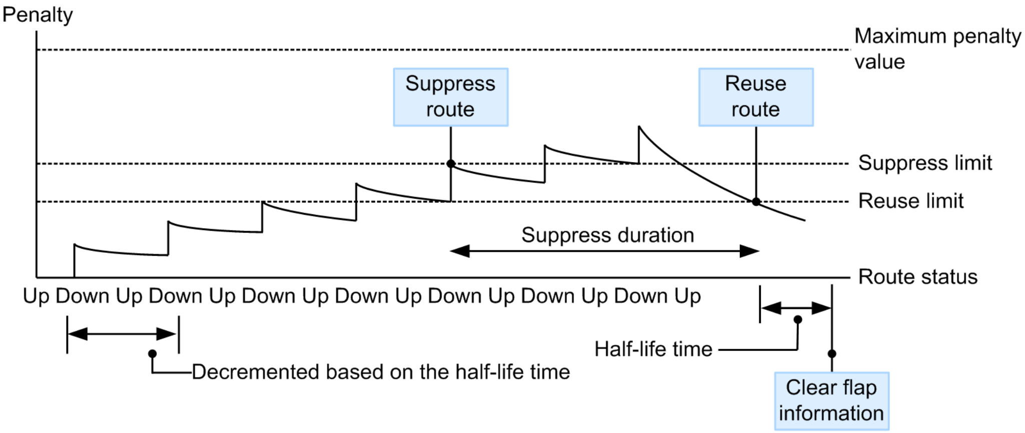
Route flap dampening minimizes network instability by temporarily suppressing the use of routes that flap frequently. Note that this functionality is not supported in VRFs. The following table describes the parameters used in route flap dampening.

Table 12-12: Parameters for route flap dampening

Parameters	Description
Penalty	A dynamic control variable for suppressing or reusing the route. The penalty value increases with each route flap, and decreases with time. The penalty is incremented by 1 every time a route flaps (becomes unreachable) and is decremented based on the half-life time. The maximum penalty value is calculated as follows: *maximum-penalty* = *reuse-limit* x 2^(*maximum-suppression-time*/*half-life-time*)
Suppress limit	Use of the route is suppressed when the penalty reaches or exceeds this value.
Reuse limit	Use of the route is allowed when the penalty falls to this value or lower.
Half-life time	The time required for the penalty to be reduced to half (50%).
Maximum suppress time	The maximum duration for the suppression of a route. This value is the time taken for the penalty value to decrease from the maximum to the reuse limit.

The following figure illustrates how route flap dampening operates.

Figure 12-22: Operation of route flap dampening



12.4.9 Route reflection

Route reflection is a means of reducing the number of internal peers in an AS. In BGP4, routing information distributed by one internal peer is not redistributed to any other internal peers. Therefore, internal peers must be logically fully meshed with each BGP speaker within the AS. Route reflection relaxes this restriction and reduces the number of internal peers in the AS by allowing redistribution of received routing information to other internal peers.

(1) Route reflection concept and flow of routing information

In route reflection, a route reflector (RR) and its clients constitute a cluster. A cluster can contain more than one RR. The other BGP speakers in the AS are referred to as non-clients.

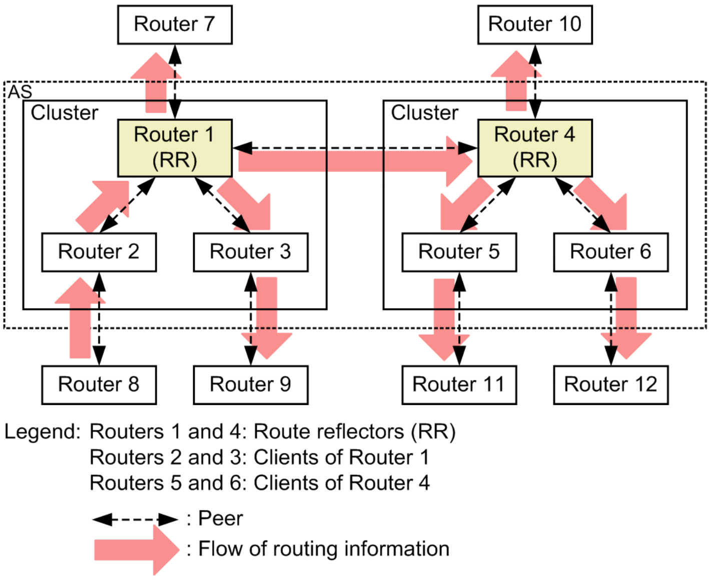
On receiving an `UPDATE` message from a client in the cluster, the RR distributes the message to all clients (including the source client) in the cluster, and to all non-clients. Upon receiving an `UPDATE` message from a non-client, the RR distributes the message to all clients in the cluster. This eliminates the need for internal peer relationships from client to non-client, or among the clients in a cluster.

Routing information distributed from external peers and member AS peers, or to external peers and member AS peers, is handled in the usual manner.

(2) Route reflection with one route reflector in the cluster

The following figure shows an example of route reflection with one RR in the cluster.

Figure 12-23: Example of route reflection with one RR


Router 7
Router 10
AS
Cluster
Router 1 (RR)
Cluster
Router 4 (RR)
Router 2
Router 3
Router 5
Router 6
Router 8
Router 9
Router 11
Router 12
Legend: Routers 1 and 4: Route reflectors (RR)
Routers 2 and 3: Clients of Router 1
Routers 5 and 6: Clients of Router 4
: Peer
: Flow of routing information

Router 1 (route reflector) and Routers 2 and 3 (clients) form a cluster. Router 4 (route reflector) and Routers 5 and 6 (clients) form another cluster. Routing information advertised from Router 2 to Router 1 is distributed to the clients (Routers 2 and 3) and all non-clients (Router 4). Routing information advertised from Router 1 to Router 4 is distributed to all clients (Routers 5 and 6).

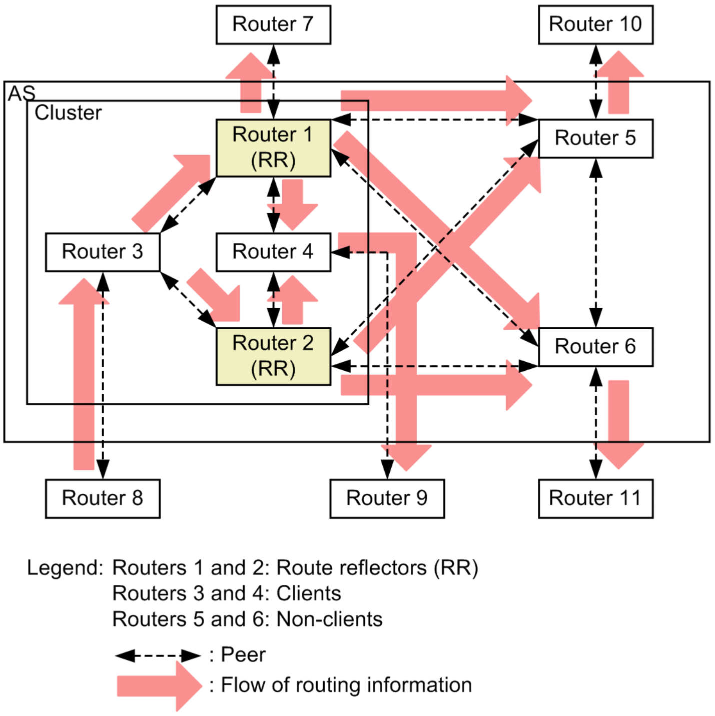
(3) Route reflection with multiple route reflectors in the cluster

A cluster can have more than one RR. This prevents the route reflection functionality from stopping if an RR fails.

Each RR forms an internal peer relationship with clients and non-clients. Routing information received from a client or non-client is redistributed by the RR as described in *Figure 12-23: Example of route reflection with one RR* above. In this way, if one RR fails, the routing information can be redistributed by the other RR. When there are multiple RRs in a cluster, the same cluster ID must be set for each RR, using the `bgp cluster-id` configuration command.

The following figure shows an example of a redundant RR configuration.

Figure 12-24: Example of a redundant configuration of RRs



The cluster contains two RRs (Routers 1 and 2). Each of which has internal peer relationships with clients (Routers 3 and 4) and with non-clients (Routers 5 and 6). For example, routing information reported from client Router 3 will be redistributed by the RRs (Routers 1 and 2) to clients (Routers 3 and 4) and non-clients (Routers 5 and 6). Should one RR fail, the routing information will be reported by the other RR. BGP speakers that do not belong to the cluster (Routers 5 and 6) can also be present in the AS.

12.4.10 Confederations

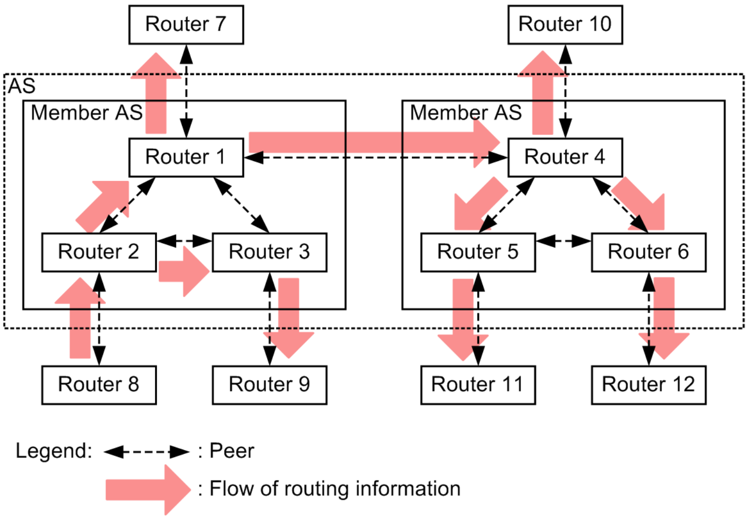
A confederation, like a route reflector, is a way of reducing the number of internal peers in an AS. In a confederation, the AS is divided into two or more member ASs, reducing the number of peers in the AS.

(1) Confederation concept and flow of routing information

A confederation is formed by dividing the AS into multiple member ASs. The BGP speakers in a member AS must be fully meshed as internal peers, and are handled as normal internal peers. Member ASs can be in peering relationships with one another, in the same way as with external peers. BGP speakers in different member ASs do not need to be fully meshed peers. In this way, deploying a confederation reduces the number of peers in an AS. In the Switch, the member ASs of a confederation in a peering relationship with one another are known as member AS peers.

The following figure shows the flow of routing information when the routers are configured in a confederation.

Figure 12-25: Flow of routing information when using a confederation



Routers 1, 2, and 3 constitute a member AS of the confederation. Routers 4, 5, and 6 constitute another member AS. Routing information from Router 8 is distributed by Router 2 to the other BGP speakers (Routers 1 and 3) in its member AS. Routing information reported from Router 2 to Router 1 is distributed to Router 4 in the other member AS, and from there to the other BGP speakers (Routers 5 and 6) in that member AS. In this way, the routing information is distributed to all BGP speakers in the AS.

(2) Route selection when using a confederation

When routers are configured in a confederation, the route selection process differs somewhat from a non-confederation topology because of the addition of another peer type (member AS peer). In the usual type of configuration, a route learned by an external peer is preferred over a route learned by an internal peer. With a confederation, however, a route learned by an external peer is preferred over a route learned by a member AS peer, which is preferred over a route learned by an internal peer.

Route selection priority when using a confederation is described in the following table.

Table 12-13: Route selection priority

Priority	Description
High	Selects the route with the greatest weight.
↑	Selects the route with the largest `LOCAL_PREF` attribute value.
	Select the route whose `AS_PATH` attribute has the smallest number of ASs.#1
	Selects by the `ORIGIN` attribute value, preferring `IGP`, `EGP`, and `Incomplete`, in that order.
	Selects the route with the smallest `MED` attribute value.#2
	Selects a route learned by an external peer, a route learned by a member AS peer, or a route learned by an internal peer, in that order.
	Selects the route with the closest next hop (the smallest metric of the IGP routes used when resolving the next-hop address).
	Selects the route whose peer has the smallest BGP identifier (router ID). For a route having the `ORIGINATOR_ID` attribute, however, the `ORIGINATOR_ID` attribute values are compared instead of the peer's BGP identifier.#3
↓	Selects the route with the smallest `CLUSTER_LIST` attribute length.#4

Priority	Description
Low	Selects the route whose learning source peer has the smallest address.#3

#1

The AS_SET path type of the AS_PATH attribute is counted as one AS. The AS_CONFED_SEQUENCE and AS_CONFED_SET path types of the AS_PATH attribute are not included in the path length.

#2

Route selection by MED attribute value applies only to redundant routes learned from the same neighboring AS. However, redundant routes learned from different neighboring ASs can also be compared by specifying the `bgp always-compare-med` configuration command.

#3

If the routes received from an external peer have different peer BGP identifiers (router IDs), the route selected at the previous step is used, and route selection based on peer BGP identifier and learning source peer address is skipped. However, routes that have different peer BGP identifiers can also be considered by specifying the `bgp bestpath compare-routerid` configuration command.

#4

When a route does not have the CLUSTER_LIST attribute, it is compared assuming a CLUSTER_LIST attribute length of 0.

(3) Handling of BGP attributes when using a confederation

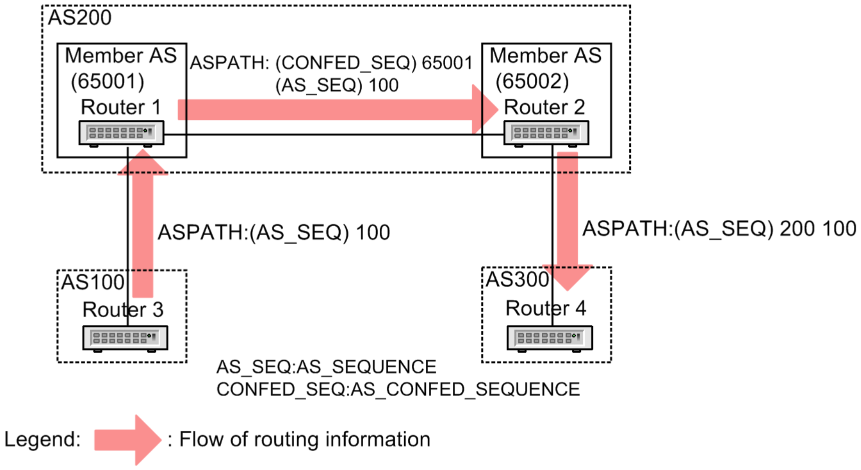
When routers are configured in a confederation, BGP attributes are handled in much the same way as in a non-confederation topology. However, there are some differences in the way the AS_PATH and COMMUNITIES attributes operate. BGP attributes in member AS peers are handled in the same way as in internal peers.

(4) Handling of the AS_PATH attribute when using a confederation

When routing information is reported to a member AS peer in a confederation, the AS number of the local member AS is appended to the AS_CONFED_SEQUENCE path type of the AS_PATH attribute. When routing information is sent to another AS (external peer), the AS_CONFED_SEQUENCE path type is removed from the AS_PATH attribute and the local AS number is appended to the AS_SEQUENCE path type. In all other respects, the AS_PATH attribute is handled in the same way as in usual configurations.

The following figure shows how the AS_PATH attribute is handled.

Figure 12-26: Handling of the AS_PATH attribute


AS200
Member AS (65001) Router 1
ASPATH: (CONFED_SEQ) 65001 (AS_SEQ) 100
Member AS (65002) Router 2
ASPATH:(AS_SEQ) 100
ASPATH:(AS_SEQ) 200 100
AS100 Router 3
AS300 Router 4
AS_SEQ:AS_SEQUENCE
CONFED_SEQ:AS_CONFED_SEQUENCE
Legend:
: Flow of routing information

In this figure, the routing information `ASPATH: (AS_SEQUENCE) 100` is reported by AS100 to Router 1. Router 1 distributes the route to Router 2 (also a member AS of this confederation), appending its own member AS number (65001) to the `AS_CONFED_SEQUENCE` path type in the `AS_PATH` attribute. Router 2 receives the routing information as `ASPATH: (AS_CONFED_SEQUENCE) 65001, (AS_SEQUENCE) 100`. When distributing this routing information to AS300, Router 2 removes the `AS_CONFED_SEQUENCE` path type and appends its own AS number (200) to the `AS_SEQUENCE` path type.

(5) Handling of the COMMUNITIES attribute when using a confederation

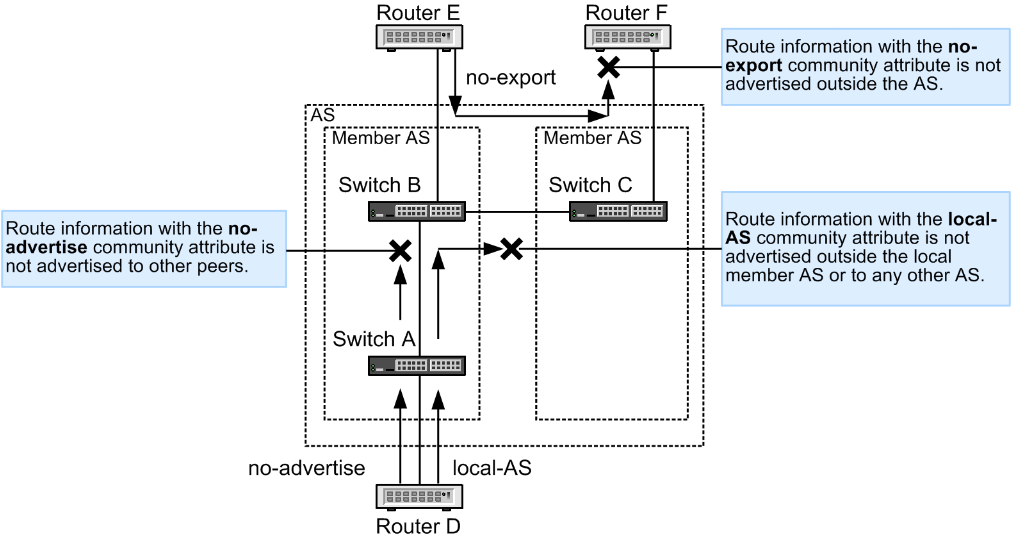
In a confederation, the well-known communities defined in RFC 1997 are handled as follows. Other communities are handled in the usual manner.

The well-known communities defined in RFC 1997 are listed in *Table 12-14: Well-known communities defined in RFC 1997*. The range of routes with the `COMMUNITIES` attribute that can be propagated in a network are shown in *Figure 12-27: Propagation of routes with the COMMUNITIES attribute specified* below.

Table 12-14: Well-known communities defined in RFC 1997

Community	Description
no-export	Do not advertise this routing information outside the AS.
no-advertise	Do not advertise this routing information to other peers.
local-AS	Do not advertise this routing information outside the local member AS.

Figure 12-27: Propagation of routes with the COMMUNITIES attribute specified



Route information with the **no-export** community attribute is not advertised outside the AS.

Route information with the **no-advertise** community attribute is not advertised to other peers.

Route information with the **local-AS** community attribute is not advertised outside the local member AS or to any other AS.

12.4.11 Graceful restart

(1) Overview

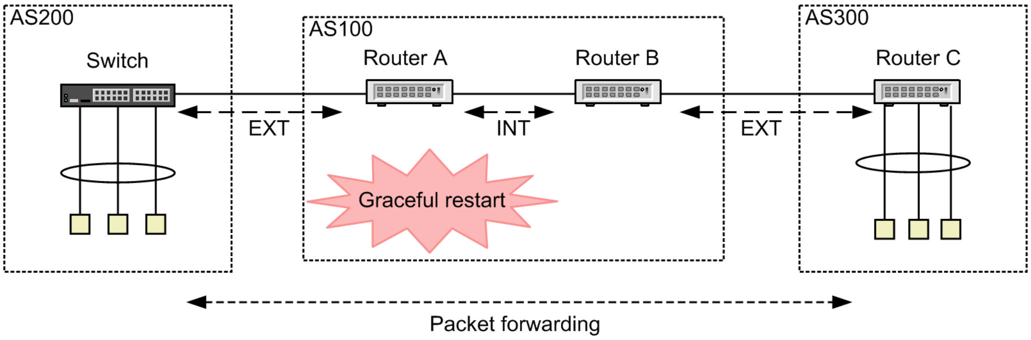
The graceful restart functionality reduces communication downtime due to routes disappearing from a network when a device is failed over or when the routing program is restarted by an operation command.

In BGP4, the router on which BGP4 is restarted by a graceful restart is known as a restarting router. A neighboring router assisting the graceful restart is known as a receiving router.

The Switch supports the receiving router functionality.

The following figure shows an example of a graceful restart with the Switch.

Figure 12-28: Example of a graceful restart



Legend: EXT: External peer
INT: Internal peer

In this figure, the Switch in AS200 and Router A in AS100 have established an external peering BGP connection in which the peer addresses are used as the interface address. In addition, an internal peer BGP connection has been established between Routers A and B, and an external peer BGP connection has been established between Routers B and C. Assume that the graceful restart functionality has been successfully negotiated on each BGP connection. When Router A performs a graceful restart, Router B and the Switch connected to Router A act as receiving routers and continue packet forwarding via Router A without interruption. This ensures continuous end-to-end communication via Router A.

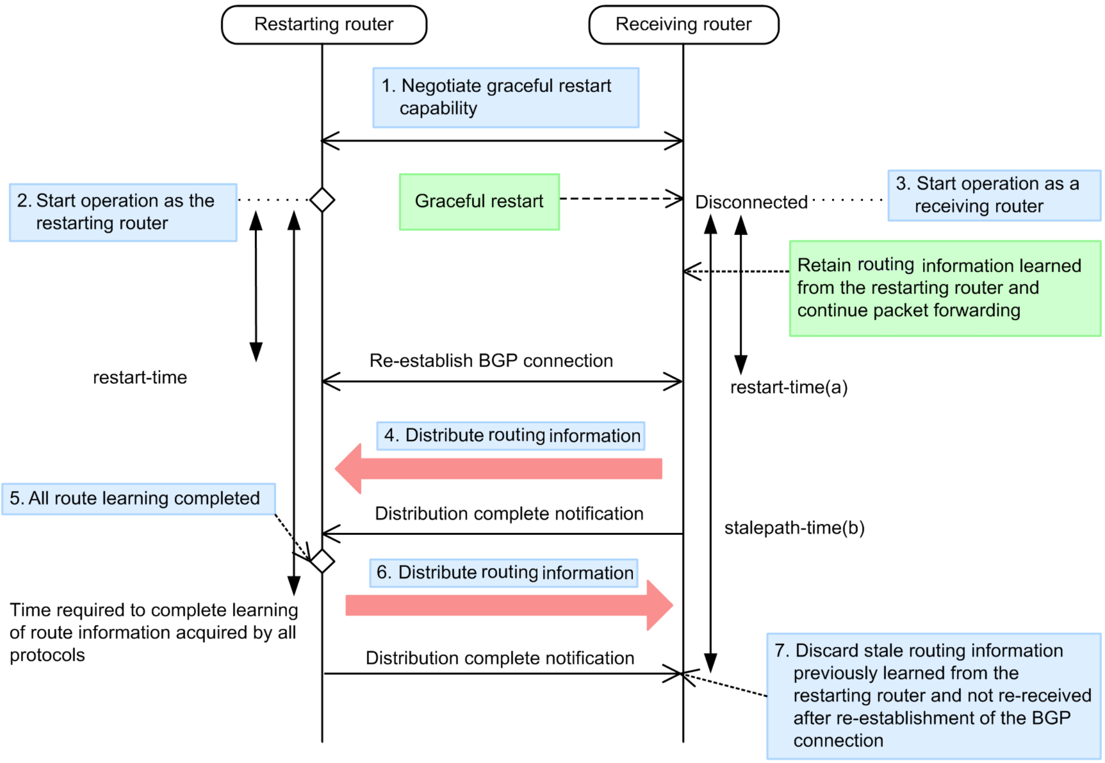
The conditions for a successful BGP4 graceful restart are as follows. If these conditions are not satisfied, a normal restart is performed and communication halts.

- The `bgp graceful-restart mode` configuration command must be set to operate the Switch as a receiving router.
- Graceful restart capability must have been successfully negotiated on the BGP connections between the router on which BGP4 will be gracefully restarted and the devices that will act as receiving routers.

(2) Performing a graceful restart

The following figure shows the sequence of steps for performing a graceful restart.

Figure 12-29: Graceful restart sequence


Restarting router
Receiving router
1. Negotiate graceful restart capability
2. Start operation as the restarting router
Graceful restart
Disconnected
3. Start operation as a receiving router
Retain routing information learned from the restarting router and continue packet forwarding
restart-time
Re-establish BGP connection
restart-time(a)
4. Distribute routing information
5. All route learning completed
Distribution complete notification
stalepath-time(b)
6. Distribute routing information
Time required to complete learning of route information acquired by all protocols
Distribution complete notification
7. Discard stale routing information previously learned from the restarting router and not re-received after re-establishment of the BGP connection

1. Prepare for a graceful restart by negotiating graceful restart functionality when establishing a BGP connection between the router to be restarted and the neighboring router.
2. The graceful restart commences when the router performing the graceful restart begins operating as the restarting router.
3. When the BGP connection closes, the neighboring router begins operating as a receiving router. It keeps the routing information it has already learned from the restarting router and continues forwarding the received packets.
4. When the BGP connection with the neighboring router is re-established, routing information is first distributed back from the receiving router to the restarting router.
5. When the restarting router has completed learning all the routing information for the protocols involved in the graceful restart, it distributes its routing information to the receiving router.
6. This step is the same as step 5.
7. Lastly, the receiving router discards the stale routing information that it previously learned from the restarting router and did not receive again after the BGP connection was re-established.

(3) Receiving router functionality

(a) Activation triggers

The BGP4 receiving router functionality is activated in the Switch when:

- The switch detects that the TCP session being used by the BGP connection has terminated, although no final `NOTIFICATION` message was received from the connected peer.
- An `OPEN` message is received from a connected peer and a new TCP session is established on the BGP connection.

(b) Receiving router functionality

The receiving router monitors re-establishment of the BGP connection for the maximum wait time set by the `bgp graceful-restart restart-time` configuration command (see *(a)* in *Figure 12-29: Graceful restart sequence*). If the BGP connection is not re-established within this time, the receiving router discards the routing information learned from the restarting router and suspends packet forwarding via the restarting router.

The `restart-time` value is exchanged between the peers when they negotiate their graceful restart functionality. If the value received from the peer by the Switch is less than its own setting, the BGP connection is monitored for the `restart-time` value reported to the switch.

To specify the maximum time that the receiving router is to retain routing information learned before the graceful restart, use the `bgp graceful-restart stalepath-time` configuration command (see *(b)* in *Figure 12-29: Graceful restart sequence*).

In most circumstances, set the parameter as follows:

- For `stalepath-time`, set a value greater than the time required for the restarting router to finish learning routing information acquired by all protocols.

  The time required to complete learning of routing information acquired by all protocols is the maximum wait time before the restarting router begins distributing routing information. In the worst case scenario, distribution to the receiving router would begin after the required completion time has elapsed. The receiving router deletes old routing information after it has learned the routes and updated its forwarding table. For this reason, for `stalepath-time` set a value approximately 120 seconds longer than the time that will be needed for the restarting router to complete its learning process. The value that you set will depend on the number of routes and the number of peers adjacent to the restarting router.

(c) Failures at the receiving router

The BGP4 graceful restart functionality fails if communication via the restarting router stops. This can happen for the following reasons:

- A BGP connection with the restarting router was not established within the restart time after commencement of a graceful restart.
- The receiving router was restarted during operation as a receiving router.
- The restarting router could not retain the routing information it learned before commencing a graceful restart.
- The BGP connection that was established after commencement of a graceful restart was lost before distribution of routing information from the restarting router was complete.
- The BGP connection was terminated again after commencement of a graceful restart because the number of BGP4 routes learned from the restarted router exceeded the limit.

(4) Notes on performing a graceful restart

1. Using TCP MD5 with a graceful restart

   Under the BGP4 protocol rules, if the peer requests a new connection after establishing a BGP connection that supports the graceful restart functionality, the established BGP connection is dropped and the new BGP connection is used. To prevent security issues due to this behavior, use TCP MD5 authentication together with the graceful restart functionality.

2. Graceful restart in an IGP environment

   Set up the graceful restart functionality for the IGP as well as for BGP4 when peer-addressed routing information is exchanged by IGP over an internal peer connection that is not directly connected, or when the `NEXT_HOP` attribute in the BGP routing information is resolved to an IGP route in an environment that uses route reflection.

12.4.12 Maximum number of learned BGP4 routes

A limit can be imposed on the number of BGP4 routes learned from a peer. The purpose of this limit is to avoid memory shortages in the Switch caused by excessive routing information, and to prevent a peer from being unable to learn routing information from other peers due to the volume of BGP4 routing information received from a particular peer. When this functionality is enabled, if the number of BGP4 routes learned from the specified peer exceeds a set threshold, a warning message is output. If the number of learned BGP4 routes exceeds the maximum, another warning message is output and the peer is disconnected. The disconnected peer is reconnected after a set interval or by input of the `clear ip bgp` operation command. The functionality can be configured so that a warning message is output when the number of learned routes exceeds the maximum, but the peer is not disconnected.

12.5 Configuration of extended functionality

12.5.1 Configuring BGP4 peer groups

(1) List of configuration commands

The following table describes the configuration commands for BGP4 peer groups.

Table 12-15: List of configuration commands

Command name	Description
neighbor peer-group (assigning members)	Assigns a peer to a peer group.
neighbor peer-group (creating)	Creates a peer group.

(2) Creating a BGP4 peer group

Points to note

The example below shows how to create a peer group by using the `neighbor peer-group` (creating) command. Settings such as the AS number, optional settings, and advertising filters apply to all peers in the peer group.

Command examples

1.
```
(config)# router bgp 65531
(config-router)# bgp router-id 172.16.2.100
(config-router)# neighbor INTERNAL-GROUP peer-group
```
Creates a peer group (group identifier: `INTERNAL-GROUP`) using the `neighbor peer-group` (creating) command.

2.
```
(config-router)# neighbor INTERNAL-GROUP remote-as 65531
(config-router)# neighbor INTERNAL-GROUP soft-reconfiguration
inbound
(config-router)# neighbor INTERNAL-GROUP timers 30 90
```
Sets the AS number (AS: 65531) and optional settings for the peer group (group identifier: `INTERNAL-GROUP`).

3.
```
(config-router)# neighbor EXTERNAL-GROUP peer-group
(config-router)# neighbor EXTERNAL-GROUP send-community
(config-router)# neighbor EXTERNAL-GROUP maximum-prefix 10000
(config-router)# exit
```
Creates a peer group (group identifier: `EXTERNAL-GROUP`) using the `neighbor peer-group` (creating) command, and performs optional settings.

4.
```
(config)# route-map SET_COM permit 10
(config-route-map)# set community 1000:1001
(config-route-map)# exit
```

Sets the community value 1000:1001 for the specified `route-map`.

5.
```
(config)# router bgp 65531
(config-router)# neighbor EXTERNAL-GROUP route-map SET_COM out
```
Sets an advertised route filter for the peer group (group identifier: `EXTERNAL-GROUP`).

(3) Assigning BGP4 peers to a peer group

Points to note

The example below shows how to assign a peer to a peer group by using the `neighbor peer-group` (assigning members) command. Settings such as the AS number, optional settings, and advertising filters apply to the new member.

Command examples

1.
```
(config-router)# neighbor 172.16.2.2 peer-group INTERNAL-GROUP
```
Assigns a peer (remote peer address: 172.16.2.2) to the peer group (group identifier: `INTERNAL-GROUP`) by using the `neighbor peer-group` (assigning members) command. The AS number 65531 set for the peer group will be used for the peer's AS number.

2.
```
(config-router)# neighbor 172.17.3.3 peer-group INTERNAL-GROUP
```
Assigns a peer (remote peer address: 172.17.3.3) to the peer group (group identifier: `INTERNAL-GROUP`) by using the `neighbor peer-group` (assigning members) command. The AS number 65531 set for the peer group will be used for the peer's AS number.

3.
```
(config-router)# neighbor 192.168.4.4 remote-as 65533
(config-router)# neighbor 192.168.4.4 peer-group
EXTERNAL-GROUP
```
Creates a peer (remote peer address: 192.168.4.4) and assigns it to a peer group (group identifier: `EXTERNAL-GROUP`). The number 65533 that was set for the peer will be used as the peer's AS number.

4.
```
(config-router)# neighbor 192.168.5.5 remote-as 65534
(config-router)# neighbor 192.168.5.5 peer-group
EXTERNAL-GROUP
```
Creates a peer (remote peer address: 192.168.5.5) and assigns it to a peer group (group identifier: `EXTERNAL-GROUP`). The number 65534 that was set for the peer will be used as the peer's AS number.

12.5.2 Configuring communities

(1) List of configuration commands

The following table describes the configuration commands for communities.

Table 12-16: List of configuration commands

Command name	Description
neighbor send-community	Specifies that the `COMMUNITIES` attribute should not be removed from route advertisements to peers.

Command name	Description
distribute-list in (BGP4)#	Specifies the route filter to be used as BGP4 learned route filter conditions.
distribute-list out (BGP4)#	Specifies the route filter to be used as BGP4 advertised route filter conditions.
neighbor in (BGP4)#	Specifies in the `route-map` parameter the route filter to be used as learned route filter conditions for a specific BGP4 peer only.
neighbor out (BGP4)#	Specifies in the `route-map` parameter the route filter to be used as advertised route filter conditions for a specific BGP4 peer only.
redistribute (BGP4)#	Specifies the protocol of learned routes advertised by BGP4.

#

See *14. Route Filters (IPv4 and IPv6)* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

(2) Creating a community

Points to note

The example below shows how to add the COMMUNITIES attribute to advertised BGP4 routes by setting the `neighbor send-community` configuration command for the concerned peers.

Command examples

1.
```
(config)# router bgp 65531
(config-router)# bgp router-id 192.168.1.100
(config-router)# neighbor 192.168.2.2 remote-as 65531
(config-router)# neighbor 172.16.2.2 remote-as 65532
(config-router)# neighbor 10.2.2.2 remote-as 65533
```
Configures the BGP4 peers.

2.
```
(config-router)# neighbor 172.16.2.2 send-community
(config-router)# neighbor 10.2.2.2 send-community
(config-router)# exit
```
Specifies that the COMMUNITIES attribute is to be added to BGP4 routes advertised to the peers.

3.
```
(config)# ip community-list 10 permit 1000:1002
(config)# ip community-list 20 permit 1000:1003
(config)# route-map SET_LOCPREF permit 10
(config-route-map)# match community 10
(config-route-map)# set local-preference 120
(config-route-map)# exit
(config)# route-map SET_LOCPREF permit 20
(config-route-map)# match community 20
(config-route-map)# set local-preference 80
```

```
(config-route-map)# exit
(config)# route-map SET_LOCPREF permit 30
(config-route-map)# exit
```

Sets 120 as the `LOCAL_PREF` attribute value of routes that have the community value 1000:1002 in their `COMMUNITIES` attribute, and 80 as the `LOCAL_PREF` attribute value of routes that have the community value 1000:1003 in their `COMMUNITIES` attribute.

4.
```
(config)# ip prefix-list MY_NET seq 10 permit 192.168.0.0/16 ge 
16 le 30
(config)# route-map SET_COM permit 10
(config-route-map)# match ip address prefix-list MY_NET
(config-route-map)# set community 1000:1001
(config-route-map)# exit
```

Sets a `COMMUNITIES` attribute that has the community value 1000:1001 for routes to destination network 192.168.0.0/16 (mask length 16 to 30).

5.
```
(config)# router bgp 65531
(config-router)# distribute-list route-map SET_LOCPREF in
(config-router)# distribute-list route-map SET_COM out
(config-router)# exit
```

Sets a learned route filter and advertised route filter for all peers.

(3) Applying filters

Points to note

The example below shows how to apply route filters as learned route and advertised route filter conditions by using the `clear ip bgp` operation command.

Command examples

1.
```
# clear ip bgp * both
```

Applies the community-based route filters to the network operation.

12.5.3 Configuring BGP4 multipath

(1) List of configuration commands

The following table describes the configuration commands for BGP4 multipaths.

Table 12-17: List of configuration commands

Command name	Description
bgp always-compare-med	Enables comparison of the `MED` attribute of routes learned from different ASs. (If this command has not been set, you cannot set the `all-as` parameter in the `maximum-paths` command.)
maximum-paths	Sets the maximum number of paths to a given destination.

(2) Configuring BGP4 multipath

Points to note

To specify the `all-as` parameter in the `maximum-paths` command, you must first set the `bgp always-compare-med` command.

Command examples

1.
```
(config)# router bgp 65531
(config-router)# bgp router-id 192.168.1.100
(config-router)# neighbor 172.16.2.2 remote-as 65532
(config-router)# neighbor 172.17.2.2 remote-as 65533
```
Configures the peers that will participate in multipath routing. In this example, the routes learned from AS65532 and AS65533 will be alternative paths in a multipath route.

2.
```
(config-router)# bgp always-compare-med
(config-router)# maximum-paths 4 all-as
(config-router)# exit
```
Specifies that a maximum of four paths, including paths learned from different ASs, can be in a multipath.

12.5.4 Configuring TCP MD5 authentication

(1) List of configuration commands

The following table describes the configuration commands for TCP MD5 authentication.

Table 12-18: List of configuration commands

Command name	Description
neighbor password	Specifies that TCP MD5 authentication is to be used for peer connection.

(2) Configuring TCP MD5 authentication

Points to note

The example below shows how to set an authentication key for TCP MD5 authentication by using the `neighbor password` configuration command.

Command examples

1.
```
(config)# router bgp 65531
(config-router)# bgp router-id 192.168.1.100
(config-router)# neighbor 172.16.2.2 remote-as 65532
(config-router)# neighbor 192.168.2.2 remote-as 65531
```
Configures the BGP4 peers.

2.
```
(config-router)# neighbor 172.16.2.2 password "authmd5_65532"
(config-router)# exit
```
Sets up TCP MD5 authentication based on the authentication key `authmd5_65532` for the peer whose remote peer address is 172.16.2.2.

12.5.5 Configuring BGP4 advertised route generation

(1) List of configuration commands

The following table describes the configuration commands used to generate BGP4 advertised routes.

Table 12-19: List of configuration commands

Command name	Description
network	Specifies the generation of a BGP4 advertised route.

(2) Configuring BGP4 advertised route generation

Points to note

The example below shows how to generate BGP4 advertised routes by using the `network` configuration command. To filter the routes generated by using the `network` command, specify `local` in the `match route-type` command in `route-map`.

Command examples

1.
```
(config)# router bgp 65531
(config-router)# bgp router-id 192.168.1.100
(config-router)# neighbor 172.16.2.2 remote-as 65532
(config-router)# neighbor 192.168.2.2 remote-as 65531
```
Configures the BGP4 peers.

2.
```
(config-router)# network 192.169.10.0/24
(config-router)# exit
```
Generates a BGP4 advertised route for 192.169.10.0/24 if the route is in the routing table.

3.
```
(config)# route-map ADV_NET permit 10
(config-route-map)# match route-type local
(config-route-map)# exit
```
Specifies the generated BGP4 advertised routes.

4.
```
(config)# route-map ADV_NET deny 20
(config-route-map)# match protocol bgp
(config-route-map)# exit
```
Specifies the BGP protocol.

5.
```
(config)# router bgp 65531
(config-router)# neighbor 172.16.2.2 route-map ADV_NET out
(config-router)# exit
```
Specifies that only the generated BGP4 advertised routes are to be announced to the peer

whose remote peer address is 172.16.2.2 (i.e., the learned BGP4 route is not announced).

6.
```
(config)# route-map DENY_NET deny 10
(config-route-map)# match route-type local
(config-route-map)# exit
```
Specifies the generated BGP4 advertised routes.

7.
```
(config)# router bgp 65531
(config-router)# neighbor 192.168.2.2 route-map DENY_NET out
(config-router)# exit
```
Specifies that the generated BGP4 advertised routes are not to be announced to the peer whose remote peer address is 192.168.2.2.

(3) Applying filters

Points to note

The example below shows how to use the `clear ip bgp` operation command to apply filter settings to generated BGP4 advertised routes.

Command examples

1.
```
# clear ip bgp * out
```
Applies the route filter to BGP4 advertised routes.

12.5.6 Configuring route flap dampening

(1) List of configuration commands

The following table describes the configuration commands for route flap dampening.

Table 12-20: List of configuration commands

Command name	Description
bgp dampening	Temporarily stops the use of an unstable route and reduces the effect of route flapping.#

#: Only specified for the global network. The `config-router` mode setting is not applied to VRF.

(2) Configuring route flap dampening

Points to note

The example below shows how to apply route flap dampening to a BGP4 route by using the `bgp dampening` command in `config-router` mode.

Command examples

1.
```
(config)# router bgp 65531
(config-router)# bgp router-id 192.168.1.100
(config-router)# neighbor 172.16.2.2 remote-as 65532
(config-router)# neighbor 172.17.2.2 remote-as 65533
```

Configures the BGP4 peers.

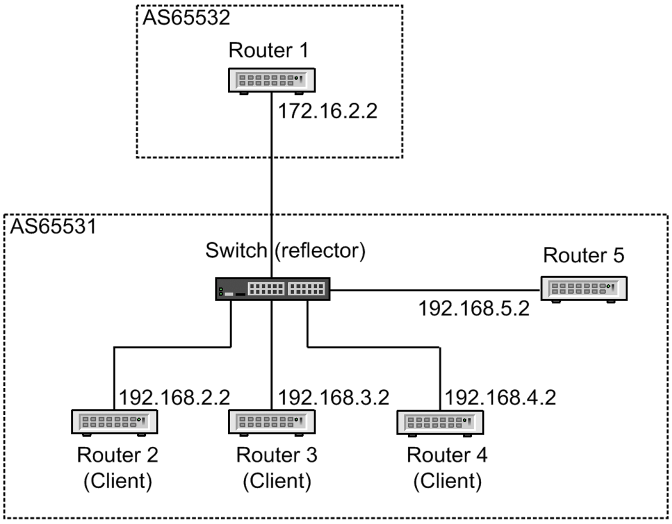
2. `(config-router)# bgp dampening`

   Applies route flap dampening.

12.5.7 Configuring route reflection

This section describes how to configure route reflection using the following figure as a reference.

Figure 12-30: Example of configuring route reflection



(1) List of configuration commands

The following table describes the configuration commands for route reflection.

Table 12-21: List of configuration commands

Command name	Description
bgp client-to-client reflection	Specifies that BGP4 routes are to be reflected between the route reflector and clients.
bgp cluster-id	Specifies the cluster ID to be used in route reflection.
bgp router-id	If `bgp cluster-id` is not set, the router ID is used as the cluster ID for route reflection.
neighbor always-nexthop-self	Specifies that the `NEXT_HOP` attribute of routes advertised to an internal peer (including route reflection) is to be forcibly changed to the local address being used for peering with the internal peer.
neighbor route-reflector-client	Specifies the route reflector client.

(2) Configuring route reflection

Points to note

The `bgp client-to-client reflection` configuration command is enabled by default and does not need to be set. If you do not want BGP4 routes to be reflected between the route reflector and clients, use the `no bgp client-to-client reflection` command in `config-router` mode.

Command examples

1.
```
(config)# router bgp 65531
(config-router)# bgp router-id 192.168.1.100
(config-router)# neighbor 172.16.2.2 remote-as 65532
(config-router)# neighbor 192.168.2.2 remote-as 65531
(config-router)# neighbor 192.168.3.2 remote-as 65531
(config-router)# neighbor 192.168.4.2 remote-as 65531
(config-router)# neighbor 192.168.5.2 remote-as 65531
```
Configures the BGP4 peers, defining Router 1 as an external peer and Routers 2 to 5 as internal peers.

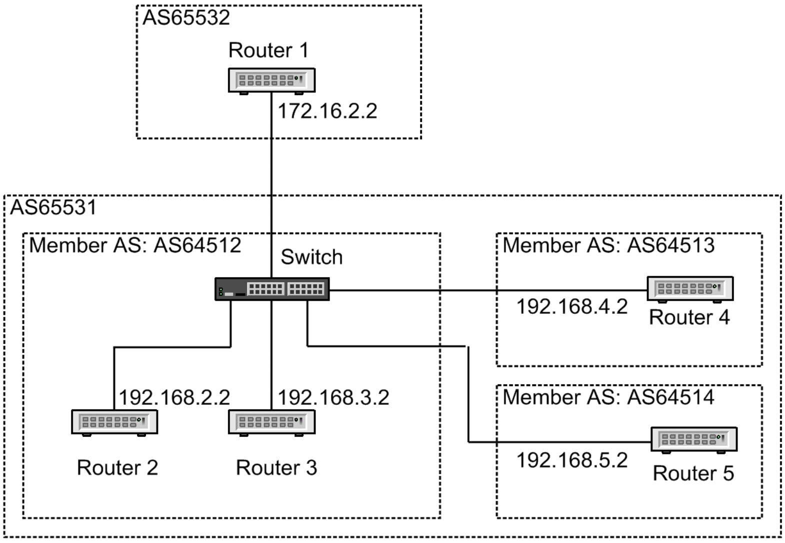
2.
```
(config-router)# bgp cluster-id 10.1.2.1
```
Sets the cluster ID.

3.
```
(config-router)# neighbor 192.168.2.2 route-reflector-client
(config-router)# neighbor 192.168.3.2 route-reflector-client
(config-router)# neighbor 192.168.4.2 route-reflector-client
```
Defines Routers 2, 3, and 4 as route reflector clients.

12.5.8 Configuring confederations

This section describes how to configure a confederation using the following figure as a reference.

Figure 12-31: Example of configuring a confederation



(1) List of configuration commands

The following table describes the configuration commands for confederations.

Table 12-22: List of configuration commands

Command name	Description
bgp confederation identifier	Specifies the AS number when configuring a confederation.#
bgp confederation peers	Specifies the member AS numbers of the ASs connected to the local member AS when configuring a confederation.
neighbor remote-as	Configures a BGP4/BGP4+ peer. Sets the local member AS number when configuring a confederation.

#: Commonly specified for both VRFs and the global network.

(2) Configuring a confederation

Points to note

The example below shows how to specify the local member AS number by using the `router bgp` command. Then, set the member AS numbers of the other ASs connected to each AS by using the `bgp confederation peers` command in `config-router` mode.

Command examples

1.
```
(config)# router bgp 64512
```
Specifies the local member AS number (64512).

2.
```
(config-router)# bgp router-id 192.168.1.100
```
Specifies the router ID.

3.
```
(config-router)# bgp confederation identifier 65531
```
Specifies the AS number (65531) of the confederation.

4.
```
(config-router)# bgp confederation peers 64513 64514
```
Specifies the numbers of the other member ASs (64513 and 64514) connected to the local member AS.

5.
```
(config-router)# neighbor 172.16.2.2 remote-as 65532
(config-router)# neighbor 192.168.2.2 remote-as 64512
(config-router)# neighbor 192.168.3.2 remote-as 64512
(config-router)# neighbor 192.168.4.2 remote-as 64513
(config-router)# neighbor 192.168.5.2 remote-as 64514
```
Configures the BGP4 peers, defining Router 1 as an external peer, Routers 2 and 3 as internal peers, and Routers 4 and 5 as member AS peers.

12.5.9 Configuring a graceful restart

(1) List of configuration commands

The following table describes the configuration commands for the graceful restart functionality.

Table 12-23: List of configuration commands

Command name	Description
bgp graceful-restart mode	Specifies use of the graceful restart functionality.#
bgp graceful-restart restart-time	Specifies the maximum time that a peer will wait to be reconnected after a neighboring router has commenced a graceful restart. #
bgp graceful-restart stalepath-time	Specifies the maximum time that a peer will keep routes received before a graceful restart after the neighboring router has commenced a graceful restart. #

#

Commonly specified for both VRFs and the global network. The `config-router` mode setting is also applied to VRF.

(2) Configuring the graceful restart functionality

Points to note

The example below shows how to use the graceful restart functionality by setting the `bgp graceful-restart mode` command in `config-router` mode. Set this command before you set the optional `bgp graceful-restart restart-time` command and `bgp graceful-restart stalepath-time` command.

Command examples

1.
```
(config)# router bgp 65531
(config-router)# bgp router-id 192.168.1.100
(config-router)# neighbor 172.16.2.2 remote-as 65532
(config-router)# neighbor 192.168.2.2 remote-as 65531
```
Configures the BGP4 peers.

2.
```
(config-router)# bgp graceful-restart mode receive
```
Specifies use of the graceful restart receiving router functionality.

12.5.10 Configuring the maximum number of learned BGP4 routes

(1) List of configuration commands

The following table describes the configuration commands for the number of learned BGP4 routes.

Table 12-24: List of configuration commands

Command name	Description
neighbor maximum-prefix	Limits the number of routes that can be learned from a specified peer.

(2) Configuring the maximum number of learned BGP4 routes

Points to note

The example below shows how to limit the number of BGP4 routes that can be learned from a peer by using the `neighbor maximum-prefix` command.

Command examples

1.
```
(config)# router bgp 65531
```

```
(config-router)# bgp router-id 192.168.1.100
(config-router)# neighbor 172.16.2.2 remote-as 65532
(config-router)# neighbor 192.168.2.2 remote-as 65531
```

Configures the BGP4 peers.

2.
   ```
   (config-router)# neighbor 172.16.2.2 maximum-prefix 10000 80
   restart 60
   ```

   Allows a maximum of 10000 routes to be learned from the external peer (remote peer address: 172.16.2.2). Sets a threshold of 80% for output of a warning message, and an interval of 60 minutes before reconnection of any peer that was disconnected because it exceeded the maximum.

3.
   ```
   (config-router)# neighbor 192.168.2.2 maximum-prefix 1000
   warning-only
   ```

   Allows a maximum of 1000 routes to be learned from the internal peer (remote peer address: 172.16.2.2), and specifies that any peer exceeding this maximum will not be disconnected.

12.6 Operation for extended functionality

12.6.1 Checking BGP4 peer groups

(1) List of operation commands

The following table describes the operation commands for BGP4 peer groups.

Table 12-25: List of operation commands

Command name	Description
show ip bgp	Shows information related to the BGP4 protocol.

(2) Checking a BGP4 peer group

To check the peering information about the peers in a peer group, use the `show ip bgp` command with the `peer-group` parameter specified.

Figure 12-32: Results of executing the show ip bgp command (with the peer-group parameter specified)

```
>show ip bgp peer-group INTERNAL-GROUP
Date 20XX/07/17 18:40:00 UTC
Local AS: 65531, Local Router ID: 172.16.2.100
BGP Peer       AS     Received    Sent        Up/Down              Status
172.16.2.2     65531  36          42          20XX/07/16 18:42:26  Established
172.16.3.3     65531  51          63          20XX/07/16 12:42:31  Established
```

(3) Checking the peer relationships in a BGP4 peer group

To display information about the peers in a peer group, use the `show ip bgp` command with the `neighbors` parameter specified.

Figure 12-33: Results of executing the show ip bgp command (with the neighbors parameter specified)

```
>show ip bgp neighbors EXTERNAL-GROUP
Date 20XX/07/17 18:45:09 UTC
Peer Address    Peer AS  Local Address    Local AS  Type      Status
192.168.4.4     65533    192.168.4.214    65531     External  Established
192.168.5.5     65534    192.168.5.189    65531     External  Active
```

(4) Checking the BGP4 peer group membership of a peer

To check which peer group a peer belongs to, use the `show ip bgp` command with the `neighbors` parameter and a *<Peer Address>* or *<Host name>* argument specified.

Figure 12-34: Results of executing the show ip bgp command (with the neighbors and <Peer Address> parameters specified)

```
>show ip bgp neighbors 172.16.2.2
Date 20XX/07/17 18:45:09 UTC
BGP Peer: 172.16.2.2, Remote AS: 65531
Remote Router ID: 172.16.2.20, Peer Group: INTERNAL-GROUP                  ...1
    BGP Status:Established          HoldTime: 90   , Keepalive: 30
    Established Transitions: 1      Established Date: 20XX/07/16 18:42:26
    BGP Version: 4                  Type: Internal
    Local Address: 172.16.2.214,  Local AS: 65531
    Local Router ID: 172.16.2.100
    Next Connect Retry:-,          Connect Retry Timer: -
    Last Keep Alive Sent: 18:42:20, Last Keep Alive Received: 18:42:20
    BGP Message  UpdateIn UpdateOut TotalIn TotalOut
                 12       14        36      42
    BGP Capability Negotiation: <IPv4-Uni Refresh Refresh(v)>>
      Send    : <IPv4-Uni Refresh Refresh(v)>
      Receive: <IPv4-Uni Refresh Refresh(v)>>
    Password : UnConfigured
```

1. The peer belongs to the peer group `INTERNAL-GROUP`.

12.6.2 Checking communities

(1) List of operation commands

The following table describes the operation commands for communities.

Table 12-26: List of operation commands

Command name	Description
show ip route	Shows routing information stored in a routing table.
show ip bgp	Shows information related to the BGP4 protocol.

(2) Displaying the learned routes in a community

To display routes that have a specific community, use the `show ip bgp` command with the `community` parameter specified.

Figure 12-35: Results of executing the show ip bgp command (with the community parameter specified)

```
> show ip bgp community 1000:1002
Date 20XX/10/20 21:07:32 UTC
Local AS: 65531, Local Router ID: 192.168.1.100
Status Codes: d dampened, * valid, > active, S Stale, r RIB failure
Origin Codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop        MED    LocalPref Weight Path
*> 10.10/16           172.16.2.2      0      -         0      65532 i
*> 10.20/16           172.16.2.2      0      -         0      65532 i
```

To display the communities that a route belongs to, use the `show ip bgp` command with the `route` parameter specified.

Figure 12-36: Results of executing the show ip bgp command (with the route parameter specified)

```
> show ip bgp route 10.10/16
Date 20XX/10/20 21:09:12 UTC
BGP Peer: 172.16.2.2    , Remote AS: 65532
Local AS: 65531, Local Router ID: 192.168.1.100
Status Codes: d dampened, * valid, > active, S Stale, r RIB failure
Route 10.10/16
*> Next Hop 172.16.2.2
     MED: -, LocalPref: 100, Weight: 0, Type: External route
     Origin: IGP, IGP Metric: 0
     Path: 65532
     Communities: 1000:1002
```

(3) Displaying the execution result of learned route filtering

To display the result of learned route filtering based on the `COMMUNITIES` attribute, use the `show ip bgp` operation command.

Figure 12-37: Results of executing the show ip bgp command

```
> show ip bgp
Date 20XX/10/20 21:10:09 UTC
Local AS: 65531, Local Router ID: 192.168.1.100
Status Codes: d dampened, * valid, > active, S Stale, r RIB failure
Origin Codes: i - IGP, e - EGP, ? - incomplete
   Network            Next Hop        MED    LocalPref Weight Path
*> 10.10/16           172.16.2.2      -      120       0      65532 i
```

```
*  10.10/16          10.2.2.2        -      80       0      65533 i
*> 10.20/16          172.16.2.2      -     120       0      65532 i
*  10.20/16          10.2.2.2        -      80       0      65533 i
*> 192.169.10/24     192.168.2.2     -     100       0      i
*> 192.169.20/24     192.168.2.2     -     100       0      i
```

(4) Displaying the advertised routes in a community

To display the `COMMUNITIES` attribute of advertised BGP4 routes, use the `show ip bgp` operation command with the `advertised-routes` parameter specified.

Figure 12-38: Results of executing the show ip bgp command (with the advertised-routes parameter specified)

```
> show ip bgp advertised-routes 192.169.10/24
Date 20XX/10/20 21:10:25 UTC
BGP Peer: 172.16.2.2    , Remote AS: 65532
Local AS: 65531, Local Router ID: 192.168.1.100
Status Codes: d dampened, * valid, > active, S Stale, r RIB failure
Route 192.169.10/24
*> Next Hop 192.168.2.2
     MED: -, LocalPref: -,  Type: Internal route
     Origin: IGP
     Path: 65531
     Next Hop Attribute: 172.16.2.1
     Communities: 1000:1001

BGP Peer: 10.2.2.2     , Remote AS: 65533
Local AS: 65531, Local Router ID: 192.168.1.100
Status Codes: d dampened, * valid, > active, S Stale, r RIB failure
Route 192.169.10/24
*> Next Hop 192.168.2.2
     MED: -, LocalPref: -, Type: Internal route
     Origin: IGP
     Path: 65531
     Next Hop Attribute: 10.1.2.1
     Communities: 1000:1001
```

12.6.3 Checking BGP4 multipath

(1) List of operation commands

The following table describes the operation commands for BGP4 multipath.

Table 12-27: List of operation commands

Command name	Description
show ip route	Shows the routes in a routing table.
show ip bgp	Shows information related to the BGP4 protocol.

(2) Displaying a BGP4 multipath

To display multipath settings, use the `show ip route` operation command.

Figure 12-39: Results of executing the show ip route command

```
> show ip route
Date 20XX/10/20 21:40:39 UTC
Total: 19 routes
Destination       Next Hop        Interface     Metric   Protocol  Age
10.10/16         172.17.2.2      VLAN0006      -/-      BGP       33m 31s  ... 1
                  172.16.2.2      VLAN0005      -        -         -
10.20/16         172.17.2.2      VLAN0006      -/-      BGP       33m 31s  ... 2
                  172.16.2.2      VLAN0005      -        -         -
127/8             ----            loopback0     0/0      Connected 42m 45s
127.0.0.1/32      127.0.0.1       loopback0     0/0      Connected 42m 45s
```

```
172.17/16         172.17.2.2       VLAN0006       0/0      Connected 42m 43s
172.17.2.1/32     172.17.2.2       VLAN0006       0/0      Connected 42m 43s
172.16/16         172.16.2.2       VLAN0005       0/0      Connected 42m 43s
172.16.2.1/32     172.16.2.2       VLAN0005       0/0      Connected 42m 43s
172.10/16         172.17.2.2      VLAN0006       -/-      BGP        3s      ... 3
                  172.16.2.2       VLAN0005       -         -          -
172.20/16         172.17.2.2      VLAN0006       -/-      BGP        3s      ... 4
                  172.16.2.2       VLAN0005       -         -          -
192.168.1.100/32  192.168.1.100    loopback0      0/0      Connected 42m 45s
```

1 to 4: Multiple routes.

12.6.4 Checking capability negotiation

(1) List of operation commands

The following table describes the operation commands for capability negotiation.

Table 12-28: List of operation commands

Command name	Description
show ip bgp	Shows information related to the BGP4 protocol.

(2) Checking negotiation

To display the result of capability negotiation, use the `show ip bgp` operation command with the `neighbors` and `detail` parameters specified.

Figure 12-40: Results of executing the show ip bgp command (with the neighbors and detail parameters specified)

```
> show ip bgp neighbor detail
Date 20XX/10/17 15:52:14 UTC
BGP Peer: 10.1.2.2       , Remote AS: 65531
Remote Router ID: 10.1.2.102
    BGP Status: Established         HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/10/17 15:51:00
    BGP Version: 4                  Type: Internal
    Local Address: 10.1.2.1         Local AS: 65531
    Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:52:00  Last Keep Alive Received: 15:52:00
    BGP Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                 0          0          2          4
    BGP Capability Negotiation: <IPv4-Uni Refresh Refresh(v)>             ...1
      Send    : <IPv4-Uni Refresh Refresh(v)>
      Receive: <IPv4-Uni Refresh Refresh(v)>
    Password: UnConfigured
BGP Peer: 192.168.2.2    , Remote AS: 65531
Remote Router ID: 192.168.1.102
    BGP Status: Established         HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/10/17 15:50:43
    BGP Version: 4                  Type: Internal
    Local Address: 192.168.2.1      Local AS: 65531
    Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:43  Last Keep Alive Received: 15:51:43
    BGP Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                 0          0          2          4
    BGP Capability Negotiation: <IPv4-Uni Refresh>                        ...2
      Send    : <IPv4-Uni Refresh Refresh(v)>
      Receive: <IPv4-Uni Refresh >
    Password: UnConfigured
BGP Peer: 10.2.2.2       , Remote AS: 65533
Remote Router ID: 10.2.2.102
    BGP Status: Established         HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/10/17 15:50:30
    BGP Version: 4                  Type: External
    Local Address: 10.1.2.1         Local AS: 65531
    Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:30  Last Keep Alive Received: 15:51:30
    BGP Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                 0          0          2          4
    BGP Capability Negotiation: <IPv4-Uni>                                ...3
      Send   : <IPv4-Uni Refresh Refresh(v)>
      Receive: <IPv4-Uni>
    Password: UnConfigured
BGP Peer: 172.16.2.2     , Remote AS: 65532
Remote Router ID: 172.16.1.102
    BGP Status: Established         HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/10/17 15:49:35
    BGP Version: 4                  Type: External
    Local Address: 172.16.2.1       Local AS: 65531
    Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:35  Last Keep Alive Received: 15:51:35
    BGP Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                 0          0          3          5
    BGP Capability Negotiation: <>                                        ...4
      Send   : <IPv4-Uni Refresh Refresh(v)>
      Receive: <>
    Password: UnConfigured
>
```

1. The following capabilities were successfully negotiated: `IPv4-Uni`: IPv4 unicast routing; `Refresh`: Route refresh capability (RFC 2918-compliant); and `Refresh(v)`: Route refresh capability (Capability Code: 128).
2. The following capabilities were successfully negotiated: `IPv4-Uni`: IPv4 unicast routing; and `Refresh`: Route refresh capability (RFC 2918-compliant).
3. The following capability was successfully negotiated: `IPv4-Uni`: IPv4 unicast routing.
4. Capabilities were not successfully negotiated.

12.6.5 Checking the route refresh functionality

(1) List of operation commands

The following table describes the operation commands for the route refresh capability.

Table 12-29: List of operation commands

Command name	Description
clear ip bgp	Clears BGP4 sessions or BGP4-related information, or filters inbound or outbound routes using new BGP filter information.
show ip route	Shows the routing information stored in a routing table.
show ip bgp	Shows information related to the BGP4 protocol.

(2) Checking negotiation of route refresh capability

First, make sure that route refresh capability has been successfully negotiated with the BGP4 peer that will be requested to re-advertise its BGP4 routes. To do so, use the `show ip bgp` operation command with the `neighbors` parameter specified. If route refresh capability has not been negotiated, route refresh requests to relearn changed routes will not be sent to the remote peer.

Figure 12-41: Results of executing the show ip bgp command (with the neighbors parameter specified)

```
> show ip bgp neighbors 172.16.2.2
```

```
Date 20XX/10/17 15:52:14 UTC
BGP Peer: 172.16.2.2     , Remote AS: 65532
Remote Router ID: 172.16.1.102
    BGP Status: Established         HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/10/17 15:49:35
    BGP Version: 4                  Type: External
    Local Address: 172.16.2.1       Local AS: 65531
    Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:35  Last Keep Alive Received: 15:51:35
    BGP Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                 1          1          4          6
    BGP Capability Negotiation: <IPv4-Uni Refresh Refresh(v)>             ...1
      Send    : <IPv4-Uni Refresh Refresh(v)>
      Receive: <IPv4-Uni Refresh Refresh(v)>
    Password: UnConfigured
```

1. Route refresh capability with the remote peer was successfully negotiated.

(3) Requesting BGP4 route re-advertisement and re-advertising the routes

To request that all BGP4 peers re-advertise their BGP4 routes, and to re-advertise the routes to all BGP4 peers, use the `clear ip bgp` operation command with the `* both` parameter specified.

Figure 12-42: Results of executing the clear ip bgp command

```
#clear ip bgp * both
```

(4) Checking the relearning and re-advertising of BGP4 routes

To check that BGP4 routes have been re-advertised and relearned by using the route refresh functionality, use the `show ip bgp` command with the `neighbors` parameter specified.

Figure 12-43: Results of executing the show ip bgp command (with the neighbors parameter specified)

```
> show ip bgp neighbors 172.16.2.2
Date 20XX/10/17 15:58:12 UTC
BGP Peer: 172.16.2.2     , Remote AS: 65532
Remote Router ID: 172.16.1.102
    BGP Status: Established         HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/10/17 15:49:35
    BGP Version: 4                  Type: External
    Local Address: 172.16.2.1       Local AS: 65531
    Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:57:35  Last Keep Alive Received: 15:57:35
    BGP Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                 2          2          11         14                    ...1
    BGP Capability Negotiation: <IPv4-Uni Refresh Refresh(v)>
      Send    : <IPv4-Uni Refresh Refresh(v)>
      Receive: <IPv4-Uni Refresh Refresh(v)>
    Password: UnConfigured
```

1. There has been an increase in the numbers of received `UPDATE` messages and sent `UPDATE` messages.

Notes

The `clear ip bgp` operation command (with `* in`, `* out`, or `* both` specified) applies the new route filtering settings and implements the route refresh capability (see *12.4.5 Route refresh capability*). If route refresh capability has not been negotiated, no route refresh requests are made in order to relearn changed routes, although the route filter changes are applied.

12.6.6 Checking TCP MD5 authentication

(1) List of operation commands

The following table describes the operation commands for TCP MD5 authentication.

Table 12-30: List of operation commands

Command name	Description
show ip bgp	Shows information related to the BGP4 protocol.

(2) Checking TCP MD5 authentication

To display whether TCP MD5 authentication has been performed, use the `show ip bgp` operation command with the `neighbors` and `detail` parameters specified.

Figure 12-44: Results of executing the show ip bgp command (with the neighbor and detail parameters specified)

```
> show ip bgp neighbor detail
Date 20XX/10/07 21:24:24 UTC
BGP Peer: 192.168.2.2    , Remote AS: 65531
Remote Router ID: 192.168.2.100
    BGP Status: Established         HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/10/07 21:23:48
    BGP Version: 4                  Type: Internal
    Local Address: 192.168.2.1      Local AS: 65531
    Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 21:23:48  Last Keep Alive Received: 21:23:48
    BGP Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                 0          0          0          3
    BGP Capability Negotiation: <IPv4-Uni Refresh Refresh(v)>
      Send   : <IPv4-Uni Refresh Refresh(v)>
      Receive: <IPv4-Uni Refresh Refresh(v)>
    Password: UnConfigured                                               ...1

BGP Peer: 172.16.2.2     , Remote AS: 65532
Remote Router ID: 172.16.2.100
    BGP Status: Established         HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/10/07 21:23:58
    BGP Version: 4                  Type: External
    Local Address: 172.16.2.1       Local AS: 65531
    Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 21:23:58  Last Keep Alive Received: 21:23:58
    BGP Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                 0          0          1          3
    BGP Capability Negotiation: <IPv4-Uni Refresh Refresh(v)>
      Send   : <IPv4-Uni Refresh Refresh(v)>
      Receive: <IPv4-Uni Refresh Refresh(v)>
    Password: Configured                                                ...2
```

1. MD5 authentication was not used for connection with the peer whose remote peer address is 192.168.2.2.
2. MD5 authentication was used for connection with the peer whose remote peer address is 172.16.2.2.

Notes

A peer relationship is not established if TCP MD5 authentication fails (if the peer's BGP Status is not `Established`). Check the logged messages to check whether TCP MD5 authentication has failed.

12.6.7 Checking BGP4 advertised route generation

(1) List of operation commands

The following table describes the operation commands for displaying information about generated BGP4 advertised routes.

Table 12-31: List of operation commands

Command name	Description
show ip bgp	Shows information related to the BGP4 protocol.
show ip route	Shows the routing information stored in a routing table.

(2) Checking BGP4 advertised routes

(a) Displaying generated BGP4 advertised routes

To display generated BGP4 advertised routes, use the `show ip bgp` operation command. In the following example, 173.16/16 and 192.169.10/24 are generated BGP4 advertised routes.

Figure 12-45: Results of executing the show ip bgp command

```
> show ip bgp
Date 20XX/10/20 22:43:26 UTC
Local AS: 65531, Local Router ID: 192.168.1.100
Status Codes: d dampened, * valid, > active, S Stale, r RIB failure
Origin Codes: i - IGP, e - EGP, ? - incomplete
   Network             Next Hop        MED    LocalPref Weight Path
*  173.16/16           ----            -      100       0      i
*  192.169.10/24       ----            -      100       0      i
```

(b) Checking whether routes have been advertised

To check whether a generated BGP4 advertised route has been advertised, use the `show ip bgp` operation command with the `advertised-routes` parameter specified.

Figure 12-46: Results of executing the show ip bgp command (with the advertised-routes parameter specified)

```
> show ip bgp advertised-routes 173.16/16
Date 20XX/10/20 22:44:54 UTC
BGP Peer: 172.16.2.2     , Remote AS: 65532
Local AS: 65531, Local Router ID: 192.168.1.100
Status Codes: d dampened, * valid, > active, S Stale, r RIB failure
Route 173.16/16
*  Next Hop ----
     MED: -, LocalPref: -, Type: Internal route
     Origin: IGP
     Path: 65531
     Next Hop Attribute: 172.16.2.1

> show ip bgp advertised-routes 192.169.10/24
Date 20XX/10/18 22:44:58 UTC
BGP Peer: 172.16.2.2      , Remote AS: 65532
Local AS: 65531, Local Router ID: 192.168.1.100
Status Codes: d dampened, * valid, > active, S Stale, r RIB failure
Route 192.169.10/24
*  Next Hop ----
     MED: -, LocalPref: -, Type: Internal route
     Origin: IGP
     Path: 65531
     Next Hop Attribute: 172.16.2.1
```

12.6.8 Checking route flap dampening

(1) List of operation commands

The following table describes the operation commands for route flap dampening.

Table 12-32: List of operation commands

Command name	Description
show ip route	Shows the routing information stored in a routing table.
show ip bgp	Shows information related to the BGP4 protocol.
clear ip bgp	Removes the suppressed status of a suppressed route and clears route flap statistics.

(2) Checking route flap dampening

To display routes suppressed by route flap dampening, use the `show ip bgp` operation command with the `dampened-routes` parameter (only for the global network) specified.

Figure 12-47: Results of executing the show ip bgp command (with the dampened-routes parameter specified)

```
>show ip bgp neighbor 172.16.2.2 dampened-routes
Date 20XX/01/17 11:53:14 UTC
Status Codes: d dampened, h history, * valid, > active
   Network            Peer Address   ReUse
 d 172.20.211/24      172.16.2.2     00:07:11                          ...1
 d 172.21.211/24      172.16.2.2     00:19:10                          ...1
```

1. Use of this route has been suppressed by route flap dampening.

To display the flap state of the routes, use the `show ip bgp` operation command with the `flap-statistics` parameter (only for the global network) specified.

Figure 12-48: Results of executing the show ip bgp command (with the flap-statistics parameter specified)

```
>show ip bgp flap-statistics
Date 20XX/01/17 11:56:28 UTC
Status Codes: d dampened, h history, * valid, > active
   Network            Peer Address    Flaps      Duration ReUse    Penalty
 d 172.20.211/24      172.16.2.2      114        00:12:30 00:07:11 5.0
 d 172.21.212/24      172.16.2.2      108        00:12:30 00:19:10 4.0
 h 172.27.119/24      192.168.2.2     2          00:11:20          1.7
 h 172.27.191/24      192.168.2.2     2          00:11:20          1.7
*> 172.30.189/24      192.168.79.188  1          00:05:10          0.6
*> 172.30.192/24      192.168.79.188  3          00:05:10          0.6
>
```

12.6.9 Checking route reflection

(1) List of operation commands

The following table describes the operation commands for route reflection.

Table 12-33: List of operation commands

Command name	Description
show ip route	Shows the routing information stored in a routing table.
show ip bgp	Shows information related to the BGP4 protocol.

(2) Checking route reflection

To display the route reflector clients, use the `show ip bgp` operation command with the `neighbors` and `detail` parameters specified.

Figure 12-49: Results of executing the show ip bgp command (with the neighbors and detail parameters specified)

```
> show ip bgp neighbors detail
Date 20XX/01/17 15:52:14 UTC
BGP Peer: 192.168.2.2       , Remote AS: 65531
Remote Router ID: 192.168.100.2
    BGP Status: Established         HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/01/17 15:51:00
    BGP Version: 4                  Type: Internal RRclient             ...1
    Local Address: 192.168.2.1      Local AS: 65531
    Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:52:00  Last Keep Alive Received: 15:52:00
    BGP Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                 0          0          2          4
    BGP Capability Negotiation: <IPv4-Uni Refresh Refresh(v)>
      Send   : <IPv4-Uni Refresh Refresh(v)>
      Receive: <IPv4-Uni Refresh Refresh(v)>
    Password: UnConfigured

BGP Peer: 192.168.3.2    , Remote AS: 65531
Remote Router ID: 192.168.1.103
    BGP Status: Established         HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/01/17 15:50:43
    BGP Version: 4                  Type: Internal RRclient            ...1
    Local Address: 192.168.3.1      Local AS: 65531
    Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:43  Last Keep Alive Received: 15:51:43
    BGP Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                 0          0          2          4
    BGP Capability Negotiation: <IPv4-Uni Refresh Refresh(v)>
      Send   : <IPv4-Uni Refresh Refresh(v)>
      Receive: <IPv4-Uni Refresh Refresh(v)>
    Password: UnConfigured

BGP Peer: 192.168.4.2    , Remote AS: 65531
Remote Router ID: 192.168.1.104
    BGP Status: Established         HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/01/17 15:50:30
    BGP Version: 4                  Type: Internal RRclient            ...1
    Local Address: 192.168.4.1      Local AS: 65531
    Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:30  Last Keep Alive Received: 15:51:30
    BGP Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                 0          0          2          4
    BGP Capability Negotiation: <IPv4-Uni Refresh>
      Send   : <IPv4-Uni Refresh Refresh(v)>
      Receive: <IPv4-Uni Refresh Refresh(v)>
    Password: UnConfigured

BGP Peer: 172.16.2.2     , Remote AS: 65532
Remote Router ID: 172.16.1.102
    BGP Status: Established         HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/01/17 15:49:35
    BGP Version: 4                  Type: External
    Local Address: 172.16.2.1       Local AS: 65531
    Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:35  Last Keep Alive Received: 15:51:35
    BGP Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                 0          0          3          5
    BGP Capability Negotiation: <IPv4-Uni Refresh Refresh(v)>
      Send   : <IPv4-Uni Refresh Refresh(v)>
      Receive: <IPv4-Uni Refresh Refresh(v)>
    Password: UnConfigured
>
```

1. Specified as a route reflector client.

To display reflected routes, use the `show ip bgp` operation command with the `advertised-routes` parameter specified.

Figure 12-50: Results of executing the show ip bgp command (with the advertised-routes parameter specified)

```
> show ip bgp advertised-routes
Date 20XX/01/17 22:44:54 UTC
BGP Peer: 192.168.3.2       , Remote AS: 65531
Local AS: 65531, Local Router ID: 192.168.1.100
Origin Codes: i - IGP, e - EGP, ? - incomplete
Network             Next Hop        MED    LocalPref Path
192.169.10/24       192.168.2.2     120    100       i
192.169.20/24       192.168.2.2     100    100       i
BGP Peer: 192.168.4.2       , Remote AS: 65531
Local AS: 65531, Local Router ID: 192.168.1.100
Origin Codes: i - IGP, e - EGP, ? - incomplete
Network             Next Hop        MED    LocalPref Path
192.169.10/24       192.168.2.2     120    100       65532 i
192.169.20/24       192.168.2.2     100    100       65532 i
```

12.6.10 Checking confederations

(1) List of operation commands

The following table describes the operation commands for confederations.

Table 12-34: List of operation commands

Command name	Description
show ip route	Shows the routing information stored in a routing table.
show ip bgp	Shows information related to the BGP4 protocol.

(2) Checking confederations

To display a confederation, use the `show ip bgp` operation command with the `neighbors` and `detail` parameters specified.

Figure 12-51: Results of executing the show ip bgp command (with the neighbors and detail parameters specified)

```
> show ip bgp neighbors detail
Date 20XX/01/17 15:52:14 UTC
BGP Peer: 192.168.2.2    , Remote AS: 64512                             ...2
Remote Router ID: 192.168.100.2
    BGP Status: Established         HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/01/17 15:51:00
    BGP Version: 4                  Type: Internal
    Local Address: 192.168.2.1      Local AS: 64512
    Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:52:00  Last Keep Alive Received: 15:52:00
    BGP Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                 0          0          2          4
    BGP Capability Negotiation: <IPv4-Uni Refresh Refresh(v)>
      Send    : <IPv4-Uni Refresh Refresh(v)>
      Receive: <IPv4-Uni Refresh Refresh(v)>
    Password: UnConfigured

Confederation ID: 65531, Member AS: 64512                             ...1
```

```
BGP Peer: 192.168.4.2    , Remote AS: 64513                            ...2
Remote Router ID: 192.168.1.104
    BGP Status: Established         HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/01/17 15:50:30
    BGP Version: 4                  Type: ConfedExt                    ...3
    Local Address: 192.168.4.1      Local AS: 64512
    Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:30  Last Keep Alive Received: 15:51:30
    BGP Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                 0          0          2          4
    BGP Capability Negotiation: <IPv4-Uni Refresh>
      Send    : <IPv4-Uni Refresh Refresh(v)>
      Receive: <IPv4-Uni Refresh Refresh(v)>
    Password: UnConfigured

Confederation ID: 65531, Member AS: 64512                              ...1
BGP Peer: 192.168.5.2    , Remote AS: 64514                            ...2
Remote Router ID: 192.168.1.104
    BGP Status: Established         HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/01/17 15:50:30
    BGP Version: 4                  Type: ConfedExt                    ...3
    Local Address: 192.168.5.1      Local AS: 64512
    Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:30  Last Keep Alive Received: 15:51:30
    BGP Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                 0          0          2          4
    BGP Capability Negotiation: <IPv4-Uni Refresh>
      Send    : <IPv4-Uni Refresh Refresh(v)>
      Receive: <IPv4-Uni Refresh Refresh(v)>
    Password: UnConfigured

BGP Peer: 172.16.2.2     , Remote AS: 65532
Remote Router ID: 172.16.1.102
    BGP Status: Established         HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/01/17 15:49:35
    BGP Version: 4                  Type: External
    Local Address: 172.16.2.1       Local AS: 65531
    Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:35  Last Keep Alive Received: 15:51:35
    BGP Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                 0          0          3          5
    BGP Capability Negotiation: <IPv4-Uni Refresh Refresh(v)>
      Send    : <IPv4-Uni Refresh Refresh(v)>
      Receive: <IPv4-Uni Refresh Refresh(v)>
    Password: UnConfigured
>
```

1. The local router belongs to a member AS of the confederation.
2. Shows the member AS number of a connected peer.
3. The peer type of the connected peer is member AS peer.

12.6.11 Checking graceful restart

(1) List of operation commands

The following table describes the operation commands for a graceful restart.

Table 12-35: List of operation commands

Command name	Description
show ip route	Shows the routing information stored in a routing table.

Command name	Description
show ip bgp	Shows information related to the BGP4 protocol.

(2) Checking the graceful restart functionality

To display whether the graceful restart functionality is in operation, use the `show ip bgp` operation command with the `neighbors` and `detail` parameters specified.

Figure 12-52: Results of executing the show ip bgp command (with the neighbors and detail parameters specified)

```
> show ip bgp neighbors detail
Date 20XX/01/17 15:52:14 UTC
BGP Peer: 192.168.2.2       , Remote AS: 65531
Remote Router ID: 192.168.100.2
    BGP Status: Established         HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/01/17 15:51:00
    BGP Version: 4                  Type: Internal
    Local Address: 192.168.2.1      Local AS: 65531
    Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:52:00  Last Keep Alive Received: 15:52:00
    Graceful Restart: Receive                                             ...1
      Receive Status : Finished    20XX/01/16 19:11:12
       Stalepath-Time: 30

    BGP Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                 0          0          2          4
    BGP Capability Negotiation: <IPv4-Uni Refresh Refresh(v) GracefulRestart >
...2
      Send   : <IPv4-Uni Refresh Refresh(v) GracefulRestart(RestartTime:120s)>
      Receive: <IPv4-Uni Refresh Refresh(v) GracefulRestart(RestartTime:120s)>
    Password: UnConfigured

BGP Peer: 172.16.2.2     , Remote AS: 65532
Remote Router ID: 172.16.1.102
    BGP Status: Established         HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/01/17 15:49:35
    BGP Version: 4                  Type: External
    Local Address: 172.16.2.1       Local AS: 65531
    Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:35  Last Keep Alive Received: 15:51:35
    Graceful Restart: Receive                                             ...1
      Receive Status : Finished    20XX/01/16 19:13:40
       Stalepath-Time: 30
    BGP Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                 0          0          3          5
    BGP Capability Negotiation: <IPv4-Uni Refresh Refresh(v) GracefulRestart >
...2
      Send   : <IPv4-Uni Refresh Refresh(v) GracefulRestart(RestartTime:120s)>
      Receive: <IPv4-Uni Refresh Refresh(v) GracefulRestart(RestartTime:120s)>
    Password: UnConfigured
```

1. The device is acting as a graceful restart receiving router.
2. The peers successfully negotiated graceful restart functionality on connection of the BGP session.

To display routes used during a graceful restart of the originating router, use the `show ip bgp` operation command.

Figure 12-53: Results of executing the show ip bgp command

```
> show ip bgp
Date 20XX/01/17 19:12:23 UTC
Local AS: 65531, Local Router ID: 192.168.1.100
```

```
Status Codes: d dampened, * valid, > active, S Stale, r RIB failure
Origin Codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop      MED   LocalPref Weight Path
S  10.10/16         172.16.2.2    -     120       20     65532 65528 i    ...1
S  10.20/16         172.16.2.2    -     80        20     65532 65528 i    ...1
*> 172.20/16        192.168.2.2   -     100       10     65530    i
*  172.30/16        192.168.2.2   100   100       10     65530 i
*  192.168.10/24    192.168.2.2   -     100       10     65530 i
*> 192.169.10/24    192.168.2.2   -     100       10           i
*> 192.169.20/24    192.168.2.2   -     100       10           i
```

1. Shows the route used during restart of the originating router.

12.6.12 Checking the maximum number of learned BGP4 routes

(1) List of operation commands

The following table describes the operation commands for limiting the number of learned BGP4 routes.

Table 12-36: List of operation commands

Command name	Description
show ip route	Shows the routing information stored in a routing table.
show ip bgp	Shows information related to the BGP4 protocol.
clear ip bgp	Reconnects a peer that was disconnected because it exceeded the maximum number of BGP4 routes that can be learned from a particular device.

(2) Checking the allowable maximum and actual number of BGP4 routes learned from a peer

To check the allowable maximum and the actual number of BGP4 routes (sum of the active and inactive paths) learned from a particular peer, use the `show ip bgp` operation command with the `neighbors` parameter and an <*AS*>, a <*Peer Address*>, or a <*Host name*> argument or the `detail` parameter specified.

Figure 12-54: Results of executing the show ip bgp command (with the neighbors and detail parameters specified)

```
>show ip bgp neighbors detail
 Date 20XX/01/17 18:45:09
 BGP Peer: 172.16.2.2, Remote AS: 65532
 Remote Router ID: 172.16.2.200
     BGP Status: Idle                HoldTime: 90   , Keepalive: 60
     Established Transitions: 1      Established Date: 20XX/01/16 18:42:26...1
     BGP Version: 4                  Type: External
     Local Address: 172.16.23.214,  Local AS: 65531
     Local Router ID: 172.16.2.100
     Next Connect Retry: -,          Connect Retry Timer: -
     Last Keep Alive Sent: 18:42:20, Last Keep Alive Received: 18:42:20
     NLRI of End-of-RIB Marker: Advertised and Received
     BGP Message  UpdateIn UpdateOut TotalIn TotalOut
                  12       14          36    42
     BGP Peer Last Error: Cease(Over Prefix Limit)                        ...2
     BGP Routes  Accepted     MaximumPrefix RestartTime Threshold         ...3
                 0            10000         60m         80%
     BGP Capability Negotiation: <IPv4-Uni>
       Send   : <IPv4-Uni>
       Receive: <IPv4-Uni>
     Password : Configured
 BGP Peer: 192.168.2.1, Remote AS: 65531
 Remote Router ID: 192.168.2.200
     BGP Status: Established         HoldTime: 90   , Keepalive: 60
     Established Transitions: 1      Established Date: 20XX/01/16 18:42:31
```

```
BGP Version: 4                  Type: Internal
Local Address: 192.168.23.214,  Local AS: 65531
Local Router ID: 192.168.2.100
Next Connect Retry: 00:32,      Connect Retry Timer: 00:32
Last Keep Alive Sent: 18:44:31, Last Keep Alive Received: 18:44:31
NLRI of End-of-RIB Marker: Advertised and Received
BGP Message  UpdateIn UpdateOut TotalIn TotalOut
             9        19        51      63
BGP Routes  Accepted     MaximumPrefix RestartTime Threshold        ...4
            942          1000          none        75%
BGP Capability Negotiation: <IPv4-Uni>
  Send   : <IPv4-Uni>
  Receive: <IPv4-Uni>
Password : Configured
```

1. The peer was disconnected at 20XX/01/16 18:42:26.
2. The peer was disconnected because it exceeded the maximum number of learned routes.
3. The disconnected peer was reconnected after 60 minutes.
4. Of the allowable maximum of 1000 routes, the peer has learned 942 routes from the specified peer.

(3) Reconnecting a BGP4 session that was disconnected because the peer exceeded the learned routes limit

To reconnect a BGP4 session that was disconnected because the peer exceeded the maximum number of learned BGP4 routes, use the `clear ip bgp` operation command with the `*` parameter specified or with a *<Peer Address>* or *<Host Name>* argument specified.

Reconnection of a BGP4 session by using the command

1. `# clear ip bgp 172.16.2.2`

   Reconnects the BGP4 session with remote peer address 172.16.2.2, which was disconnected after exceeding the learned routes limit.

Chapter

13. Route Filtering (IPv4)

This chapter provides an overview of IPv4 route filtering and explains how to use it.

13.1 Description of route filtering

13.1.1 Overview of route filtering

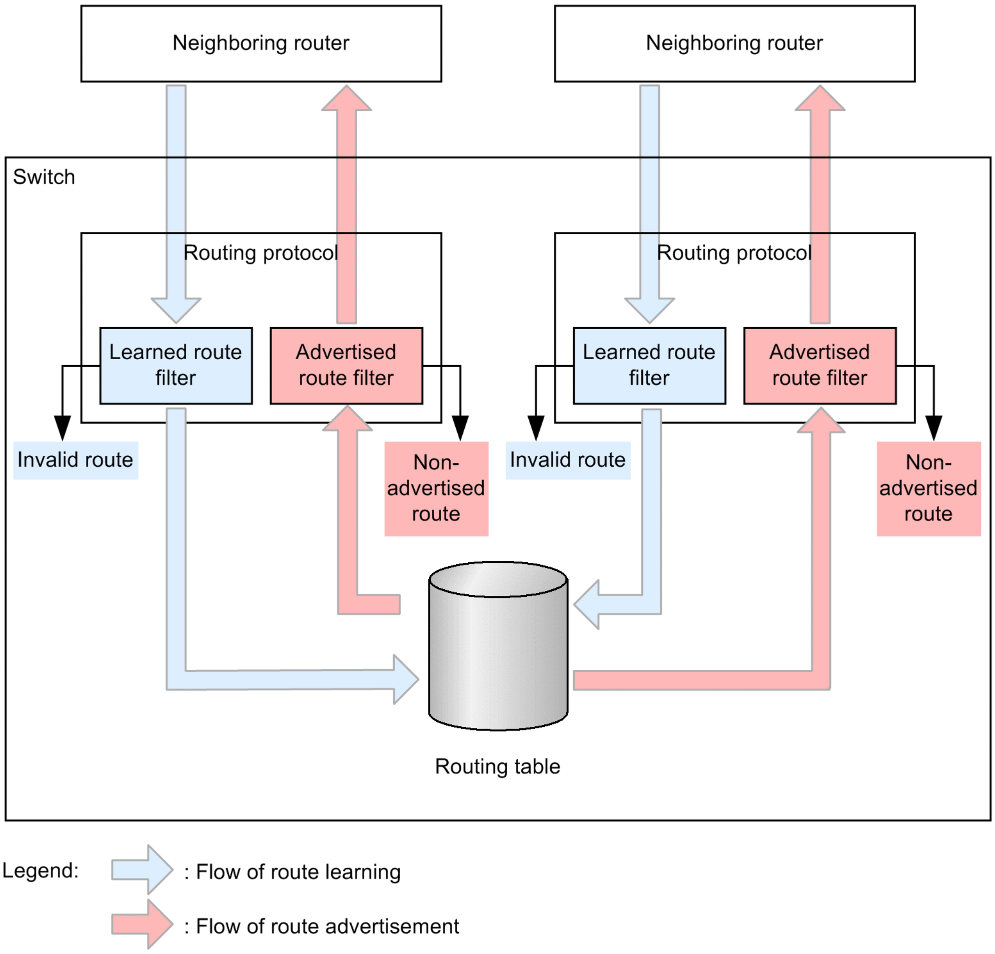
Route filtering allows you to control which routes to accept, by passing routing information through a filter.

There are three types of route filtering: learned route filtering, advertised route filtering, and extranet route filtering.

(1) Learned route filtering and advertised route filtering

The following figure shows the concepts of learned route filtering and advertised route filtering.

Figure 13-1: Concept of route filtering



(a) Learned route filtering

Learned route filtering works between the protocol and routing table to filter the routes the protocol learns. This allows you to control which learned routes the protocol should deem valid, or change the attributes of routes that meet certain conditions.

If you do not configure learned route filtering, all learned routes are considered valid.

(b) Advertised route filtering

Advertised route filtering works between the routing table and protocol to filter the routes in the routing table. This allows you to control which routes a protocol advertises, or change the attributes of advertised routes that meet certain conditions.

If you do not configure advertised route filtering, routes are advertised according to the policy of the associated protocol.

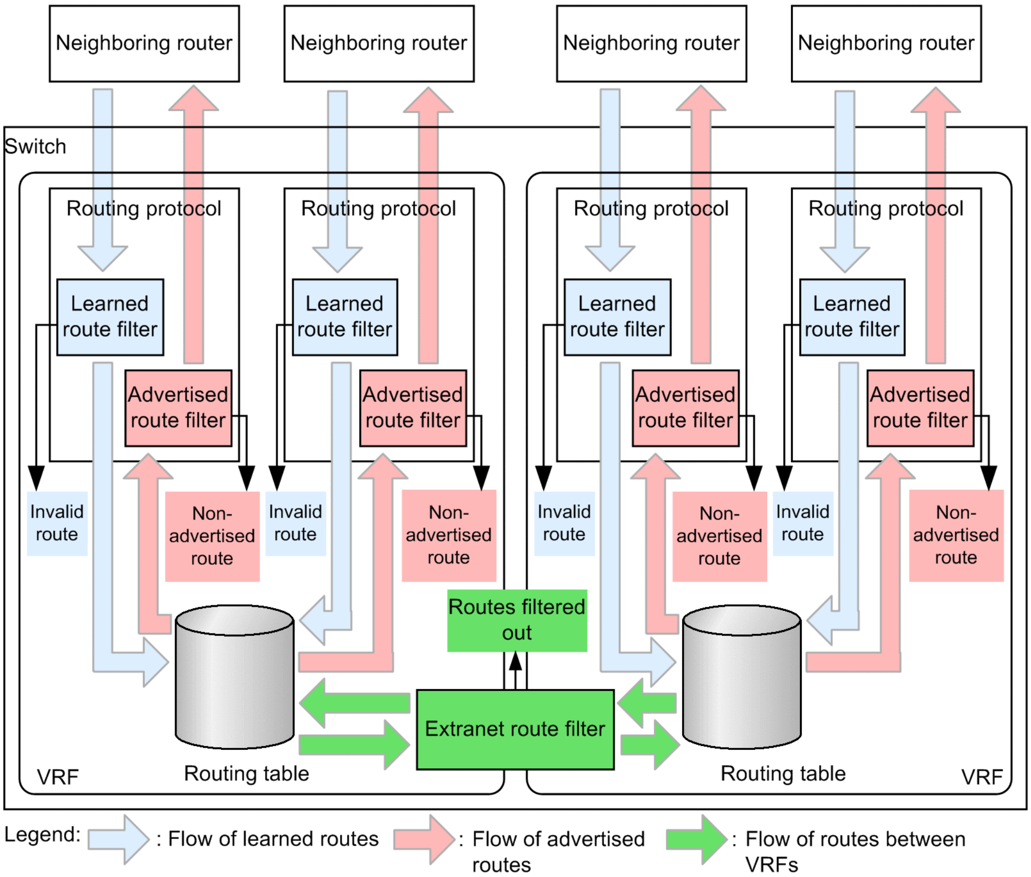
(2) Extranet route filtering [OS-L3SA]

Implementation of an extranet requires technology that allows access between different VRFs. For the Switch, one method for implementing an extranet is to exchange routing information between VRF routing tables. When the routing information is exchanged between VRF routing tables, extranet route filtering is used to filter the routes to be exchanged. Using this filter, you can control whether to exchange the routing information and change the attributes of routing information to be exchanged.

If extranet route filtering has not been set up, routing information is not exchanged between VRFs.

The following figure shows the concept of extranet route filtering.

Figure 13-2: Concept of extranet route filtering



Route filtering performed between VRFs in extranets is called inter-VRF route filtering.

13.1.2 Filtering methods

A filter is a list of conditions to be fulfilled. You apply a learned route or advertised route filter to routing traffic by specifying a filter ID in the route filtering configuration.

There are two main filter types you can use to filter routes in the Switches: `prefix-list` and `access-list`, which filter routing traffic based only on the destination network, and `route-map`, which filters by using most of the key route attributes and allows those attributes to be modified. Other filters include `ip as-path access-list` and `ip community-list`, which filter routes based on BGP4 routing attributes. The `ip as-path access-list` and `ip community-list` filters are called from within `route-map`.

To configure a filter, a filter ID, filter conditions, and the action to take when the conditions are met need to be specified. The actions are `permit` or `deny`.

You can assign multiple filters to a single filter ID. When filtering a piece of routing information, the switch evaluates the filters that match the specified ID in the order in which they appear in the configuration, and then adopts the action of the first filter whose conditions match the route. Filters that can be assigned a sequence number are evaluated in sequence number order. Filters that lack a sequence number are evaluated in the order in which they are configured.

If none of the filter conditions associated with the specified ID match, the process ends with a `deny` action. This is called an implicit deny. Filters that specify conditions always end with an implicit deny statement.

Filters that do not specify any conditions end with a `permit` action.

(1) Filtering by destination network

(a) ip prefix-list

The `ip prefix-list` filter specifies a list of prefixes as filter conditions. When `ip prefix-list` is used as a route filter, the destination network of the route is compared with the prefixes in the filter.

In addition to prefixes, you can specify minimum and maximum mask lengths in the filter conditions. A route matches the conditions if its destination network is within the address range specified in the filter and the mask length of the address is within the specified mask length range. If you do not specify a mask length range, a route matches the filter conditions only when the mask length of the prefix matches exactly. The following table describes examples of `ip prefix-list` comparisons.

Table 13-1: Examples of prefix comparison with ip prefix-list

Compared prefix	**Conditions of ip prefix-list**		
	192.168.0.0/16 **Matches when mask length is 16**	**192.168.0.0/16 ge 16 le 24** **Matches when mask length is between 16 and 24**	**192.168.0.0/16 ge 8 le 24** **Matches when mask length is between 8 and 24**
0.0.0.0/0	N	N	N
192.0.0.0/8	N	N	Y
193.0.0.0/8	N	N	N
192.168.0.0/16	Y	Y	Y
192.169.0.0/16	N	N	N
192.168.43.0/24	N	Y	Y
192.168.42.3/32	N	N	N

Legend: Y: Matches, N: Does not match

An `ip prefix-list` filter can also be referenced as route destination conditions from the `match ip address` command in `route-map`. The same method of comparison applies as if it were used as a standalone route filter.

The `match ip route-source` command in `route-map` can invoke an `ip prefix-list` filter as conditions for the learning source router. In this case, the conditions are the IPv4 address of the learning source router with a 32-bit mask applied.

(b) ip access-list standard

The `ip access-list standard` and `access-list` filters numbered 1 to 99 and 1300 to 1999 are used primarily for packet filtering and to control login access, but can also be used in route filtering.

When `ip access-list standard` is used for route filtering, the address portion of the destination network is compared with the address conditions.

An `ip access-list standard` filter can be referenced by the `match ip address` command in `route-map`. The same method of comparison applies as if it were used as a standalone route filter.

Moreover, the `match ip route source` command in `route-map` can invoke an `ip access-list standard` filter as conditions for the learning source router. In this case, the conditions are compared against the IPv4 address of the learning source router.

(c) ip access-list extended

The `ip access-list extended` and `access-list` filters numbered 100 to 199 and 2000 to 2699 are used primarily to filter packets, but can also be used in route filtering.

When you use `ip access-list extended` for route filtering, the address portion of the destination network is compared with the destination address conditions. All non-address conditions such as upper-layer protocols and port numbers are ignored.

You can reference an `ip access-list extended` filter as route destination conditions from the `match ip address` command in `route-map`. The same method of comparison applies as if it were used as a standalone route filter.

The `match ip route-source` command in `route-map` can invoke an `ip access-list extended` filter as conditions for the learning source router. In this case, the IPv4 address of the route source router is compared with the destination address conditions, and compares the 32-bit mask 255.255.255.255 against the source address conditions.

(2) route-map

A `route-map` filter is used to specify a number of different conditions. This kind of filter can also change route attributes when certain conditions are met.

Statements in `route-map` each have a sequence number. For each sequence number, you can specify one line of filter conditions for each condition. Multiple filter conditions can be specified in that line. The conditions within a given line are related by an OR condition. Conditions that share the same sequence number but appear on different lines are subject to an AND condition.

The statement represented by a sequence number is considered to be satisfied when the route matches every one of its filter conditions. When the conditions are satisfied, the action associated with the sequence number is taken and `route-map` terminates the filter.

If there is even one type of filter conditions for which none of the conditions match, the statement represented by the sequence number is considered not to be satisfied. In this case, the next sequence number in `route-map` is evaluated.

The tables below list the types of filter conditions you can specify in a `route-map` filter and the attributes the filter can change.

Notes

> When a series of `route-map` filters to a route are applied in succession, changes to route attributes will affect route filtering by subsequent `route-map` filters.
>
> Suppose you use the RIP `redistribute` command to apply a `route-map` filter that changes a tag value, and then use the RIP `distribute-list out` command to apply a `route-map` filter that uses that tag value as a condition. First, the tag is modified by the `redistribute` command, and then a comparison is made using the modified tag value when the `route-map` filter of the `distribute-list out` command is applied.

Table 13-2: Types of route-map filter conditions

Route attribute used as conditions	Description	Configuration commands
Destination network	With the ID of a `prefix-list` or `access-list` filter specified as filter conditions, uses the specified filter to filter the destination network of a route. A match is assumed if the filter action is permit. If the action is *deny*, the attribute is assumed not to match.	match ip address ip prefix-list ip access-list
Protocol type	Uses the specified routing protocol name as match conditions for the learning source protocol type of the route.	match protocol
Neighboring routers	With the ID of a `prefix-list` or `access-list` filter specified as its conditions, uses the specified filter to filter the address of the learning source router. A match is assumed if the filter action is permit. If the action is deny, the attribute is assumed not to match. Only RIP routes and BGP4 routes include the address of the learning source router. Other route types cannot match these conditions.	match ip route-source ip access-list ip prefix-list
Interface	Uses interfaces as conditions, and compares the interface with the interface of the next routing network hop. Routes with no next hop do not match the conditions. With BGP4 learned route filtering, routes do not match any interface.	match interface
Tag value	Uses the specified tag value as match conditions for a tag value of the route. Routes with no tags are assumed to have a tag value of 0.	match tag
AS_PATH attribute	With the ID of `ip as-path access-list` specified as filter conditions, uses the specified `ip as-path access-list` to filter the `AS_PATH` attribute of the route. A match is assumed if the action is permit. If the action is deny, the attribute is assumed not to match. Routes with no `AS_PATH` attribute are assumed to have an `AS_PATH` length of 0.	match as-path ip as-path access-list
COMMUNITIES attribute	With the ID of `ip community-list` specified as filter conditions, uses the specified `ip community-list` to filter the `COMMUNITIES` attribute of the route. A match is assumed if the action is permit. If the action is deny, the attribute is assumed not to match. Routes with no `COMMUNITIES` attribute are assumed to lack community affiliations.	match community ip community-list
ORIGIN attribute	Uses the specified value (`IGP`, `EGP`, or `INCOMPLETE`) as a match conditions for the `ORIGIN` attribute of the route. Routes with no `ORIGIN` attribute are assumed to have an IGP origin.	match origin
Route type	Specifies an OSPF route type or `local` (indicating a route generated by the BGP `network` command) as filter conditions and compares with the route's protocol-dependent route type.	match route-type
VRF ID	Uses the specified VRF ID as match conditions for a VRF ID of the route.	match vrf

Note: If the conditions for an interface condition specify an interface that is not used for IPv4 or

IPv6, the interface conditions match any route.

Table 13-3: Route attributes changeable by a route-map filter

Changeable attribute	Description	Configuration commands
Distance	Changes the route priority (distance) in the routing table. Valid only for learned route filtering.	set distance
Metric	Changes the metric or `MED` attribute. Values can be added to or subtracted from as well as replaced. For route filtering in BGP4, the route can inherit the metric of the route to the BGP `NEXT_HOP` attribute.	set metric `set metric-type internal` (inherits the metric of the route to the `NEXT_HOP` attribute)
MED attribute		
Tag value	Changes the tag value of the route.	set tag
LOCAL_PREF attribute	Changes the `LOCAL_PREF` attribute of the route. Values can be added to or subtracted from as well as replaced. Used with BGP4 route filtering.	set local-preference
AS_PATH attribute	Changes the `AS_PATH` attribute of the route. The filter is limited to adding the AS number of the sending peer. Used with route filtering for BGP4 routes learned and advertised by external peers.	set as-path prepend count
COMMUNITIES attribute	Changes the `COMMUNITIES` attribute of the route. The filter can replace, add, and delete communities. Used with BGP4 route filtering.	set community set community-delete
ORIGIN attribute	Changes the `ORIGIN` attribute of the route. Used with BGP4 route filtering.	set origin
OSPF metric type	Changes the metric type. Used with OSPF advertised route filtering.	set metric-type

(3) Other filters

In addition to the filters above, you can apply filters that use BGP4 route attributes as conditions. You use the filters below by referencing them as `route-map` filter conditions.

(a) ip as-path access-list

This filter applies exclusively to the `AS_PATH` attribute. It compares the string representation of the `AS_PATH` attribute against conditions specified by a regular expression. You call this filter with the `match as-path` command in `route-map`. For details about regular expressions, see *(d)Regular expressions*.

The `AS_PATH` attribute is a string of decimal AS numbers separated by spaces.

You cannot specify the path type with the `AS_PATH` attribute as a filter condition. Filtering performed on all path types included in the `AS_PATH` attribute for the AS number is specified as a filter condition. In the following example, a route with the following `AS_PATH` attribute is filtered:

AS_PATH attribute

```
    AS_SEQ: 100 200 300, AS_SET: 1000 2000 3000, AS_CONFED_SEQUENCE: 65001 65002
```

Display format of the AS_PATH attribute for operation commands

```
    100 200 300 {1000 2000 3000} (65001 65002)
```

With the above `AS_PATH` attribute, any of the following AS numbers will match the filter:

- "100 200 300"

- "1000 2000 3000"
- "65001 65002"
- "300 1000"

Note that special characters such as curly brackets ({}) and parentheses (()) are used as regular expressions of path type notation for operation commands and cannot be used to specify a path type.

Because the `AS_SET` attributes are sorted in ascending order when receiving a BGP4 route, the sorting result is filtered.

(b) ip community-list standard

This filter applies exclusively to the `COMMUNITIES` attribute. You can specify multiple communities as filter conditions. The filter matches if the `COMMUNITIES` attribute of the route contains every community you specify. You call this filter with the `match community` command in `route-map`.

(c) ip community-list expanded

This filter applies exclusively to the `COMMUNITIES` attribute. It compares the string representation of the `COMMUNITIES` attribute against conditions specified by a regular expression. You call this filter with the `match community` command in `route-map`. For details about regular expressions, see *(d)Regular expressions*.

The string representation of the `COMMUNITIES` attribute consists of community values converted to character strings and separated by spaces. The values appear in order from smallest to largest. The following table describes the notation used for community values:

Table 13-4: String representations of the COMMUNITIES attribute

Community value	Character string
0xFFFFFF01 (hexadecimal)	no-export
0xFFFFFF02 (hexadecimal)	no-advertise
0xFFFFFF03 (hexadecimal)	local-AS
All other cases	*<AS number>*:*<last 2 octets>* *<AS number>* and *<last 2 octets>* are both written in decimal notation.

(d) Regular expressions

A regular expression is a means of describing a text pattern. You can use regular expressions to represent patterns like repeating strings. Regular expressions can be used as filter conditions for the `AS_PATH` and `COMMUNITIES` attributes.

In regular expressions, you can use simple characters such as numerals, upper- and lower-case letters, and symbols (excluding double quotation marks), and special characters. Simple characters match their equivalent in the character string, as do special characters if preceded with the \ symbol. Each special character represents a pattern. The following table describes the special characters and the patterns they represent:

Table 13-5: Special characters and patterns

Special character	Pattern
.	Represents any single character including spaces.
*	Indicates that the preceding character or set of characters repeats zero or more times.
+	Indicates that the preceding character or set of characters repeats one or more times.

Special character	Pattern
?	Represents 0 or 1 occurrence of the preceding character or set of characters (press **Ctrl** + **V**, and then enter `?` during command entry).
^	Indicates the first character in the string.
$	Indicates the last character in the string.
_	Represents the first or last character of a string, a space, an underscore (_), a comma (`,`), a left parenthesis (`(`) and a simple character, a right parenthesis (`)`) and a simple character, a left curly bracket ({), a right curly bracket (}), a left angled bracket (`<`), or a right angled bracket (`>`).
[ ]	Represents any single character from the character range inside `[]`. Except for the following, special characters act as simple characters within square brackets. `^`: When a caret is used as the first character in square brackets, the expression matches any character except those in the brackets. `-`: Indicates the beginning and end of a character range. Make sure that the character before the hyphen has a lower character code than the character after it. For details about character codes, see *Table 1-3 List of character codes* in the manual *Configuration Command Reference Vol. 1 For Version 11.10*. Example: `[6-8]` matches any one of 6, 7, or 8. `[^6-8]` matches any single character other than 6, 7, or 8.
( )	Indicates a group of characters. You can specify a maximum of nine character groups in a nested structure.
|	Represents an OR condition.
\	Treats a special character preceded by a backslash as a simple character.

The following table describes the priority of operator characters in regular expressions.

Table 13-6: Operator priority in regular expressions

Priority	Character
High	( )
↑	* + ?
↓	Simple characters, `.`, `[`, `]`, `^`, and `$`
Low	|

When you specify a regular expression in a configuration command or operation command, enclose it in double quotation marks (`"`).

Example 1:

```
> show ip bgp aspath-regexp "^$"
```

Example 2:

```
(config)# ip as-path access-list 10 permit "_100_"
```

13.1.3 RIP

(1) RIP learned route filtering

In RIP, you can filter every route that the protocol has learned. Routes denied by the filter are not added to the routing table.

(a) Method and procedure for applying filters

Learned routes are filtered according to conditions specified by the `distribute-list in` command. You can solely filter routes that the protocol learns from a specific interface or router by specifying the interface or router in the command parameters. The table below describes the configuration commands used to filter learned routes in RIP.

When the switch learns a route, it applies the specified filters in the order shown in the table below. If there are no applicable filters or every filter gives a permit result, the route is entered into the routing table as a valid route. The learned route does not enter the routing table if it is denied by even one filter.

Table 13-7: Configuration commands related to RIP learned route filtering

Command name	Parameter	Filtered routes
distribute-list in (RIP)	gateway *<IPv4>*	Filters RIP routes learned from the specified neighboring router.
	<Interface>	Filters RIP routes learned from the specified IPv4 interface.
	None	Filters all learned RIP routes.

(b) Route attributes changeable by learned route filtering

The table below describes the attributes that can be changed by RIP learned route filtering.

The modified metric is used to define route priority in RIP. The modified distance is used to define the relative priority of routing protocols.

Table 13-8: Route attributes changeable by RIP learned route filtering

Attribute	Default value
Distance	The value specified by the distance command (RIP). If no value is specified, 120 is used.
Metric	The attribute of the received route
Tag value	The attribute of the received route

Notes

- We recommend that you do not use methods other than addition to modify the metric. Replacing or subtracting from the metric might cause routing loops and prevent packets from routing correctly.
- You can configure a route filter to change the metric of a route to 16 or greater. However, a RIP route with a metric of 16 or greater will be deemed invalid.
- Changes to metrics made using the `metric-offset` configuration command take effect after learned route filtering has taken place. You can use the `metric-offset` command to further modify a metric that was changed by a route filter. The route will be deemed invalid if its metric is 16 or greater after modification by the `metric-offset` command.
- You can change tag values regardless of the version of RIP that learned the route. However, when advertising the modified route, only RIP version 2 advertises it with its tag information intact.

  The tag can have a maximum value of 4294967295. However, when RIP version 2 advertises the modified route, it uses the lowest 16 bits of the binary expression and discards the rest.

(2) RIP advertised route filtering

The RIP only advertises the prioritized routes in the routing table. However, it does not advertise routes that are subject to a split horizon, or routes that do not meet the conditions for route advertisement in RIP version 1.

If you do not configure advertised route filtering, the protocol will advertise RIP routes and direct routes to RIP interfaces.

Notes

When advertising OSPF or BGP4 routes, configure the switch to change the metric in the course of advertised route filtering, or assign an advertised metric. These routes have a default metric of 16 and would otherwise not be advertised.

(a) Route attributes changeable by advertised route filtering

The following table describes the attributes that can be changed by RIP advertised route filtering.

Table 13-9: Route attributes changeable by RIP advertised route filtering

Attribute	Learning source protocol	Default value
Metric	Directly connected route Summarized route	1
	Static route	Uses the value specified by `default-metric`. If no value is specified, 1 is used.
	RIP Route	Inherits the metric of the routing information.
	OSPF route BGP4 route Route imported from another VRF or a global network	Inherits the metric of the routing information if the `inherit-metric` command is configured. 16 is used if the routing information has no metric. If `inherit-metric` is not configured, the value specified by `default-metric` is used. If neither `inherit-metric` nor `default-metric` are configured, `16` is used as the metric.
Tag value	Common to all protocols	Inherits the tag value of the routing information.

Notes

- When using RIP to advertise a RIP route, we recommend that you do not use methods other than addition to modify the metric. Replacing or subtracting from the metric might cause routing loops and prevent packets from routing correctly.
- You can configure a route filter to change the metric to a value of 16 or greater. However, the protocol will not advertise routes with a metric of 16 or greater.
- Metric changes made using the `metric-offset` configuration command take effect after advertised route filtering has taken place. You can use the `metric-offset` command to further modify a metric that was changed by a route filter. A route will not be advertised if its metric is 16 or greater after modification by the `metric-offset` command.
- Only version 2 of RIP can advertises tag values. If you change a tag to a value greater than 65535, the protocol advertises the lowest 16 bits of the binary expression and discards the rest.

(b) Method and procedure for applying filters

The application of advertised route filtering involves the following steps:

1. First, select the routes to be advertised by RIP. Specify the learning source protocol of the routes you want to advertise. To specify the protocol, use the `redistribute` configuration command. By specifying a route type in the `redistribute` command, you can limit advertised routes to those of a certain type. By specifying `route-map`, you can advertise only those routes that the associated filters permit. The `redistribute` command compares the route attributes in the routing table against the conditions in `route-map`.

RIP routes and directly connected routes of a RIP interface are advertised regardless of whether they are specified in the `redistribute` command.

You can also change the attributes of advertised routes by specifying the new values directly in the `redistribute` command, or by specifying `route-map` in the `redistribute` command that changes the route attributes.

2. The advertised route takes on the default metric of the protocol. If the `redistribute` command changes the metric, the route uses the metric assigned by the command.

   For details about the default metrics for RIP routes, see *Table 13-9: Route attributes changeable by RIP advertised route filtering*.

3. Use the parameters of the `distribute-list out` command to filter the routes selected by the `redistribute` command. If you specify an interface or router in the command parameters, the filter applies only when advertising routes to the designated destination. If you specify a protocol, the filters apply only to routes learned by the specified protocol. The table below describes the configuration command and its parameters.

   When advertising routes to a RIP interface or a specific neighboring router, applicable filters are selected according to the destination and learning source protocol, and applied in the order shown in the table. If there are no applicable filters or every filter gives a permit result, the route is advertised to the specified destination. The route is not advertised to the destination if it is denied by even one filter.

   If you specify `route-map` in the `distribute-list out` command, routes are filtered according to the default advertising attributes and the attributes after modification by the `redistribute` command.

   You can also change an attribute of an advertised route by specifying `route-map` in the `distribute-list out` command that performs the desired change.

Table 13-10: Configuration command used for RIP advertised route filtering

Command name	Parameter	Filtered routes
`distribute-list out` (RIP)	gateway *<IPv4><Protocol>*	Filters routes advertised to a specific neighboring router using a specific protocol.
	gateway *<IPv4>*	Filters routes advertised to a specific neighboring router.
	<Interface>	Filters routes advertised from a specific IPv4 interface.
	<Protocol>	Filters routes matching a specific protocol regardless of their advertising destination.
	None	Filters all routes regardless of their advertising destination.

13.1.4 OSPF [OS-L3SA]

(1) OSPF learned route filtering

Of the routes computed by the SPF algorithm, OSPF only allows external AS routes and NSSA routes to be filtered. External AS routes and NSSA routes that do not pass the filter are entered into the routing table as invalid routes.

Intra-area and inter-area routes are not filtered before entering the routing table.

Even when a route is disabled by the learned route filtering process, corresponding routes are still formed in other routers. This is because the LSA that originates the route is propagated to the other routers within the OSPF domain. Learned route filtering filters the external AS routes and NSSA routes generated from the LSA, but not the LSA itself.

(a) Method and procedure for applying filters

The switch filters the external AS routes and NSSA routes subject to the filters specified in the `distribute-list in` command. The table below describes the configuration command used to filter learned routes in OSPF.

If there are no applicable filters or every filter gives a permit result, the route is entered into the routing table as a valid route. The route is deemed an invalid route if it is denied by a filter.

Table 13-11: Configuration command for OSPF learned route filtering

Command name	Filtered routes
distribute-list in (OSPF)	Filters external AS routes and NSSA routes calculated in the designated OSPF domain.

(b) Route attributes changeable by learned route filtering

The table below describes the attribute that can be changed by OSPF learned route filtering.

OSPF learned route filtering can only change distances. The modified distance is used to define the relative priority of routing protocols.

Table 13-12: Route attributes changeable by OSPF learned route filtering

Attribute	Default value
Distance	The value specified by distance ospf (OSPF). If no value is specified, 110 is used.

(2) OSPF advertised route filtering

In OSPF, a directly connected route of an OSPF interface is advertised as an intra-area or inter-area route. This behavior is outside the control of advertised route filtering.

OSPF routes are also propagated to other routers. This also cannot be controlled by route filtering. This is because an LSA that originates a route is propagated unconditionally regardless of what filters are in place.

You can use advertised route filtering to advertise prioritized routes other than the above to OSPF. Such routes will be advertised as external AS routes or NSSA routes.

If you do not configure advertised route filtering, the OSPF protocol advertises no routes except OSPF routes and directly connected routes of OSPF interfaces.

(a) Route attributes changeable by advertised route filtering

The following table describes the attributes that can be changed by OSPF advertised route filtering.

Table 13-13: External AS route attributes changeable by OSPF advertised route filtering

Attribute	Source protocol	Default value
Metric	Directly connected route	20
	BGP4 route	The value specified by `default-metric` (OSPF). If no value is specified, 1 is used.
	Other	The value specified by `default-metric` (OSPF). If no value is specified, 20 is used.
OSPF route type	Common to all protocols	Type 2 external AS or NSSA route
Tag value	Common to all protocols	Inherits the tag value of the routing information.

Notes

You can configure a route filter to change the metric of a route to 16777215 or greater. However, such a route will not be advertised.

(b) Method and procedure for applying filters

The application of advertised route filtering involves the following steps:

1. First, select the routes to be advertised by OSPF. Specify the learning source protocol of the routes you want to advertise. To specify the protocol, use the `redistribute` configuration command. However, you cannot re-advertise the routes of a given OSPF domain by specifying it in the `redistribute` command.

   By specifying a route type in the `redistribute` command, you can limit advertised routes to those of a certain type. By specifying `route-map`, you can advertise only those routes that the associated filters permit. The `redistribute` command compares the route attributes in the routing table against the conditions in `route-map`.

   You can also change the attributes of advertised routes by specifying the new values directly in the `redistribute` command, or by specifying `route-map` in the `redistribute` command that changes the route attributes.

2. The advertised route takes on the default metric and OSPF route type configured for the protocol. If you used the `redistribute` command to change the attribute value, the route retains the attribute value assigned by the command.

   For details about the default metrics for advertised OSPF routes, see *Table 13-13: External AS route attributes changeable by OSPF advertised route filtering*.

3. Use the parameters of the `distribute-list out` command to filter the routes selected by the `redistribute` command. If you specify a protocol in the command parameters, the filter applies only to routes learned by the specified protocol. The table below describes the configuration command and its parameters.

   When advertising routes to an OSPF domain, applicable filters are selected according to the learning source protocol, and applied in the order shown in the table. The route is advertised if there are no applicable filters or every filter gives a permit result. The route is not advertised if it is denied by even one filter.

   If you specify `route-map` in the `distribute-list out` command, routes are filtered according to the default advertising attributes and the attributes after modification by the `redistribute` command.

   You can also change an attribute of an advertised route by specifying `route-map` in the `distribute-list out` command that performs the desired change.

Notes

If you execute the `match route-type` command while performing advertised route filtering by means of the `distribute-list out` command in step 3, routes will match `external`, and `external 1` or `external 2`. This is because the OSPF route type in the route attribute will already have been rewritten to indicate a type 1 or type 2 external route.

Table 13-14: Configuration command used for OSPF advertised route filtering

Command name	Parameter	Filtered routes
distribute-list out (OSPF)	*<Protocol>*	Filters routes matching a specific protocol regardless of their advertising destination.
	None	Filters all routes regardless of their advertising destination.

13.1.5 BGP4 [OS-L3SA]

(1) BGP4 learned route filtering

In BGP4, you can filter every route that the protocol learns. By default, routes that are denied by the filter are not added to the routing table.

Notes

After you specify or change the BGP4 learned route filtering settings, execute the `clear ip bgp * in` or `clear ip bgp * both` operation command at the appropriate time. Route filtering will operate according to its previous settings until you execute the command.

If you execute `clear ip bgp * in`, the new route filtering settings apply to learned route filtering only. If you execute `clear ip bgp * both`, the new settings apply to learned route filtering and advertised route filtering.

(a) Method and procedure for applying filters

Learned routes are filtered according to the filters specified in the `distribute-list in` and `neighbor in` commands. The filters specified in `neighbor in` apply only to routes learned from specific peers or peers belonging to a specific peer group. The table below describes the configuration commands used in BGP4 learned route filtering.

When the switch learns a route, it applies the configured filters in the order shown in the table below. If there are no applicable filters or every filter gives a permit result, the route is entered into the routing table as a valid route. The learned route is considered an invalid route if it is denied by even one filter.

Table 13-15: Configuration commands for BGP4 learned route filtering

Command name	Parameter	Filtered routes
neighbor in (BGP4) (with `route-map` specified)	*<IPv4>* (peer address)	Filters routes learned from the specified peers.
neighbor in (BGP4) (with `access-list` or `prefix-list` specified)	*<IPv4>* (peer address)	Filters routes learned from the specified peers.
neighbor in (BGP4) (with `route-map` specified)	*<Peer-Group>* (peer group)	Filters routes learned from peers belonging to the specified peer groups.
neighbor in (BGP4) (with `access-list` or `prefix-list` specified)	*<Peer-Group>* (peer group)	Filters routes learned from peers belonging to the specified peer groups.
distribute-list in (BGP4)	None	Filters all routes learned by BGP4.

(b) Route attributes changeable by learned route filtering

The table below describes the attributes that can be changed by BGP4 learned route filtering.

Of the values below, BGP4 uses values other than the distance to select prioritized routes. The distance defines the relative priority of routing protocols.

Table 13-16: Route attributes changeable by BGP4 learned route filtering

Attribute	Default value
Distance	The value specified by `distance bgp`. If no value is specified, the following value is used: Internal peer: 200 External peer: 20 Member AS peer: 200
MED attribute	The attribute value of the received route

Attribute	Default value
LOCAL_PREF attribute	Internal peer: The attribute value of the received route. External peer: The value specified by `bgp default local-preference`. If no value is specified, 100 is used. Member AS peer: The attribute value of the received route.
AS_PATH attribute	The attribute value of the received route
COMMUNITIES attribute	The attribute value of the received route
ORIGIN attribute	The attribute value of the received route

Notes

An AS can only be added to the `AS_PATH` attribute of routes learned from an external peer. You cannot add an AS to the `AS_PATH` attribute of a route learned from an internal peer or a member AS peer.

(2) BGP4 advertised route filtering

In addition to the prioritized route in the routing table, the BGP4 protocol can advertise BGP4 routes that have been superseded by higher-prioritized routes from other routing protocols, and BGP4 routes generated by the `network` command. If a situation arises in which all three types of routes with the same destination network are to be advertised, the protocol selects one route to advertise, in the following order: the prioritized route, the superseded BGP4 route, and then the `network`-generated route.

If you do not configure advertised route filtering, only BGP4 routes are advertised. However, the protocol cannot advertise a route back to the peer from which it learned the route.

Notes

After you specify or change the BGP4 advertised route filtering settings, execute the `clear ip bgp * out` or `clear ip bgp * both` operation command at the appropriate time. Route filtering will operate according to its previous settings until you execute the command.

If you execute `clear ip bgp * out`, the new route filtering settings apply to advertised route filtering only. If you execute `clear ip bgp * both`, the new settings apply to learned route filtering and advertised route filtering.

(a) Route attributes changeable by advertised route filtering

The following table describes the attributes that can be changed by BGP4 advertised route filtering.

Table 13-17: Route attributes changeable by BGP4 advertised route filtering

Attribute	Default value
MED attribute	Differs depending on the destination peer type and learning source protocol. When advertising to an internal peer: If a BGP4 route is used, the metric is inherited. If a non-BGP4 route is used, the value specified by `default-metric` is inherited. If no value is specified, the route is advertised without a `MED` attribute value. When advertising to an external peer: The value specified by `default-metric` is used. If no value is specified, the route is advertised without a `MED` attribute value. When advertising to a member AS peer: If a BGP4 route is used, the metric is inherited. If a non-BGP4 route is used, the value specified by `default-metric` is inherited. If no value is specified, the route is advertised without a `MED` attribute value.

Attribute	Default value
LOCAL_PREF attribute	If a BGP4 route is used, the `LOCAL_PREF` attribute is inherited. If a non-BGP4 route is used, the value specified by `bgp default local-preference` is inherited. If no value is specified, 100 is used. If the advertising destination peer is an external peer, the advertisement will not include a `LOCAL_PREF` attribute.
AS_PATH attribute	The value of the route in the routing table is inherited.
ORIGIN attribute	
COMMUNITIES attribute	

Notes

- If you do not configure `neighbor send-community`, advertisements will not include a `COMMUNITIES` attribute.

(b) Method and procedure for applying filters

The application of advertised route filtering involves the following steps:

1. First, select the routes to be advertised by BGP4. Specify the learning source protocol of the routes you want to advertise. By specifying a route type in the `redistribute` command, you can limit advertised routes to those of a certain type. By specifying `route-map`, you can advertise only those routes that the associated filters permit. The `redistribute` command compares the route attributes in the routing table against the conditions.

   BGP4 routes are advertised regardless of whether you specify the `redistribute` command.

   You can also change the attributes of advertised routes by specifying the new values directly in the `redistribute` command, or by specifying `route-map` in the `redistribute` command that changes the route attributes.

2. The `MED` and `LOCAL_PREF` attributes take on the default values determined by the protocol. If you used the `redistribute` command to change the attribute value, the route retains the attribute value assigned by the command.

   For details about the default attribute values for advertised BGP routes, see *Table 13-17: Route attributes changeable by BGP4 advertised route filtering*.

3. The routes selected by the redistribute command are filtered by applying the filters specified in the `distribute-list out` and `neighbor out` commands. The filters specified in `neighbor out` apply only to routes advertised to specific peers or peers belonging to a specific peer group. If you specify a protocol, the filters apply only to routes learned by the specified protocol. The table below describes the configuration commands and the routes to which they apply.

   When advertising a route to a peer, the switch selects the applicable filters according to the advertising destination and learning source protocol, and then applies them in the order shown in the table. If there are no applicable filters or every filter gives a permit result, the route is advertised to the specified peer. The route is not advertised to the peer if it is denied by even one filter.

   If you specify `route-map` in a `neighbor out` or `distribute-list out` command, filtering takes place according to the default advertising attributes and the attributes after modification by the `redistribute` command.

   You can also change an attribute of an advertised route by specifying `route-map` in the `neighbor out` or `distribute-list out` command that performs the desired change.

Table 13-18: Configuration commands used related to BGP4 advertised route filtering

Command name	Parameter	Filtered routes
neighbor out (BGP4) (with `route-map` specified)	*<IPv4>* (peer address) *<Protocol>*	Filters routes advertised to a specific peer using a specific protocol.
neighbor out (BGP4) (with `access-list` or `prefix-list` specified)	*<IPv4>* (peer address) *<Protocol>*	
neighbor out (BGP4) (with `route-map` specified)	*<IPv4>* (peer address)	Filters routes advertised to a specific peer.
neighbor out (BGP4) (with `access-list` or `prefix-list` specified)	*<IPv4>* (peer address)	
neighbor out (BGP4) (with `route-map` specified)	*<Peer-Group>* (peer group) *<Protocol>*	Filters routes advertised to a member of a specific peer group using a specific protocol.
neighbor out (BGP4) (with `access-list` or `prefix-list` specified)	*<Peer-Group>* (peer group) *<Protocol>*	
neighbor out (BGP4) (with `route-map` specified)	*<Peer-Group>* (peer group)	Filters routes advertised to a member of a specific peer group.
neighbor out (BGP4) (with `access-list` or `prefix-list` specified)	*<Peer-Group>* (peer group)	
distribute-list out (BGP4)	*<Protocol>*	Filters routes matching a specific protocol regardless of their advertising destination.
	None	Filters all routes regardless of their advertising destination.

13.1.6 Extranet [OS-L3SA]

(1) Inter-VRF route filtering

Routes between VRFs can be filtered. Routes denied by the filter are not added to the routing table.

(a) Method of applying filters

The routes between VRFs are filtered by `import inter-vrf`.

The routes permitted by this filtering are added to the routing table. If there are no applicable filters or every filter gives a deny result, the route is not added.

The table below shows the configuration command used to filter the route between VRFs.

Table 13-19: Configuration command used for inter-VRF route filtering

Command name	Filtered routes
import inter-vrf	Filters routes from a VRF specified in `route-map`.

(b) Route attributes changeable by inter-VRF route filtering

The following table describes the changeable attributes of routes that have been imported from another VRF or the global network.

Table 13-20: Changeable attributes used for inter-VRF route filtering

Attribute	Defautl
Distance	210
Tag value	Inherits the value of the route in the routing table.
AS_PATH attribute	

(c) Configuring inter-VRF routes

Apply an inter-VRF route filter. Routes imported from another VRF or the global network are added to the local VRF routing table according to the filter conditions. For an imported route, the VRF ID in the source routing table is also used in the destination routing table. Note that the protocol type of imported routes becomes `extra-vrf`.

If the `match vrf` configuration command is specified in the inter-VRF route filter, the VRF ID is compared against the source routing table. If you do not specify the `match vrf` command, the same filter conditions are applied to all other VRFs and global networks.

(d) Advertising inter-VRF routes by using protocols

If an advertised route filter is specified for a protocol, routes are advertised from the routing table of the VRF where the protocol is running. To use routes imported from another VRF or the global network, specify `extra-vrf` as the protocol in the `redistribute` configuration command.

13.2 Configuration

13.2.1 List of configuration commands

The following table describes the configuration commands for route filtering.

Table 13-21: List of configuration commands

Command name	Description
distribute-list in (BGP4)	Filters which BGP4-learned routes are added to the routing table.
distribute-list in (OSPF)	Filters which OSPF-learned routes are added to the routing table.
distribute-list in (RIP)	Filters which RIP-learned routes are added to the routing table.
distribute-list out (BGP4)	Filters which BGP4 routes are advertised.
distribute-list out (OSPF)	Filters which OSPF routes are advertised.
distribute-list out (RIP)	Filters which RIP routes are advertised.
ip as-path access-list	Configures `access-list` to filter routes based on their `AS_PATH` attribute.
ip community-list	Configures a community list to filter routes based on their `COMMUNITIES` attribute.
ip prefix-list	Configures an IPv4 prefix list.
match as-path	Configures `route-map` to use the `AS_PATH` attribute as filter conditions.
match community	Configures `route-map` to use the `COMMUNITIES` attribute as filter conditions.
match interface	Configures `route-map` to use the interface of the route as filter conditions.
match ip address	Configures `route-map` to use the IPv4 address prefix as filter conditions.
match ip route-source	Configures `route-map` to use the source IPv4 address as filter conditions.
match origin	Configures `route-map` to use the `ORIGIN` attribute as filter conditions.
match protocol	Configures `route-map` to use the routing protocol as filter conditions.
match route-type	Configures `route-map` to use the route type as filter conditions.
match tag	Configures `route-map` to use the tag value as filter conditions.
match vrf	Configures `route-map` to use a VRF as a filter conditions.
neighbor in (BGP4)	Specifies the filter to be used for BGP4 learned route filtering.
neighbor out (BGP4)	Sets the filters used for BGP4 advertised route filtering.
redistribute (BGP4)	Sets the protocol types of routes advertised from BGP4.
redistribute (OSPF)	Sets the protocol types of routes advertised from OSPF.
redistribute (RIP)	Sets the protocol types of routes advertised from RIP.
route-map	Configures `route-map`.
set as-path prepend count	Sets the number of `AS_PATH` numbers added to the routing information.

Command name	Description
set community	Replaces the COMMUNITIES attribute of the route.
set community-delete	Deletes the COMMUNITIES attribute of the route.
set distance	Sets the priority of the routing information.
set local-preference	Sets the LOCAL_PREF attribute of the routing information.
set metric	Assigns a metric to the routing information.
set metric-type	Sets the metric type or metric of the routing information.
set origin	Sets the ORIGIN attribute of the routing information.
set tag	Sets the tag value of the routing information.
access-list#1	Configures an access list to serve as an IPv4 filter.
deny (ip access-list extended)#1	Specifies the conditions by which the IPv4 packet filter denies access.
deny (ip access-list standard)#1	Specifies the conditions by which the IPv4 address filter denies access.
ip access-list extended#1	Configures an access list to serve as an IPv4 packet filter.
ip access-list resequence#1	Re-sequences the sequence numbers that determine the order in which the IPv4 address filter and IPv4 packet filter apply filter conditions.
ip access-list standard#1	Configures an access list to serve as an IPv4 address filter.
permit (ip access-list extended)#1	Specifies the conditions by which the IPv4 packet filter permits access.
permit (ip access-list standard)#1	Specifies the conditions by which the IPv4 address filter permits access.
router rip#2	Configures router settings related to the RIP routing protocol.
router ospf#3	Configures router settings related to the OSPF routing protocol.
router bgp#4	Configures router settings related to the BGP (BGP4 and BGP4+) routing protocol.
import inter-vrf#5	Controls routes that are imported from another VRF or the global network according to the filter.

#1

See *19. Access Lists* in the manual *Configuration Command Reference Vol. 1 For Version 11.10*.

#2

See *11. RIP* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

#3

See *12. OSPF [OS-L3SA]* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

#4

See *13. BGP4 [OS-L3SA]* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

#5

See *30. VRF [OS-L3SA]* in the manual *Configuration Command Reference Vol. 2 For Version*

11.10.

13.2.2 RIP learned route filtering

(1) Learning routes to specific destination networks

Configure RIP to learn RIP routes destined for only 192.168.0.0/16, but disregard RIP routes destined for other networks.

Points to note

To apply learned route filtering, configure the `distribute-list in` command. To filter routes by their destination networks, use an `ip prefix-list` filter.

First, configure an `ip prefix-list` filter to permit routes to the 192.168.0.0/16 address range. When this filter is referenced from the `distribute-list in` command, the learned RIP routes are filtered according to their destination network.

Command examples

1.
```
(config)# ip prefix-list ONLY192168 seq 10 permit 192.168.0.0/
16
```

Configures `prefix-list` to permit only routes in the 192.168.0.0/16 range. Because `ONLY192168` has no other conditions, the filter denies routes with any other destination address or mask length.

2.
```
(config)# router rip
(config-router)# distribute-list prefix ONLY192168 in
```

Applies `ONLY192168` to routes learned by RIP.

(2) Learning routes from specific interfaces and to specific destination networks

Configure RIP to learn only those RIP routes from VLAN 10 that have 192.168.0.0/16 as their destination. Routes learned from interfaces other than VLAN 10 will not be filtered.

Points to note

To apply RIP learned route filtering on a per-interface basis, specify the <*Interface*> in the parameter of the `distribute-list in` command.

First, configure an `ip prefix-list` filter to permit routes to the 192.168.0.0/16 address range. When this filter is referenced from `distribute-list in VLAN 10`, RIP routes learned from VLAN 10 are filtered according to their destination network.

Command examples

1.
```
(config)# ip prefix-list ONLY192168 seq 10 permit 192.168.0.0/
16
```

Configures `prefix-list` to permit only routes in the 192.168.0.0/16 range. Because `ONLY192168` has no other conditions, the filter denies routes with any other destination address or mask length.

2.
```
(config)# router rip
(config-router)# distribute-list prefix ONLY192168 in vlan 10
```

Applies `ONLY192168` to routes learned from VLAN 10.

(3) Filtering learned routes by a combination of tag value and destination network

Configure RIP not to learn routes that have a destination address in the 192.168.0.0/16 range and a tag value other than 15. All other RIP routes will be learned.

Points to note

The example below shows how to use `route-map` to filter a route by an attribute other than its destination network, or to modify some of its attributes. You can reference this `route-map` from the `distribute-list in` command.

First, configure an `ip prefix-list filter` to permit the routes whose prefixes are in the 192.168.0.0/16 range. Next, configure `route-map` to deny any routes that are permitted by this `prefix-list` filter and also have a tag value other than 15.

Finally, by referencing `route-map` from the `distribute-list in` command, you can configure RIP learned route filtering based on a combination of tag value and destination network.

You must be using RIP version 2 to filter based on tag values. Keep in mind that RIP version 1 does not support tags.

Command examples

1.
```
(config)# ip prefix-list PERMIT192168LONGER seq 10 permit
192.168.0.0/16 ge 16 le 32
```

Configures `prefix-list` to permit routes in the 192.168.0.0/16 range.

2.
```
(config)# route-map TAG permit 10
(config-route-map)# match ip address prefix-list
PERMIT192168LONGER
(config-route-map)# match tag 15
(config-route-map)# exit
```

Configures `route-map` to permit routes in the 192.168.0.0/16 range that have a tag value of 15.

3.
```
(config)# route-map TAG deny 20
(config-route-map)# match ip address prefix-list
PERMIT192168LONGER
(config-route-map)# exit
```

Configures `route-map` to deny routes in the 192.168.0.0/16 range if they do not match the conditions associated with sequence number 10.

4.
```
(config)# route-map TAG permit 30
(config-route-map)# exit
```

Configures `route-map` to permit any route that does not match the conditions associated with sequence numbers 10 and 20.

5.
```
(config)# router rip
(config-router)# distribute-list route-map TAG in
```

Applies the above filter to RIP learned route filtering. This means that RIP does not learn routes that are in the 192.168.0.0/16 range and have a tag value other than 15.

(4) Changing distances based on destination networks

Assign a distance 50 to RIP-learned routes whose destination network is in the 192.168.0.0/16 range, to give such routes priority over OSPF routes.

Points to note

First, configure an `ip prefix-list` filter to permit routes in the 192.168.0.0/16 range. Next, configure `route-map` to assign a distance of 50 to routes permitted by the `prefix-list` filter.

Finally, by referencing `route-map` from the `distribute-list in` command, configure RIP learned route filtering to change a route's distance based on its destination network.

Command examples

1.
```
(config)# ip prefix-list PERMIT192168LONGER seq 10 permit
192.168.0.0/16 ge 16 le 32
```

Configures `prefix-list` to permit routes in the 192.168.0.0/16 range.

2.
```
(config)# route-map Distance50 permit 10
(config-route-map)# match ip address prefix-list
PERMIT192168LONGER
(config-route-map)# set distance 50
(config-route-map)# exit
```

Configures `route-map` to assign a distance of 50 to routes in the 192.168.0.0/16 range, and permits those routes.

3.
```
(config)# route-map Distance50 permit 20
(config-route-map)# exit
```

Configures `route-map` to permit routes that do not match the conditions associated with sequence number 10, without changing any of their attributes.

4.
```
(config)# router rip
(config-router)# distribute-list route-map Distance50 in
```

Applies the above filter to RIP learned route filtering. This means that a distance of 50 is assigned to RIP-learned routes in the 192.168.0.0/16 range.

13.2.3 RIP advertised route filtering

(1) Advertising routes associated with specific protocols

Configure the advertisement of static routes and OSPF domain 1 routes by RIP.

Points to note

The example below shows how to use the `redistribute` command to advertise routes that would not be advertised by default. In the `redistribute` command, specify the protocols that you want advertised.

When configuring the advertisement of OSPF routes, you must also specify a metric. OSPF and BGP4 routes cannot be advertised without a metric.

Command examples

1.
```
(config)# router rip
(config-router)# redistribute static
```
Advertises static routes to RIP.

2.
```
(config-router)# redistribute ospf 1 metric 2
```
Advertises OSPF domain 1 routes, assigning them a metric of 2.

(2) Advertising routes by specific protocols to specific destination networks

Configure RIP to advertise static routes, and only those OSPF routes that have a destination network in the 192.168.0.0/16 range.

Points to note

To filter advertised routes based on their learning source protocol, specify `route-map` in the `redistribute` command. Use an `ip prefix-list` filter to supply the destination network conditions for `route-map`.

First, configure an `ip prefix-list` filter to permit only routes in the 192.168.0.0/16 range. Next, configure `route-map` to use this filter as its conditions. Finally, use `redistribute` commands to specify static routes and OSPF routes. In the `redistribute` command for the OSPF routes, specify `route-map`.

Command examples

1.
```
(config)# ip prefix-list ONLY192168 seq 10 permit 192.168.0.0/
16
```
Configures `prefix-list` to permit only routes in the 192.168.0.0/16 range. Because `ONLY192168` has no other conditions, the filter denies routes with any other destination address or mask length.

2.
```
(config)# route-map ONLY192168 permit 10
(config-route-map)# match ip address prefix-list ONLY192168
(config-route-map)# exit
```
Configures `route-map` to permit routes whose destination network is in the 192.168.0.0/16 range.

3.
```
(config)# router rip
(config-router)# redistribute static
```
Configures RIP to advertise static routes.

4.
```
(config-router)# redistribute ospf 1 metric 2 route-map
ONLY192168
```
Configures RIP to filter OSPF domain 1 routes by `ONLY192168` and advertise permitted routes, assigning them a metric of 2.

(3) Suppressing advertisement of routes to specific destination networks

You can prevent RIP from advertising routes destined for the 192.168.0.0/16 address range.

Points to note

The example below shows how to use the `distribute-list out` command to filter advertised routes regardless of their learning source protocol.

First, configure an `ip prefix-list` filter to deny routes to the 192.168.0.0/16 address range. By referencing this filter from the `distribute-list out` command, you can configure RIP to filter learned routes according to their destination network.

Command examples

1.
   ```
   (config)# ip prefix-list OMIT192168 seq 10 deny 192.168.0.0/16
   ```
   Configures `prefix-list` to deny routes in the 192.168.0.0/16 range.

2.
   ```
   (config)# ip prefix-list OMIT192168 seq 100 permit 0.0.0.0/0 ge 0 le 32
   ```
   Configures an `ip prefix-list` filter to permit routes with any destination address and mask length. Because `OMIT192168` has no other conditions, the filter denies routes to 192.168.0.0/16 only.

3.
   ```
   (config)# router rip
   (config-router)# distribute-list prefix OMIT192168 out
   ```
   Configures RIP to apply the `OMIT192168` filter to every route it advertises.

(4) Filtering advertised routes to individual destination interfaces

Configure the Switch to use RIP interface VLAN 10 to advertise routes only to 192.168.0.0/16, and RIP interface VLAN 20 to advertise all other routes. In this scenario, no interface-level filtering is applied to the other RIP interfaces.

Points to note

To apply route filtering at the level of individual RIP interfaces, specify the <*Interface*> in the parameters of the `distribute-list out` command.

First, configure a `prefix-list` filter to permit routes in the 192.168.0.0/16 range, and another to permit any route not in the 192.168.0.0/16 range. Next, specify the `distribute-list out` <*Interface*> command for RIP interfaces VLAN 10 and VLAN 20. In the `distribute-list out` <*Interface*> command, specify the `prefix-list` filter appropriate to that RIP interface.

Command examples

1.
   ```
   (config)# ip prefix-list ONLY192168 seq 10 permit 192.168.0.0/16
   ```
   Configures an `ip prefix-list` to permit routes in the 192.168.0.0/16 range. Because `ONLY192168` has no other conditions, the filter denies routes with any other destination address or mask length.

2.
   ```
   (config)# ip prefix-list OMIT192168 seq 10 deny 192.168.0.0/16
   ```
   Configures an `ip prefix-list` to deny routes in the 192.168.0.0/16 range.

3.
```
(config)# ip prefix-list OMIT192168 seq 100 permit 0.0.0.0/0 ge
0 le 32
```

Configures `prefix-list` to permit routes with any destination address and mask length. Because `OMIT192168` has no other conditions, the filter denies routes to 192.168.0.0/16 only.

4.
```
(config)# router rip
(config-router)# distribute-list prefix ONLY192168 out vlan 10
```

Configures RIP to apply the `ONLY192168` filter to routes advertised from VLAN 10.

5.
```
(config-router)# distribute-list prefix OMIT192168 out vlan 20
```

Configures RIP to apply the `OMIT192168` filter to routes advertised from VLAN 20.

(5) Controlling route advertisement based on tag values

Configure the switch to assign a tag value of 210 to any directly-connected routes to be advertised, and advertise static routes only if they have a tag value of 211. You then configure the switch not to advertise routes that have a tag value of 210 or 211 by RIP. This process prevents RIP advertised routes from looping through the Switch.

You must be using RIP version 2 to filter based on tag values. Keep in mind that RIP version 1 does not support tags.

Points to note

The example below shows how to use `route-map` to filter a route by an attribute other than its destination network, or you wish to change a route attribute other than the metric. You can reference this `route-map` from `redistribute` and `distribute-list out` among other commands.

The commands below do the following: configure `route-map` to set the tag value of directly connected routes to 210, configure `route-map` to permit static routes with a tag value of 211, and configure `route-map` to deny RIP routes with tag values of 210 or 211.

Command examples

1.
```
(config)# route-map ConnectedToRIP permit 10
(config-route-map)# set tag 210
(config-route-map)# exit
```

Configures `route-map` to assign a tag value of 210.

2.
```
(config)# route-map StaticToRIP permit 10
(config-route-map)# match tag 211
(config-route-map)# exit
```

Configures `route-map` to permit routes with a tag value of 211.

3.
```
(config)# route-map RIPToRIP deny 10
(config-route-map)# match tag 210 211
```

```
(config-route-map)# exit
(config)# route-map RIPToRIP permit 20
(config-route-map)# exit
```

Configures `route-map` to deny routes with a tag value of 210 or 211 while permitting all others.

4.
```
(config)# router rip
(config-router)# version 2
(config-router)# redistribute connected route-map
ConnectedToRIP
```

Advertises direct routes into RIP. Specify `ConnectedToRIP` as the advertising conditions.

5.
```
(config-router)# redistribute static route-map StaticToRIP
```

Advertises static routes into RIP. Specify `StaticToRIP` as the advertising conditions.

6.
```
(config-router)# redistribute rip route-map RIPToRIP
```

Advertises RIP routes into RIP. Specify `RIPToRIP` as the advertising conditions.

13.2.4 OSPF learned route filtering [OS-L3SA]

(1) Learning routes to specific destination networks

Configure OSPF to only learn routes to addresses in the 192.168.0.0/16 range.

Points to note

To apply learned route filtering, configure the `distribute-list in` command. To filter routes by their destination networks, use an `ip prefix-list` filter.

First, configure an `ip prefix-list` filter to permit routes to the 192.168.0.0/16 address range. By referencing this filter from the `distribute-list in` command, you can configure OSPF to filter learned routes based on the destination networks.

Command examples

1.
```
(config)# ip prefix-list ONLY192168 seq 10 permit 192.168.0.0/
16
```

Configures `prefix-list` to permit only routes in the 192.168.0.0/16 range. Because `ONLY192168` has no other conditions, the filter denies routes with any other destination address or mask length.

2.
```
(config)# router ospf 1
(config-router)# distribute-list prefix ONLY192168 in
```

Configures OSPF to apply the conditions `ONLY192168` to the OSPF external AS routes and NSSA routes it learns.

(2) Filtering learned routes by tag value

Configure the switch not to learn routes with a tag value of 15. Other routes are learned.

Points to note

The example below shows how to use `route-map` to filter a route by an attribute other than its destination network, or to modify some of its attributes. You can reference this `route-map` from the `distribute-list in` command.

First, configure `route-map` to deny routes with a tag value of 15. Next, configure OSPF learned route filtering by tag value by referencing `route-map` from the `distribute-list in` command.

Command examples

1.
```
(config)# route-map TAG15DENY deny 10
(config-route-map)# match tag 15
(config-route-map)# exit
```
Configures `route-map` to deny routes with a tag value of 15.

2.
```
(config)# route-map TAG15DENY permit 20
(config-route-map)# exit
```
Configures `route-map` to permit routes that do not match the conditions associated with sequence number 10.

3.
```
(config)# router ospf 1
(config-router)# distribute-list route-map TAG15DENY in
```
Applies the filter to OSPF learned route filtering. This configures OSPF not to learn external AS routes and NSSA routes that have a tag value of 15.

(3) Changing distances based on destination networks

Configure OSPF to assign a distance of 150 to external AS routes and NSSA routes whose destination network is in the 192.168.0.0/16 range, thereby giving priority to RIP routes.

Points to note

The example below shows how to use `route-map` if you want to filter a route by an attribute other than its destination network, or to modify some of its attributes. You can reference this `route-map` from the `distribute-list in` command.

First, configure a `prefix-list` filter to permit routes in the 192.168.0.0/16 range. Next, configure `route-map` to assign a distance of 150 to routes permitted by this `prefix-list`.

Finally, by referencing `route-map` from the `distribute-list in` command, configure the switch to change distances based on destination networks when performing OSPF learned route filtering.

Command examples

1.
```
(config)# ip prefix-list PERMIT192168LONGER seq 10 permit
192.168.0.0/16 ge 16 le 32
```
Configures `prefix-list` to permit routes in the 192.168.0.0/16 range.

2.
```
(config)# route-map Distance150 permit 10
(config-route-map)# match ip address prefix-list
PERMIT192168LONGER
(config-route-map)# set distance 150
(config-route-map)# exit
```

Configures `route-map` to assign a distance of 150 to routes in the 192.168.0.0/16 range, and permits those routes.

3.
```
(config)# route-map Distance150 permit 20
(config-route-map)# exit
```

Configures `route-map` to permit routes that do not match the conditions associated with sequence number 10, without changing any of their attributes.

4.
```
(config)# router ospf 1
(config-router)# distribute-list route-map Distance150 in
```

Applies the above filter to OSPF learned route filtering. This configures OSPF to assign a distance of 150 to external AS routes and NSSA routes in the 192.168.0.0/16 range.

13.2.5 OSPF advertised route filtering [OS-L3SA]

(1) Advertising routes associated with specific protocols

Configure OSPF to advertise static routes and RIP routes to OSPF domain 1.

Points to note

The example below shows how to use the `redistribute` command to advertise routes that would not be advertised by default. In the `redistribute` command, specify the protocols that you want advertised.

Command examples

1.
```
(config)# router ospf 1
(config-router)# redistribute static
```

Configures OSPF to advertise static routes.

2.
```
(config-router)# redistribute rip
```

Configures OSPF to advertise RIP routes.

(2) Advertising routes by specific protocols to specific destination networks

You can configure the switch to advertise static routes, and RIP routes whose destination network is 192.168.0.0/16, into OSPF domain 1.

Points to note

To filter advertised routes based on their learning source protocol, specify `route-map` in the `redistribute` command. Configure an `ip prefix-list` filter to supply the destination network conditions for `route-map`, and call the list from the `match ip address` command.

First, configure an `ip prefix-list` filter to permit only routes in the 192.168.0.0/16 range. Next, configure `route-map` to use this filter as its conditions. Finally, use the `redistribute` command to configure the switch to advertise static routes and RIP routes. In the `redistribute` command for the RIP routes, specify `route-map`.

Command examples

1.
```
(config)# ip prefix-list ONLY192168 seq 10 permit 192.168.0.0/
16
```

Configures `prefix-list` to permit only routes in the 192.168.0.0/16 range. Because `ONLY192168` has no other conditions, the filter denies routes with any other destination address or mask length.

2.
```
(config)# route-map ONLY192168 permit 10
(config-route-map)# match ip address prefix-list ONLY192168
(config-route-map)# exit
```

Configures `route-map` to permit routes whose destination network is in the 192.168.0.0/16 range.

3.
```
(config)# router ospf 1
(config-router)# redistribute static
```

Configures OSPF to advertise static routes into OSPF domain 1.

4.
```
(config-router)# redistribute rip route-map ONLY192168
```

Configures OSPF to filter RIP routes by `ONLY192168` and only advertise those routes permitted by the filter.

(3) Suppressing advertisement of routes to specific destination networks

Configure OSPF to advertise static routes and RIP routes to OSPF domain 1 , except for routes destined for the 192.168.0.0/16 address range.

Points to note

The example below shows how to use the `distribute-list out` command to filter advertised routes regardless of their learning source protocol.

First, configure an `ip prefix-list` filter to deny routes to the 192.168.0.0/16 address range. By referencing this filter from the `distribute-list out` command, you can configure the switch to perform advertised route filtering by destination network.

Finally, use the `redistribute` command to configure the switch to advertise static routes and RIP routes.

Command examples

1.
```
(config)# ip prefix-list OMIT192168 seq 10 deny 192.168.0.0/16
```

Configures `prefix-list` to deny routes in the 192.168.0.0/16 range.

2.
```
(config)# ip prefix-list OMIT192168 seq 100 permit 0.0.0.0/0 ge
0 le 32
```

Configures `prefix-list` to permit routes with any destination address and mask length. Because `OMIT192168` has no other conditions, the filter denies routes to 192.168.0.0/16 only.

3.
```
(config)# router ospf 1
(config-router)# distribute-list prefix OMIT192168 out
```
Configures OSPF to filter advertised routes by `OMIT192168`.

4.
```
(config-router)# redistribute static
(config-router)# redistribute rip
```
Configures OSPF to advertise static routes and RIP routes.

(4) Advertising routes between OSPF domains

The procedure below configures the reciprocal exchange of routes between OSPF domain 1 and OSPF domain 2.

Routes associated with OSPF domain 1 are tagged with a value of 1001 and advertised to OSPF domain 2. In turn, the domain 2 does not advertise routes that have a tag value of 1001 to the domain 1. This process prevents routing loops by stopping OSPF domain 2 from advertising OSPF domain 1 routes back to OSPF domain 1.

Similarly, routes associated with OSPF domain 2 are tagged with a value of 1002 and advertised to OSPF domain 1. In turn, the domain 1 does not advertise routes that have a tag value of 1002 to the domain 2.

Points to note

The example below shows how to use `route-map` if you want to filter a route by an attribute other than its destination network, or you wish to change a route attribute other than the metric. You can reference this `route-map` from `redistribute` and `distribute-list out` among other commands.

Configure advertising to OSPF domain 1 by specifying a `route-map` filter to deny routes with a tag value of 1001 and permits the advertisement of all other routes after assigning them a tag value of 1002. Then, specify this filter in the `redistribute` command that configures the advertisement of the domain 1 routes to the domain 2.

Similarly, configure advertisement to OSPF domain 2 by specifying `route-map` to deny routes with a tag value of 1002 but permits the advertisement of all other routes after assigning them a tag value of 1001. Then, specify this filter in the `redistribute` command that configures the advertisement of the domain 2 routes to the domain 1.

Command examples

1.
```
(config)# route-map OSPF2to1 deny 10
(config-route-map)# match tag 1001
(config-route-map)# exit
```
Configures the `route-map` filter `OSPF2to1` to deny routes with a tag value of 1001.

2.
```
(config)# route-map OSPF2to1 permit 20
(config-route-map)# set tag 1002
(config-route-map)# exit
```

Configures route-map to assign a tag value of 1002 to routes that do not satisfy the above conditions.

3.
```
(config)# router ospf 1
(config-router)# redistribute ospf 2 route-map OSPF2to1
(config-router)# exit
```
Configures OSPF to advertise routes in OSPF domain 2 to OSPF domain 1. Specify OSPF2to1 as a filter.

4.
```
(config)# route-map OSPF1to2 deny 10
(config-route-map)# match tag 1002
(config-route-map)# exit
(config)# route-map OSPF1to2 permit 20
(config-route-map)# set tag 1001
(config-route-map)# exit
```
Configures the route-map filter OSPF1to2 to deny routes with a tag value of 1002, and assign a tag value of 1001 to all other routes.

5.
```
(config)# router ospf 2
(config-router)# redistribute ospf 1 route-map OSPF1to2
(config-router)# exit
```
Configures OSPF to advertise routes in OSPF domain 1 to OSPF domain 2. Specify OSPF2to1 as a filter.

13.2.6 BGP4 learned route filtering [OS-L3SA]

(1) Conditional route learning across all peers

Configure the switch to learn BGP4 routes to any destination network except the routes in the 192.168.0.0/16 range.

Points to note

The example below shows how to use the distribute-list in command to apply learned route filtering consistently among all peers. To filter routes by destination network, use an ip prefix-list filter.

First, configure an ip prefix-list filter to deny routes in the 192.168.0.0/16 range. Then, by referencing this filter from the distribute-list in command, configure BGP4 to filter learned routes by destination network.

Command examples

1.
```
(config)# ip prefix-list DENY192168LONGER seq 10 deny
192.168.0.0/16 ge 16 le 32
(config)# ip prefix-list DENY192168LONGER seq 20 permit
0.0.0.0/0 ge 0 le 32
```
Configures prefix-list to deny prefixes in the 192.168.0.0./16 range but permits all other

prefixes.

2. `(config)# router bgp 65531`

   `(config-router)# distribute-list prefix DENY192168LONGER in`

   Configures the switch to apply learned route filtering by the specified `prefix-list` filter to all peers.

3. `(config-router)# end`

   `# clear ip bgp * in`

   Applies the changes to the learned route filtering configuration.

(2) Conditional route learning for individual peers

The following shows how to configure BGP4 to learn routes received from external peers that have an `AS_PATH` attribute of `65532 65533` and are not destined for a private address (10.0.0.0/8, 172.16.0.0/12, or 192.168.0.0/16). A value of 200 is assigned to the `LOCAL_PREF` attribute of learned routes. Other routes are not learned.

Points to note

Use the `neighbor in` command to apply learned route filtering to routes learned from individual BGP4 peers. Use `route-map` if you want to filter a route by conditions other than its destination network, or to modify some of its attributes.

First, configure a `prefix-list` filter to permit private addresses, and an `ip as-path access`-list filter to permit routes with an `AS_PATH` attribute of `65532 65533`. Next, configure `route-map` to combine these two conditions. Finally, set the `neighbor in` command for peers that you want to filter by these conditions.

Command examples

1. `(config)# ip prefix-list PRIVATE seq 10 permit 10.0.0.0/8 ge 8 le 32`

   `(config)# ip prefix-list PRIVATE seq 20 permit 172.16.0.0/12 ge 12 le 32`

   `(config)# ip prefix-list PRIVATE seq 30 permit 192.168.0.0/16 ge 16 le 32`

   Configures `prefix-list` to permit private addresses.

2. `(config)# ip as-path access-list 2 permit "^65532_65533$"`

   Configures an `ip as-path access-list` filter to permit routes with an `AS_PATH` attribute of `65532 65533`.

3. `(config)# route-map BGP65532IN deny 10`

   `(config-route-map)# match ip address prefix-list PRIVATE`

   `(config-route-map)# exit`

   Configures the `route-map` filter `BGP65532IN`, which denies the private addresses.

4.
```
(config)# route-map BGP65532IN permit 20
(config-route-map)# match as-path 2
(config-route-map)# set local-preference 200
(config-route-map)# exit
```

Configures `route-map` to assign a value of 200 to the `LOCAL_PREF` attribute of routes whose `AS_PATH` attribute matches `65532 65533`, and permit those routes. Because `BGP65532IN` has no other conditions, the filter will deny routes that do not match any of the conditions set so far.

5.
```
(config)# router bgp 65531
(config-router)# neighbor 172.17.1.1 remote-as 65532
(config-router)# neighbor 172.17.1.1 route-map BGP65532IN in
```

Configures BGP4 to use the `route-map` filter `BGP65532IN` to filter routes received from external peers.

6.
```
(config-router)# end
# clear ip bgp * in
```

Applies the changes to the learned route filtering configuration.

13.2.7 BGP4 advertised route filtering [OS-L3SA]

(1) Advertising routes of other protocols

Among directly connected and static routes, you can configure BGP4 to advertise only those routes whose destination network is the local AS network (192.169.0.0/16).

Points to note

The example below shows how to use the `redistribute` command to advertise routes that would not be advertised by default. In the `redistribute` command, specify the protocols that you want advertised.

To define conditions for route advertisement, specify `route-map` in the `redistribute` command. Use a `prefix-list` filter to supply the destination network conditions for `route-map`.

Command examples

1.
```
(config)# ip prefix-list PERMIT192169LONGER seq 10 permit
192.169.0.0/16 ge 16 le 32
```

Configures `prefix-list` to permit only routes in the 192.169.0.0/16 range.

2.
```
(config)# route-map PERMIT192169LONGER permit 10
(config-route-map)# match ip address prefix-list
PERMIT192169LONGER
(config-route-map)# exit
```

Configures `route-map` to permit routes in the 192.169.0.0/16 range.

3.
```
(config)# router bgp 65531
```

```
(config-router)# redistribute connected route-map
PERMIT192169LONGER
```

```
(config-router)# redistribute static route-map
PERMIT192169LONGER
```

Configures the `redistribute` command to advertise only those static and directly connected routes permitted by the `route-map` filter `PERMIT192169LONGER`.

4. `(config-router)# end`

   ```
   # clear ip bgp * out
   ```

   Applies the changes to the advertised route filtering configuration.

(2) Changing advertised routes for individual peers

You can restrict which routes are advertised to external peers. This example restricts route advertisement to BGP4 routes received from AS100 that have one AS path, and directly connected routes and static routes for which the local AS network is the destination (192.169.0.0/16). When advertising routes, the switch adds two AS numbers to the `AS_PATH` of peer 172.18.1.1. Only BGP4 routes are advertised to internal peers.

Points to note

The example below shows how to use the `neighbor out` command if you need to apply route filtering to individual peers.

Here, configure a total of four `route-map` filters: one to redistribute static and directly connected routes, one for advertising to the peer 172.18.1.1, one for advertising to external peers other than 172.18.1.1, and one for internal peers.

For static and directly connected routes, configure an `ip prefix-list` filter to permit routes in the 192.169.0.0/16 range, and a `route-map` filter from which to call the filter.

For the peer 172.18.1.1, configure a `route-map` filter that adds two AS numbers to direct and static routes.

For external peers other than 172.18.1.1, configure an `ip as-path access-list` filter to permit routes whose `AS_PATH` attribute contains one AS, and a `route-map` filter that references the filter.

For internal peers, configure a `route-map` filter to permit BGP4 routes and denies all others.

Command examples

1.
   ```
   (config)# ip prefix-list PERMIT192169LONGER seq 10 permit
   192.169.0.0/16 ge 16 le 32
   ```

   ```
   (config)# route-map PERMIT192169LONGER permit 10
   ```

   ```
   (config-route-map)# match ip address prefix-list
   PERMIT192169LONGER
   ```

   ```
   (config-route-map)# exit
   ```

   Configures `route-map` to permit routes in the 192.169.0.0/16 range. This filter is used to redistribute static and direct routes.

2.
   ```
   (config)# ip as-path access-list 1 permit "^[0-9]+$"
   ```

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

```
(config-route-map)# match protocol connected static
(config-route-map)# exit
(config)# route-map BGPEXTOUT permit 20
(config-route-map)# match protocol bgp
(config-route-map)# match as-path 1
(config-route-map)# exit
```

Configures `route-map` to permit direct routes, static routes, and only those BGP routes whose `AS_PATH` attribute contains one AS. This `route-map` filter is used to filter advertisement to external peers.

3.
```
(config)# route-map BGP1721811OUT permit 10
(config-route-map)# match protocol connected static
(config-route-map)# set as-path prepend count 2
(config-route-map)# exit
(config)# route-map BGP1721811OUT permit 20
(config-route-map)# match protocol bgp
(config-route-map)# match as-path 1
(config-route-map)# set as-path prepend count 2
(config-route-map)# exit
```

Configures `route-map` to permit direct routes, static routes, and only those BGP routes whose `AS_PATH` attribute contains one AS, and adds two AS numbers to those routes. This filter is used to filter advertisement to peer 172.18.1.1.

4.
```
(config)# route-map BGPINTOUT permit 10
(config-route-map)# match protocol bgp
(config-route-map)# exit
```

Configures `route-map` to permit only BGP4 routes. This filter is used to filter advertisement to internal peers.

5.
```
(config)# router bgp 65531
(config-router)# redistribute connected route-map 
PERMIT192169LONGER
(config-router)# redistribute static route-map 
PERMIT192169LONGER
```

Configures the `redistribute` command to advertise only those static and directly connected routes permitted by the `route-map` filter `PERMIT192169LONGER`.

6.
```
(config-router)# neighbor 172.17.1.1 remote-as 65532
(config-router)# neighbor 172.17.1.1 route-map BGPEXTOUT out
```

Applies the filter `BGPEXTOUT` when advertising routes to external peers.

7.
```
(config-router)# neighbor 172.18.1.1 remote-as 65533
(config-router)# neighbor 172.18.1.1 route-map BGP1721811OUT
out
```
Applies the filter BGP1721811OUT when advertising routes to external peer 172.18.1.1.

8.
```
(config-router)# neighbor 192.169.1.1 remote-as 65531
(config-router)# neighbor 192.169.1.1 route-map BGPINTOUT out
```
Applies the filter BGPINTOUT when advertising routes to internal peers.

9.
```
(config-router)# end
# clear ip bgp * out
```
Applies the changes to the advertised route filtering configuration.

13.2.8 Extranet [OS-L3SA]

To communicate from a given VRF to a network in a different VRF, configure a route filter so that a specific route in the remote VRF is imported into the local VRF.

(1) Importing a specific VRF route

Configure route filters to allow communication between VRFs. To do this, a VRF 2 route (172.16.1.0/24) is imported into VRF 3, and a VRF 3 route (172.16.3.0/24) is imported to VRF 2.

Points to note

Use `import inter-vrf` to perform inter-VRF route filtering. Use `route-map` to filter routes by VRF ID. Use `prefix-list` to supply the destination network conditions for `route-map`.

Configure `route-map` so that it permits only the VRF 2 route. This `route-map` is referenced from `import inter-vrf` of VRF 3. Next, configure `route-map` to permit only the VRF 3 route. This `route-map` is referenced from `import inter-vrf` of VRF 2.

Command examples

1.
```
(config)# route-map VRF2PERMIT permit 10
(config-route-map)# match vrf 2
(config-route-map)# exit
```
Configures `route-map` to permit the VRF 2 route.

2.
```
(config)# vrf definition 3
(config-vrf)# import inter-vrf VRF2PERMIT
(config-vrf)# exit
```
Applies the filter settings in step 1 to the VRF 3 extranet so that the VRF 2 route is imported into VRF 3.

3.
```
(config)# route-map VRF3PERMIT permit 10
(config-route-map)# match vrf 3
(config-route-map)# exit
```

Configures `route-map` to permit the VRF 3 route.

4.
```
(config)# vrf definition 2
(config-vrf)# import inter-vrf VRF3PERMIT
(config-vrf)# exit
```
Applies the filter settings in step 3 to the VRF 2 extranet so that the VRF 3 route is imported into VRF 2.

Notes

If `route-map` referenced by `import inter-vrf` has not been configured, all routes in the other VRF or the global network are imported. To prevent unwanted routes from being imported, always configure `route-map` first, and then configure `import inter-vrf`.

(2) Advertising routes between VRFs using a protocol

Implement the route (172.16.3.0/24) for VRF 3 into the network for VRF 2. Use OSPF for VRF 2 to advertize the implemented route for VRF 3.

Points to note

Use `import inter-vrf` to perform inter-VRF route filtering. Use `route-map` to filter routes by VRF. Use `prefix-list` to supply the destination network conditions for `route-map`. To advertise a route imported by OSPF from another VRF or the global network, configure `redistribute`.

Configure `route-map` to permit only the VRF 3 route. Next, configure `import inter-vrf` to reference `route-map` so that the VRF 3 route is imported into VRF 2. Finally, configure `redistribute` for the OSPF of VRF 2 to advertise the route imported from the other VRF or global network.

Command examples

1.
```
(config)# ip prefix-list PERMITVRF3 seq 10 permit 172.16.3.0/24
(config)# route-map VRF3TO2 permit 10
(config-route-map)# match vrf 3
(config-route-map)# match ip address prefix-list PERMITVRF3
(config-route-map)# exit
```
Configures `route-map` to permit the VRF 3 route.

2.
```
(config)# vrf definition 2
(config-vrf)# import inter-vrf VRF3TO2
(config-vrf)# exit
```
Applies the filter settings in step 1 to the VRF 2 extranet so that the VRF 3 route is imported into VRF 2.

3.
```
(config)# router ospf 1 vrf 2
(config-router)# redistribute extra-vrf
```
Advertises the route imported from the other VRF or the global network in VRF 2 OSPF domain 1.

Notes

If `route-map` referenced by `import inter-vrf` has not been configured, all routes in the other VRF or the global network are imported. To prevent unwanted routes from being imported, always configure `route-map` first, and then configure `import inter-vrf`.

(3) Changing the distance for a specific VRF

VRF 2 and VRF 3 routes are imported into the global network. For the VRF 2 route only, the distance is set to 150.

Points to note

Use `import inter-vrf` to perform inter-VRF route filtering. Use `route-map` to filter routes by VRF.

First, configure `route-map` to permit the VRF 2 route and changes its distance to `150`. Next, in this `route-map`, specify a setting to permit a VRF 3 route with a different sequence number.

Finally, configure a filter that changes the distance of a specific VRF by ensuring that `import inter-vrf` references `route-map`.

Command examples

1.
```
(config)# route-map VRF2AND3PERMIT permit 10
(config-route-map)# match vrf 2
(config-route-map)# set distance 150
(config-route-map)# exit
```
Configures `route-map` to permit the VRF 2 route and change its distance to `150`.

2.
```
(config)# route-map VRF2AND3PERMIT permit 20
(config-route-map)# match vrf 3
(config-route-map)# exit
```
Configures `route-map` to permit the VRF 3 route.

3.
```
(config)# vrf definition global
(config-vrf)# import inter-vrf VRF2AND3PERMIT
```
Applies the filter settings in steps 1 and 2 to the extranet of the global network so that the VRF 2 and VRF 3 routes are imported into the global network, and the distance of the VRF 2 route is changed to `150`.

Notes

If `route-map` referenced by `import inter-vrf` has not been configured, all routes in the other VRF or the global network are imported. To prevent unwanted routes from being imported, always configure `route-map` first, and then configure `import inter-vrf`.

13.3 Operation

13.3.1 List of operation commands

The following table describes the operation commands for route filtering..

Table 13-22: List of operation commands

Command name	Description
show ip route	Lists the IPv4 unicast routes in the routing table.
show ip rip	Shows information about the RIP protocol.
show ip ospf	Shows information about the OSPF protocol.
show ip bgp	Shows information about the BGP protocol.
clear ip bgp	Clears BGP4 sessions or BGP4-related information, or filters inbound or outbound routes using new BGP filter information.
show ip vrf	Shows the IPv4 information of a VRF.
restart unicast#	Restarts the unicast routing program.
dump protocols unicast#	Outputs the trace information and control table information collected by the unicast routing program to a file.
erase protocol-dump unicast#	Deletes the file of trace information and control table information generated by the unicast routing program.

#

See *8. Routing Protocol Common to IPv4 and IPv6* in the manual *Operation Command Reference Vol. 2 For Version 11.10.*

13.3.2 Checking routes received by RIP (prior to learned route filtering)

To check routes received by RIP, use the `show ip rip` operation command with the `received-routes` parameter specified.

Figure 13-3: Example of displaying RIP-received routes

```
> show ip rip received-routes
Date 20XX/07/14 12:00:00 UTC
Status Codes: * valid, > active, r RIB failure

Neighbor Address: 192.168.1.145
   Destination        Next Hop        Interface      Metric   Tag   Timer
*> 172.10.1/24        192.168.1.145   VLAN0007       1        0     23s
```

Notes

The command output does not include routes that are excluded by learned route filtering or do not have priority under RIP.

13.3.3 Checking OSPF routes computed by SPF [OS-L3SA]

If a filter invalidates an external AS route or NSSA route computed by the SPF algorithm of the OSPF protocol, the route is still entered into the routing table but as an invalid route. You can check the external AS routes and NSSA routes that the OSPF protocol has generated, including invalid routes, by executing the `show ip route` operation command with the `all-routes` and `-T ospf external` parameters specified.

Figure 13-4: Example of displaying OSPF external AS routes and NSSA routes

```
> show ip route all-routes -T ospf external
Date 20XX/07/14 12:00:00 UTC
Status Codes: * valid, > active, r RIB failure
Total: 2 routes
   Destination       Next Hop        Interface   Metric   Protocol  Age
*> 200.1/24          192.168.1.145   VLAN0007    1/1      OSPF ext2 52s, Tag: 10
*  200.200.1/24      192.168.1.145   VLAN0007    1/1      OSPF ext2 52s, Tag:  0
```

13.3.4 Checking routes received by BGP4 (prior to learned route filtering) [OS-L3SA]

To check routes received by the BGP4 protocol, use the `show ip bgp` operation command with the `received-routes` parameter specified.

Figure 13-5: Example of displaying BGP4 received routes

```
> show ip bgp received-routes
Date 20XX/07/14 12:00:00 UTC
BGP Peer: 177.7.7.145    , Remote AS: 1000
Local AS: 200, Local Router ID: 200.201.202.203
Status Codes: d dampened, * valid, > active, S Stale, r RIB failure
Origin Codes: i - IGP, e - EGP, ? - incomplete
   Network              Next Hop        MED    LocalPref Path
*> 200.1/24             192.168.1.145   -      -         1000 i
*  200.200.1/24         192.168.1.145   -      -         1000 i
```

Notes

The command output does not include routes that are excluded by learned route filtering or do not have priority under BGP4.

To include detailed information about route attributes in the command output, use the `show ip bgp` operation command with the `received-routes` and `-F` parameters specified. Use this method to check the `ORIGIN`, `AS_PATH`, `MED`, `LOCAL_PREF`, and `COMMUNITIES` attributes of the routes.

Figure 13-6: Example of displaying detailed information about BGP4 received routes

```
> show ip bgp received-routes -F
Date 20XX/07/14 12:00:00 UTC
BGP Peer: 192.168.1.145    , Remote AS: 1000
Local AS: 200, Local Router ID: 192.168.1.1
Status Codes: d dampened, * valid, > active, S Stale, r RIB failure
Route 200.1/24
*> Next Hop 192.168.1.145
     MED: -, LocalPref: -, Type: External route
     Origin: IGP
     Path: 1000
     Next Hop Attribute: 192.168.1.145
     Communities: 120:200
Route 200.200.1/24
*  Next Hop 192.168.1.145
     MED: -, LocalPref: -, Type: External route
     Origin: IGP
     Path: 1000
     Next Hop Attribute: 192.168.1.145
     Communities: 120:200
```

Notes

The command output does not include routes that are excluded by learned route filtering or do not have priority under BGP4.

13.3.5 Checking routes resulting from route filtering

Routes permitted by learned route filtering are entered into the routing table. You can check the results of learned route filtering by viewing the routes in the routing table.

To display every route in the routing table including invalid routes, execute the `show ip route` operation command with the `all-routes` parameter specified.

Figure 13-7: Example of displaying routes in routing table (including invalid routes)

```
> show ip route all-routes
Date 20XX/07/14 12:00:00 UTC
Status Codes: * valid, > active, r RIB failure
Total: 12 routes
   Destination     Next Hop       Interface    Metric  Protocol    Age    _
*> 127/8           ----           localhost    0/0     Connected    1h 32s |
*> 127.0.0.1/32    127.0.0.1      localhost    0/0     Connected    1h 32s |
*> 172.10.1/24     192.168.1.145  VLAN0007     2/0     RIP         12s     |
*> 192.168.1/24    192.168.1.1    VLAN0007     0/0     Connected    2s     |
   192.168.1/24    192.168.1.1    VLAN0007     1/-     OSPF intra  48m  3s |
*> 192.168.1.1/32  192.168.1.1    VLAN0007     0/0     Connected    1h 31s #
*> 200.1/24        192.168.1.145  VLAN0007     -/-     BGP         11m 26s |
*> 201.110/24      192.168.1.145  VLAN0007     1/1     OSPF ext2   52s     |
*> 200.200.1/24    192.168.1.145  VLAN0007     0/0     Static      46m 58s |
*  200.200.1/24    192.168.1.145  VLAN0007     -/-     BGP         50m 14s |
*  200.200.1/24    192.168.1.145  VLAN0007     1/1     OSPF ext2   48m 52s |
*  200.200.1/24    192.168.1.145  VLAN0007     2/0     RIP         12s    _|
```

#

The characters `*` and `>` at the beginning of an entry signify the following:

*: Signifies a valid route. Its absence indicates an invalid route.

>: Signifies a prioritized route. Only prioritized routes are used for packet transfer.

To check only those routes that were learned by a specific protocol, execute the `show ip route` operation command with the `all-routes` parameter and the protocol specified.

Figure 13-8: Example of displaying routes in the routing table (RIP only, includes invalid routes)

```
> show ip route all-routes rip
Date 20XX/07/14 12:00:00 UTC
Status Codes: * valid, > active, r RIB failure
Total: 2 routes
   Destination        Next Hop        Interface      Metric   Protocol  Age
*> 172.10.1/24        192.168.1.145   VLAN0007       2/0      RIP       12s
*  200.200.1/24       192.168.1.145   VLAN0007       2/0      RIP       12s
```

If the various protocols generate more than one route to the same destination network, you will need to check which protocol supplied the prioritized route, and the priority of each route. The priority of each route is determined by its distance.

To display distances, execute the `show ip route` operation command with the `all-routes` and `-P` parameters specified. The distance is the first value in the `Distance` column at the end of each line.

Figure 13-9: Example of displaying distances of routes in the routing table

```
> show ip route all-routes -P
Date 20XX/07/14 12:00:00 UTC
Status Codes: * valid, > active, r RIB failure
Total: 12 routes
   Destination        Next Hop       Interface    Metric   Protocol    Age
*> 127/8              ----           localhost    0/0      Connected    1h 36m,
Distance: 0/0/0
```

```
*> 127.0.0.1/32       127.0.0.1      localhost    0/0      Connected    1h 36m,
Distance: 0/0/0
*> 172.10.1/24        192.168.1.145  VLAN0007     2/0      RIP         12s,
Distance: 120/0/0
*> 192.168.1/24       192.168.1.1    VLAN0007     0/0      Connected    0s,
Distance: 0/0/0
   192.168.1/24       192.168.1.1    VLAN0007     1/-      OSPF intra  52m 32s,
Distance: -110/1/0
*> 192.168.1.1/32     192.168.1.1    VLAN0007     0/0      Connected    1h 35m,
Distance: 0/0/0
*> 200.1/24           192.168.1.145  VLAN0007     -/-      BGP         12m 37s,
Distance: 20/0/0
*> 201.110/24         192.168.1.145  VLAN0007     1/1      OSPF ext2    6m 11s,
Distance: 110/1/0
*> 200.200.1/24       192.168.1.145  VLAN0007     0/0      Static      50m 27s,
Distance: 2/0/0
*  200.200.1/24       192.168.1.145  VLAN0007     -/-      BGP         54m 43s,
Distance: 20/0/0
*  200.200.1/24       192.168.1.145  VLAN0007     1/1      OSPF ext2   52m 21s,
Distance: 110/1/0
*  200.200.1/24       192.168.1.145  VLAN0007     2/0      RIP         12s,
Distance: 120/0/0
```

To display the distances for routes to a specific destination, execute the `show ip route` operation command with the `all-routes` parameter and the destination network specified. The route's distance is the first value on the `Distance` line in the detailed routing information.

Figure 13-10: Example of displaying routes in the routing table (includes invalid routes, specified destination only)

```
> show ip route all-routes 200.200.1/24
Date 20XX/07/14 12:00:00 UTC
Route codes: *  = active,   + = changed to active recently
            ' ' = inactive, - = changed to inactive recently
             r  = RIB failure

Route 200.200.1/24
Entries 4 Announced 1 Depth 0 <>

* NextHop 192.168.1.145   , Interface   : VLAN0007
     Protocol <Static>
     Source Gateway ----
     Metric/2    : 0/0
     Distance/2/3: 2/0/0
     Tag : 0, Age : 58m 29s
     AS Path : IGP (Id 1)
     Communities: -
     LocalPref : -
     RT State: <Remote Int Active Gateway>

  NextHop 192.168.1.145   , Interface   : VLAN0007
     Protocol <BGP>
     Source Gateway 192.168.1.145
     Metric/2    : -/-
     Distance/2/3: 20/0/0
     Tag : 0, Age :  1h  2m
     AS Path : 1000 IGP (Id 2)
     Communities: -
     LocalPref : 100
     RT State: <Ext Gateway>
```

To check detailed attribute information for routes, use the `show ip route` operation command with the `all-routes` and `-F` parameters specified.

Figure 13-11: Example of displaying routes in the routing table (including invalid routes, and detailed attributes)

```
> show ip route all-routes -F
Date 20XX/07/14 12:00:00 UTC
Status Codes: * valid, > active, r RIB failure
Total: 12 routes
   Destination        Next Hop       Interface    Metric    Protocol    Age
*> 127/8              ----           localhost    0/0       Connected    1h 46m,
Distance: 0/0/0, Tag: 0, AS-Path: IGP (Id 1), Communities: -, LocalPref: -,
<NoAdvise Active Retain Reject>
*> 127.0.0.1/32       127.0.0.1      localhost    0/0       Connected    1h 46m,
Distance: 0/0/0, Tag: 0, AS-Path: IGP (Id 1), Communities: -, LocalPref: -,
<NoAdvise Active Retain>
*> 172.10.1/24        192.168.1.145  VLAN0007     2/0       RIP         19s,
Distance: 120/0/0, Tag: 0, AS-Path: IGP (Id 1), Communities: -, LocalPref: -,<Int
Active Gateway>
*> 192.168.1/24       192.168.1.1    VLAN0007     0/0       Connected    7s,
Distance: 0/0/0, Tag: 0, AS-Path: IGP (Id 1), Communities: -, LocalPref: -,
<Active Retain>
   192.168.1/24       192.168.1.1    VLAN0007     1/-       OSPF intra   1h  2m,
Distance: -110/1/0, Tag: 0, AS-Path: IGP (Id 1), Communities: -, LocalPref: -,
<NotInstall NoAdvise Int Hidden Gateway>
*> 177.7.7.1/32       192.168.1.1    VLAN0007     0/0      Connected    1h 45m,
Distance: 0/0/0, Tag: 0, AS-Path: IGP (Id 1), Communities: -, LocalPref: -,
<NoAdvise Active Retain>
*> 200.1/24           192.168.1.145  VLAN0007     -/-      BGP         12m 57s,
Distance: 20/0/0, Tag: 0, AS-Path: 1000 IGP (Id 3), Communities: 120:200,
LocalPref: 100, <Ext Active Gateway>
*> 201.110.1/24       192.168.1.145  VLAN0007     1/1      OSPF ext2    3m 34s,
Distance: 110/1/0, Tag: 10, AS-Path: IGP (Id 1), Communities: -, LocalPref: -,
<Int Ext Active Gateway>
*> 200.200.1/24       192.168.1.145  VLAN0007     0/0      Static       1h  0m,
Distance: 2/0/0, Tag: 0, AS-Path: IGP (Id 1), Communities: -, LocalPref: -,
<Remote Int Active Gateway>
*  200.200.1/24       192.168.1.145  VLAN0007     -/-      BGP          1h  5m,
Distance: 20/0/0, Tag: 0, AS-Path: 1000 IGP (Id 2), Communities: -, LocalPref:
100, <Ext Gateway>
*  200.200.1/24       192.168.1.145  VLAN0007     1/1      OSPF ext2    1h  2m,
Distance: 110/1/0, Tag: 0, AS-Path: IGP (Id 1), Communities: -, LocalPref: -,<Int
Ext Gateway>
*  200.200.1/24       192.168.1.145  VLAN0007     2/0       RIP         19s,
Distance: 120/0/0, Tag: 0, AS-Path: IGP (Id 1), Communities: -, LocalPref: -,<Int
Gateway>
```

13.3.6 Checking routes prior to advertised route filtering

In basic terms, the routes subject to advertisement are the prioritized routes contained in the routing table. You can check which routes are subject to advertised route filtering by displaying the routes in the routing table.

To display the prioritized routes in the routing table, execute the `show ip route` operation command.

Figure 13-12: Example of displaying routes in the routing table

```
> show ip route
Date 20XX/07/14 12:00:00 UTC
Total: 8 routes
Destination        Next Hop        Interface      Metric   Protocol  Age
127/8              ----            localhost      0/0      Connected  1h 32s
127.0.0.1/32       127.0.0.1       localhost      0/0      Connected  1h 32s
172.10.1/24        192.168.1.145   VLAN0007       2/0      RIP       12s
192.168.1/24       192.168.1.1     VLAN0007       0/0      Connected  2s
192.168.1.1/32     192.168.1.1     VLAN0007       0/0      Connected  1h 31s
200.1/24           192.168.1.145   VLAN0007       -/-      BGP       11m 26s
201.110/24         192.168.1.145   VLAN0007       1/1      OSPF ext2 52s
```

```
200.200.1/24       192.168.1.145   VLAN0007       0/0      Static    46m 58s
```

To limit the command output to prioritized routes learned by a specific protocol, execute the `show ip route` operation command with the protocol specified as a parameter.

Figure 13-13: Example of displaying routes in the routing table (RIP only)

```
> show ip route rip
Date 20XX/07/14 12:00:00 UTC
Total: 5 routes
Destination        Next Hop        Interface      Metric   Protocol  Age
172.10.1/24        192.168.1.145   VLAN0007       2/0      RIP       12s
```

To check detailed attribute information for the prioritized routes in the routing table, execute the `show ip route` operation command with the `-F` parameter specified.

Figure 13-14: Example of displaying routes in the routing table (detailed)

```
> show ip route -F
Date 20XX/07/14 12:00:00 UTC
Total: 8 routes
Destination        Next Hop        Interface      Metric   Protocol  Age
127/8              ----            localhost      0/0      Connected  1h 46m,
Distance: 0/0/0, Tag: 0, AS-Path: IGP (Id 1), Communities: -, LocalPref: -,
<NoAdvise Active Retain Reject>
127.0.0.1/32       127.0.0.1       localhost      0/0      Connected  1h 46m,
Distance: 0/0/0, Tag: 0, AS-Path: IGP (Id 1), Communities: -, LocalPref: -,
<NoAdvise Active Retain>
172.10.1/24        192.168.1.145   VLAN0007       2/0      RIP       19s,
Distance: 120/0/0, Tag: 0, AS-Path: IGP (Id 1), Communities: -, LocalPref: -,<Int
Active Gateway>
192.168.1/24       192.168.1.1     VLAN0007       0/0      Connected  7s,
Distance: 0/0/0, Tag: 0, AS-Path: IGP (Id 1), Communities: -, LocalPref: -,
<Active Retain>
177.7.7.1/32       192.168.1.1     VLAN0007       0/0      Connected  1h 45m,
Distance: 0/0/0, Tag: 0, AS-Path: IGP (Id 1), Communities: -, LocalPref: -,
<NoAdvise Active Retain>
200.1/24           192.168.1.145   VLAN0007       -/-      BGP       12m 57s,
Distance: 20/0/0, Tag: 0, AS-Path: 1000 IGP (Id 3), Communities: 120:200,
LocalPref: 100, <Ext Active Gateway>
201.110.1/24       192.168.1.145   VLAN0007       1/1      OSPF ext2  3m 34s,
Distance: 110/1/0, Tag: 10, AS-Path: IGP (Id 1), Communities: -, LocalPref: -,
<Int Ext Active Gateway>
200.200.1/24       192.168.1.145   VLAN0007       0/0      Static     1h  0m,
Distance: 2/0/0, Tag: 0, AS-Path: IGP (Id 1), Communities: -, LocalPref: -,
<Remote Int Active Gateway>
```

The BGP4 protocol sometimes advertises routes that do not have priority in the routing table. To include non-priority BGP4 routes in the output of the `show ip route` operation command, execute the command with the `all-routes` and `bgp` parameters specified.

Figure 13-15: Example of displaying routes in the routing table (includes invalid routes, BGP only)

```
> show ip route all-routes bgp
Date 20XX/07/14 12:00:00 UTC
Status Codes: * valid, > active, r RIB failure
Total: 12 routes
   Destination   Next Hop       Interface    Metric    Protocol   Age    _
*> 200.1/24      192.168.1.145  VLAN0007     -/-       BGP        11m 26s¯|
*  200.200.1/24  192.168.1.145  VLAN0007     -/-       BGP        50m 14s_|
                                                                          #

#
```

The characters `*` and `>` at the beginning of an entry signify the following:

*: Signifies a valid route. Its absence indicates an invalid route.

>: Signifies a prioritized route. Only prioritized routes are used for packet transfer.

13.3.7 Checking RIP advertised routes

To check the routes that RIP advertises, execute the `show ip rip` operation command with the `advertised-routes` parameter specified. The command output includes the target address, the route advertised to that address, and the attributes of the route. If the target address corresponds to an interface, the command output shows its broadcast address.

Figure 13-16: Example of displaying RIP advertised routes

```
> show ip rip advertised-routes
Date 20XX/10/20 16:47:36 UTC

Target Address: 177.7.7.255
Destination        Next Hop        Interface      Metric   Tag   Age
192.158.1/24       192.158.1.1     VLAN0006       1        0      5s
```

13.3.8 Checking OSPF advertised routes [OS-L3SA]

AS External LSAs and NSSA External LSAs contain the OSPF routes selected for advertisement by advertised route filtering.

To check which of the AS External LSAs have been generated by the Switch, execute the `show ip ospf` operation command with the `database`, `external`, and `self-originate` parameters specified.

Figure 13-17: Example of displaying AS External LSAs (generated by local device)

```
> show ip ospf database external self-originate
Date 20XX/07/14 12:00:00 UTC
Domain: 1
Local Router ID : 200.199.198.197
Area : 0
Address         State Priority Cost  Neighbor   DR              Backup DR
177.7.7.1       BackupDR 1     1     1          1.4.8.0         200.199.198.197

LS Database: AS External Link
Network Address: 192.168.1/24, AS Boundary Router: 200.199.198.197      ...1
    LSID: 192.168.1.0
    Age:  221, Length: 36 , Sequence: 80000001, Checksums: BB9C
     -> Type: 2, Metric: 20, Tag: 00000000, Forward: 0.0.0.0
```

1. `Network Address` (192.168.1/24) is the route's destination network.

To check which of the NSSA External LSAs have been generated by the Switch, execute the `show ip ospf` operation command with the `database`, `nssa`, and `self-originate` parameters specified.

Figure 13-18: Example of displaying NSSA External LSAs

```
> show ip ospf database nssa self-originate
Date 20XX/07/14 12:00:00 UTC
Domain: 1
Local Router ID : 200.199.198.197
Area : 0
Address         State Priority Cost  Neighbor   DR              Backup DR
177.7.7.1       BackupDR 1     1     1          1.4.8.0         200.199.198.197

LS Database: NSSA AS External Link
Network Address: 192.168.1/24, AS Boundary Router: 200.199.198.197      ...1
    LSID: 192.168.1.0
    Age:   39, Length: 36 , Sequence: 80000001, Checksums: 9FB6
     -> Type: 2, Metric: 20, Tag: 00000000, Forward: 0.0.0.0
```

1. `Network Address` (192.168.1/24) is the route's destination network.

13.3.9 Checking BGP4 advertised routes [OS-L3SA]

To check which routes are advertised by BGP4, execute the `show ip bgp` operation command with the `advertised-routes` parameter specified.

Figure 13-19: Example of displaying BGP4 advertised routes

```
> show ip bgp advertised-routes
Date 20XX/07/14 12:00:00 UTC
BGP Peer: 177.7.7.145    , Remote AS: 2000
Local AS: 1000, Local Router ID: 192.168.1.1
Origin Codes: i - IGP, e - EGP, ? - incomplete
Network              Next Hop        MED    LocalPref Path
200.1/24             177.2.2.1       0      -         1000 2100 i
200.200.1/24         177.2.2.1       0      -         1000 2100 i
```

To include detailed information about route attributes in the command output, use the `show ip bgp` operation command with the `advertised-routes` and `-F` parameters specified. Use this method to check the `ORIGIN`, `AS_PATH`, `MED`, `LOCAL_PREF`, and `COMMUNITIES` attributes, of the routes.

Figure 13-20: Example of displaying BGP4 advertised routes (detailed)

```
> show ip bgp advertised-routes -F
Date 20XX/07/14 12:00:00 UTC
BGP Peer: 177.7.7.145    , Remote AS: 2000
Local AS: 1000, Local Router ID: 192.168.1.1
Status Codes: d dampened, * valid, > active, S Stale, r RIB failure
Route 200.1/24
*> Next Hop 177.2.2.1
     MED:0, LocalPref: -, Type: External route
     Origin: IGP
     Path: 1000 2100
     Next Hop Attribute: 177.2.2.1
     Communities: 1020:1200
Route 110.10/24
*> Next Hop 2.2.2.2
     MED: 0, LocalPref: -, Type: External route
     Origin: IGP
     Path: 1000 2100
     Next Hop Attribute: 177.2.2.1
     Communities: 1020:1200
```

13.3.10 Checking extranet [OS-L3SA]

To display only the imported routes, specify `extra-vrf` as the protocol in the `show ip route` operation command.

Figure 13-21: Example of show ip route command output

```
> show ip route vrf 2 extra-vrf
Date 20XX/12/20 12:00:00 UTC
Total: 1 routes
Destination        Next Hop        Interface      Metric  Protocol   Age
172.16.3.0/24      10.3.1.1        VLAN0030       0/0     Extra-Vrf  365d
>
> show ip route vrf 3 extra-vrf
Date 20XX/12/20 12:00:00 UTC
Total: 1 routes
Destination        Next Hop        Interface      Metric  Protocol   Age
172.16.1.0/24      10.1.1.1        VLAN0010       0/0     Extra-Vrf  365d
```

Chapter

14. Description of IPv4 Multicasting

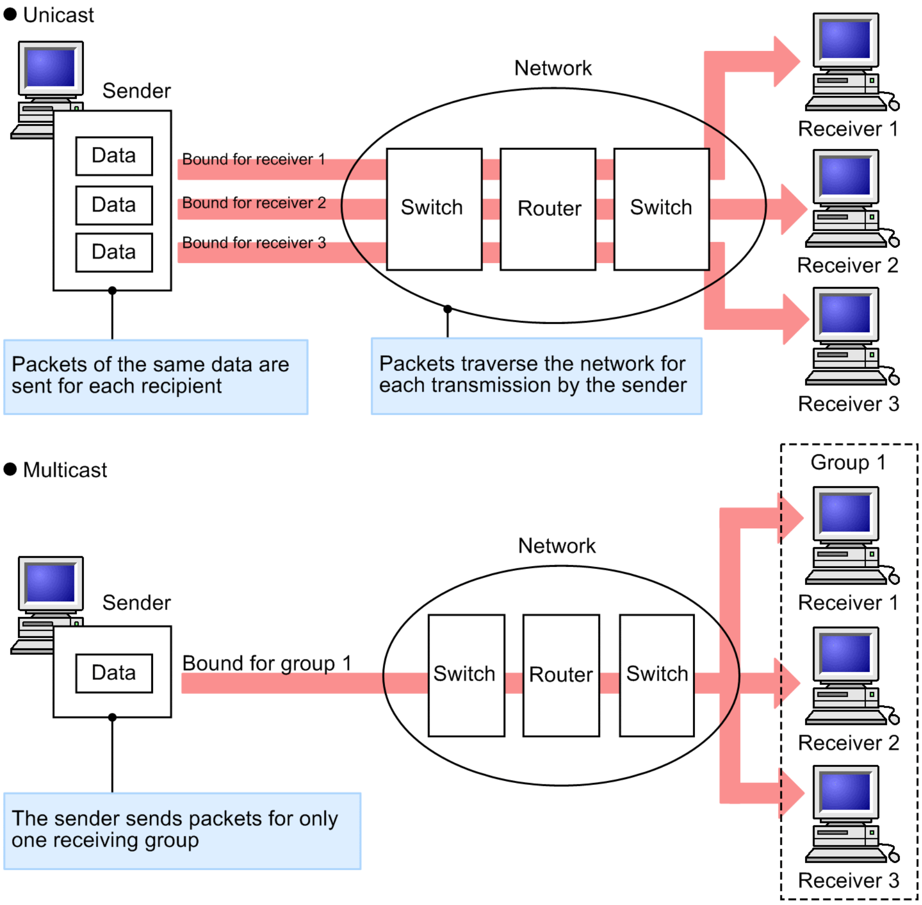
Multicasting is a term used to describe sending the same information to selected groups within a network. This chapter describes multicasting as implemented by IPv4 networks.

14.1 Overview of IPv4 multicasting

Using multiple unicasts to send the same information increases load on both the sender and the network. In comparison, because multicasting sends the same information to selected groups within a network without the sender needing to replicate data for each receiver, the network load can be reduced regardless of the number of receivers.

The following diagram provides an overview of multicasting.

Figure 14-1: Overview of multicasting (IPv4)



Packets of the same data are sent for each recipient

Packets traverse the network for each transmission by the sender

The sender sends packets for only one receiving group

14.1.1 IPv4 multicast addresses

Class D IP addresses are used for multicast communication. A multicast address is a logical group address that exists only within a group participating in the sending and receiving of multicast data. Addresses range from 224.0.0.0 to 239.255.255.255. However, 224.0.0.0 to 224.0.0.255 are reserved. The following figure shows the multicast address format.

Figure 14-2: Multicast address format

0	1	2	3	4 ... 31
1	1	1	0	Multicast Group ID

14.1.2 IPv4 multicast routing functionality

The Switch forwards received multicast packets based on multicast forwarding entries. Multicast routing functionality consists of the following three types of functionality:

- Multicast group management functionality

  This functionality sends and receives group membership information to learn about the existence of multicast groups. For the Switch, IGMP (Internet Group Management Protocol) is used.

- Route control function

  This functionality sends and receives routing information to determine forwarding paths and to create multicast routing information and multicast forwarding entries. PIM-SM (including PIM-SSM) is used to collect routing information.

- Forwarding function

  This functionality forwards multicast packets using hardware and software based on multicast forwarding entries.

14.2 IPv4 multicast group management functionality

Multicast group management functionality learns about the existence of multicast group members on networks to which the router is directly connected by sending and receiving group membership information between routers and hosts. The Switch supports IGMP as the management protocol for implementing multicast group management functionality.

IGMP is the multicast group management protocol used between routers and hosts. The router uses queries to join multicast groups and reports of joining/leaving multicast groups from hosts to determine when a host multicast group is joined or left. The router then uses this information to forward or isolate multicast packets.

IGMPv3 is a protocol extended from IGMPv2 to implement IPv4 multicast group management functionality by introducing functionality to filter senders so that multicast packets are only received from specified senders. Because senders can be specified when the joining or leaving of an IPv4 multicast group is reported, IGMPv3 and PIM-SSM can be used in combination to implement more efficient IPv4 multicast forwarding.

The format and setting values for IGMPv2 messages sent by the Switch follows RFC 2236. Likewise, the format and setting values for IGMPv3 messages follow RFC 3376.

14.2.1 Supported specifications for IGMP messages

(1) Supported specifications for IGMPv2 messages

The following table describes the specifications that the Switch supports for IGMPv2 messages.

Table 14-1: Supported specifications for IGMPv2 messages

Type		Meaning	Supported	
			Sending	Receiving
Membership Query		Multicast group join query	--	--
--	General Query	To all groups	Y	Y
	Group-Specific Query	To a specific group	Y	Y
Version2 Membership Report		Report to joined multicast groups (IGMPv2-compatible)	N	Y
Leave Group		Report of multicast group departure	N	Y
Version1 Membership Report		Report to joined multicast groups (IGMPv1-compatible)	N	Y

Legend: Y: Supported, N: Not supported, --: Not applicable

(2) Supported specifications for IGMPv3 message

IGMPv3 enables source filtering functionality by specifying a filter mode and sender list. There are two filter modes:

- INCLUDE: Only packets from senders included in the specified sender list are forwarded.
- EXCLUDE: Only packets from senders that are not included in the specified sender list are forwarded.

The following table describes the specifications that the Switch supports for IGMPv3 messages.

Table 14-2: Supported specifications for IGMPv3 messages

Type		Meaning	Supported	
			Send ing	Rece iving
Version 3 Multicast Membership Query	General Query	IPv4 multicast group join request (to all groups)	Y	Y
	Group-Specific Query	IPv4 multicast group join request (to a specific group)	Y	Y
	Group-and-Source-Specific Query	IPv4 multicast group join request (to a specific sender and group)	Y	Y
Version 3 MulticastMembership Report	Current StateReport	Filter mode report to joined IPv4 multicast groups	N	Y
	State ChangeReport	Filter mode update report to joined IPv4 multicast groups	N	Y

Legend: Y: Supported, N: Not supported

The filter mode and sender list can be changed after a group is joined, by specification in a Group Record within a Report message. The following table describes the Group Record types supported by the Switch.

Table 14-3: Group Record types

Type		Meaning	Supported
Current State Report	MODE_IS_INCLUDE	Indicates that the mode is INCLUDE	Y
	MODE_IS_EXCLUDE	Indicates that the mode is EXCLUDE	Y (the sender list is disregarded)
State Change Report	CHANGE_TO_ INCLUDE_MODE	Indicates that the filter mode has changed to INCLUDE	Y
	CHANGE_TO_ EXCLUDE_MODE	Indicates that the filter mode has changed to EXCLUDE	Y (the sender list is disregarded)
	ALLOW_NEW_ SOURCES	Indicates that a sender wishing to receive data has been added	Y
	BLOCK_OLD_ SOURCES	Indicates that a sender wishing to receive data has been deleted	Y

Legend: Y: Supported

14.2.2 IGMP operation

The following explains IGMPv2 operation using IGMPv2 messages:

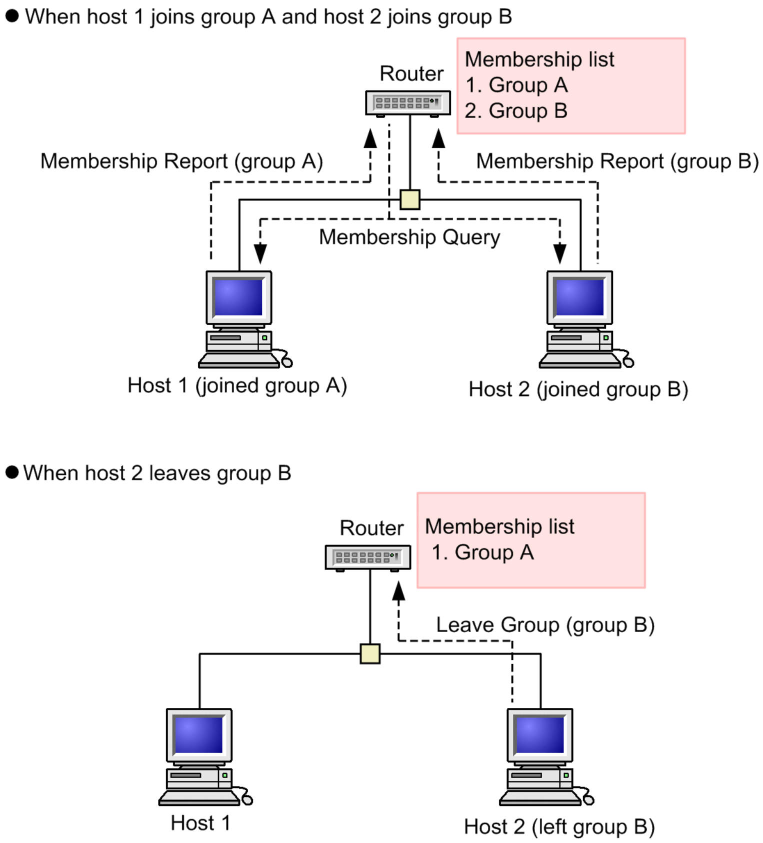
- IPv4 multicast routers obtain information about IPv4 multicast membership by regularly sending Multicast Membership Query (General Query) messages on directly connected interfaces to all multicast hosts at 224.0.0.1.
- When a host receives a Multicast Membership Query, it sends a Multicast Membership Report to the corresponding group to report the join status to the group.
- When a Multicast Membership Report is received from a host, the IPv4 multicast router adds

the group to the membership list.

- When a Multicast Leave Group message is received, the group is deleted from the membership list.

The following figure shows how IGMPv2 groups are joined and left.

Figure 14-3: Joining and leaving IGMPv2 groups



● When host 1 joins group A and host 2 joins group B

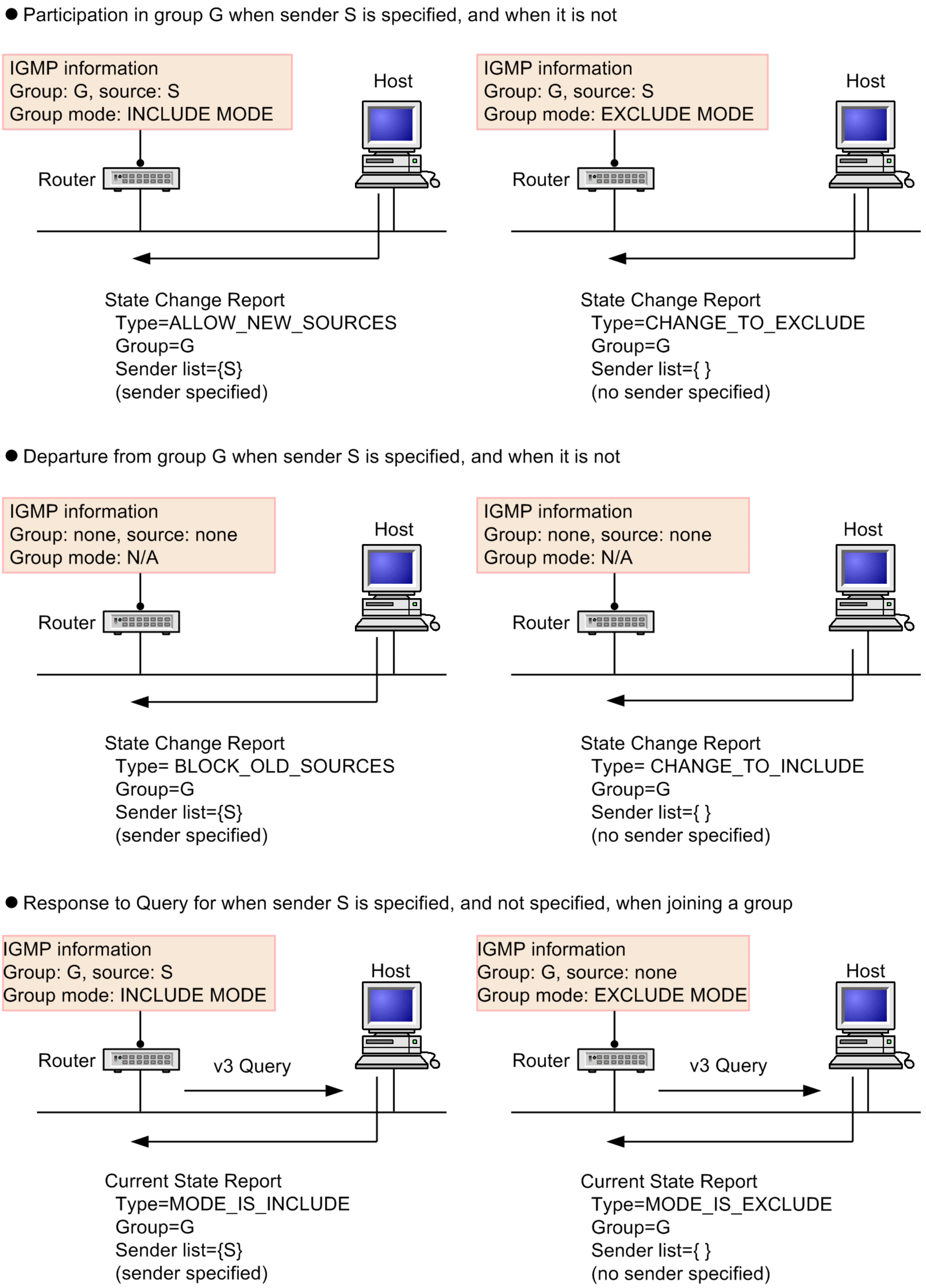
● When host 2 leaves group B

The following describes IGMPv3 operation using IGMPv3 messages:

- IPv4 multicast routers obtain information about IPv4 multicast membership by regularly sending Version 3 Multicast Membership Query (General Query) messages on directly connected interfaces to all multicast hosts at 224.0.0.1.
- When a host receives a Version 3 Multicast Membership Query, it sends a Version 3 Multicast Membership Report (Current State Report) to the corresponding group to report the join status to the group.
- When an IPv4 multicast router receives a Version 3 Multicast Membership Report (State Change Report) message from a host, it adds or removes the group from membership based on the Group Record type.

The following figure shows how IGMPv3 Report messages are sent from hosts.

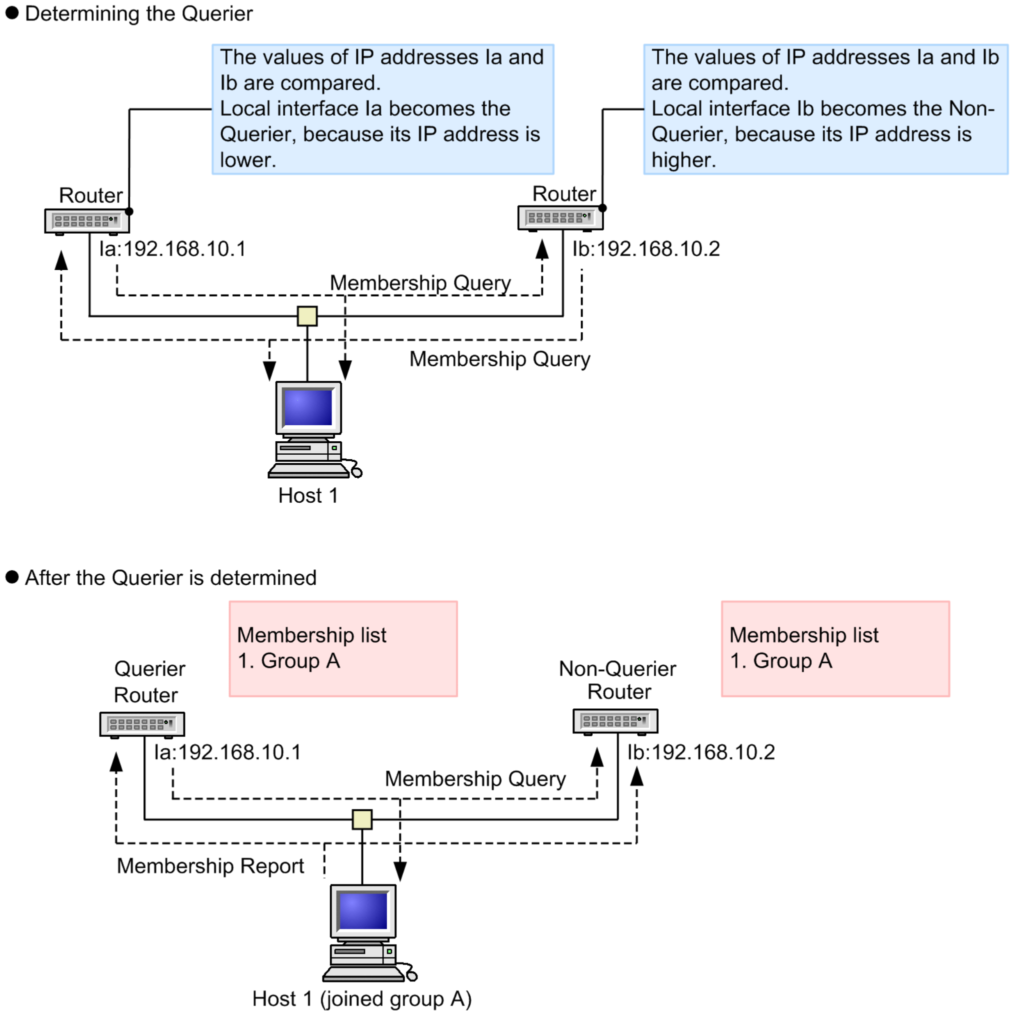
Figure 14-4: Joining and leaving IGMPv3 groups



14.2.3 Determining the Querier

IGMP routers act as either Queriers or Non-Queriers. If multiple routers exist on the same network, a Querier for regularly sending Membership Query messages is determined by comparing the sender IP address and local interface IP address of each Membership Query received from IGMP routers on the same network, and the one with the lower local interface address runs as the Querier. The one with the higher local interface address becomes a Non-Querier, and does not send Membership Queries. This means that only one Querier can exist on the same network. The following figure shows how the Querier and Non-Querier routers are determined.

Figure 14-5: Determining the Querier and Non-Querier routers



● Determining the Querier

The values of IP addresses Ia and Ib are compared.
Local interface Ia becomes the Querier, because its IP address is lower.

The values of IP addresses Ia and Ib are compared.
Local interface Ib becomes the Non-Querier, because its IP address is higher.

● After the Querier is determined

Membership list
1. Group A

Membership list
1. Group A

When a router becomes the Querier, it runs as the Querier until a Membership Query with a sender IP address lower than the local interface is received, regularly sending Membership Queries (every 125 seconds). A Non-Querier performs monitoring by receiving Querier Membership Queries, and when a Membership Query with sender IP address higher than the local interface is received, or no Membership Queries are received within a set time (255 seconds), it becomes the Querier.

14.2.4 Managing group members

(1) Managing IPv4 group members when IGMPv2 is used

Group members are registered when a Membership Report is received from a host. Non-Queriers are also registered as group members as with Queriers when a Membership Report is received from a host.

When a Querier receives a Leave Group message reporting that a host is leaving a group, it checks for the existence of other hosts that have joined the group member receiving the departure report. The Querier does this by sending continuous Membership Query (Group-Specific Query) messages to the corresponding group every second. After these messages are sent twice, if no Membership Report is received within one second, the corresponding group is deleted. Note that Leave Group messages are disregarded by Non-Queriers.

(2) Managing IPv4 group members when IGMPv3 is used

The following explains IPv4 group member registration and deletion for when IGMPv3 is used.

When a host sends a Report message indicating a request to join a multicast group and a router receives the message, the router registers group information This group information consists of a group address and the source address to the group address. Group information is registered when Reports are received for both Queriers and Non-Queriers.

When the Querier receives a Report message indicating a leave request from a multicast group, the Querier checks for the existence of other hosts that have joined the group member by sending a message every second indicating the following, based on whether a sender list is specified.

- If no sender list is specified: Group-Specific Query message
- If a sender list is specified: Group-and-Source-Specific Query message

If the Switch is the Querier, after these messages are sent twice, if no Report is received within one second, the corresponding group information is deleted. If the Switch is a Non-Querier, corresponding group information is deleted after a message is received from the Querier.

14.2.5 IGMP timers

The following table describes the IGMPv2 timer values used by the Switch.

Table 14-4: IGMPv2 timer values

Timer	Description	Timer value (in seconds)	Remarks
Query Interval	Membership Query sending interval time	125	--
Query Response Interval	Maximum response wait time for Membership Reports	10	--
Other Querier Present Interval	Querier monitoring interval	255	2 x *Query Interval* + *Query Response Interval*/2
Group Membership Interval	Group member retention time	260	2 x *Query Interval* + *Query Response Interval*
Startup Query Interval	Interval for sending a General Query message during startup	30	--
Last Member Query Interval	Specific Query sending interval after leave request reception	1	--

Legend: --: Not applicable

The following table describes the IGMPv3 timer values used by the Switch.

Table 14-5: IGMPv3 timer values

Timer	Description	Timer value (in seconds)	Remarks
Query Interval	Membership Query sending interval time	125	--
Query Response Interval	Multicast Membership Report maximum response wait time	10	--

Timer	Description	Timer value (in seconds)	Remarks
Other Querier Present Interval	Querier monitoring interval	255	*Robustness Variable* x *Query Interval* + *Query Response Interval*/2#
Startup Query Interval	Interval for sending a General Query message during startup	30	--
Last Member Query Interval	Specific Query sending interval after leave request reception	1	--
Group Membership Interval	Group member retention time	260	*Robustness Variable* x *Query Interval* + *Query Response Interval*#
Older Host Present Interval	Time for switching to IGMPv3 multicast address compatibility mode	260	*Robustness Variable* x *Query Interval* + *Query Response Interval*#

Legend: --: Not applicable

#: The Robustness Variable is 2 if the Switch is the Querier, and the same as that of the Querier if the Switch a non-Querier.

14.2.6 Connecting to IGMPv1, IGMPv2, and IGMPv3 devices

The Switch supports IGMPv2 and IGMPv3. The `ip igmp version` configuration command can be used to set the version of IGMP used by each interface. The following table describes the operation status corresponding to each specified version. Version 3 is used by default.

Table 14-6: Operation if the IGMP version is specified

Specified version	Operation when the version is specified
version 2	IGMPv2 is used for operation. IGMPv1 and IGMPv2 are used for operation for each group address. IGMPv3 packets are disregarded.
version 3	Both IGMPv2 and IGMPv3 can be used for operation. IGMPv1, IGMPv2 and IGMPv3 are used for operation for each group address.
version 3 only	IGMPv3 is used for operation. IGMPv1 packets and IGMPv2 packets are disregarded.

(1) Connecting to IGMPv2 and IGMPv3 routers

If multiple IGMP routers exist in the same network such as due to redundant configurations, the Querier is determined by mutual Query reception (for details, see *14.2.3 Determining the Querier*). The Switch does not support connections to IGMPv2 routers through interfaces for which the IGMP version is set to version 3 or version 3 only. This is because v2 Queries are disregarded, preventing the Querier from being determined. To connect to an IGMPv2 router, set the IGMP version of the corresponding interface to version 2.

(2) Mixed usage with IGMPv1 routers

The Switch supports only IGMPv2 and IGMPv3. Do not use IGMPv1 routers on the same network as these switches.

(3) Operation during mixed usage with IGMPv1, IGMPv2, and IGMPv3 hosts

When connecting to a network to which IGMPv1 hosts, IGMPv2 hosts, and IGMPv3 hosts are used together, use the default IGMP version of the corresponding interface. However, IGMPv1 hosts and IGMPv2 hosts need to be able to receive IGMPv3 Queries (as specified by the RFC). Also, if the IGMP version of the corresponding interface is set to version 2, IGMPv1 hosts and IGMPv2 hosts can be used together, and IGMPv3 hosts are disregarded.

If IGMPv1 hosts, IGMPv2 hosts, and IGMPv3 hosts are used together, group member registrations differ depending on the IGMP version for which the group join is requested. The following table describes the ways in which group members are registered when IGMPv1 hosts, IGMPv2 hosts, and IGMPv3 hosts are used together.

Table 14-7: Group member registration when IGMPv1 hosts, IGMPv2 hosts, and IGMPv3 hosts are used together

Group join request	Group member registration
Received by IGMPv1	Group members are registered in IGMPv1 mode
Received by IGMPv2	Group members are registered in IGMPv2 mode
Received by IGMPv3	Group members are registered in IGMPv3 mode
Received by IGMPv1 and IGMPv2	Group members are registered in IGMPv1 mode
Received by IGMPv1 and IGMPv3	Group members are registered in IGMPv1 mode
Received by IGMPv2 and IGMPv3	Group members are registered in IGMPv2 mode
Received by IGMPv1, IGMPv2, and IGMPv3	Group members are registered in IGMPv1 mode

14.2.7 Static group joins

Use static group joining functionality to forward IP multicast packets on networks without hosts supporting IGMP.

Interfaces for which static group joining is set run the same as those that have joined groups, without receiving any Membership Reports.

Because this functionality belongs to IGMPv2, it does not work if the IGMP version of the corresponding interface is set to version 3 only. Likewise, if version 3 is set, this functionality runs the same as that for group joining under IGMPv2.

14.2.8 Notes that apply when IGMP is used

- When static group joining is set due to a configuration change, as many as 125 seconds might be needed to create a (*,G) entry for a PIM-SM group, or a (S,G) entry for a PIM-SSM group.
- When an IGMPv3 Report with a sender specified is received for a group for which the SSM address set in the configuration is out of range, multicast packets for all senders are forwarded.

14.3 IPv4 multicast forwarding functionality

Forwarding of multicast packets is performed in hardware and software based on multicast forwarding entries. Forwarding information for forwarded multicast packets is registered in hardware multicast forwarding entries upon forwarding. Packets registered in multicast forwarding entries are forwarded by hardware, and unregistered packets are forwarded based on multicast forwarding entries generated by software multicast routing information.

(1) Hardware-based multicast packet forwarding

Hardware-based multicast packet forwarding provides the following functionality:

- Searching within multicast forwarding entries

  When a packet bound for a multicast group is received, the hardware multicast forwarding entries are searched for corresponding entries.

- Receiving interface validity checking for multicast packets

  If multicast forwarding entry search returns any entries, the packet is checked for whether it was received from a valid interface.

- Filtering multicast packets

  Forwarding judgment is performed based on the information registered in the filtering table.

(2) Software-based multicast packet forwarding

- If no hardware-based multicast forwarding entries exist

  When a packet from a given sender and bound for a given multicast group is first received, it is forwarded by software based on a multicast forwarding entry generated from multicast routing information. At the same time, a multicast forwarding entry is registered in hardware.

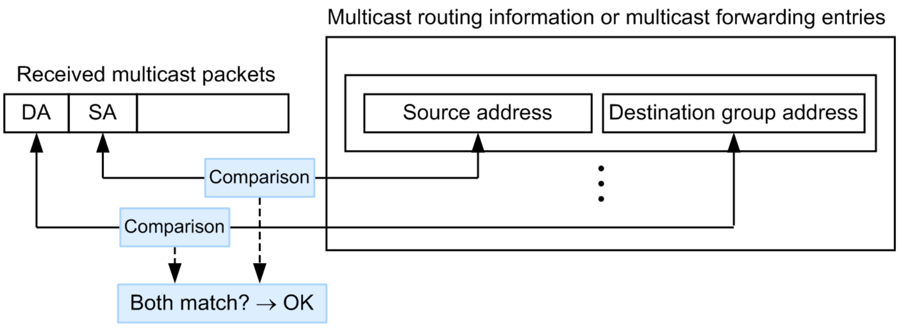
- When IP encapsulation processing is performed

  Forwarding is performed with temporary IP encapsulation for PIM-SM bound for the rendezvous point, and then decapsulation is performed at the rendezvous point to each forwarding destination.

(3) Searching for multicast routing information or multicast forwarding entries

Entries corresponding to the DA (destination group address) and SA (source address) of received multicast packets are searched from multicast routing information or multicast forwarding entries. The following figure shows how multicast routing information or multicast forwarding entries are searched.

Figure 14-6: How multicast routing information or multicast forwarding entries are searched


Multicast routing information or multicast forwarding entries
Received multicast packets
DA
SA
Source address
Destination group address
Comparison
Comparison
Both match? → OK

(4) Negative caching

Negative caching is functionality that uses hardware to discard multicast packets that cannot be forwarded. A negative cache is a forwarding entry for which no forwarding destination interface exists. When a negative cache receives a multicast packet that cannot be forwarded, it registers it in hardware. Then, when a multicast packet is received with the same address as the registered multicast packet, the packet is discarded by using hardware. This prevents undue load when many multicast packets that cannot be forwarded are received.

(5) VRF functionality [OS-L3SA]

When an IPv4 multicast is performed over multiple VRFs, IPv4 multicast forwarding entries can be set for each individual VRF. For different VRFs, IPv4 multicast forwarding entries that have the same IP address can be created. If an IPv4 multicast extranet is used, multicast communication can be performed between different VRFs.

(6) Note on IPv4 multicast forwarding functionality

Note the following points when using the IPv4 multicast forwarding functionality.

- MTU length of the interface used in IPv4 multicasting

  If the MTU length of the forwarding destination interface is shorter than that of the receiving interface, IPv4 multicast packets whose MTU length is longer than the forwarding destination interface cannot be forwarded properly. Make sure that the same MTU length value is used for all interfaces that are used in IPv4 multicasting.

14.4 IPv4 path control functionality

Path control functionality creates multicast routing information and multicast forwarding entries based on neighbor information and group information collected using the multicast routing protocol.

14.4.1 Overview of IPv4 multicast routing protocols

Multicast routing protocols are used for path control. The Switch supports the multicast routing protocols listed below.

- PIM-SM (Protocol Independent Multicast-Sparse Mode)

  This protocol uses the unicast IPv4 path structure to perform multicast path control. After packets have been sent to the rendezvous point, communication is performed on the shortest path.

- PIM-SSM (Protocol Independent Multicast-Source Specific Multicast)

  PIM-SSM is extended PIM-SM functionality. that communicates on the shortest path without using a rendezvous point.

The following table describes appropriate networks for multicast protocols.

Table 14-8: Appropriate networks for multicast routing protocols

Multicast protocol	Appropriate networks
PIM-SM	Networks on which multicast group members are sparsely distributed
PIM-SSM	

PIM-SM and PIM-SSM can operate simultaneously. but cannot use the same group. Also, if routers running PIM-SM, routers running PIM-DM, and routers running DVMRP exist on the same network, multicast packets are not forwarded between each router. To forward multicast packets within the same network, set the same multicast protocol for all routers. For details about the appropriate networks for each protocol, see *14.5.3 Examples of appropriate network configurations*.

14.4.2 IPv4 PIM-SM

PIM-SM exchanges neighbor information and join/prune requests to the multicast delivery tree through the multicast routing protocol used between routers to forward and discard received multicast packets. PIM-SM first forwards multicast packets through the rendezvous point. It then uses existing unicast routing information to switch to the shortest path from the multicast packet sender, and continues forwarding multicast packets.

The format and setting values for PIM-SM frames sent by the Switch follow RFC 2362.

(1) Supported specifications for PIM-SM messages

The following table describes supported specifications for PIM-SM messages.

Table 14-9: Supported specifications for PIM-SM messages

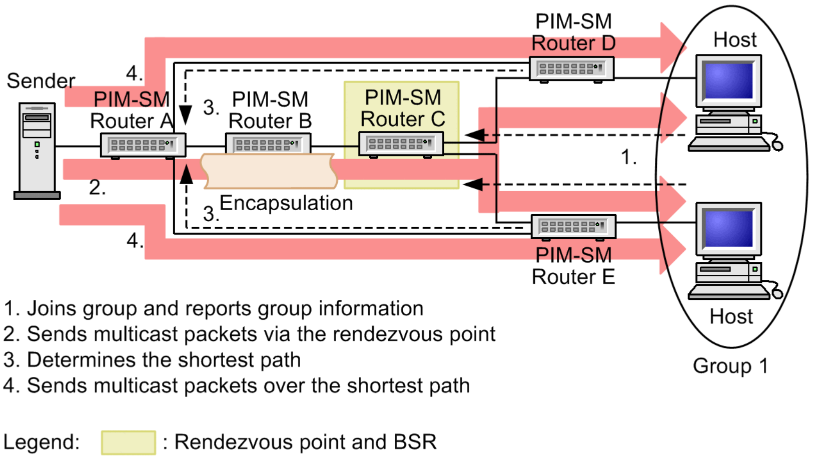
Message type	Functionality
PIM-Hello	Detecting nearby PIM routers
PIM-Join/Prune	Joining and pruning multicast delivery trees
PIM-Assert	Determining the forwarder

Message type	Functionality
PIM-Register	Performing IP encapsulation to rendezvous points for multicast packets
PIM-Register-stop	Suppressing Register messages
PIM-Bootstrap	Determining the BSR and distributing rendezvous point information.
PIM-Candidate-RP-Advertisement	Reporting local rendezvous point information from the rendezvous point to the BSR

(2) Operation

Each PIM-SM router reports information learned using IGMP to rendezvous points. Rendezvous points acknowledge the existence of each group by receiving group information from each PIM-SM router. Therefore, PIM-SM first forms a delivery tree of rendezvous points based on sender in order to deliver multicast packets from sender networks to all group members via the rendezvous points. It then uses existing unicast routing information to form a shortest-path delivery tree from senders, to arrive at the shortest path from the senders to each group. This is how it performs multicast packet forwarding from senders to each group member over the shortest path. The following figure provides an overview of PIM-SM operation.

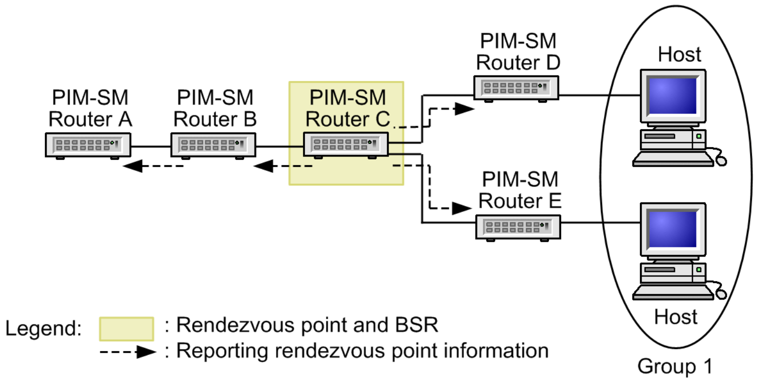
Figure 14-7: Overview of PIM-SM operation


PIM-SM
Router D
Host
Sender
4.
PIM-SM
Router A
3.
PIM-SM
Router B
PIM-SM
Router C
1.
2.
3.
Encapsulation
4.
PIM-SM
Router E
Host
Group 1
1. Joins group and reports group information
2. Sends multicast packets via the rendezvous point
3. Determines the shortest path
4. Sends multicast packets over the shortest path
Legend: ▨ : Rendezvous point and BSR

(a) Rendezvous points and bootstrap routers (BSRs)

Rendezvous point and bootstrap routers (BSRs) can be set up through configuration. The Switch can use a maximum of 16 BSRs in each network (VPN). The BSR reports the IP address and other rendezvous point information to all multicast interfaces. These notifications are reported hop by hop to all multicast routers. The following figure shows the roles of rendezvous points and BSRs.

Figure 14-8: Roles of rendezvous points and bootstrap routers (BSRs)

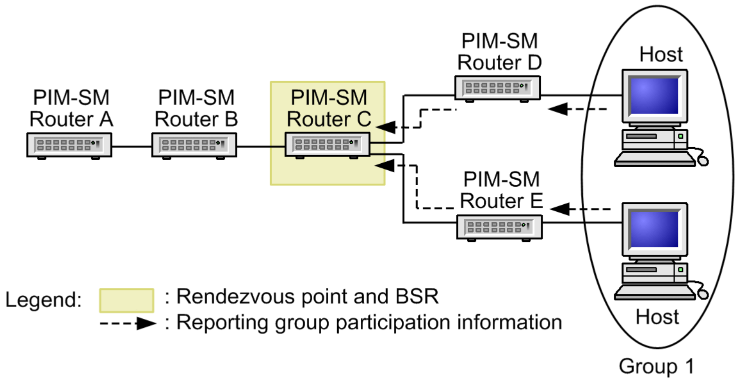


In this figure, the BSR (PIM-SM router C) reports rendezvous point information to all multicast interfaces. Routers receiving rendezvous point information learn the IP address of rendezvous points, and report rendezvous point information to all interfaces for which multicast routers exist other than the received interface.

(b) Reporting group joining information to the rendezvous point

Each router reports the group joining information learned by IGMP to the rendezvous point, which use this information to understand the existence of groups for each interface. The following figure shows how group joining information is reported to rendezvous points.

Figure 14-9: How group-joining information is reported to the rendezvous point

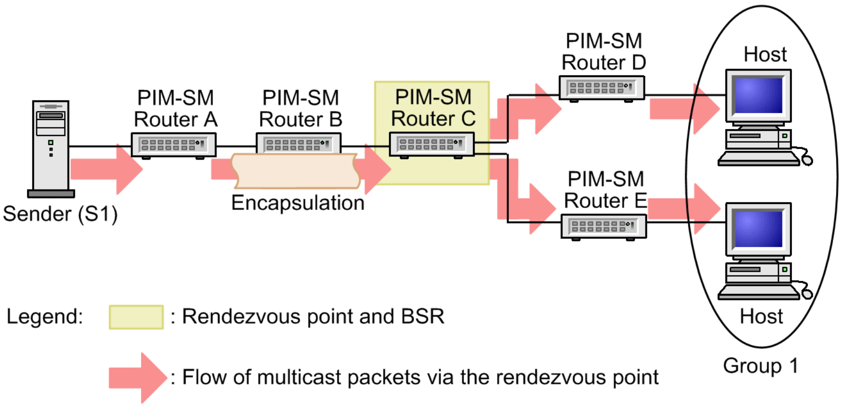


In this diagram, each host joins group 1 using IGMP. PIM-SM router D and PIM-SM router E learn information about group 1, and report this information to the rendezvous point (PIM-SM router C). The rendezvous point (PIM-SM router C) receives this information, and learns that group 1 exists on the interface from which reception occurred.

(c) Multicast packet communication via rendezvous points (encapsulation)

When sender S1 sends a multicast packet bound for group 1, PIM-SM router A IP-encapsulates the multicast packet (creating a Register packet) and sends it to the rendezvous point (PIM-SM router C), because the IP address of the rendezvous point was learned in (a). When the rendezvous point (PIM-SM router C) receives the IP-encapsulated packet, it decapsulates the packet and forwards the multicast packet bound for group 1 to the interface on which group 1 exists, because the existence of group 1 was learned in (b). When PIM-SM router D and PIM-SM router E receive the multicast packet bound for group 1, the packet is forwarded to the interface on which group 1 exists, because the existence of group 1 was learned by IGMP in (b). The following figure shows multicast packet communication (encapsulation) via rendezvous point.

Figure 14-10: Communicating multicast packets via rendezvous points (encapsulation)

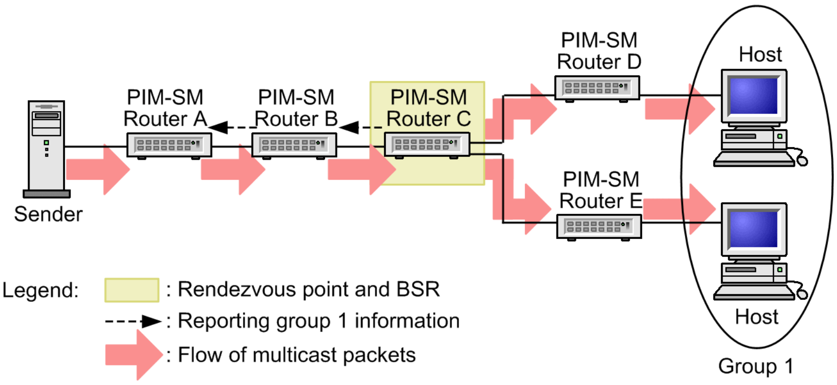


(d) Multicast packet communication (decapsulation) via rendezvous point

When the rendezvous point (PIM-SM router C) receives IP-encapsulated packets, it decapsulates them and forwards the multicast packet bound for group 1 to the interface on which group 1 exists.

After this processing is completed, the rendezvous point reports information about group 1 to the sending server over the shortest path. When PIM-SM router B and PIM-SM router A receive information about group 1, they learn that group 1 exists on the interface from which the information was received. PIM-SM router A forwards to the corresponding interface all multicast packets bound for group 1 as sent by the sending server, without performing IP encapsulation. When PIM-SM router B, PIM-SM router C, PIM-SM router D, and PIM-SM router E receive multicast packets bound for group 1, they forward them to the interface on which group 1 exists. The following figure shows multicast packet communication (decapsulation) via rendezvous points.

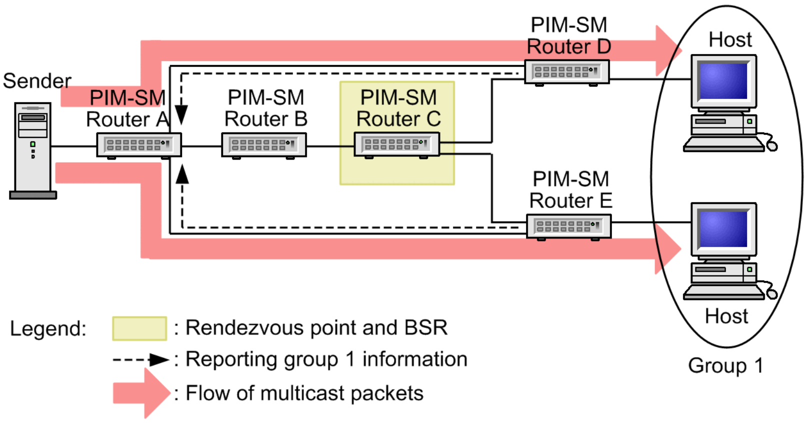
Figure 14-11: Multicast packet communication (decapsulation) via rendezvous points



(e) Multicast packet communication over the shortest path

When PIM-SM router D and PIM-SM router E receive multicast packets bound for group 1 from the sending server (as explained in (c)), PIM-SM router D and PIM-SM router E report information about group 1 to sender S1 over the shortest path (based on the existing unicast routing information). When PIM-SM router A receives information about group 1 from PIM-SM router D and PIM-SM router E, it learns that group 1 exists on the interface from which the information was received, and when it receives multicast packets bound for group 1 from the sending server, it forwards them to the corresponding interface. The following figure shows multicast packet communication for shortest paths.

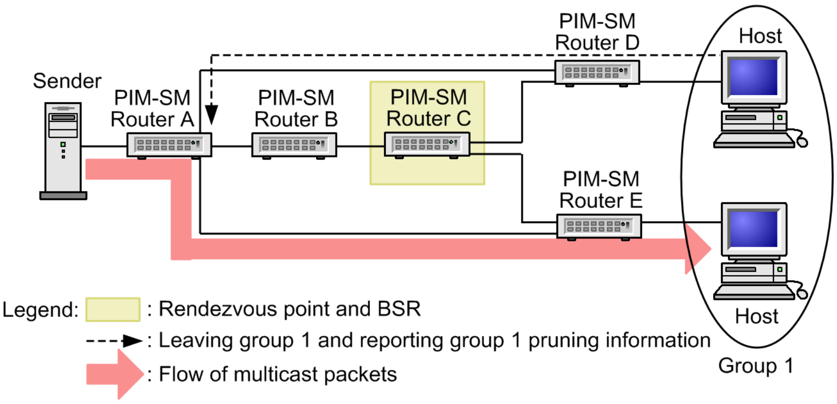
Figure 14-12: Multicast packet communication over the shortest path



(f) Pruning multicast delivery trees

When a host leaves group 1 over IGMP, PIM-SM router D reports pruning information for group 1 to interfaces to which information about group 1 was being reported. When PIM-SM router A receives a pruning notification for group 1, it stops forwarding multicast packets bound for group 1 to the interface from which the information was received. The following figure shows multicast delivery tree pruning.

Figure 14-13: Pruning multicast delivery trees


Sender
PIM-SM Router A
PIM-SM Router B
PIM-SM Router C
PIM-SM Router D
PIM-SM Router E
Host
Host
Group 1
Legend:
: Rendezvous point and BSR
: Leaving group 1 and reporting group 1 pruning information
: Flow of multicast packets

(3) Detecting neighbors

PIM-SM routers regularly send PIM-Hello messages to all interfaces capable of multicast. PIM-Hello messages are sent to the address for the All-PIM-RoutersIP multicast group (224.0.0.13). When these messages are received, neighboring PIM routers can be dynamically detected. The Switch supports the Generation ID option for PIM-Hello messages (in accordance with RFC 4601 and draft-ietf-pim-sm-bsr-07).

A Generation ID is a 32-bit random number for each multicast interface that is appended when PIM-Hello messages are sent. Generation IDs are regenerated when the status of the multicast interface changes to Up. When the Generation ID option is added to a received PIM-Hello message, the Generation ID is recorded to enable neighboring switches to detect interface failure when the Generation ID changes. When a Generation ID change is detected, updates to neighbor switch information, PIM-Hello messages, PIM Bootstrap messages, and PIM Join/Prune messages are sent without waiting for regular advertisements. This process allows multicast routing information to be relearned quickly.

(4) Determining the forwarder

If multiple PIM-SM routers are connected on the same LAN, multicast packets might be duplicated and forwarded on the network.

If multiple PIM-SM routers exist on the same LAN, and two or more of these routers forward multicast packets to the LAN, the PIM-SM routers use PIM-Assert messages to compare the multicast path preferences and metrics, and choose the most appropriate router on the sender network as the forwarder.

Only the router that becomes the forwarder forwards multicast packets on the LAN, in order to prevent duplicate multicast packet forwarding.

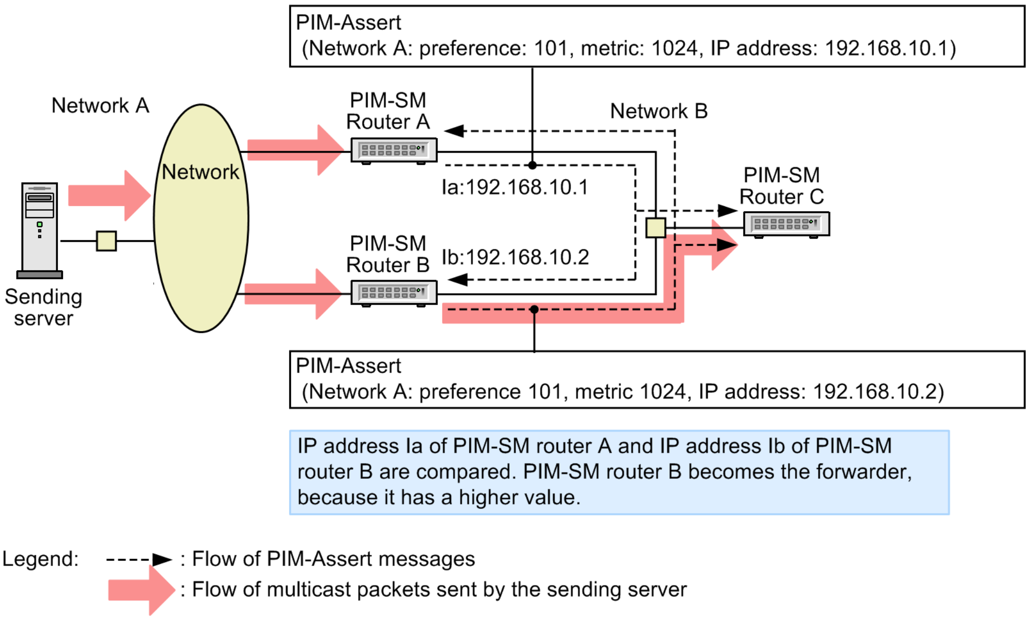
The flow for determining the forwarder by PIM-Assert messages is as follows:

1. Preferences are compared, and the router with the lowest value becomes the forwarder.
2. If the preferences are the same, the metrics are compared, and the router with the lowest value becomes the forwarder.
3. If the metrics are the same, the IP addresses of the routers are compared, and the router with the highest IP address becomes the forwarder.

The Switch sends PIM-Assert messages with the multicast path preference fixed at 101, and the metric fixed at 1024. However, for direct connections with a sender, PIM-Assert messages are sent with the preference and metric both fixed at 0.

The following figure shows how the forwarder is determined.

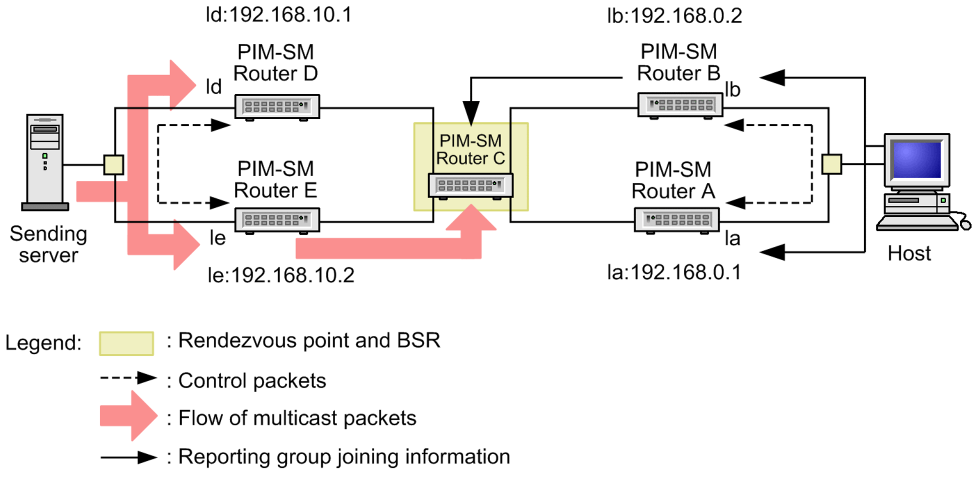
Figure 14-14: Determining the forwarder


PIM-Assert
(Network A: preference: 101, metric: 1024, IP address: 192.168.10.1)
Network A
PIM-SM
Router A
Network B
Network
Ia:192.168.10.1
PIM-SM
Router C
PIM-SM
Router B
Ib:192.168.10.2
Sending
server
PIM-Assert
(Network A: preference 101, metric 1024, IP address: 192.168.10.2)
IP address Ia of PIM-SM router A and IP address Ib of PIM-SM router B are compared. PIM-SM router B becomes the forwarder, because it has a higher value.
Legend:
: Flow of PIM-Assert messages
: Flow of multicast packets sent by the sending server

(5) Determining and running the DR

If multiple PIM-SM routers exist on the same LAN, a forwarding delegate router (DR) for the LAN is determined. The router with the highest IP address becomes the DR. The DR reports group-joining information from the receiving host and reports it to the rendezvous point. The DR IP-encapsulates multicast packets sent by the sending server, and then sends them to the rendezvous point. The following figure shows DR operation.

Figure 14-15: DR operation

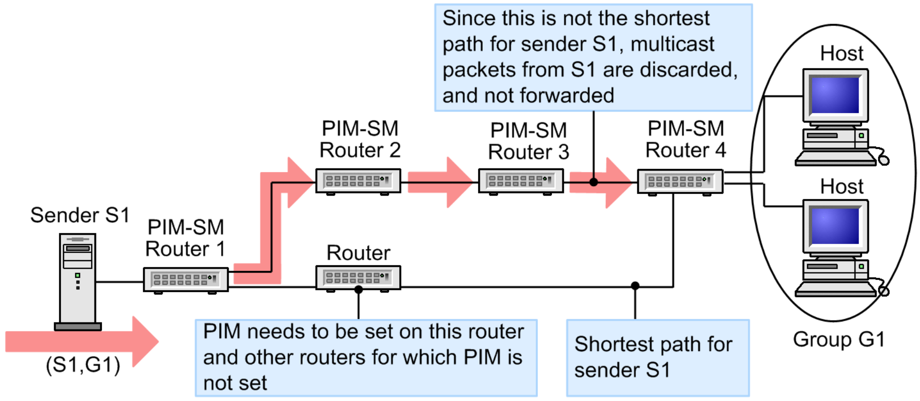


The IP addresses of PIM-SM router A and PIM-SM router B are compared, and if that of PIM-SM router B is higher, PIM-SM router B becomes the DR and reports group joining information to the rendezvous point. Likewise, the IP addresses of PIM-SM router D and PIM-SM router E are compared, and if that of PIM-SM router E is higher, PIM-SM router E becomes the DR and forwards IP encapsulated packets to the rendezvous point.

(6) Notes on redundant routes

Keep in mind that multicast packets are not forwarded in redundant configurations such as that shown in the figure below. PIM settings are required for all routers on any redundant routes.

Figure 14-16: Note regarding redundant paths


Since this is not the shortest path for sender S1, multicast packets from S1 are discarded, and not forwarded
PIM-SM Router 2
PIM-SM Router 3
PIM-SM Router 4
Host
Host
Group G1
Sender S1
PIM-SM Router 1
Router
(S1,G1)
PIM needs to be set on this router and other routers for which PIM is not set
Shortest path for sender S1

(7) PIM-SM timer specification

The following table describes the timer values used by PIM-SM.

Table 14-10: PIM-SM timers

Timer name	Description	Default value (in seconds)	Range of values that can be set in configuration mode (in seconds)	Remarks
Hello-Period	Hello sending interval	30	5 to 3600	--
Hello-Holdtime	Adjacency retention period	105	3.5 x *Hello-Period*	Calculated based on the formula to the left.

Timer name	Description	Default value (in seconds)	Range of values that can be set in configuration mode (in seconds)	Remarks
Assert-Timeout	Assert-based forwarding suppression period	180	--	--
Join/Prune-Period	Join/Prune sending interval	60	30 to 3600	A variance as high as +50% might occur.
Join/Prune-Holdtime	Retention period for routing information and forwarding destination interfaces	210	3.5 x *Join/ Prune-Period*	Calculated based on the formula to the left.
Deletion-Delay-Time	Retention period for multicast forwarding destination interfaces after a Prune message is received[#1]	1/3 x *Join/ Prune - Holdtime*	0 to 300	[#2]
Data-Timeout	Forwarding entry retention period	210	0 (infinite), 60 to 43200	A gap of as much as +90 seconds might occur.
Register-Supression-Timer	Encapsulation sending suppression period	60	--	A variance as high as 30 seconds might occur.
Probe-Time	Time for sending restart checks for encapsulation sending	5	5 to 60	With the default of 5 seconds, a restart check (Null-Register) for encapsulation sending is sent once, 5 seconds before Register-Suppression-Timer is up.[#3]
C-RP-Adv-Period	Notification interval for rendezvous point candidates	60	--	--
RP-Holdtime	Rendezvous point retention period	150	2.5 x *C-RP-Adv-Period*	Calculated based on the formula to the left.
Bootstrap-Period	BSR message sending interval	60	--	--
Bootstrap-Timeout	BSR message retention period	130	2 x *Bootstrap-Period* + 10	Calculated based on the formula to the left.
BS_Rand_Override	BSR switching delay	5 to 23	--	--
Negative-Cache-Holdtime (PIM-SM)	Negative cache retention period	210	10 to 3600	For PIM-SSM, this is fixed at 3600 seconds.

Legend: --: Not applicable

#1

A configured timer value is used if it set in the configuration, however, to the forwarding destination interface, set a value that does not exceed the Join/Prune-Holdtime value included in the PIM-Join/Prune message received with the last Join as the retention period in the interface.

#2

Because the value set in the configuration is given priority for this timer value, operation differs from the standard in RFC 2362. However, if no value is specified in the configuration, operation complies with RFC 2362.

#3

If this timer value is set to 10 or more, the restart check for encapsulation sending is sent multiple times every five seconds. If no value is specified in the configuration, the check is sent only once.

(8) Notes on using PIM-SM

Keep the following limitations in mind when configuring a network using PIM-SM. The Switch conforms to RFC 2362 (the PIM-SM specification), but differs from some parts of the RFC due to software functionality limitations. The following table describes the differences with the RFC.

Table 14-11: Differences with the RFC

Item	RFC	Switch
Packet format	The RFC contains a field for setting a mask length for the encoding group address and encoding source address.	The mask length of encoding addresses is fixed at 32 for the Switch.
	The RFC contains fields for setting the address family and encoding type for encoding group addresses and encoding source addresses.	The address family for encoding addresses is fixed at 1 (IPv4), and the encoding type is fixed at 0 for the Switch. Connections cannot be established outside of IPv4 to PIM-SM.
	The RFC contains a field for setting the PIM version of a PIM message header.	The PIM version is fixed at 2 for the Switch. Connections to PIM version 1 cannot be established.
Join/Prune fragments	Join/Prune messages can be fragmented even when exceeding the network MTU.	If the size of a Join/Prune message to be sent is large, the Switch splits it into 8 KB parts before sending. Also, Join/Prune messages split before sending are sent using IP fragments based on the network MTU length.
Connections to PMBR	The RFC contains a specification for connections to PMBRs (PIM Border Routers) and for (*, *, RP) entries.	The Switch does not support connections to PMBR. Also, (*, *, RP) entries are not supported.
Switching to shortest paths	The RFC contains a switching method based on the data rate for switching to the shortest path.	When the Switch first receives data for the last-hop-router, it switches to the shortest path without checking the data rate.

14.4.3 IPv4 PIM-SSM

PIM-SSM is extended PIM-SM functionality. PIM-SM and PIM-SSM can run at the same time. The multicast addresses used by PIM-SSM are assigned by the IANA. The Switches allow the address range of multicast addresses (group addresses) for which PIM-SSM runs to be specified by configuration. PIM-SM runs on addresses other than those specified.

Whereas PIM-SM requires multicast forwarding packets when creating multicast entries, PIM-SSM creates multicast forwarding entries by exchanging multicast routing information (PIM-Join), and forwards multicast packets based on the corresponding entries. Note that PIM-SSM does not require a rendezvous point or bootstrap router. This means that packet encapsulation and decapsulation is not needed when multicast packets are forwarded, enabling more efficient multicast forwarding. PIM-SSM runs if IGMPv3 (INCLUDE mode) hosts are connected. The Switch provides measures to enable PIM-SSM from IGMPv2 or IGMPv3

(EXCLUDE mode) hosts.

(1) Supported specifications for PIM-SSM messages

The supported specifications for PIM-SM messages are the same as those in *14.4.2 IPv4 PIM-SM (1) Supported specifications for PIM-SM messages*.

(2) Prerequisites for running PIM-SSM

The following settings are required to configure the Switch:

- Settings for each switch

  Set the range of group addresses for which PIM-SSM runs.

- Switches directly connected to hosts running IGMPv3 (INCLUDE mode)

  Set up IGMPv3 for the connected interfaces.

- Switches directly connected to hosts running IGMPv2 or IGMPv3 (EXCLUDE mode)

  Set up IGMPv2 or IGMPv3 for the connected interfaces.

  Set the source address for the group addresses used.

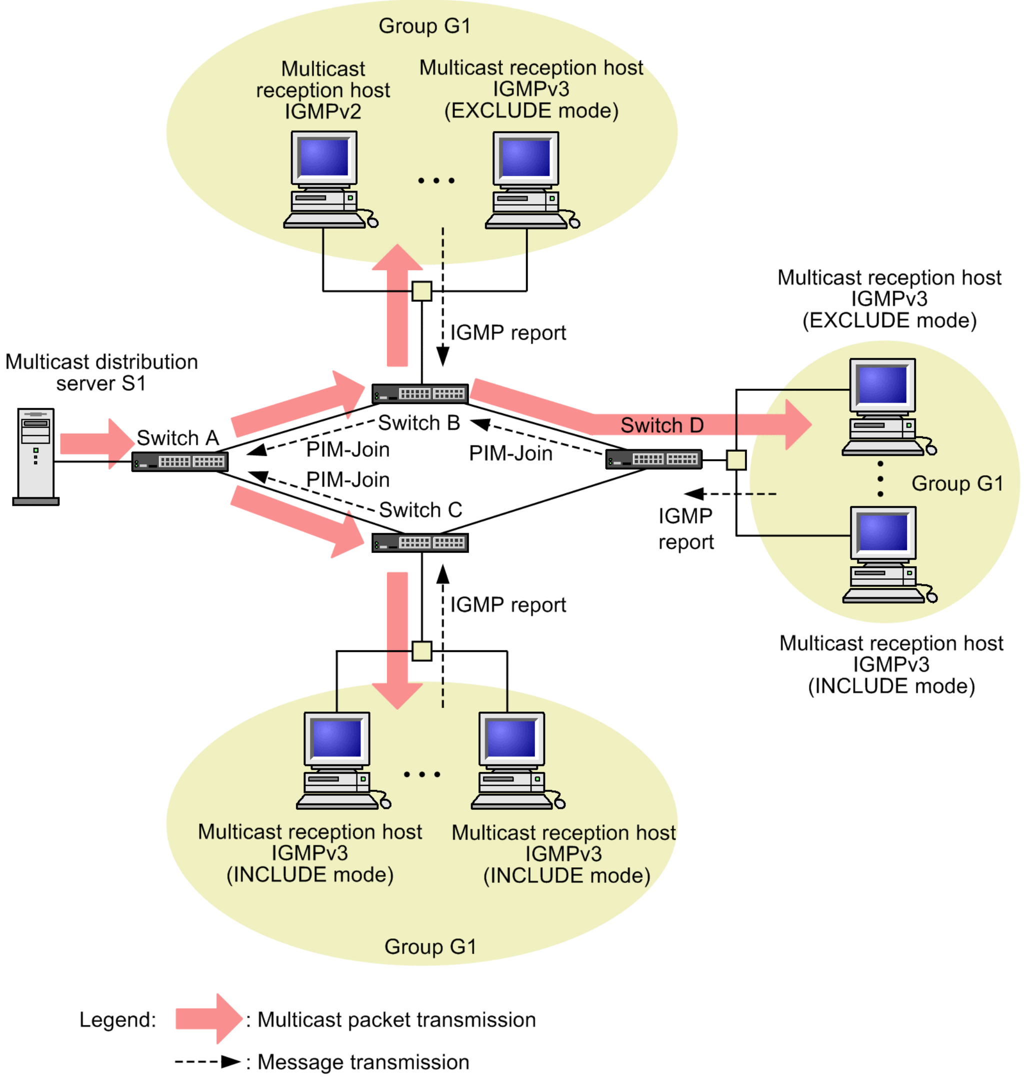
(3) PIM-SSM operation (for IGMPv3 (INCLUDE mode) hosts)

Operation is as follows when the multicast packet distribution server (source address: S1) distributes multicast packets to group 1 (group address: G1).

1. A request to join the multicast group (IGMPv3 (INCLUDE mode)) is received from hosts.
2. The switch receiving the join request (IGMPv3 (INCLUDE mode)) sends PIM-Join from the reported group address (G1) and source address (S1) in the direction of the source address (S1, as determined by unicast routing information). In this case, the PIM-Join message contains information about the source address (S1) and group address (G1). Each switch receiving PIM-Join sends PIM-Join hop by hop in the direction of the source address (S1). Switches receiving PIM-Join learn multicast routing information for the source address (S1) and group address (G1).
3. The multicast packet distribution server (S1) sends multicast packets bound for group 1 (G1). Switches receiving multicast packets forward them based on the multicast forwarding entries generated by the learned multicast routing information.

The following figure shows an overview of PIM-SSM operation.

Figure 14-17: Overview of PIM-SSM operation


Group G1
Multicast reception host IGMPv2
Multicast reception host IGMPv3 (EXCLUDE mode)
IGMP report
Multicast distribution server S1
Switch A
Switch B
Switch C
Switch D
PIM-Join
PIM-Join
PIM-Join
Multicast reception host IGMPv3 (EXCLUDE mode)
Group G1
IGMP report
Multicast reception host IGMPv3 (INCLUDE mode)
IGMP report
Multicast reception host IGMPv3 (INCLUDE mode)
Multicast reception host IGMPv3 (INCLUDE mode)
Group G1
Legend:
: Multicast packet transmission
: Message transmission

(4) PIM-SSM operation (for IGMPv2 or IGMPv3 (EXCLUDE mode) hosts)

Operation is as follows when the multicast packet distribution server (source address: S1) distributes multicast packets to group 1 (group address: G1).

1. A request to join the multicast group (IGMPv2 or IGMPv3 (EXCLUDE mode)) is received from hosts.
2. The switch receiving the join request (IGMPv2 or IGMPv3 (EXCLUDE mode)) compares the reported group address (G1) and that set in the configuration. If these group addresses match, a PIM-Join message is sent to the source address (S1) set in configuration over the shortest path (determined by the unicast routing information). In this case, the PIM-Join message contains information about the source address (S1) and group address (G1). Each switch receiving the PIM-Join message sends it hop-by-hop to the source address (S1) over the shortest path. Switches receiving PIM-Join learn multicast routing information for the source address (S1) and group address (G1).
3. The multicast packet distribution server (S1) sends multicast packets bound for group 1 (G1). Switches receiving multicast packets forward them based on the multicast forwarding entries generated by the learned multicast routing information.

For an overview of PIM-SSM operation, see *Figure 14-17: Overview of PIM-SSM operation*.

(5) Detecting neighbors

Operation is the same as for PIM-SM (see *14.4.2 IPv4 PIM-SM (3) Detecting neighbors*).

(6) Determining the forwarder

Operation is the same as for PIM-SM (see *14.4.2 IPv4 PIM-SM (4) Determining the forwarder*).

(7) Determining and running the DR

Operation is the same as for PIM-SM (see *14.4.2 IPv4 PIM-SM (5) Determining and running the DR*).

(8) Notes on redundant routes

Operation is the same as for PIM-SM (see *14.4.2 IPv4 PIM-SM (6) Notes on redundant routes*).

14.4.4 IPv4 path control operation when IGMPv3 is used

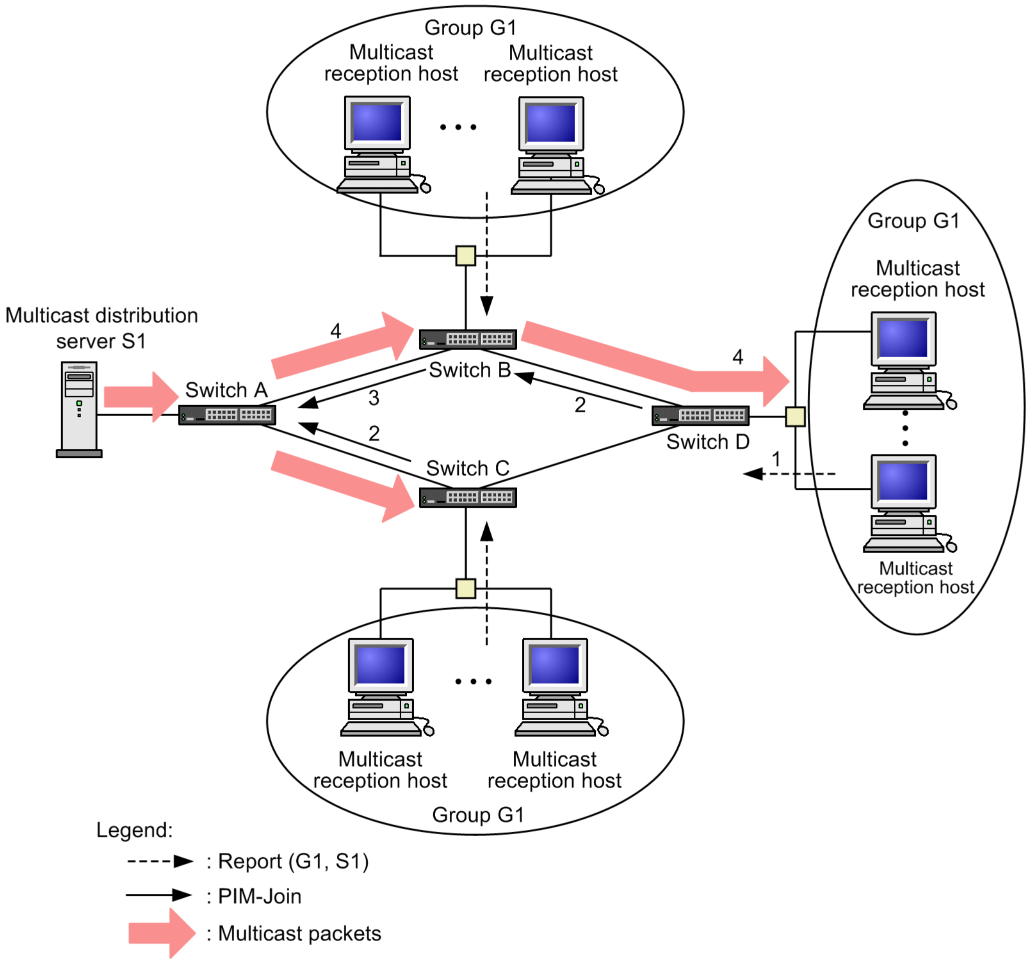
(1) IPv4 PIM-SSM operation when IGMPv3 is used

Sender information is needed to use PIM-SSM. The Switch allows PIM-SSM to be used by setting the sender during configuration when IGMPv2 is used. For IGMPv3, PIM-SSM can be used without setting a sender in the configuration (PIM-SSM needs to be set in the configuration).

IPv4 PIM-SSM operation is as follows when the multicast distribution server (source address S1) sends multicast packets to multicast group G1:

1. An IGMPv3 Report (G1, S1) to join the multicast group is received from hosts.
2. Switches receiving the IGMPv3 Report (G1, S1) compare the group address (G1) reported by the Report with the SSM group address (range) specified in the configuration. If the group addresses match, a PIM-Join message including the group address (G1) and source address (S1) is sent to the source address (S1) reported in the Report, over the shortest path.
3. The switches receiving the PIM-Join message forward the message to source address S1 hop-by-hop over the shortest path. These switches form an S1-to-G1 delivery tree so that multicast packets from source address S1 are forwarded only to the interfaces from which the PIM-Join message was received.
4. Switches receiving multicast packets sent by multicast distribution server S1 to group G1 forward the multicast packets based on multicast forwarding information.

Figure 14-18: Overview of IPv4 PIM-SSM operation when IGMPv3 is used

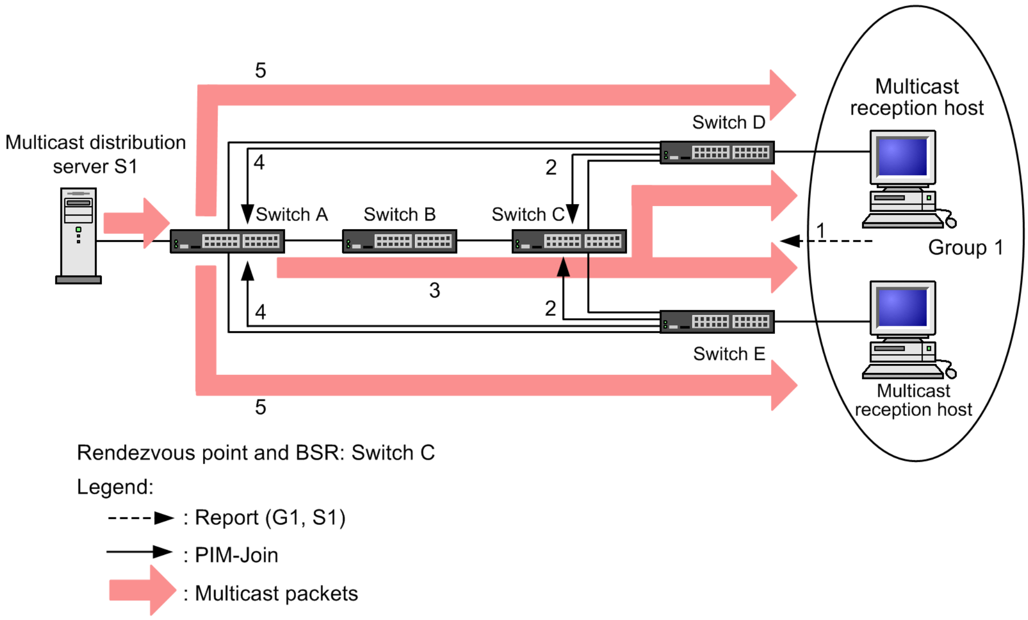


(2) IPv4 PIM-SM operation when IGMPv3 is used

If PIM-SSM is not configured, PIM-SM operation is performed. IPv4 PIM-SM operation is as follows when the multicast distribution server (source address S1) sends multicast packets to multicast group G1:

1. An IGMPv3 Report (G1, S1) to join the multicast group is received from hosts.
2. Switches receiving the IGMPv3 Report (G1, S1) send a PIM-Join message including the group address (G1) to the rendezvous point over the shortest path.
3. The rendezvous point receiving the PIM-Join message learns about the existence of the group. A delivery tree from the sender via the rendezvous point is formed so that multicast packets can be delivered from the sender network to each group member via the rendezvous point.
4. The shortest path from the sender to each group member is determined by using the existing unicast routing information (PIM-Join messages are sent to the sender over the shortest path to the sender to form a shortest-path delivery tree).
5. Switches receiving multicast packets sent by multicast distribution server S1 bound for group G1 forward the multicast packets based on the shortest path delivery tree.

Figure 14-19: Overview of IPv4 PIM-SM operation when IGMPv3 is used



(3) IPv4 path control when IGMPv1/IGMPv2 hosts and IGMPv3 hosts are mixed

The following explains IPv4 path control operation if PIM-SSM is set to be used on IGMPv2, and IGMPv1 hosts, IGMPv2 hosts, and IGMPv3 hosts are used together.

PIM-SSM is used when a join request is received for a group address included in the PIM-SSM target address range specified in the configuration (for details, see *Table 14-12: IPv4 path control operation when IGMPv1/IGMPv2 and IGMPv3 hosts are used together.*). When a join request is received through an IGMPv1 Report or IGMPv2 Report, the sender list uses the source address set in the configuration. When a join request is received for the same group address through an IGMPv1 Report, IGMPv2 Report, and IGMPv3 Report (EXCLUDE), the used sender list combines the source addresses set in the configuration and the sender list contained in the IGMPv3 Report (INCLUDE).

The following table describes the IPv4 path control operations performed when IGMPv1/IGMPv2 and IGMPv3 hosts are used together.

Table 14-12: IPv4 path control operation when IGMPv1/IGMPv2 and IGMPv3 hosts are used together.

Subscription group addresses	IGMPv1 Report IGMPv2 Report IGMPv3 Report(EXCLUDE)	IGMPv3 Report(INCLUDE)
Within the SSM address range	PIM-SSM	PIM-SSM
Outside the SSM address range	PIM-SM	PIM-SM

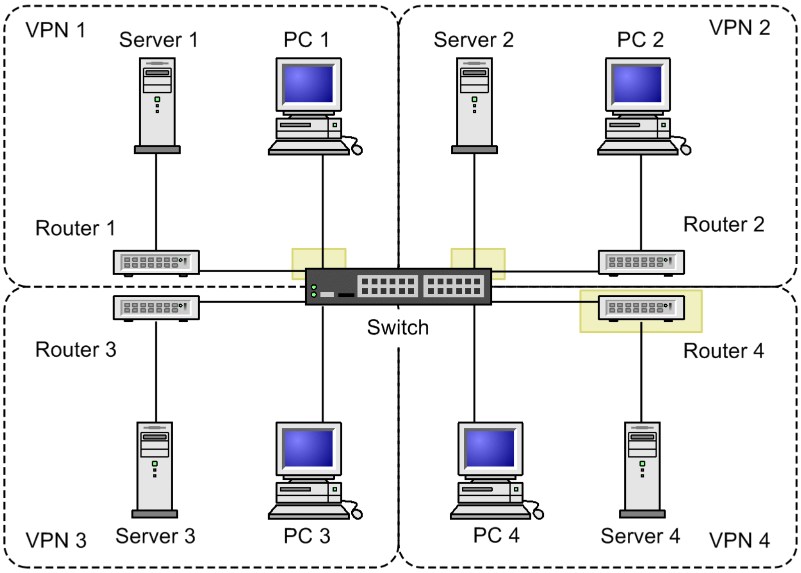
14.4.5 IPv4 multicasting for a VRF [OS-L3SA]

(1) IPv4 multicasting VRF

You can connect the Switch to multiple VPNs and use IPv4 multicast in each VPN. Configure VRF for each VPN and run IPv4 multicast in each VRF. You can set different rendezvous points, BSRs, timers, and SSM address ranges for VRF-based IPv4 multicasts.

The figure below shows an example configuration in which the Switch is connected to four VPNs. The settings on the Switch in this example are given in the table that follows.

Figure 14-20: IPv4 multicasting in a VRF



Legend: 🟨 : Rendezvous point and BSR

Table 14-13: Settings on the Switch

VPN	Operation protocol	Loopback address	Rendezvous point (The values in parentheses are rendezvous point addresses)	SSM address
1	PIM-SM	1.1.1.1	Switch (1.1.1.1)	Not used
2	PIM-SM/ PIM-SSM	2.2.2.2	Switch (2.2.2.2)	232.0.0.0/8
3	PIM-SSM	2.2.2.2	None	232.10.0.0/16
4	PIM-SM	3.3.3.3	Router 4 (1.1.1.1)	Not used

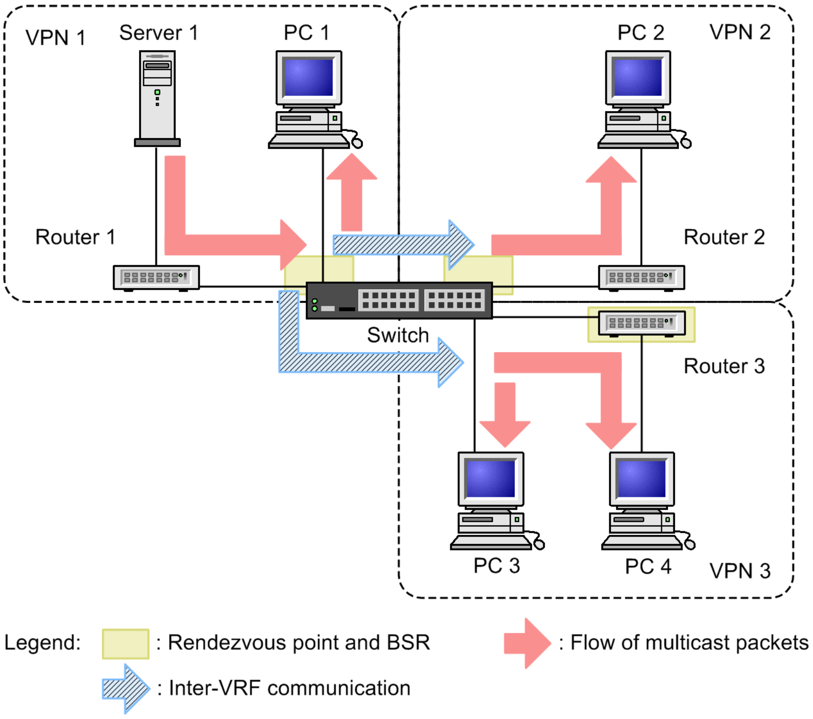
(2) IPv4 multicast extranet

If an IPv4 multicast extranet is used, IPv4 multicast forwarding can be performed between VRFs. In addition, if IPv4 multicast route filtering is used, you can limit the range of group addresses used in the extranet and the VRFs for which forwarding requests from the downstream are permitted.

Note that a unicast route to the sender must exist in the unicast extranet to establish the shortest path from the last-hop-router.

The following figure shows the overview of IPv4 multicast extranet operation.

Figure 14-21: Overview of IPv4 multicast extranet operation


VPN 1
Server 1
PC 1
PC 2
VPN 2
Router 1
Router 2
Switch
Router 3
PC 3
PC 4
VPN 3
Legend:
: Rendezvous point and BSR
: Flow of multicast packets
: Inter-VRF communication

(3) PIM-SM VRF gateways

To perform multicast communication by using the PIM-SM protocol, IPv4 multicast packets must be forwarded to the last-hop-router. If PIM-SM is used in an extranet environment, IPv4 multicast packets must also be forwarded to all the last-hop-routers in the VRFs.

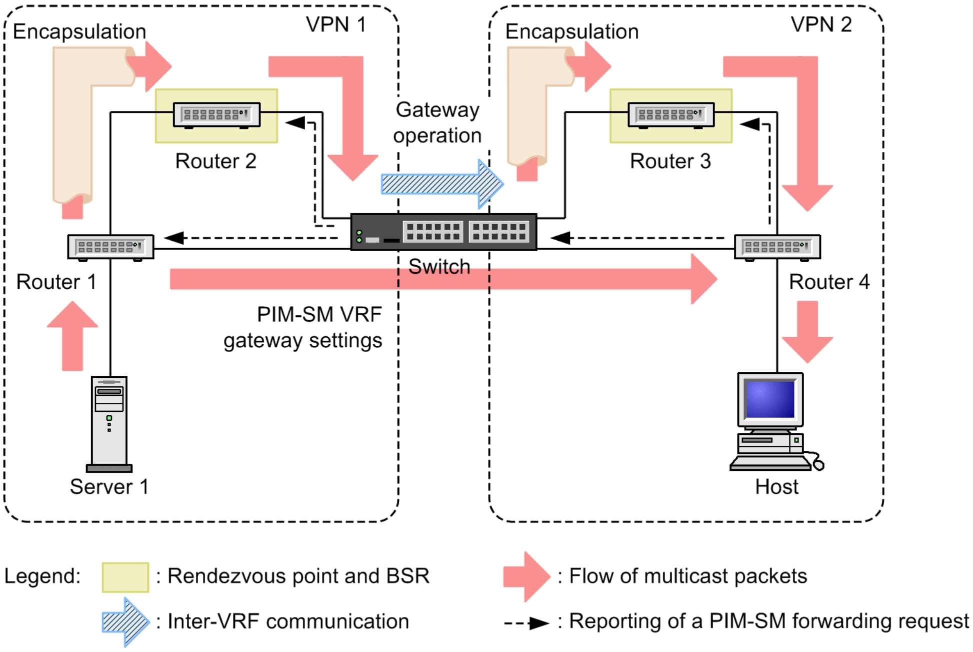
Use a PIM-SM VRF gateway to enable the Switch to forward IPv4 multicast packets to the rendezvous point of each VRF.

The PIM-SM VRF gateway set in the VRF containing the sender (multicast server) operates as the last-hop-router for the specified group in the target VPN, and requests the rendezvous point to forward packets. Upon receiving IPv4 multicast packets from the rendezvous point, the gateway forwards the packets to the destination VRFs.

The PIM-SM VRF gateway in a destination VPN operates as the first-hop-router. The gateway encapsulates IPv4 multicast packets, and sends the encapsulated packets (called register packets) to the rendezvous point of each VRF. When a rendezvous point receives register packets, the rendezvous point decapsulates the packets and transfers the decapsulated IPv4 multicast packets to the last-hop-router the same way as in normal PIM-SM operation. Next, the shortest-path delivery tree from the last-hop-router to the sender is generated. At the same time, IPv4 multicast packets in the extranet are also forwarded by hardware.

As described above, use of PIM-SM VRF gateways allows a PIM-SM-based extranet to be created without changing the settings of any components other than the Switch. The following figure shows the overview of PIM-SM VRF gateway operation.

Figure 14-22: Overview of PIM-SM VRF gateway operation


Encapsulation
VPN 1
Encapsulation
VPN 2
Gateway operation
Router 2
Router 3
Switch
Router 1
Router 4
PIM-SM VRF gateway settings
Server 1
Host
Legend:
: Rendezvous point and BSR
: Inter-VRF communication
: Flow of multicast packets
: Reporting of a PIM-SM forwarding request

(4) Notes on using IPv4 multicast extranet

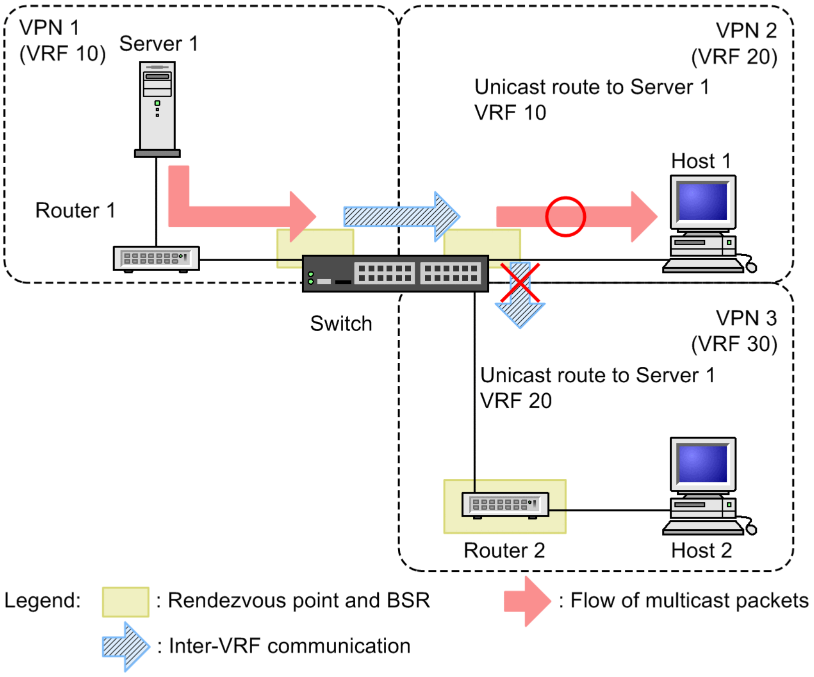
(a) Multi-level VRF forwarding with one device

The IPv4 multicast extranet does not permit multi-level VRF forwarding with one device.

For an IPv4 multicast route of a VRF, the same VRF must be used from the upstream interfaces to the downstream interfaces. Any forwarding requests from an upstream multicast routing information with a different VRF are ignored. When the downstream interfaces of a multicast route use a VRF and the VRF of an upstream interface is switched to another VRF, the downstream interfaces are detached from the target multicast route.

As indicated in the following figure, when there is a unicast route from VPN 3 to VPN 1 via VPN 2, host 1 in VPN 2 can receive IPv4 multicast packets from server 1 in VPN 1, but host 2 in VPN 3 cannot.

Figure 14-23: Example of an IPv4 multicast extranet in which VRF forwarding is not permitted with one device


VPN 1
(VRF 10)
Server 1
Router 1
Switch
VPN 2
(VRF 20)
Unicast route to Server 1
VRF 10
Host 1
VPN 3
(VRF 30)
Unicast route to Server 1
VRF 20
Router 2
Host 2
Legend:
: Rendezvous point and BSR
: Flow of multicast packets
: Inter-VRF communication

(b) Interconnection of PIM-SM and PIM-SSM

If you want to use an IPv4 multicast extranet to perform multicast forwarding between VRFs, make sure that all group addresses used in the multicast forwarding use the same protocol (for IPv4 multicast VRFs, the group addresses to be used with PIM-SSM can be specified for each VRF).

In an IPv4 multicast extranet, multicast forwarding between different protocols is impossible. The figure below shows an example in which inter-VRF forwarding cannot be performed because the protocols do not match. The settings on the Switch in this example are given in the table that follows.

Figure 14-24: Example of an IPv4 multicast extranet in which inter-VRF forwarding cannot be performed because the protocols do not match

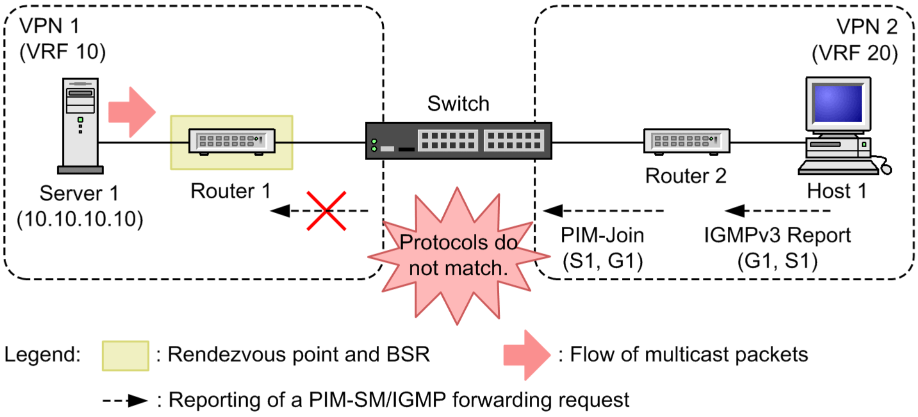


Table 14-14: Settings on the Switch

VPN	Operation protocol	Rendezvous point (The values in parentheses are rendezvous point addresses)	SSM address
1	PIM-SM	Switch (1.1.1.1)	Not used
2	PIM-SSM	None	232.0.0.0/8

G1: 232.10.10.10
S1: 10.10.10.10

14.5 Approaches to network design

14.5.1 IPv4 multicast forwarding

Keep the following in mind when using the Switch to forward multicast packets.

(1) Notes applying to both PIM-SM and PIM-SSM

(a) Operation interface

Operation is performed on interfaces whose IP address mask length is between 8 bits and 30 bits.

(b) Chronological packet overtaking

When the Switch receives multicast data from a sender and a PIM-Join message from the receiver at the same time, some packets might overtake others due to timing, causing the order of the packets to be switched.

(c) Stopped forwarding due to a routing program restart

Keep in mind that when the `restart ipv4-multicast` command is used to restart the IP multicast routing program, multicast communication is stopped until multicast routing information is relearned.

(d) Multihoming

IPv4 multicast is not performed on multihomed interfaces.

(2) Using PIM-SM

Keep the following in mind when using PIM-SM.

(a) Packet loss during software forwarding

The Switch sets a multicast forwarding entry in hardware when the first multicast packet is received to perform multicast communication. Because multicast packets are forwarded in software until the multicast forwarding entry is created, some packets might be temporarily lost depending on the amount of multicast communication traffic.

(b) Duplicate forwarding or packet loss during path switching

When the Switch switches from sending packets by forwarding multicast packets via a rendezvous point to sending packets via the shortest path, duplicate forwarding or packet loss might occur temporarily.

For details about switching from sending packets by forwarding multicast packets via a rendezvous point to sending packets via the shortest path, see *14.4.2 IPv4 PIM-SM*.

(c) Packet overtaking during a switchover from software forwarding to hardware forwarding

When a multicast forwarding entry is finished being set in hardware, the Switch switches from software-based multicast packet forwarding to hardware forwarding. Some packets might be overtaken at this point, causing changes in packet order.

(d) Device address reachability

If the Switch is used as a rendezvous point and bootstrap router, the IPv4 address set as the local address in switch management information is used as the address for the rendezvous point and bootstrap router. The local address in this device management information must be reachable for route recognition and communication over unicast for all switches using multicast communication.

(e) PIM-Register message checksums

In a system configuration that contains both the Switch and other switches, multicast communication might not be possible depending on differences in calculated checksum ranges for

PIM-Register messages (encapsulation packets). If multicast forwarding is not performed due to checksum errors for Register messages on rendezvous points, use the `ip pim register-checksum` configuration command for the Switch to change the range for calculating PIM checksums.

(f) Static rendezvous point

Static rendezvous point functionality allows the rendezvous point to be specified without using a BSR. Static rendezvous points are set by configuration.

The static rendezvous point can also exist with rendezvous point candidates advertised by Bootstrap messages from a BSR. In that case, the static rendezvous point is given priority over the rendezvous point candidates advertised by Bootstrap messages from a BSR.

If a rendezvous point candidate router recognizes that its local address is the rendezvous point router address, the router functions as the rendezvous point. Therefore, if a network using the static rendezvous point is designed without using a BSR, the static rendezvous point settings must also be specified on all rendezvous point candidate routers.

Also, if the static rendezvous point is used, the same settings need to be specified for all routers on the same network.

14.5.2 Redundant paths (path switching due to failure and other reasons)

The following explains the point to keep in mind for the Switch when multicast paths are redundant.

(1) Using PIM-SM

Keep in mind that for PIM-SM, it might take some time for multicast communication to restart for the following route switching. In the following time indications, the switching time for sender network information (unicast routing information) is represented as *U*.

The time shown here indicates how long it takes the Switch to perform switching. Because of this, join notification time (from the connection request from the Switch to the upstream router to the arrival of the multicast data from the upstream) is required to actually restart multicast forwarding.

- When a prioritized route is switched, the following time might be required until communication restarts:

  U + 20 seconds

- When a prioritized route is switched to a redundant route due to line failure, the following time is required until communication restarts:

  When *U* < 5: 5 to 10 seconds
  When *U* ≥ 5: *U* + 0 to 60 seconds

- When a redundant route is switched to a prioritized route due to line restoration, the following time might be required until the prioritized route is used in communication:

  0 seconds

  The following time is necessary to perform switching back.
  U + (*PIM-Hello-message-sending-interval-for-sender-direction* + 20) seconds
  (Default: *U* + 30 + 20 = *U* + 50 seconds)

- When a rendezvous point and BSR are switched to the Switch (the Switch becomes the rendezvous point and BSR, such as due to failure or configuration), the following time is required until communication restarts.

  The time until communication restarts differs depending on the rendezvous point or BSR. The default values are shown in parentheses.

- For rendezvous point switching: 285 seconds

RP-Holdtime `(150 seconds) +` *Query-interval* `(125 seconds) +` *Query Response Interval* `(10 seconds)`

- For BSR switching: At most 348 seconds

Bootstrap-Timeout `(130 seconds) +` *BS_Rand_Override* `(5 to 23 seconds) +`
Bootstrap-Period `(60 seconds)`
`+` *Query-interval* `(125 seconds) +` *Query Response Interval* `(10 seconds)`

- When a DR is switched to the Switch, the following time is required until communication restarts. The default values are shown in parentheses.
  - For DR switching: 240 seconds

Hello-Holdtime `(105 seconds) +` *Query-interval* `(125 seconds) +` *Query Response Interval* `(10 seconds)`

Multicast communication might stop at these times not only for redundant route switching due to failure, but also when path switching is performed explicitly due to a configuration change. Plan ahead when changing the system configuration.

(2) Using PIM-SSM

Keep in mind that for PIM-SSM, it might take some time for multicast communication to restart for the following path switching. In the following time indications, the switching time for sender network information (unicast routing information) is represented as U.

The time shown here indicates how long it takes the Switch to perform switching. Because of this, join notification time (from the connection request from the Switch to the upstream router to the arrival of the multicast data from the upstream) is required to actually restart multicast forwarding.

- When a prioritized route is switched, the following time might be required until communication restarts:

U `+ 20 seconds`

- When a prioritized route is switched to a redundant route due to line failure, the following time is required until communication restarts:

`When` *U* `< 5: 5 to 10 seconds`
`When` *U* `≥ 5:` *U* `+ 0 to 135 seconds`

- When a redundant route is switched to a prioritized route due to line restoration, the following time might be required until the prioritized route is used in communication:

`0 seconds`

The following time is necessary to perform switching back.
U `+ (`*PIM-Hello-message-sending-interval-for-sender-direction* `+ 20) seconds`
`(Default:` *U* `+ 30 + 20 =` *U* `+ 50 seconds)`

- When a DR is switched to the Switch, the following time is required until communication restarts. The default values are shown in parentheses.
  - For DR switching: 240 seconds

Hello-Holdtime `(105 seconds) +` *Query-interval* `(125 seconds) +` *Query Response Interval* `(10 seconds)`

14.5.3 Examples of appropriate network configurations

(1) Configuration using PIM-SM

This configuration is appropriate in the following cases:

- The number of users sending multicast packets is not limited.
- The number of users sending multicast packets is large.

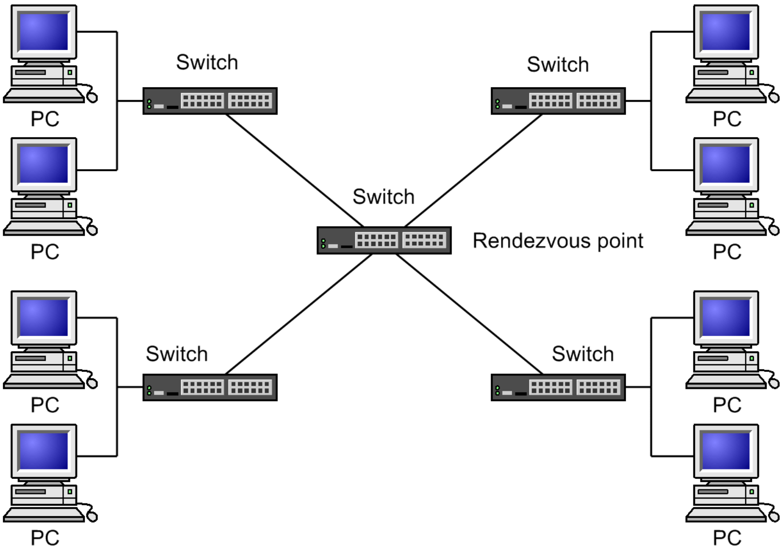
Network environment

1. The IP unicast routing protocol runs on all routers.
2. PIM-SM is used as the multicast routing protocol between the Switches.
3. IGMP is used for group management control between each group and the Switches.
4. One switch is used as the rendezvous point and BSR.
5. A static rendezvous point can be specified as the rendezvous point to reduce the time needed to determine the rendezvous point upon system startup.

Configuration diagram

The following diagram shows a configuration.

Figure 14-25: Configuration using PIM-SM



(2) Configuration using PIM-SSM

This configuration is appropriate in the following cases:

- The number of users sending multicast packets is limited (such as for distribution servers).
- Broadband multicast communication is used.
- Multi-channel multicast communication is used.

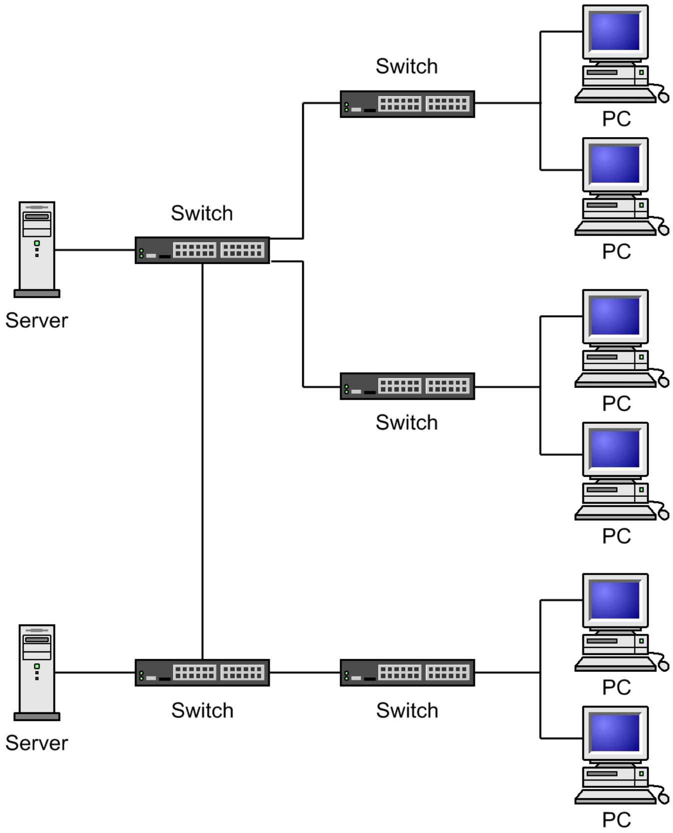
Network environment

1. The IP unicast routing protocol runs on all routers.
2. PIM-SSM is used as the multicast routing protocol between the Switches. PIM-SSM is extended PIM-SM functionality.
3. IGMPv2 is used for group management control (linked operation with SSM is needed for IGMPv2).

Configuration diagram

The following diagram shows a configuration.

Figure 14-26: Configuration using PIM-SSM



14.5.4 Notes on network configurations

Multicasting is suited to 1: N unidirectional communication in which data is distributed from one server (sender) to multiple groups (receivers). The following gives notes on network configurations appropriate for IPv4 multicasting.

(1) Notes applying to both PIM-SM and PIM-SSM

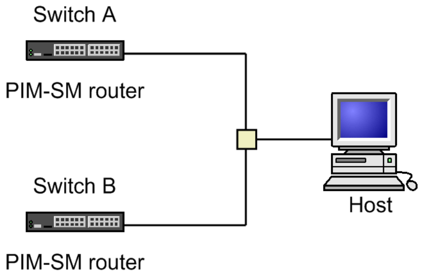
(a) Configurations requiring special attention

Take care of the following if using PIM-SM or PIM-SSM in the following configurations.

- In a configuration such as the one in the figure below, in which multiple routers are directly connected to the host on the same network, make sure that PIM-SM always runs at the interface.

  If IGMP runs without running PIM-SM at an interface on which multiple routers exist on the same network, multicast data might be forwarded twice.

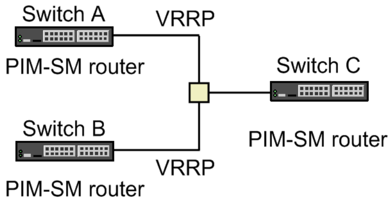
Figure 14-27: Configuration requiring special attention (when connecting multiple routers to a host)



- In configurations such as the one in the figure below, in which the PIM protocol cannot detect the upstream router, multicast communication cannot be performed. This is because the environment has a static route for which the virtual interface set as the gateway has VRRP set by Switch C for Switch A and for Switch B.

  To perform multicast communication in this configuration, a static route needs to be set for Switch C, with the real address of Switch A or Switch B used as the gateway address to the rendezvous point address, BSR address, and multicast data source address.

  Figure 14-28: Configuration requiring special attention (when configuring VRRP setting)

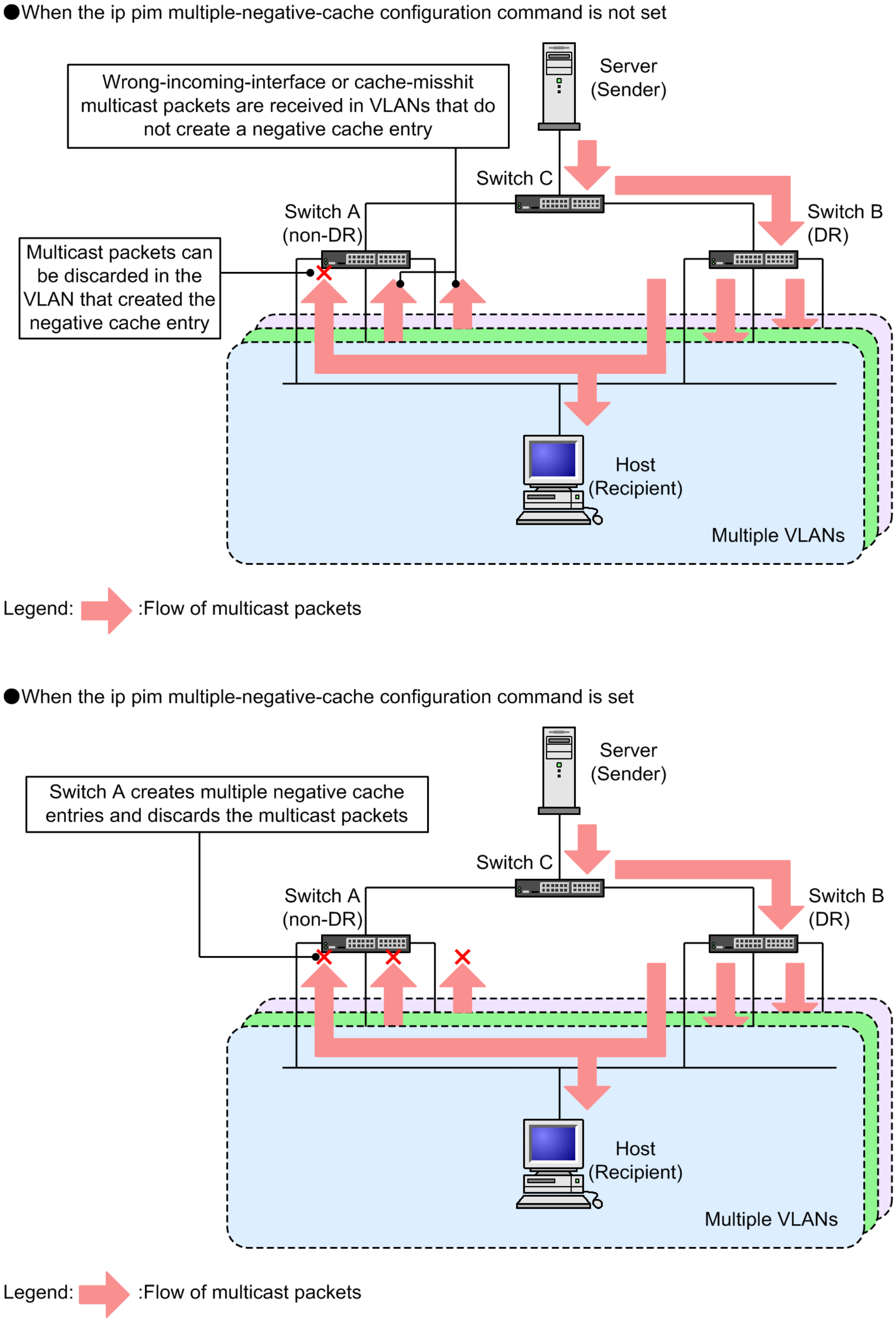


(b) Load due to receiving unnecessary multicast packets from multiple VLANs

When two routers exist in a network for multicast network configurations such as in the following figure, the non-DR (Switch A) receives multicast packets that are forwarded to the user from DR (Switch B). At this time, Switch A creates negative cache entries and discards the hardware for the unnecessary packets.

The following figure illustrates the discarding of multicast packets when multiple VLANs are set.

Figure 14-29: Discarding of multicast packets when multiple VLANs are set


●When the ip pim multiple-negative-cache configuration command is not set
Wrong-incoming-interface or cache-misshit multicast packets are received in VLANs that do not create a negative cache entry
Server (Sender)
Switch C
Switch A (non-DR)
Switch B (DR)
Multicast packets can be discarded in the VLAN that created the negative cache entry
Host (Recipient)
Multiple VLANs
Legend: :Flow of multicast packets
●When the ip pim multiple-negative-cache configuration command is set
Switch A creates multiple negative cache entries and discards the multicast packets
Server (Sender)
Switch C
Switch A (non-DR)
Switch B (DR)
Host (Recipient)
Multiple VLANs
Legend: :Flow of multicast packets

When the `ip pim multiple-negative-cache` configuration command is not set, one negative cache entry is created for one multicast communication (S, G) of each global network or VRF. If multiple VLANs are set in the same port, only negative cache entries that consider the first VLAN that received a packet as a receiving interface will be created. Other VLANs will not create negative cache entries even if the same multicast packet arrives.

For this reason, although multicast packets can be discarded in the VLAN that created the negative cache entry, if a large number of multicast packets of wrong-incoming interface or cache-misshit is received in another VLAN, packet congestion occurs in the CPU processing.

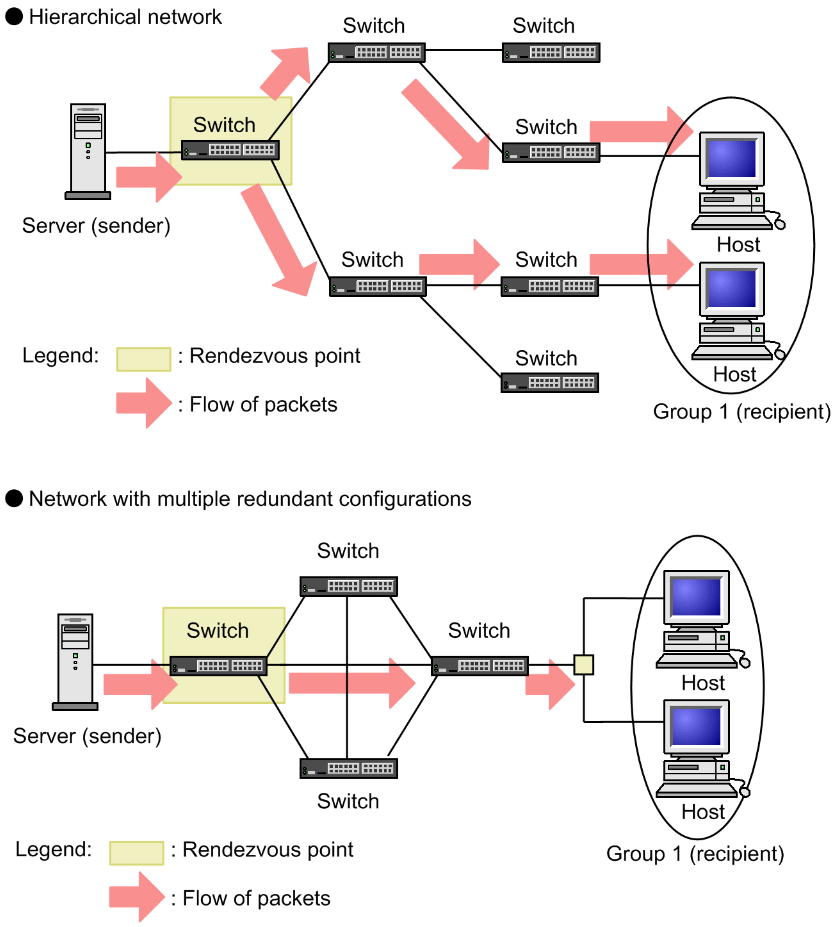
When the `ip pim multiple-negative-cache` configuration command is set, two or more negative cache entries are created for each multicast communication (S, G). Because negative cache entries can be created in each VLAN, packet congestion in the CPU processing can be prevented even if a large number of multicast packets of wrong-incoming interface or cache-misshit is received.

(2) PIM-SM

(a) Recommended configurations

We recommend network configurations based on PIM-SM for hierarchical network configurations and those with redundant routes. However, special care is needed for the placement of the rendezvous point. The following figure shows a recommended network configuration for PIM-SM.

Figure 14-30: Recommended network configuration for PIM-SM



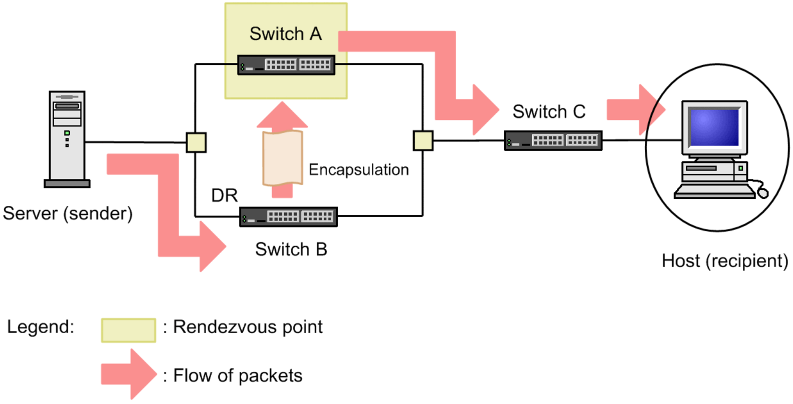
(b) Configurations requiring special attention

The following configuration needs special attention.

- In configurations such as the one shown in the figure below, in which two or more routers connected directly to the sender exist on the same network, when choosing one as the rendezvous point, make sure that the DR and the rendezvous point are the same.

  If a switch other than the rendezvous point is used as the DR, the load on Switches A and B will increase because PIM-Register messages are sent from the DR to the rendezvous point. Also, when multicast packets within the PIM-Register messages are forwarded, packet loss might occur on the rendezvous point. Note that if the rendezvous point is the DR, encapsulation is not performed using PIM-Register messages.

Figure 14-31: Configuration requiring special attention (when connecting a sender to multiple routers)



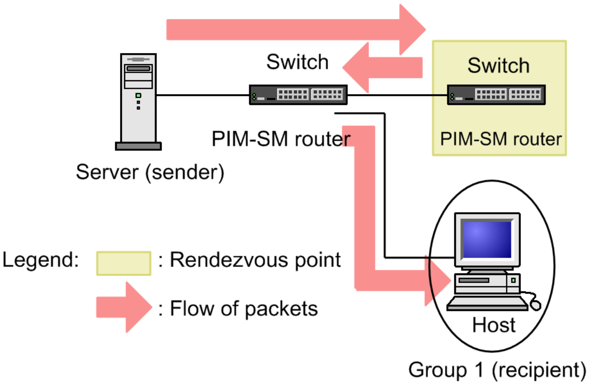
(c) Inappropriate configurations

Do not use PIM-SM for configurations such as the following.

- Configuration in which a recipient exists between the sender and rendezvous point

  When multicast communication with group 1 is performed from the server in the following configuration, forwarding via the rendezvous point cannot be performed efficiently.

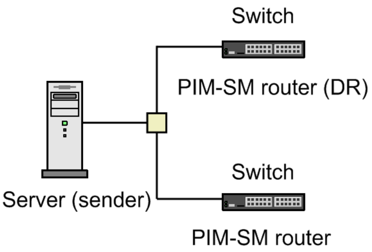
  Figure 14-32: Inappropriate configuration (when a recipient exists between a sender and a rendezvous point)



- Multiple PIM-SM routers are running on the same line as the sender

  When the server sends multicast data in the configuration below, undue load is placed on the non-DR PIM-SM router, which might significantly impact other functionality on the Switch. In this case, split the line as appropriate.

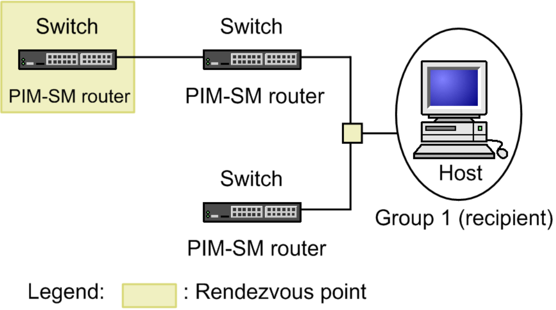
  Figure 14-33: Inappropriate configuration (when connecting a sender to multiple routers)



- Multiple PIM-SM routers are running on the same line as the multicast group (receiver), and one of the PIM-SM routers does not connect to the rendezvous point

When multicast communication to group 1 is performed in the following configuration, the shortest path between the sender and group 1 might not be established. Connect PIM-SM router 1 and PIM-SM router 2 to the rendezvous point.

Figure 14-34: Inappropriate configuration (when a router that is not connected to a rendezvous point exists)



(3) PIM-SSM

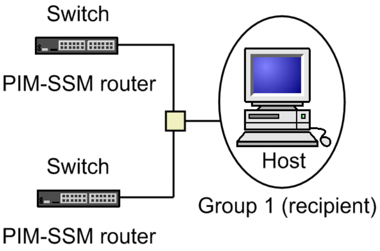
(a) Configurations requiring special attention

The following configuration needs special attention.

- Multiple PIM-SSM routers are running on the same line as the multicast group (receiver)

  If running IGMPv2 PIM-SSM in the following configuration, execute the `ip pim ssm` and `ip igmp ssm-map static` configuration commands on all routers on the same line.

  Figure 14-35: Configuration requiring special attention (when connecting multiple routers to a host)



(4) PIM-SM VRF gateways [OS-L3SA]

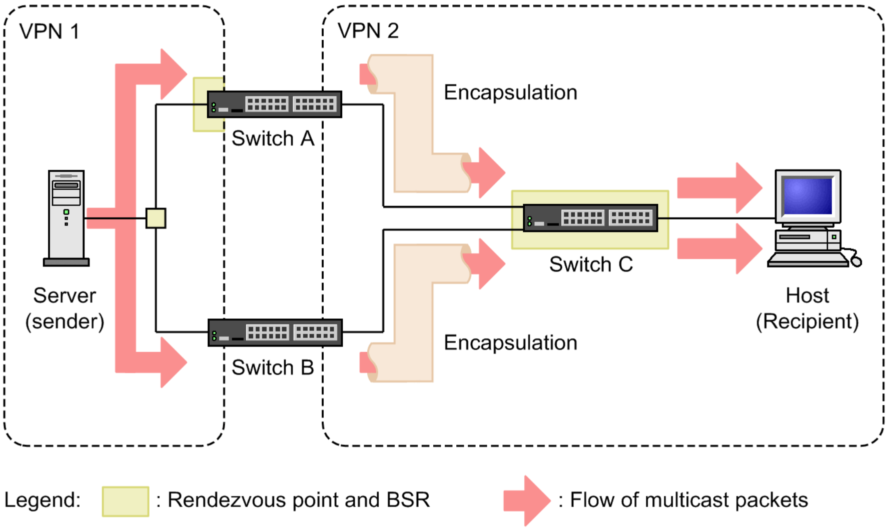
(a) Configurations requiring special attention

The following configuration needs special attention:

- In an IPv4 multicast extranet that uses PIM-SM VRF gateways, if two or more VRF border routers are used to create a redundant configuration, make sure that one of the VRF border routers is set as the rendezvous point.

  If the rendezvous point is not a VRF border router, each VRF border router forwards the same IPv4 multicast packets until the shortest path delivery tree is generated.

Figure 14-36: Configuration requiring special attention (when deploying more than one VRF boundary router for a redundant configuration)



Chapter

15. Settings and Operation for IPv4 Multicasting

This chapter describes how to set up and check the status of an IPv4 multicast configuration.

15.1 Configuration

15.1.1 List of configuration commands

The following table describes the configuration commands for IPv4 multicasting.

Table 15-1: List of configuration commands

Command name	Description
ip igmp group-limit	Specifies the maximum number of groups that can run on an interface.
ip igmp router	Runs IGMP on the interface.
ip igmp source-limit	Specifies the maximum number of sources during group participation.
ip igmp ssm-map enable	Enables IPv4 PIM-SSM mapping operation to be used with IGMPv2 or IGMPv3 (EXCLUDE mode).
ip igmp ssm-map static	Sets the group address and source address for which PIM-SSM runs.
ip igmp static-group	Enables static additions to IGMP groups.
ip igmp version	Changes the IGMP version.
ip multicast-routing	Enables the IPv4 multicast functionality to be used.
ip pim bsr-candidate	Sets the BSR.
ip pim deletion-delay-time	Changes the deletion delay time.
ip pim keep-alive-time	Changes the keep-alive time.
ip pim max-interface	Changes the maximum number of interfaces that can run IPv4 PIM.
ip pim mcache-limit	Specifies the maximum number of multicast forwarding entries.
ip pim message-interval	Changes the sending interval for join or prune messages.
ip pim mroute-limit	Specifies the maximum number of multicast routing information entries.
ip pim multiple-negative-cache	Specifies that the same (S, G) multiple negative cache entries can be created for each VLAN.
ip pim negative-cache-time	Changes the negative cache time.
ip pim query-interval	Changes the sending interval for Hello messages.
ip pim register-checksum	Changes the checksum range for PIM-Register messages.
ip pim register-probe-time	Specifies the register probe time.
ip pim rp-address	Sets the static rendezvous point.
ip pim rp-candidate	Sets a rendezvous point candidate.
ip pim rp-mapping-algorithm	Specifies the rendezvous point selection algorithm.
ip pim sparse-mode	Sets the IPv4 PIM-SM.
ip pim ssm	Sets the IPv4 PIM-SSM address.
ip pim vrf-gateway	Sets the PIM-SM VRF gateway.

15.1.2 Overview of configuration

Refer to the setting examples below according to the configuration used.

- When PIM-SM is used
  - Configuring IPv4 multicast routing
  - Configuring IPv4 PIM-SM
  - Configuring an IPv4 PIM-SM rendezvous point candidate (when using the local switch as a rendezvous point)
  - Configuring an IPv4 PIM-SM BSR candidate (when using the local switch as the BSR)
  - Configuring IGMP
- When PIM-SM (static rendezvous point) is used
  - Configuring IPv4 multicast routing
  - Configuring IPv4 PIM-SM
  - Configuring an IPv4 PIM-SM rendezvous point candidate (when using the local switch as a rendezvous point)
  - Configuring an IPv4 PIM-SM static rendezvous point
  - Configuring IGMP
- When PIM-SSM is used
  - Configuring IPv4 multicast routing
  - Configuring IPv4 PIM-SM
  - Configuring IPv4 PIM-SSM
  - Configuring IGMP
- When PIM-SM is used in a VRF
  - Configuring IPv4 multicast routing for a VRF
  - Configuring IPv4 PIM-SM for a VRF
  - Configuring an IPv4 PIM-SM rendezvous point candidate for a VRF (when using the local switch as a rendezvous point in the target VPN)
  - Configuring an IPv4 PIM-SM BSR candidate for a VRF (when using the local switch as the BSR in the target VPN)
  - Configuring IGMP for a VRF
- When PIM-SM (static rendezvous point) is used in a VRF
  - Configuring IPv4 multicast routing for a VRF
  - Configuring IPv4 PIM-SM for a VRF
  - Configuring an IPv4 PIM-SM rendezvous point candidate for a VRF (when using the local switch as a rendezvous point in the target VPN)
  - Configuring an IPv4 PIM-SM static rendezvous point for a VRF
  - Configuring IGMP for a VRF
- When PIM-SSM is used for a VRF
  - Configuring IPv4 multicast routing for a VRF
  - Configuring IPv4 PIM-SM for a VRF

  - Configuring IPv4 PIM-SSM for a VRF
  - Configuring IGMP for a VRF
- ■ When PIM-SM is used for a VRF (extranet) (PIM-SM VRF gateway)
  - Configuring IPv4 multicast routing for a VRF
  - Configuring IPv4 PIM-SM for a VRF
  - Configuring an IPv4 PIM-SM rendezvous point candidate for a VRF (when using the local switch as a rendezvous point)
  - Configuring an IPv4 PIM-SM BSR candidate for a VRF (when using the local switch as the BSR)
  - Configuring a PIM-SM VRF gateway
  - Configuring IGMP for a VRF
- ■ When PIM-SSM is used for a VRF (extranet)
  - Configuring IPv4 multicast routing for a VRF
  - Configuring IPv4 PIM-SM for a VRF
  - Configuring IPv4 PIM-SSM for a VRF
  - Configuring an IPv4 multicast extranet
  - Configuring IGMP for a VRF

15.1.3 Configuring IPv4 multicast routing

Points to note

The example below shows how to configure the settings to run IPv4 multicast routing on the Switch. Specify the settings in global configuration mode.

Note that, in addition to the settings described here, you must configure IPv4 PIM on at least one interface using the `ip pim sparse-mode` command.

Command examples

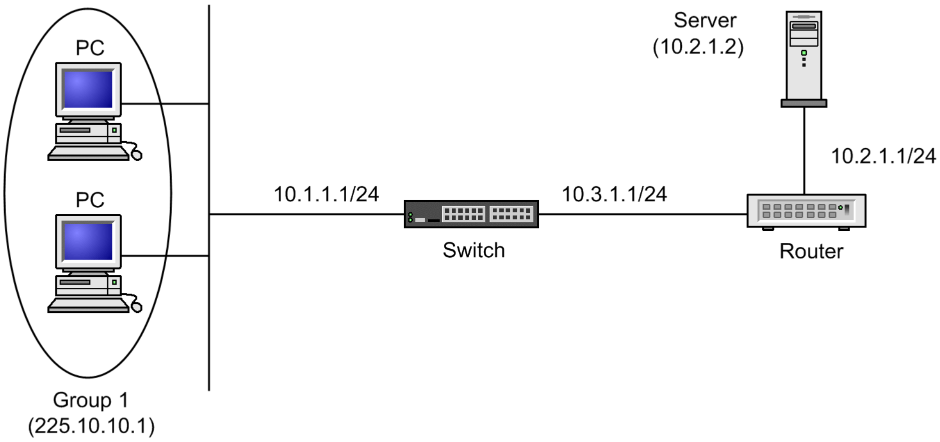
1.
```
(config)# ip multicast-routing
```
Enables IPv4 multicast functionality.

15.1.4 Configuring IPv4 PIM-SM

Points to note

IPv4 PIM-SM (sparse mode) needs to be set on interfaces running IPv4 multicast routing. The settings for IPv4 PIM-SM (sparse mode) are specified in interface configuration mode. The following figure shows a PIM-SM configuration example in which 10.3.1.1/24 is used as the interface IP address.

Figure 15-1: PIM-SM configuration example



Command examples

1. `(config)# interface vlan 20`

   Sets the VLAN.

2. `(config-if)# ip address 10.3.1.1 255.255.255.0`

   Sets the IP address.

3. `(config-if)# ip pim sparse-mode`

   Specifies operation as an IPv4 PIM-SM.

15.1.5 Configuring an IPv4 PIM-SM rendezvous point candidate

Points to note

When the Switch is used as rendezvous point candidates, the address to the loopback 0 interface is set as the rendezvous point address, and the following settings are specified in global configuration mode. In this example, 225.10.10.0/24 is used as the address of the managed multicast group, and 10.10.10.10 is used as the loopback address for the Switch.

Command examples

1.
```
(config)# interface loopback 0
(config-if)# ip address 10.10.10.10
(config-if)# exit
```
   Sets the loopback address.

2.
```
(config)# access-list 1 permit 225.10.10.0 0.0.0.255
(config)# exit
```
   Creates an access list for the multicast group addresses to be managed.

3. `(config)# ip pim rp-candidate loopback 0 group-list 1`

   Sets the Switch as a rendezvous point candidate, (The access list created in step 2 is used to specify the multicast group address to be managed.)

15.1.6 Configuring an IPv4 PIM-SM BSR candidate

Points to note

When the Switch is used as a BSR candidate, the address to the loopback 0 interface is set as the BSR address, and the following settings are specified in global configuration mode. In this example, 10.10.10.10 is used as the loopback address for the Switch.

Command examples

1.
```
(config)# interface loopback 0
(config-if)# ip address 10.10.10.10
(config-if)# exit
```
Sets the loopback address.

2.
```
(config)# ip pim bsr-candidate loopback 0
```
Sets the Switch as a BSR candidate.

15.1.7 Configuring an IPv4 PIM-SM static rendezvous point

Points to note

The example below shows how to specify the static rendezvous point by using global configuration mode for the configuration. In this example, 10.10.10.1 is used as the switch address for the static rendezvous point.

Command examples

1.
```
(config)# ip pim rp-address 10.10.10.1
```
Specifies 10.10.10.1 as the rendezvous point.

15.1.8 Configuring IPv4 PIM-SSM

(1) IPv4 PIM-SSM address settings

Points to note

The example below shows how to specify the settings shown below in global configuration mode, to use IPv4 PIM-SSM for the Switch. These settings cause IPv4 PIM-SSM to run within the specified SSM address range on interfaces for which IPv4 PIM-SM is set. Only one SSM address setting can be used on the Switch. In this example, the default (232.0.0.0/8) is used for the SSM address range for which PIM-SSM runs. To specify the SSM address range, use the `ip pim ssm range` command.

Command examples

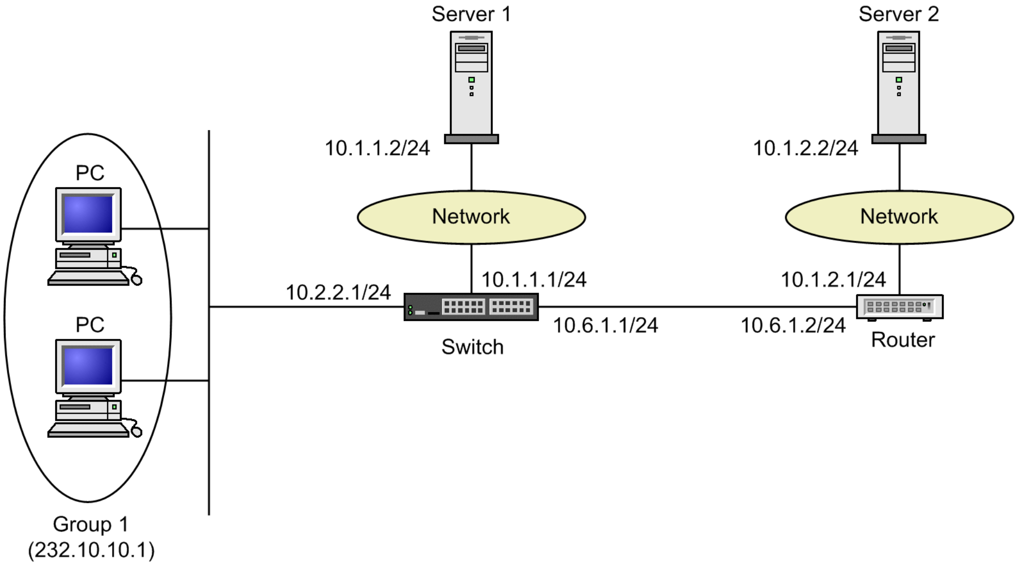
1.
```
(config)# ip pim ssm default
```
Enables use of IPv4 PIM-SSM (with an SSM address range of 232.0.0.0/8).

(2) Settings run to enable linked operation of IPv4 PIM-SSM for IGMPv2/IGMPv3 (EXCLUDE mode)

Points to note

Because source addresses cannot be differentiated with IGMPv2 or IGMPv3 (EXCLUDE mode), linkage to the PIM-SSM cannot be performed. With the Switch, the group address and source address for which PIM-SSM is run can be set to link with PIM-SSM. The group address for which PIM-SSM is run needs to be within the SSM address range specified by the IPv4 PIM-SSM address setting. In the following PIM-SSM configuration example, 232.10.10.1 is used as the group address when two servers are used with 10.1.1.2 as the source address of server 1, and 10.1.2.2 as the source address of server 2.

Figure 15-2: PIM-SSM configuration example



Command examples

1. `(config)# access-list 2 permit 232.10.10.1`

   Creates an access list for which the group address is specified.

2. `(config)# ip igmp ssm-map static 2 10.1.1.2`

   `(config)# ip igmp ssm-map static 2 10.1.2.2`

   Sets the group address for which PIM-SSM is run, and the source addresses for server 1 and server 2 (The access list created in step 1 is used to specify the group address).

3. `(config)# ip igmp ssm-map enable`

   Enables IPv4 PIM-SSM to be used for IGMPv2/IGMPv3 (EXCLUDE mode).

15.1.9 Configuring IGMP

Points to note

IGMP must be configured for interfaces on which IGMP will run. However, IGMP also can run if IPv4 PIM-SM (`sparse` mode) is set on an interface.

By default, IGMP uses a mixed mode of versions 2 and 3. Use the `ip igmp version` configuration command to change the IGMP version.

Command examples

1. `(config-if)# ip igmp router`

Specifies that IGMP runs on the target interface in mixed mode for versions 2 and 3 (default).

15.1.10 Configuring IPv4 multicast routing for a VRF [OS-L3SA]

Points to note

To use IPv4 multicast in VRFs, run IPv4 multicast routing for each VRF. Specify the settings in global configuration mode. The example below configures IPv4 multicast routing for VRF 10.

Note that, in addition to the settings described here, you must configure IPv4 PIM on at least one interface for each VRF by using the `ip pim sparse-mode` command.

Command examples

1.
```
(config)# vrf definition 10
(config-vrf)# exit
```
Configures VRF 10.

2.
```
(config)# ip multicast-routing vrf 10
```
Enables IPv4 multicasting in VRF 10.

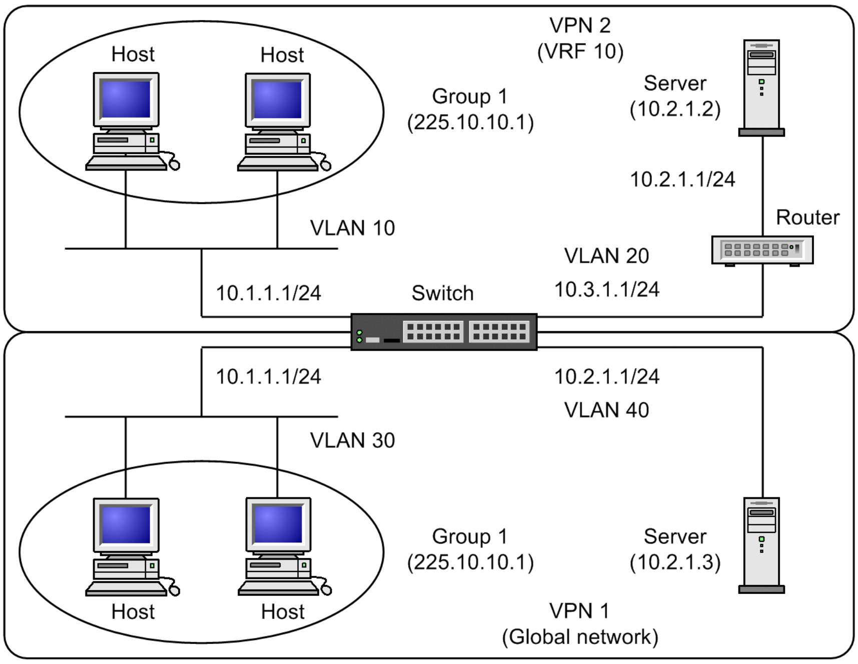
15.1.11 Configuring IPv4 PIM-SM for a VRF [OS-L3SA]

Points to note

To use IPv4 PIM-SM in a VRF, configure the IPv4 multicast routing functionality for the VRF, and configure IPv4 PIM-SM (`sparse` mode) on the interface of the VRF.

The settings for IPv4 PIM-SM (sparse mode) are specified in interface configuration mode. The figure below shows an IPv4 PIM-SM configuration example in which VPN 2 is associated with VRF 10 and 10.3.1.1/24 is used as the IP addresses of the VLAN 20 interface for VRF 10.

Figure 15-3: IPv4 PIM-SM configuration example for a VRF



Command examples

1. `(config)# interface vlan 20`

   Configures VLAN 20.

2. `(config-if)# vrf forwarding 10`

   Configures VLAN 20 in VRF 10.

3. `(config-if)# ip address 10.3.1.1 255.255.255.0`

   Sets an IP address for VLAN 20.

4.
   ```
   (config-if)# ip pim sparse-mode
   (config-if)# exit
   ```

   Sets IPv4 PIM-SM for VLAN 20.

15.1.12 Configuring an IPv4 PIM-SM rendezvous point candidate for a VRF [OS-L3SA]

Points to note

When the Switch is used as a rendezvous point candidate, the address to the loopback interface of the target VRF is set as the rendezvous point address, and the following settings are specified in global configuration mode. In this example, 225.10.10.0/24 is used as the multicast group address managed for VRF 10, loopback 30 is used as the loopback interface for the target VRF, and 10.10.10.10 is used as the loopback address.

Command examples

1. `(config)# interface loopback 30`

```
(config-if)# vrf forwarding 10
(config-if)# ip address 10.10.10.10
(config-if)# exit
```

Sets a loopback address for the `loopback 30` loopback interface of VRF 10.

2. `(config)# access-list 1 permit 225.10.10.0 0.0.0.255`

   Creates an access list for the multicast group addresses to be managed in VRF 10.

3. `(config)# ip pim vrf 10 rp-candidate loopback 30 group-list 1`

   Sets the Switch as the rendezvous point candidate for VRF 10. (The access list created in step 2 is used to specify the multicast group address to be managed.)

15.1.13 Configuring an IPv4 PIM-SM BSR candidate for a VRF [OS-L3SA]

Points to note

When the Switch is used as a BSR candidate, the address to the loopback 30 interface is set as the BSR address, and the following settings are specified in global configuration mode. In this example, 10.10.10.10 is used as the loopback address for the Switch.

Command examples

1.
```
(config)# interface loopback 30
(config-if)# vrf forwarding 10
(config-if)# ip address 10.10.10.10
(config-if)# exit
```

   Sets a loopback address for the loopback 30 loopback interface of VRF 10.

2. `(config)# ip pim vrf 10 bsr-candidate loopback 30`

   Sets the Switch as a BSR candidate of VRF 10.

15.1.14 Configuring an IPv4 PIM-SM static rendezvous point for a VRF [OS-L3SA]

Points to note

The configuration below is performed in global configuration mode to specify a static rendezvous point for a VRF. In this example, 10.10.10.1 is used as the IP address for the static rendezvous point of the VRF 10.

Command examples

1. `(config)# ip pim vrf 10 rp-address 10.10.10.1`

   Specifies 10.10.10.1 as the rendezvous point for VRF 10.

15.1.15 Configuring IPv4 PIM-SSM for a VRF [OS-L3SA]

(1) IPv4 PIM-SSM address settings

Points to note

The settings below must be specified in global configuration mode to use IPv4 PIM-SSM for the Switches in a VRF. These settings enable IPv4 PIM-SSM to run within the specified SSM address range on the VRF interfaces for which IPv4 PIM-SM is set. The Switch can set only one SSM address for each VRF. In this example, the default (232.0.0.0/8) is used for the SSM address range for which PIM-SSM runs in VRF 10.

Command examples

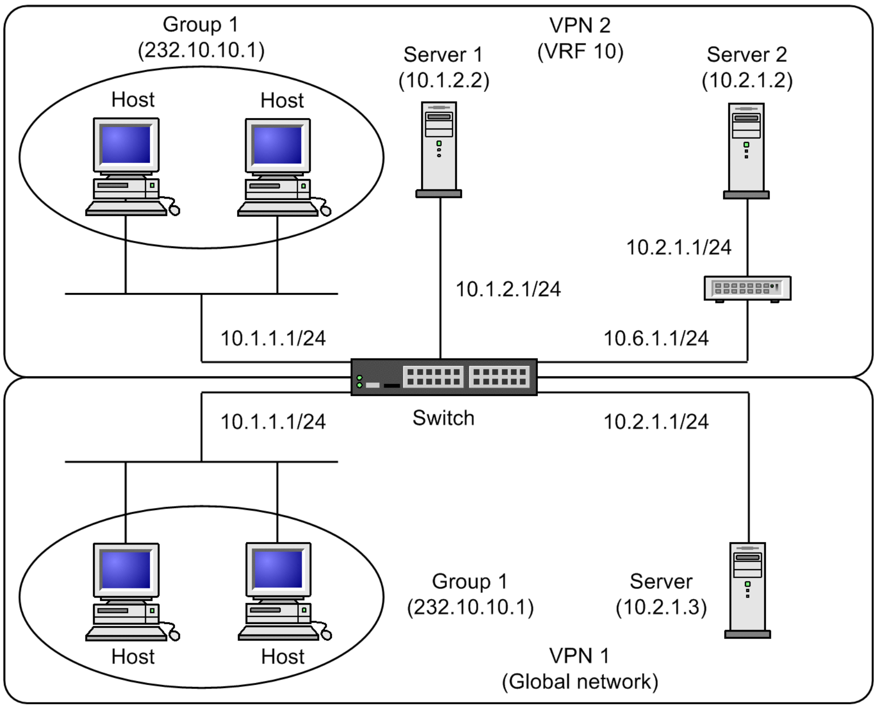
1.
```
(config)# ip pim vrf 10 ssm default
```
Enables use of IPv4 PIM-SSM in VRF 10 (with an SSM address range of 232.0.0.0/8).

(2) Settings run to enable linked operation of IPv4 PIM-SSM for IGMPv2/IGMPv3 (EXCLUDE mode)

Points to note

Because source addresses cannot be differentiated with IGMPv2 or IGMPv3 (EXCLUDE mode), linkage to the IPv4 PIM-SSM cannot be performed. With the Switches, the group address and source address for which IPv4 PIM-SSM is run can be set to link with IPv4 PIM-SSM. This functionality must be set for each VRF. The group address for which IPv4 PIM-SSM is run needs to be within the SSM address range specified for the target VRF in the IPv4 PIM-SSM address setting. The figure below shows an IPv4 PIM-SSM configuration example in which VPN 2 is associated with VRF 10 and 232.10.10.1 is used as the group address for VPN 2. When two servers are used in the same VPN, 10.1.2.2 is used as the source address of server 1 and 10.2.1.2 is used as the source address of server 2.

Figure 15-4: IPv4 PIM-SSM configuration example for a VRF



Command examples

1.
```
(config)# access-list 2 permit 232.10.10.1
```
Creates an access list for which the group address is specified.

2.
```
(config)# ip igmp ssm-map vrf 10 static 2 10.1.2.2
(config)# ip igmp ssm-map vrf 10 static 2 10.2.1.2
```
For VRF 10, sets the group address for which IPv4 PIM-SSM is run in VPN 2 and the source addresses of server 1 and server 2 (The access list created in step 1 is used to specify the group address).

3.
```
(config)# ip igmp vrf 10 ssm-map enable
```
For VRF 10, enables IPv4 PIM-SSM to be used for IGMPv2/IGMPv3 (EXCLUDE mode).

15.1.16 Configuring IGMP for a VRF [OS-L3SA]

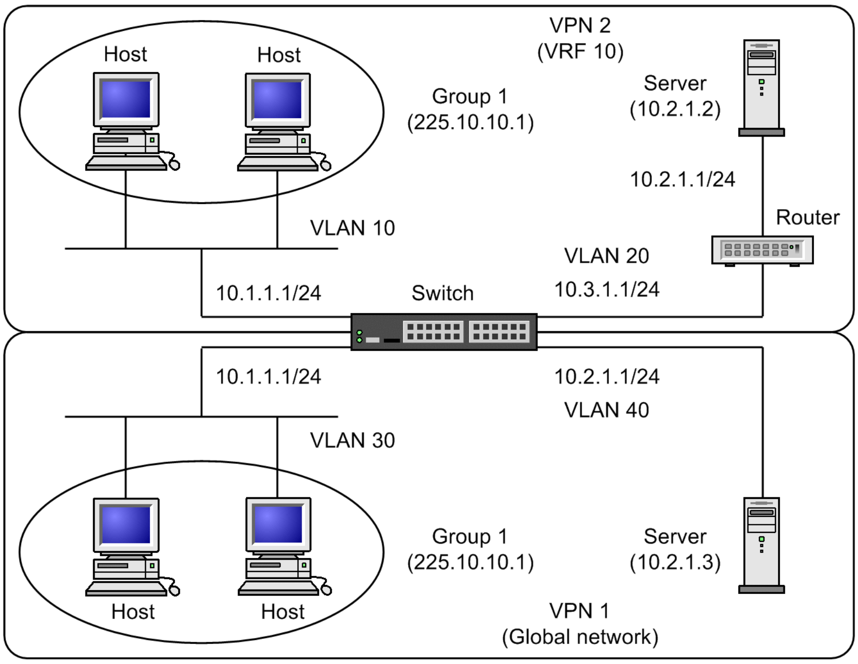
Points to note

Configure IGMP for a VRF to run IGMP on the VRF. However, IGMP can also run if IPv4 PIM-SM (sparse mode) is set on the interface of the VRF.

By default, IGMP uses a mixed mode of versions 2 and 3. Use the `ip igmp version` configuration command to change the IGMP version.

The following figure shows an example configuration in which VPN 2 is associated with VRF 10 and 10.1.1.1/24 is used as the IP addresses of the VLAN 10 interface for VRF 10.

Figure 15-5: IGMP configuration example for a VRF



Command examples

1.
```
(config)# interface vlan 10
```
Configures VLAN 10.

2. `(config-if)# vrf forwarding 10`

   Sets VLAN 10 for VRF 10.

3. `(config-if)# ip address 10.1.1.1 255.255.255.0`

   Sets an IP address for VLAN 10.

4.
   ```
   (config-if)# ip igmp router
   (config-if)# exit
   ```

   Specifies IGMP for VLAN 10.

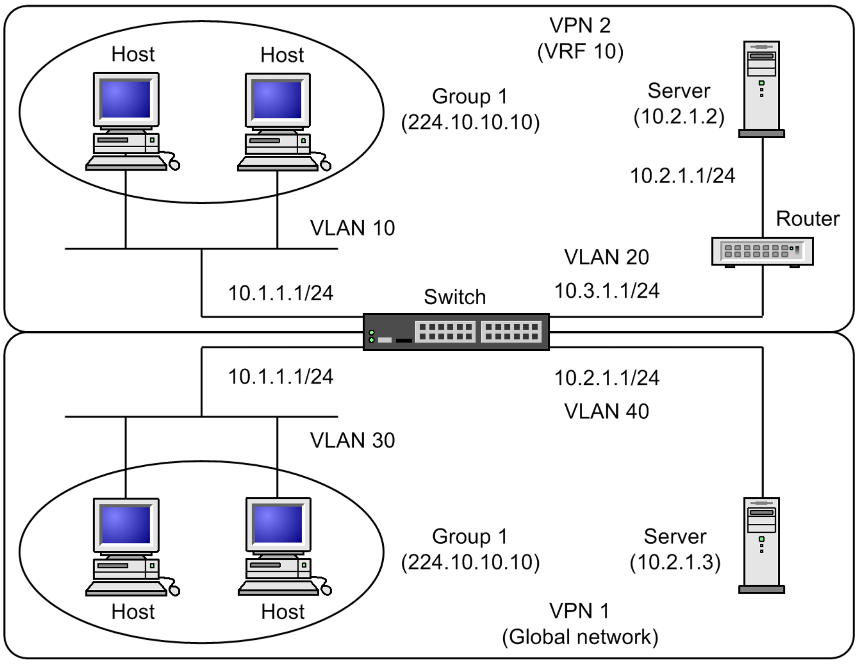
15.1.17 Configuring an IPv4 multicast extranet [OS-L3SA]

Points to note

In an IPv4 multicast extranet, a unicast extranet to the sender must be configured for the forwarding destination VRF and a unicast route must exist.

Configure multicast route filtering for the VRF that contains the sender. If no conditions are specified for the route filtering, all multicast addresses can be forwarded to all VRFs on which multicasts run. Specify all the multicast route filtering settings in global configuration mode. The following figure shows a PIM-SSM configuration example in which VPN 2 is associated with VRF 10, and 10.1.1.1/24 and 10.3.1.1/24 are used as the interface IP addresses for VRF 10. In this case, a unicast route to the server (10.2.1.2) in VPN 2 (VRF 10) must exist in VPN 1 (global network).

Figure 15-6: PIM-SSM configuration example for a VRF (IPv4 multicast extranet)



Command examples

1.
   ```
   (config)# route-map MLTEXNET permit 10
   (config-route-map)# exit
   ```

Creates a `route-map` that permits all multicast forwarding requests.

2.
```
(config)# vrf definition 10
(config-vrf)# import multicast inter-vrf MLTEXNET
(config-vrf)# exit
```

Applies the setting that permits multicast forwarding requests from all VRFs to VRF 10.

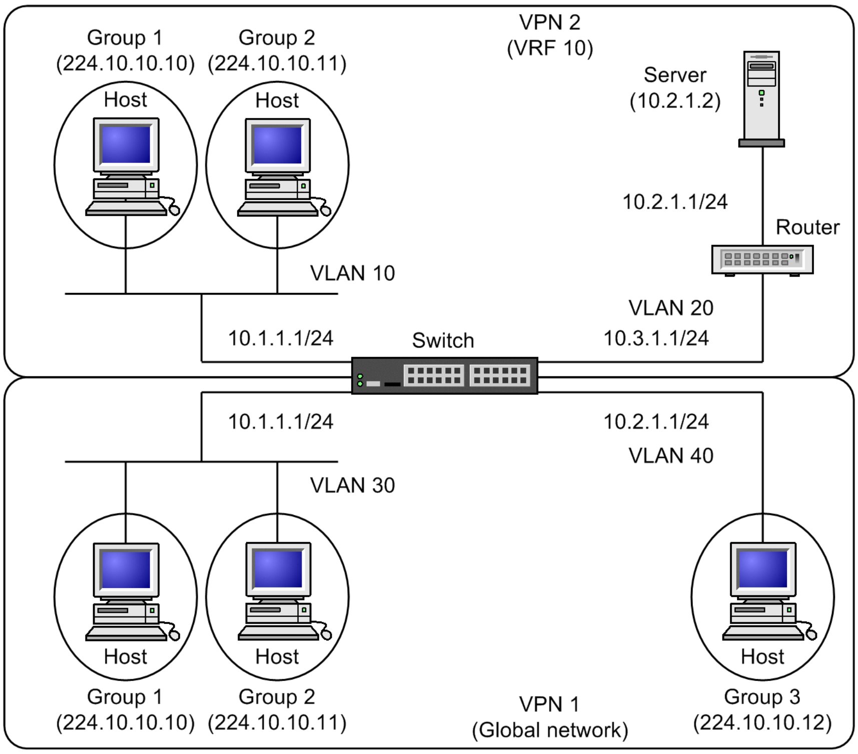
15.1.18 Configuring a PIM-SM VRF gateway [OS-L3SA]

Points to note

In an IPv4 multicast extranet, a unicast extranet to the sender must be configured for the forwarding destination VRF and a unicast route must exist.

To use multicast extranets with PIM-SM to perform communication between multicast VRFs, a PIM-SM VRF gateway must be configured. The PIM-SM VRF gateway must be set for the VRF that contains the multicast sender. Specify the settings in global configuration mode. The PIM-SM VRF gateway is configured by specifying all group addresses used for the extranet in multicast route filtering as host addresses. At this time, the group addresses specified in a range by using a wildcard mask are not subject to control by the PIM-SM VRF gateway. The figure below shows a configuration example in which packets of group addresses 224.10.10.10, 224.10.10.11, and 224.10.10.12 are forwarded from VRF 10 to the global network. In this case, a unicast route to the server (10.2.1.2) in VPN 2 (VRF 10) must exist in VPN 1 (global network).

Figure 15-7: PIM-SM configuration example for a VRF (PIM-SM VRF gateway)



Command examples

1.
```
(config)# ip access-list standard MLTGROUP
(config-std-nacl)# permit host 224.10.10.10
```

```
(config-std-nacl)# permit host 224.10.10.11
(config-std-nacl)# permit host 224.10.10.12
(config-std-nacl)# exit
(config)# route-map MLTEXNET permit 10
(config-route-map)# match ip address MLTGROUP
(config-route-map)# exit
```

Specifies 224.10.10.10, 224.10.10.11, and 224.10.10.12 as the group addresses used for the PIM-SM VRF gateway.

2.
```
(config)# vrf definition 10
(config-vrf)# import multicast inter-vrf MLTEXNET
(config-vrf)# exit
```

Specifies the groups that will be forwarded from VRF 10 to other VRFs.

3.
```
(config)# ip pim vrf 10 vrf-gateway
```

Configures the PIM-SM VRF gateway for VRF 10.

15.2 Operation

15.2.1 List of operation commands

The following table describes the operation commands for IPv4 multicasting.

Table 15-2: List of operation commands

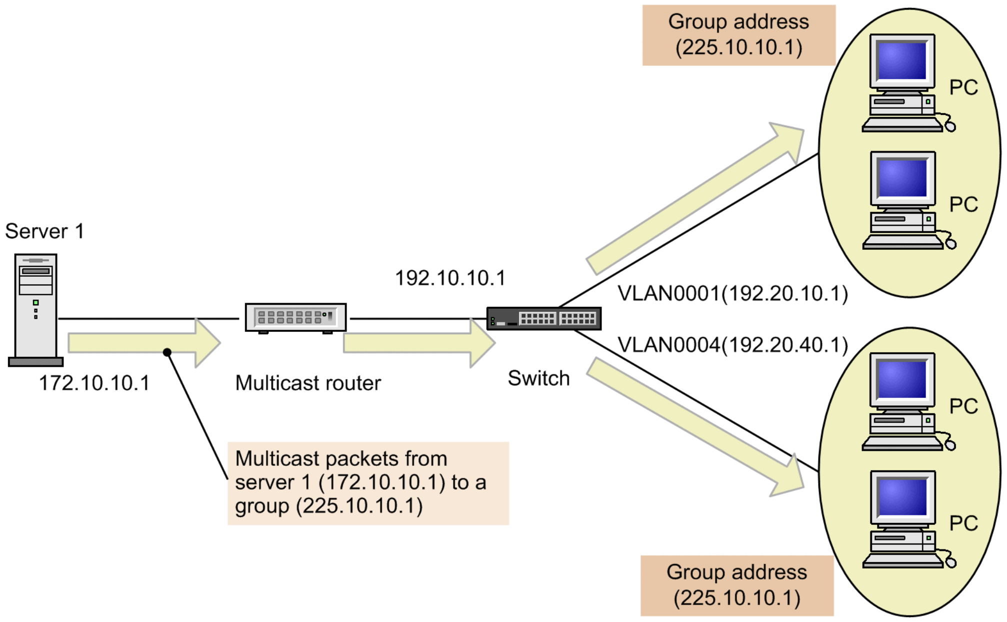
Command name	Description
show ip mcache	Shows a list of all multicast paths.
show ip mroute	Shows PIM-SM/SSM multicast routing information.
show ip pim interface	Shows the status of the PIM-SM/SSM interface.
show ip pim neighbor	Shows neighbor information for the PIM-SM/SSM interface.
show ip pim mcache	Shows multicast forwarding entries for PIM-SM/SSM.
show ip pim bsr	Shows the PIM-SM BSR information.
show ip pim rp-mapping	Shows the PIM-SM rendezvous point information.
show ip pim rp-hash	Shows the rendezvous point information for each PIM-SM group.
show ip igmp interface	Shows the status of the IGMP interface.
show ip igmp group	Shows IGMP group information.
show ip rpf	Shows PIM RPF information.
show ip multicast statistics	Shows IPv4 multicast statistics.
clear ip multicast statistics	Clears IPv4 multicast statistics.
show ip multicast resources	Shows the number of entries used in IPv4 multicast routing.
restart ipv4-multicast	Restarts the IPv4 multicast routing program (mrp).
dump protocols ipv4-multicast	Collects a dump of event trace information and control table information.
erase protocol-dump ipv4-multicast	Deletes event trace information, control table information, and core file dumps.

15.2.2 Checking routes to IPv4 multicast group addresses

To use IPv4 multicast with the Switch, execute the `show ip mcache` command to check that a route to the destination address exists. If a route does not exist and `outgoing` is invalid, see *15.2.3 Checking IPv4 PIM-SM information* and *15.2.4 Checking IGMP information*.

Figure 15-8: Results of executing the show ip mcache command

```
> show ip mcache
Date 20XX/04/01 15:20:00 UTC
Total: 1 route
- Forwarding entry ----------------------------------------------------------
Group Address  Source Address  Flags  Uptime  Expires
225.10.10.1    172.10.10.1             01:00   02:00
     incoming:
         VLAN0012(192.10.10.1)
     outgoing:
         VLAN0001(192.20.10.1)
         VLAN0004(192.20.40.1)
>
```



15.2.3 Checking IPv4 PIM-SM information

The following are the types of information you can check in the IPv4 multicast routing information for the Switch when the PIM-SM functionality is set.

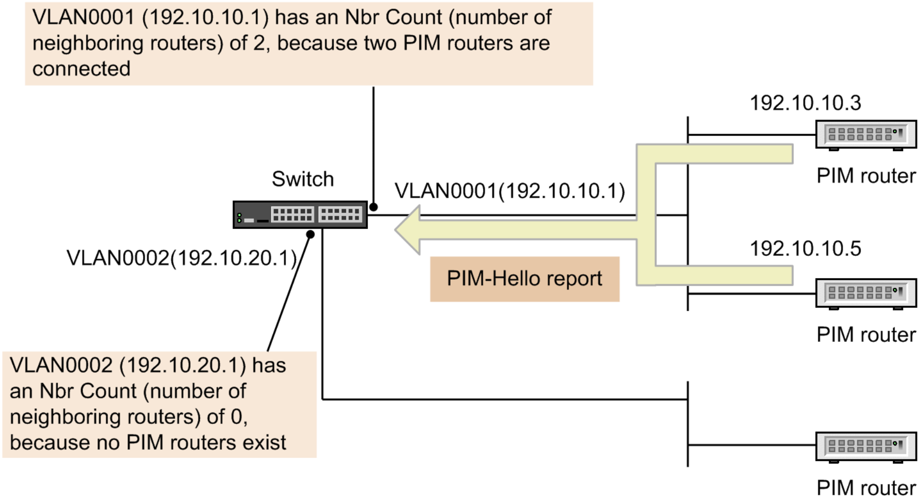
(1) Interface information

Execute the `show ip pim interface` command and check the following.

- Check the interface within Address. If none exists, PIM-SM will not run on the interface. In the configuration, make sure that PIM is enabled on the interface. Also, make sure that a failure has not occurred for the interface.
- Check the `Nbr Count` value (number of neighboring PIM routers) for the target interface. If this value is 0, neighboring routers either do not exist, or are not reporting PIM Hello messages. Check the neighboring routers.

Figure 15-9: Results of executing the show ip pim interface command

```
> show ip pim interface
Date 20XX/08/01 15:20:00 UTC
Address         Interface       Component Vif Nbr   Hello DR
                                              Count Intvl Address
192.10.10.1     VLAN0001        PIM-SM      1     2    30 192.10.10.5
192.10.20.1     VLAN0002        PIM-SM      2     0    30 This system
>
```


VLAN0001 (192.10.10.1) has an Nbr Count (number of neighboring routers) of 2, because two PIM routers are connected
Switch
VLAN0001(192.10.10.1)
VLAN0002(192.10.20.1)
192.10.10.3
PIM router
192.10.10.5
PIM-Hello report
PIM router
VLAN0002 (192.10.20.1) has an Nbr Count (number of neighboring routers) of 0, because no PIM routers exist
PIM router

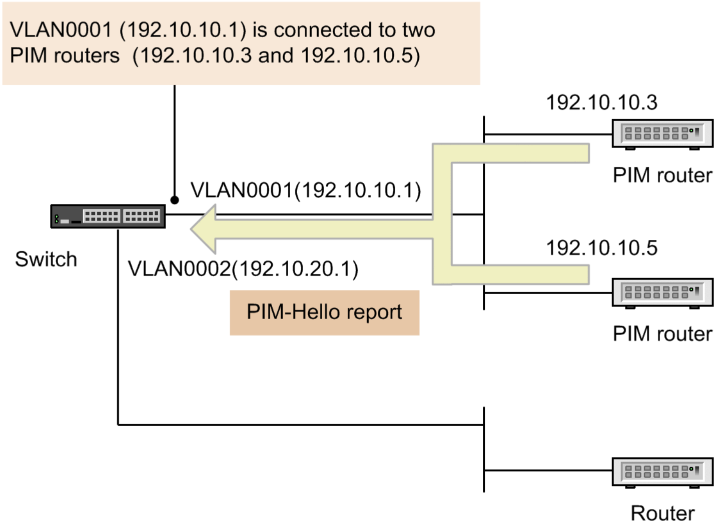
(2) Neighbor information

Execute the `show ip pim neighbor` command to check the presence of neighbors at the IP address for `Neighbor Address` of the corresponding interface. If a particular neighbor does not exist, the neighboring routers might not be reporting PIM Hello messages. Check the neighboring routers.

Figure 15-10: Results of executing the show ip pim neighbor command

```
> show ip pim neighbor
Date 20XX/08/01 15:20:00 UTC
Address          Interface        Neighbor Address Uptime   Expires
192.10.10.1      VLAN0001         192.10.10.3      00:05    01:40
                                  192.10.10.5      00:10    01:35

>
```

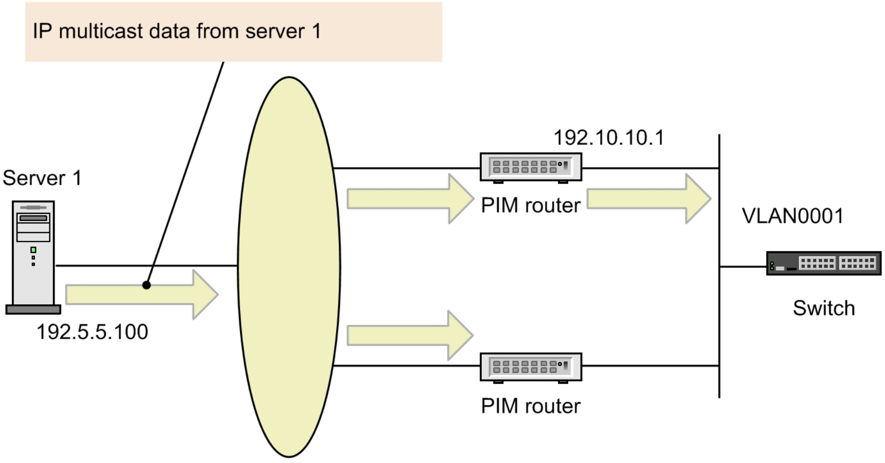


(3) Sender routing information

Execute the `show ip rpf` command to check sender routing information.

Figure 15-11: Results of executing the show ip rpf command

```
> show ip rpf 192.5.5.100
Date 20XX/08/01 15:20:00 UTC
Incoming: VLAN0001(192.5.5.200) Upstream: 192.10.10.1
```



(4) PIM-SM BSR information

Execute the `show ip pim bsr` command and check whether the BSR address is displayed. If `----` is displayed, the BSR might not be sending bootstrap messages, or might not exist. Check the BSR. Note that BSRs cannot be used for PIM-SSM.

Figure 15-12: Results of executing the show ip pim bsr command

```
> show ip pim bsr
Date 20XX/08/01 15:20:00 UTC
Status : Not Candidate Bootstrap Router
BSR Address : 192.10.10.10
    Priority: 100    Hash mask length: 30
    Uptime   : 03:00
    Bootstrap Timeout : 130 seconds
>
```

(5) PIM-SM rendezvous point information

Execute the `show ip pim rp-mapping` command to make sure that the C-RP address is displayed for the corresponding IPv4 multicast group address. If it is not displayed, the BSR might not be sending bootstrap messages, or the rendezvous point or BSR might not exist. Check the rendezvous point and BSR. Note that rendezvous points cannot be used for PIM-SSM.

Figure 15-13: Results of executing the show ip pim rp-mapping command

```
> show ip pim rp-mapping
Date 20XX/04/20 15:20:00 UTC
Status : Not Candidate Rendezvous Point
Total: 2 routes, 2 groups, 1 RP
Group/Masklen       C-RP Address Priority Uptime   Expires
224.100.100.0/24    192.1.1.1         100 02:00    02:30
224.100.200.0/24    192.1.1.1         100 02:00    02:30
>
```

(6) PIM-SM routing information

Execute the `show ip mroute` command to check whether a path to the corresponding destination address exists. If no `(S,G)` entry exists, check whether a `(*,G)` entry exists. If no `(*,G)` entry exists, and both incoming and outgoing are invalid, check the neighboring router. Note that `(*,G)` cannot be used for PIM-SSM, since it does not exist.

Figure 15-14: Displaying PIM-SM multicast routing information

```
> show ip mroute
Date 20XX/04/20 15:20:00 UTC
Total: 5 routes, 4 groups, 2 sources

(S,G) 3 routes  -----------------------------------------------------------
```

```
Group Address    Source Address   Protocol Flags  Uptime   Expires  Assert
224.100.100.10   192.1.1.1        SM       F      02:00    02:30    01:00
    incoming: VLAN0001(192.1.1.3)        upstream: Direct, reg-sup: 30s
    outgoing: VLAN0002(192.1.2.3)        uptime 02:30, expires 00:40

224.100.100.20   192.1.1.1        SM       F      02:00    02:30    01:00
    incoming: VLAN0001(192.1.1.3)        upstream: Direct
    outgoing: register <Register to 192.1.5.1>

224.100.100.30   192.1.4.1        SM       F      02:00    02:30    01:00
    incoming: VLAN0001(192.1.1.3)        upstream: 192.1.1.5
    outgoing: VLAN0002(192.1.2.3)        uptime 02:30, expires 00:40

(*,G) 2 routes  -----------------------------------------------------------
Group Address    RP Address       Protocol Flags  Uptime   Expires  Assert
225.100.100.10   192.1.5.1        SM       R      02:00    02:30    01:00
    incoming: register                   upstream: This System
    outgoing: VLAN0002(192.1.2.3)        uptime 02:30, expires 00:40

225.100.100.10   192.1.5.1        SM       R      02:00    02:30    01:00
    incoming: VLAN0001(192.1.1.3)        upstream: 192.1.1.2
    outgoing: VLAN0003(192.1.3.3)        uptime 02:30, expires 00:40
>
```

15.2.4 Checking IGMP information

The following are the types of information you can check in the IPv4 multicast routing information for the Switch when the IGMP functionality is set.

(1) Interface information

Execute the `show ip igmp interface` command and check the following.

- Check the interface within Address. If none exists, IGMP will not run on the interface. In the configuration, make sure that IGMP setting and IPv4 PIM-SM setting are enabled on the corresponding interface. Also, make sure that a failure has not occurred for the interface.
- Check the `Group Count` value (number of groups hosts have joined) for the target interface. If the value is 0, either groups hosts have joined might not exist, or the group member hosts might not advertise IGMP-Report. Check the hosts.
- Check whether the version displayed in the `Version` field allows connection with the host used by the interface.
- If a code is displayed in the Notice field, IGMP packets have been discarded. Use the code to determine why the packets were discarded.

Figure 15-15: Results of executing the show ip igmp interface command

```
> show ip igmp interface
Date 20XX/04/10 15:10:00 UTC
Total: 5 Interfaces
Address     Interface  Version  Flags  Querier     Expires  Group Count  Notice
192.10.1.2  VLAN0001      2     S      192.10.1.2     -               2
192.20.2.2  VLAN0002      2     S      192.20.2.1   02:30             0
192.30.3.2  VLAN0003      3            192.30.3.1   00:50             2
202.30.3.2  VLAN0004     (3)           202.30.3.2     -               0   Q
210.40.4.2  VLAN0005      3            210.40.4.1   03:15             3  L
```

(2) Group information

Execute the `show ip igmp group` command and check the groups displayed in the `Group Address` field. If there are none, check for the following:

- The group members (hosts) might not advertise IGMP-Report. Check the hosts.
- Check the version of the IGMP interfaces for the Switch and for each host, and make sure that

the Switch can connect to the hosts.

- If the hosts ignore IGMPv3-Query, IGMPv3 cannot be used. Set the IGMP version of the corresponding interface to 2.

Figure 15-16: Results of executing the show ip igmp group command

```
> show ip igmp group brief
Date 20XX/08/01 15:10:00 UTC
Total: 7 groups
Group Address   Interface       Version  Mode      Source Count
224.1.1.1       VLAN0001           2     EXCLUDE              0
232.1.1.2       VLAN0001           2     EXCLUDE              2
234.1.1.1       VLAN0003           2     EXCLUDE              1
234.1.1.2       VLAN0003           3     INCLUDE              1
232.1.1.1       VLAN0004           3     INCLUDE              1
232.1.1.3       VLAN0004           3     INCLUDE              2
235.1.1.1       VLAN0004           3     EXCLUDE              3
```

Chapter

16. IPv4 Multicast Route Filtering [OS-L3SA]

This chapter provides an overview of IPv4 multicast route filtering and explains how to use it.

16.1 Description of IPv4 multicast route filtering

16.1.1 Overview of IPv4 multicast route filtering

IPv4 multicast route filtering controls IPv4 multicast routes by filtering them. This functionality is used only for IPv4 multicast extranets.

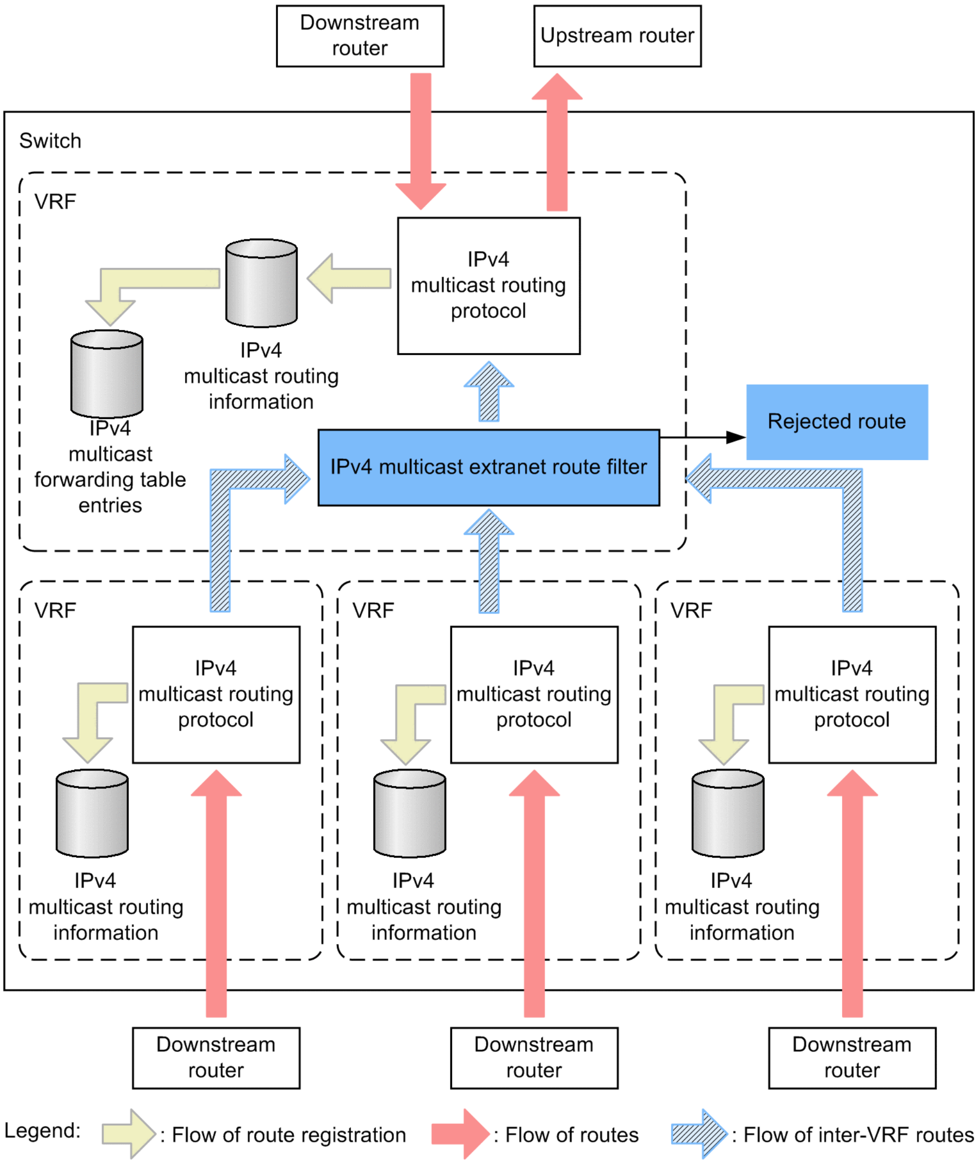
(1) Filtering routes in IPv4 multicast extranets

To enable an IPv4 multicast extranet, forwarding requests need to be exchanged between different VRFs. The Switch uses a method in which routing protocols running in VRFs exchange forwarding requests. When routes are filtered in an IPv4 multicast extranet, the routing protocols running in VRFs filter the forwarding requests. This functionality enables each VRF to determine whether to receive forwarding requests for each destination IP address in a multicast packet. Note that IPv4 multicast routing protocols follow the practice of the route filters used in a unicast extranet because they reference the routing information in the unicast extranet regarding the source IP addresses.

When route filtering is not configured in an IPv4 multicast extranet, the Switch discards all the forwarding requests between the VRFs.

The following figure shows the concept of the IPv4 multicast extranet route filtering.

Figure 16-1: Concept of IPv4 multicast extranet route filtering


Downstream router
Upstream router
Switch
VRF
IPv4 multicast routing protocol
IPv4 multicast routing information
IPv4 multicast forwarding table entries
IPv4 multicast extranet route filter
Rejected route
VRF
IPv4 multicast routing protocol
IPv4 multicast routing information
VRF
IPv4 multicast routing protocol
IPv4 multicast routing information
VRF
IPv4 multicast routing protocol
IPv4 multicast routing information
Downstream router
Downstream router
Downstream router
Legend:
: Flow of route registration
: Flow of routes
: Flow of inter-VRF routes

16.1.2 IPv4 multicast route filtering

For details about IPv4 multicast route filtering, see *13.1.2 Filtering methods*.

The following table describes how IPv4 multicast route filtering behaves when a configuration command is executed.

Table 16-1: Behavior of IPv4 multicast route filtering when a configuration command is executed

Configuration commands	Description
ip prefix-list	This command is not supported, and is ignored if specified.
ip access-list standard	Only `permit` is used. The IP addresses specified with `deny` are ignored.
ip access-listt extended	This command is not supported, and is ignored if specified.
route-map	Only `permit` is used. `route-map` specified with `deny` is ignored.

Configuration commands	Description
ip as-path access-list	This command is not supported, and is ignored if specified.
ip community-list standard	This command is not supported, and is ignored if specified.
ip community-list extended	This command is not supported, and is ignored if specified.

16.1.3 IPv4 multicast extranets

(1) Inter-VRF route filtering

Routes between VRFs can be filtered. If a Switch decides not to include a route in the routing table as a result of filtering, no IPv4 multicast routing information is generated for the route.

(a) Method of applying filters

The filter is set at the upstream VFR, and routes notified from the forwarding destination VRF are filtered using the `import multicast inter-vrf` configuration command to select permitted group addresses. The routes permitted by this filtering are added to the IPv4 multicast routing information. The routes to which no filters are applicable are not included.

The table below describes the configuration command used in the route filtering between IPv4 multicast VRFs.

Table 16-2: Configuration command used in the route filtering between IPv4 multicast VRFs

Command name	Filtered routes
import multicast inter-vrf	Forwarding requests from a VRF specified as `route-map` are filtered.

The following table describes `route-map` filter conditions in an IPv4 multicast extranet. Other conditions are ignored.

Table 16-3: Filter conditions for route-map in an IPv4 multicast extranet

Route attribute used as conditions	Description	Configuration commands
Destination IPv4 multicast group address	Specifies the identifier of an access list as a condition and uses the specified filter to filter the IPv4 multicast group address of the destination. A match is assumed if the filter action is permit. If this condition is not specified, all IPv4 multicast group addresses can be permitted.	match ip address ip access-list standard
VRF ID	Specifies a VRF ID as a condition and compares it with the VRF ID in a route. The forwarding requests from the specified VRF is permitted when the IDs match. If the same VRF ID is specified as a VRF already specified with this command, that ID is ignored. This enables grouping of multiple VRFs and use of the same `route-map` for the grouped VRFs. If this condition is not specified, the forwarding requests from all VRFs are permitted.	match vrf

(b) Setting inter-VRF routes

Specify an inter-VRF route filter. The routes which received the forwarding request from another VRF or the global network are included in the local VRF IPv4 multicast routing information according to the filter conditions. The included route is added as a forwarding-destination interface in the IPv4 multicast routing information. If you execute the `match vrf` configuration command for an IPv4 multicast inter-VRF route filter, the filter compares the VRF ID of the route source VRF against the VRF ID specified by the command. If you do not specify the `match vrf` command, the same filter conditions are applied to all other VRFs and global networks.

(c) Advertising inter-VRF routes by using protocols

If you apply a route filter to a VRF, the filter determines whether to permit forwarding requests from other VRFs or the global network. When a VRF receives a forwarding request from another VRF or the global network and finds a match after it runs the route through a filter, the VRF creates IPv4 multicast routing information and sends the route to the upstream router (if there is one).

For a VRF or the global network to be able to send a forwarding request to the local VRF, in the unicast extranet, configure the local VRF so that it contains the source IP address in the packets sent to the VRFs and global networks from which routes are requested.

16.2 Configuration

16.2.1 List of configuration commands

The following table describes the configuration commands for IPv4 multicast route filtering.

Table 16-4: List of configuration commands

Command name	Description
access-list#1	Configures an access list used as an IPv4 filter.
ip access-list standard#1	Configures an access list to serve as an IPv4 address filter.
match ip address#2	Configures `route-map` to use the IPv4 address prefix as filter conditions.
match vrf#2	Configures `route-map` to use the VRF value as a filter condition.
import multicast inter-vrf#3	Uses a filter to control IPv4 multicast forwarding requests from other VRFs or global networks.

#1

See *19. Access Lists* in the manual *Configuration Command Reference Vol. 1 For Version 11.10*.

#2

See *14. Route Filters (IPv4 and IPv6)* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

#3

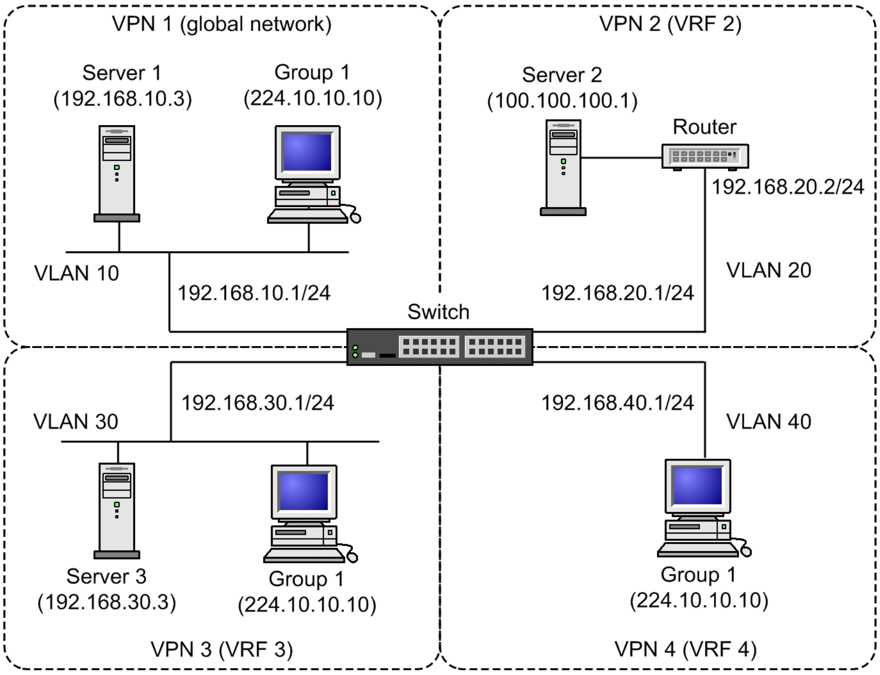
See *30. VRF [OS-L3SA]* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

16.2.2 IPv4 multicast extranets

Configure an IPv4 multicast extranet as described in the figure below.

You can use IPv4 multicast extranet route filtering to apply restrictions.

Figure 16-2: IPv4 multicast extranet configuration example



(1) Permitting requests from all VRFs

Configure VRF 2 so that VRF 2 permits the IPv4 multicast forwarding requests from all VRFs and the global network.

Configure a unicast extranet in advance, and then configure VRF 2 so that it contains the source IP address in the IPv4 multicast packets sent to forwarding destination VRFs and the global network.

Points to note

If filter conditions are not specified in `route-map`, all conditions are permitted.

Command examples

1.
```
(config)# route-map MLTEXNET permit 10
(config-route-map)# exit
```
Permits all filter conditions.

2.
```
(config)# vrf definition 2
(config-vrf)# import multicast inter-vrf MLTEXNET
(config-vrf)# exit
```
Applies the filter created in step 1 to VRF 2 in the IPv4 multicast extranet, and permits the IPv4 multicast forwarding requests from all VRFs and the global network.

(2) Permitting forwarding requests from specific VRFs

Configure VRF 2 so that VRF 2 permits IPv4 multicast forwarding requests from VRF 3 and VRF 4.

Configure a unicast extranet in advance, and configure VRF 2 so that it contains the source IP address in the IPv4 multicast packets sent to VRF 3 and VRF 4.

Points to note

If this setting is not specified, VRF 2 receives the IPv4 multicast forwarding requests from all VRFs.

Command examples

1.
```
(config)# route-map MLTEXNET permit 10
(config-route-map)# match vrf 3 4
(config-route-map)# exit
```

Permits the IPv4 multicast forwarding requests from only VRF 3 and VRF 4.

2.
```
(config)# vrf definition 2
(config-vrf)# import multicast inter-vrf MLTEXNET
(config-vrf)# exit
```

Applies the filter created in step 1 to VRF 2 in the IPv4 multicast extranet, and permits IPv4 multicast forwarding requests from VRF 3 and VRF 4.

(3) Permitting forwarding requests from specific group addresses

Set the following configuration to permit IPv4 multicast forwarding requests from only the group addresses in the 224.10.0.0/16 range.

Points to note

When you set a range of group addresses for use in an extranet, other group addresses are assigned for communication only within VRFs. The locally used group addresses can be used for different purposes in each VRF.

If you do not set this configuration, all group addresses (224.0.0.0/4) are used in an extranet.

Command examples

1.
```
(config)# ip access-list standard MLTGROUP
(config-std-nacl)# permit 224.10.0.0 0.0.255.255
(config-std-nacl)# exit
(config)# route-map MLTEXNET permit 10
(config-route-map)# match ip address MLTGROUP
(config-route-map)# exit
```

Sets 224.10.0.0/16 as the range of group addresses to be used in the extranet.

2.
```
(config)# vrf definition 2
(config-vrf)# import multicast inter-vrf MLTEXNET
(config-vrf)# exit
```

Applies the filter created in step 1 to VRF 2 in the IPv4 multicast extranet, and allows VRF 2 to accept forwarding requests only from the group addresses in the 224.10.0.0/16 range.

(4) Configuring a bidirectional IPv4 multicast extranet

This configuration allows mutual communication among the global network, VRF 2, VRF 3, and

VRF 4 in the IPv4 multicast extranet.

Configure a unicast extranet in advance, and then configure the VRF or global network you want to connect so that it contains the source IP address in IPv4 multicast packets sent to the global network, VRF 2, VRF 3, and VRF 4.

Points to note

The VRFs set in the `match vrf` command executed for `route-map` can share the same filter.

Command examples

1.

```
(config)# ip access-list standard MLTGROUP
(config-std-nacl)# permit 224.10.10.0 0.0.0.255
(config-std-nacl)# exit
(config)# route-map MLTEXNET permit 10
(config-route-map)# match vrf global 2 3 4
(config-route-map)# match ip address MLTGROUP
(config-route-map)# exit
```

Permits IPv4 multicast forwarding requests from the global network, VRF 2, VRF 3, and VRF 4 to group address 224.10.10.0/24.

2.

```
(config)# vrf definition global
(config-vrf)# import multicast inter-vrf MLTEXNET
(config-vrf)# exit
(config)# vrf definition 2
(config-vrf)# import multicast inter-vrf MLTEXNET
(config-vrf)# exit
(config)# vrf definition 3
(config-vrf)# import multicast inter-vrf MLTEXNET
(config-vrf)# exit
(config)# vrf definition 4
(config-vrf)# import multicast inter-vrf MLTEXNET
(config-vrf)# exit
```

Applies the filter created in step 1 to the global network, VRF 2, VRF 3 and VRF 4 in the IPv4 multicast extranet, and permits IPv4 multicast forwarding requests among them.

16.3 Operation

16.3.1 List of operation commands

The following table describes the operation commands for IPv4 multicast route filtering.

Table 16-5: List of operation commands

Command name	Description
show ip mcache#	Lists the IPv4 multicast forwarding entries.
show ip mroute#	Lists the IPv4 multicast routing information.
show ip multicast resources#	Shows the number of entries used in IPv4 multicast routing.

#

See *7. IPv4 Multicast Routing Protocols* in the manual *Operation Command Reference Vol. 2 For Version 11.10*.

16.3.2 Checking an IPv4 multicast extranet

Use the `show ip mroute` and `show ip mcache` commands to reference the inter-VRF forwarding entries used in an IPv4 multicast extranet. For the entries that issue forwarding requests to different VRFs, the VRF IDs of the forwarding destinations are displayed in the `incoming` section. For the entries that permit forwarding requests from different VRFs, the VRF IDs are displayed in the `outgoing` section.

Figure 16-3: Results of executing the show ip mroute command

```
> show ip mroute vrf all group 224.10.10.10 source 100.100.100.1
Date 20XX/12/10 02:34:22 UTC
Total: 4 routes
VRF: global  Total: 1 route , 1 group , 1 source

(S,G) 1 routes  -----------------------------------------------------
Group Address  Source Address  Protocol Flags  Uptime  Expires  Assert
224.10.10.10   100.100.100.1   SM               00:03   03:27    00:00
    incoming: VRF 2     upstream: Extra  reg-sup: 0s
    outgoing: VLAN0010(192.168.10.1)     uptime 00:03  expires 03:27
              VLAN0011(192.168.11.1)     uptime 00:03  expires 03:27

VRF: 2  Total: 1 routes, 1 groups, 1 source

(S,G) 1 routes  -----------------------------------------------------
Group Address  Source Address  Protocol Flags  Uptime  Expires  Assert
224.10.10.10   100.100.100.1   SM       L       00:03   03:27    00:00
    incoming: VLAN0020(192.168.20.1)     upstream: 192.168.20.2
    outgoing: VLAN0021(192.168.21.1)     uptime 00:03  expires 03:27
              global                     uptime --:--
              VRF 3                      uptime --:--
              VRF 4                      uptime --:--

VRF: 3  Total: 1 routes, 1 groups, 1 source

(S,G) 1 routes  -----------------------------------------------------
Group Address  Source Address  Protocol Flags  Uptime  Expires  Assert
224.10.10.10   100.100.100.1   SM       L       00:03   03:27    00:00
    incoming: VRF 2     upstream: Extra  reg-sup: 0s
    outgoing: VLAN0030(192.168.30.1)     uptime 00:03  expires --:--

VRF: 4  Total: 1 routes, 1 groups, 1 source

(S,G) 1 routes  -----------------------------------------------------
Group Address  Source Address  Protocol Flags  Uptime  Expires  Assert
224.10.10.10   100.100.100.1   SM               00:03   03:27    00:00
    incoming: VRF 2     upstream: Extra  reg-sup: 0s
    outgoing: VLAN0040(192.168.40.1)     uptime 00:03  expires 03:27
```

Figure 16-4: Results of executing the show ip mcache command

```
> show ip mcache vrf all source 100.100.100.1 group 224.10.10.10
Date 20XX/12/10 02:34:43 UTC
Total: 4 routes
VRF: global  Total: 1 route
- Forwarding entry --------------------------------------------------
Group Address   Source Address  Flags  Uptime  Expires
224.10.10.10    100.100.100.1   D       00:19   03:27
     incoming:
         VRF 2
     outgoing:
         VLAN0010(192.168.10.1)
         VLAN0011(192.168.11.1)

VRF: 2  Total: 1 routes
- Forwarding entry --------------------------------------------------
Group Address   Source Address  Flags  Uptime  Expires
224.10.10.10    100.100.100.1   U       00:19   03:27
     incoming:
         VLAN0020(192.168.20.1)
     outgoing:
         VLAN0021(192.168.21.1)
         VLAN0010(192.168.10.1)     global
         VLAN0011(192.168.11.1)     global
         VLAN0030(192.168.30.1)     VRF 3
         VLAN0040(192.168.40.1)     VRF 4

VRF: 3  Total: 1 routes
- Forwarding entry ---------------------------------------------------
Group Address   Source Address  Flags  Uptime  Expires
224.10.10.10    100.100.100.1   D       00:19   03:27
     incoming:
         VRF 2
     outgoing:
         VLAN0030(192.168.30.1)

VRF: 4  Total: 1 routes
- Forwarding entry ---------------------------------------------------
Group Address   Source Address  Flags  Uptime  Expires
232.0.0.1       10.2.0.100      D       00:18   03:27
     incoming:
         VRF 2
     outgoing:
         VLAN0040(192.168.40.1)
```

Chapter
17. Description of IPv6, NDP, and ICMPv6

In IPv6 networks, the Switch performs various functions, including IP packet forwarding based on communication protocols, filtering, and load balancing. This chapter describes IPv6 packet forwarding.

17.1 Addressing

IPv6 includes the following features that are not available in IPv4:

- Extended address space

  In IPv6, the address space is extended to 128 bits from the 32-bit address space available in IPv4. Because the number of addresses that can be assigned to nodes is almost limitless in IPv6, the problem of address exhaustion in IPv4 is removed. In addition, longer addresses make possible a hierarchical allocation of addresses so that addresses can be assigned to more nodes.

- Simpler header format

  The IPv6 header fields are simpler than IPv4 header fields, reducing protocol processing overhead.

- Strengthening of extended and optional headers

  Transfer efficiency has improved, restrictions on the option length are looser, and options are easy to add.

- Flow labels

  Flow labels allow a specific flow of traffic to be identified.

The following subsections provide an overview of the IPv6 addressing used in the Switch.

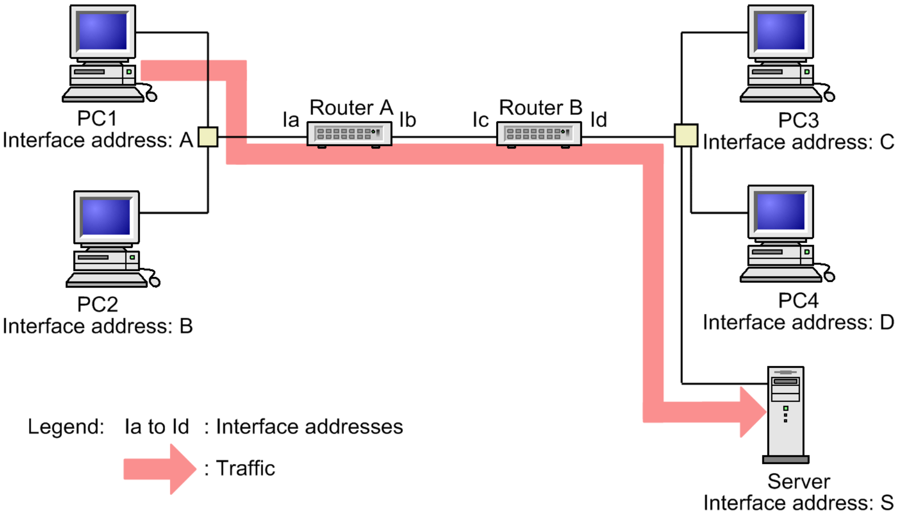
17.1.1 IPv6 addresses

IPv6 addresses can have one of three formats: unicast, anycast, and multicast.

(1) Unicast addresses

A unicast address identifies a single interface. Packets addressed to a unicast address used as the endpoint address are delivered to the interface indicated by the address. The following figure shows how communication using a unicast address works.

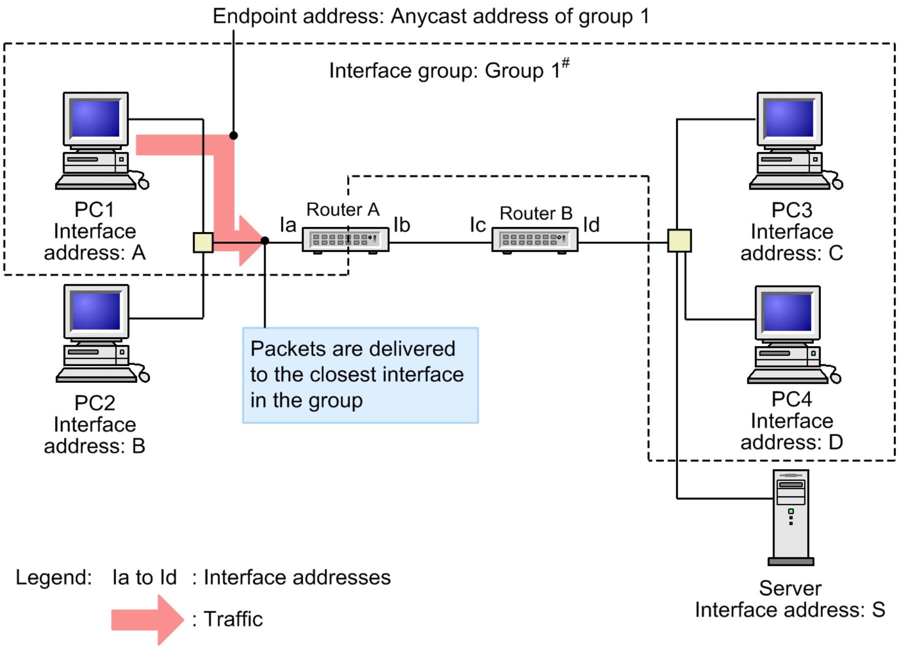
Figure 17-1: Communication using a unicast address



(2) Anycast addresses

An anycast address identifies a group of interfaces. Packets addressed to an anycast address used as the endpoint address are delivered to the closest interface in the group based on the distance calculated by the routing protocol. Note that the Switches does not support anycast addresses. The following figure shows how communication using an anycast address works.

Figure 17-2: Communication using an anycast address

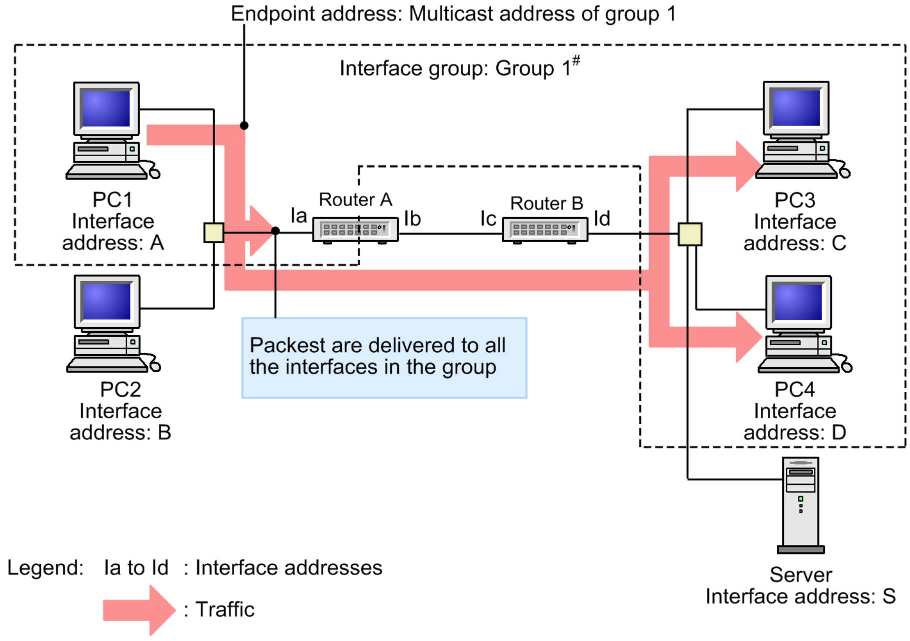

Endpoint address: Anycast address of group 1
Interface group: Group 1#
PC1
Interface address: A
Ia
Router A
Ib
Ic
Router B
Id
PC3
Interface address: C
PC2
Interface address: B
Packets are delivered to the closest interface in the group
PC4
Interface address: D
Server
Interface address: S
Legend: Ia to Id : Interface addresses
➡ : Traffic

#: Group 1 contains interface addresses A, C, D, and Ia.

(3) Multicast addresses

A multicast address also identifies a group of interfaces. Packets addressed to a multicast address used as the endpoint address are delivered to all the interfaces in the group identified by the address. The following figure shows how communication using a multicast address works.

Figure 17-3: Communication using a multicast address


Endpoint address: Multicast address of group 1
Interface group: Group 1#
PC1
Interface address: A
Ia
Router A
Ib
Ic
Router B
Id
PC3
Interface address: C
PC2
Interface address: B
Packest are delivered to all the interfaces in the group
PC4
Interface address: D
Server
Interface address: S
Legend: Ia to Id : Interface addresses
➡ : Traffic

#: Group 1 contains interface addresses A, C, D, and Ia.

17.1.2 Address notation

An IPv6 address is 128 bits. The notation is as follows:

- In hexadecimal notation, an IPv6 address is written as eight 16-bit groups, each separated by a colon.

  Example: 3ffe:0501:0811:ff02:0000:08ff:fe8b:3090

- You can omit leading 0s in each of the individual blocks in hexadecimal notation.

  Example: 3ffe:501:811:ff02:0:8ff:fe8b:3090

- You can replace consecutive blocks of 0s with two colons (::). However, only one such replacement is allowed.

  Example: You can change the full IPv6 address on the left to the format on the right:

  fe80:0000:0000:0000:0000:0000:0000:3090 -> fe80::3090

  Example: Only one set of two colons is permitted.

  fe80:0000:0000:0000:0000:0000:0000:3090 -> fe80::0::3090 (incorrect)

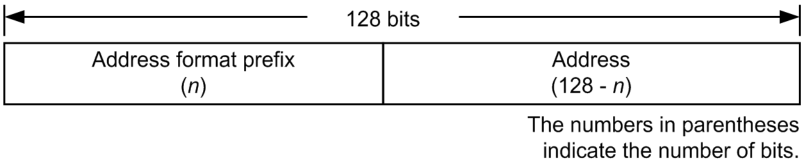
- You can use the following formats to specify an address and the prefix length:
  - IPv6 address/prefix length
  - IPv6 address prefixlen prefix-length

  The prefix length indicates in decimal the number of bits from the leftmost end of an address that are used for the prefix.

17.1.3 Address format prefix

A 128-bit IPv6 address is divided into subfields. The first bits are called the address format prefix and identify the type of the IPv6 address. The following figure shows an example of an address format prefix.

Figure 17-4: Address format prefix



The following table describes the types of address format prefixes.

Table 17-1: Address format prefix types

Prefix (binary)	Assignment
0000 0000	Reserved
0000 0001	Unassigned
0000 001	Reserved for NSAP assignment
0000 010	Reserved for IPX assignment
0000 011	Unassigned
0000 1	Unassigned
0001	Unassigned
001	Aggregation-possible global unicast addresses

Prefix (binary)	Assignment
010	Unassigned
011	Unassigned
100	Unassigned
101	Unassigned
110	Unassigned
1110	Unassigned
1111 0	Unassigned
1111 10	Unassigned
1111 110	Unassigned
1111 1110 0	Unassigned
1111 1110 10	Link-local unicast addresses
1111 1110 11	Site-local unicast addresses
1111 1111	Multicast addresses

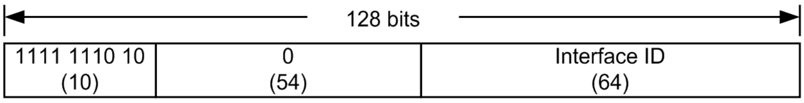
17.1.4 Unicast addresses

(1) Link-local addresses

If the upper 64 bits of the format prefix are fe80:: and the interface ID is 64 bits, the address is an IPv6 link-local address. IPv6 link-local addresses are valid only on a link and are used for address auto-configuration and neighbor discovery, and for networks without routers. When the source address or destination address of a packet is an IPv6 link-local address, the Switch does not forward the packet to other links.

Each interface that uses IPv6 on the Switch is assigned an IPv6 link-local address. You cannot assign multiple link-local addresses per interface. The following figure shows the format of an IPv6 link-local address.

Figure 17-5: IPv6 link-local address

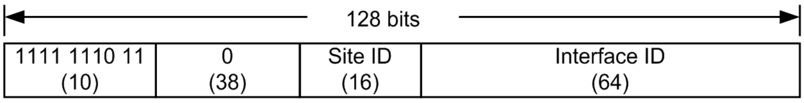


The numbers in parentheses indicate the number of bits.

(2) Site-local addresses

If the upper 10 bits of the format prefix are 1111 1110 11 and the interface ID is 64 bits, the address is an IPv6 site-local address. The use of site-local addresses is not recommended because they are made obsolete in RFC 3879. The Switch treats IPv6 site-local addresses as IPv6 global addresses discussed in *(3) Global addresses*. Therefore, when you assign an IPv6 site-local address to an interface, configure routing or filtering so that the information about the IPv6 site-local address does not leave the site. The following figure shows the format of an IPv6 site-local address.

Figure 17-6: IPv6 site-local address

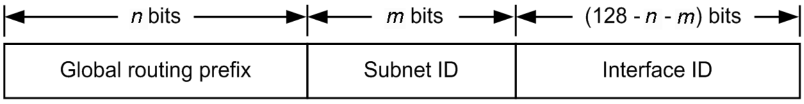


The numbers in parentheses indicate the number of bits.

(3) Global addresses

If the upper three bits of the format prefix are 001, the address is an IPv6 global address. IPv6 global addresses are unique in the world and are used for traffic on the Internet. When the source address of a packet is an IPv6 global address, the packet is forwarded based on routing information. The following figure shows the format of an IPv6 global address.

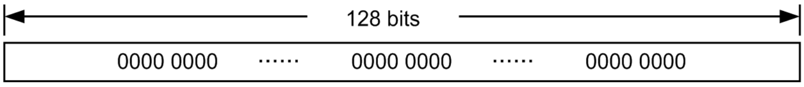
Figure 17-7: IPv6 global address



(4) Unspecified address

Address 0:0:0:0:0:0:0:0 (0::0 or ::), with all bits set to 0, is defined as the unspecified address. The unspecified address indicates that the interface has no address. This address is used when you start connecting to a node that is not assigned an address. You cannot intentionally assign the unspecified address to a node. The following figure shows the format of the unspecified address.

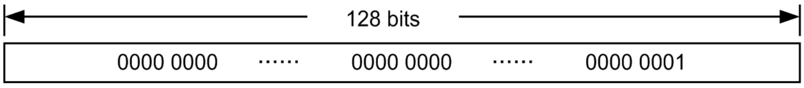
Figure 17-8: Unspecified address



(5) Loopback address

Address 0:0:0:0:0:0:0:1 (0::1 or ::1) is defined as the loopback address. The loopback address is used as the destination address of a packet when the packet is sent to the local node. You cannot assign the loopback address to an interface. When the destination address of an IPv6 packet is the loopback address, the packet is not allowed to be sent to destinations other than the local node nor allowed to be forwarded by routers. The following figure shows the format of the loopback address.

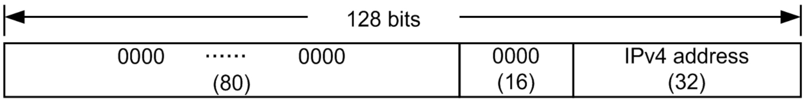
Figure 17-9: Loopback address



(6) IPv4-compatible addresses

IPv4-compatible IPv6 addresses are used to allow two IPv6 nodes to communicate on an IPv4 routed network. An IPv4-compatible address is a special unicast address whose lower 32 bits contain an IPv4 address. The prefix is 96 bits, all bits of which are set to 0. The following figure shows the format of an IPv4-compatible address.

Figure 17-10: IPv4-compatible address

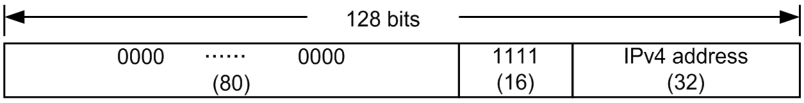


The numbers in parentheses
indicate the number of bits.

(7) IPv4-mapped addresses

IPv4-mapped IPv6 addresses are used to communicate with IPv4-only nodes that do not support IPv6. When an IPv6 host needs to send packets to a host that only supports IPv4, the IPv6 host uses an IPv4-mapped IPv6 address. The prefix is 96 bits, the 80 upper bits of which are 0s and the remaining 16 bits are 1s. The following figure shows the format of an IPv4-mapped address.

Figure 17-11: IPv4-mapped address

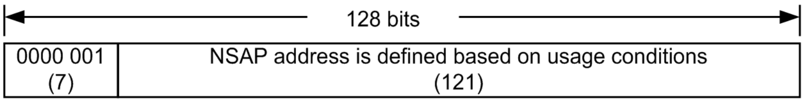


The numbers in parentheses
indicate the number of bits.

(8) NSAP-compatible addresses

This address format is used to convert an NSAP address so that it can be used in IPv6. To support NSAP, the upper 7 bits of the address format prefix are 0000 001. The following figure shows the format of an NSAP-compatible address.

Figure 17-12: NSAP-compatible address

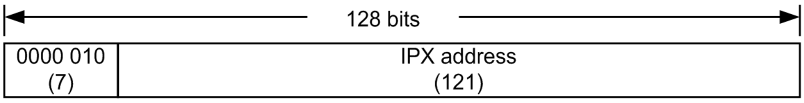


The numbers in parentheses
indicate the number of bits.

(9) IPX-compatible addresses

This address format is used to convert an IPX address so that it can be used in IPv6. To support IPX, the upper 7 bits of the address format prefix are 0000 010. The following figure shows the format of an IPX-compatible address.

Figure 17-13: IPX-compatible address



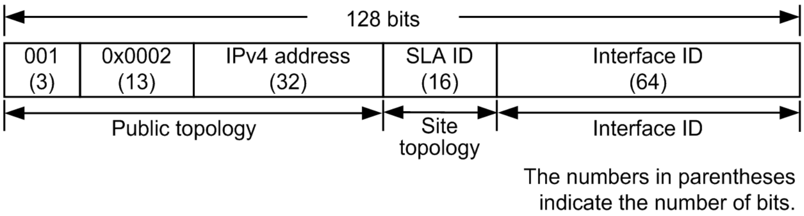
The numbers in parentheses
indicate the number of bits.

(10) 6to4 addresses

This address format is used for 6to4 tunneling. For 6to4 tunneling, the Internet Assigned Numbers Authority (IANA) assigned 0x0002 for the TLA ID, one of aggregation identifiers for IPv6 global addresses. For the NLA ID, a global or unicast IPv4 address owned by the site that uses 6to4 tunneling is defined.

The following figure shows the format of a 6to4 address.

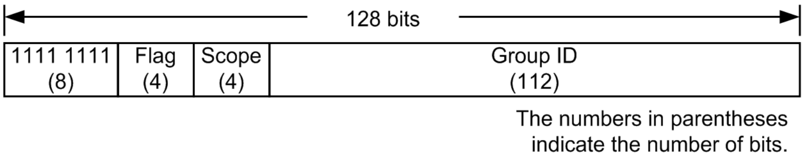
Figure 17-14: 6to4 address



17.1.5 Multicast addresses

A multicast address identifies a group of nodes. The upper 8 bits of the address format prefix are ff. A node can belong to multiple multicast groups. Multicast addresses cannot be used as the source addresses of packets. In a multicast address, the address format prefix is followed by the flag field (4 bits), scope field (4 bits), and group ID field (112 bits). The following figure shows the format of an IPv6 multicast address.

Figure 17-15: IPv6 multicast address



Each of the four bits in the flag field is a flag. The fourth bit is defined as the transient (T) flag bit, and its values have the following meaning:

1. If the T flag bit is set to 0: The address is a known multicast address permanently assigned by IANA.
2. If the T flag bit is set to 1: The address is a temporarily used (non-permanent) multicast address

The scope field is a 4-bit flag and used to limit the scope of a multicast group. The following table describes the scope field values for multicast addresses.

Table 17-2: Scope field values for multicast addresses

Value	Scope
0	Reserved
1	Node-local
2	Link-local
3	Unassigned
4	Unassigned
5	Site-local
6	Unassigned
7	Unassigned
8	Organization-local
9	Unassigned
A	Unassigned
B	Unassigned

Value	Scope
C	Unassigned
D	Unassigned
E	Global
F	Reserved

(1) Reserved multicast addresses

The following multicast addresses are reserved and cannot be assigned to any multicast group.

1. ff00:0:0:0:0:0:0:0
2. ff01:0:0:0:0:0:0:0
3. ff02:0:0:0:0:0:0:0
4. ff03:0:0:0:0:0:0:0
5. ff04:0:0:0:0:0:0:0
6. ff05:0:0:0:0:0:0:0
7. ff06:0:0:0:0:0:0:0
8. ff07:0:0:0:0:0:0:0
9. ff08:0:0:0:0:0:0:0
10. ff09:0:0:0:0:0:0:0
11. ff0a:0:0:0:0:0:0:0
12. ff0b:0:0:0:0:0:0:0
13. ff0c:0:0:0:0:0:0:0
14. ff0d:0:0:0:0:0:0:0
15. ff0e:0:0:0:0:0:0:0
16. ff0f:0:0:0:0:0:0:0

(2) All-nodes addresses

An all-nodes address identifies a group of IPv6 nodes in the specified scope. When the destination address of a packet is an all-nodes address, the packet is received by all the nodes in the specified scope. The following all-nodes addresses are available:

1. ff01:0:0:0:0:0:0:1 Node-local all-nodes address
2. ff02:0:0:0:0:0:0:1 Link-local all-nodes address

(3) All-routers addresses

An all-routers address identifies a group of IPv6 routers in the specified scope. When the destination address of a packet is an all-routers address, the packet is received by all the routers in the specified scope. The following all-routers addresses are available:

1. ff01:0:0:0:0:0:0:2 Node-local all-routers address
2. ff02:0:0:0:0:0:0:2 Link-local all-routers address
3. ff05:0:0:0:0:0:0:2 Site-local all-routers address

(4) Solicited-node addresses

A solicited-node address is created by using the unicast address or anycast address of a node. The lower 24 bits in the unicast or anycast address of the solicited node are added to the 104-bit prefix, ff02:0:0:0:0:1:ff00::/104. The range of solicited-node addresses is as follows:

ff02:0:0:0:0:1:ff00:0000 to ff02:0:0:0:0:1:ffff:ffff

If solicited-node addresses need to be created from IPv6 addresses that have a few different upper bits and those bits are different only because each provider of aggregation identifiers uses different bits in the format prefix, all such solicited-node addresses are the same. This decreases the number of multicast groups that nodes need to participate in.

17.1.6 IPv6 addresses used by the Switch

(1) Addresses that can be assigned

The following IPv6 addresses can be assigned to the interfaces of the Switch:

1. Global unicast addresses
2. Link-local unicast addresses

Although the following IPv6 addresses can be assigned, they are treated as global unicast addresses:

1. Site-local unicast addresses
2. Anycast addresses
3. Unicast addresses with address format prefix unassigned
4. NSAP-compatible addresses
5. IPX-compatible addresses

(2) Addresses that cannot be assigned

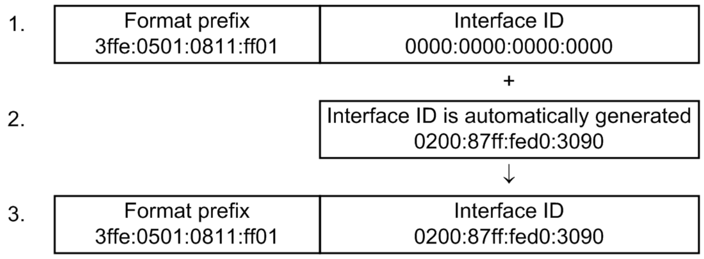
The following IPv6 addresses cannot be assigned to the interfaces of the Switch:

1. Multicast addresses
2. The unspecified address
3. The loopback address
4. IPv4-compatible addresses
5. IPv4-mapped addresses
6. Addresses with 1111 1110 10 as upper 10 bits and all of bits 11 to 64 not set to 0s
7. Addresses with 1111 1111 10 as upper 10 bits and all later bits set to 0s
8. Addresses with prefix length other than 64 and interface ID set to all 0s

(3) Automatically generating an address when an interface ID is omitted

In the Switch, when you assign an IPv6 address to an interface, you can specify only the format prefix and omit the interface ID. When you specify the format prefix and the prefix length is 64, the Switch automatically generates the interface ID based on the MAC address. The following figure shows how an address is automatically generated.

Figure 17-16: Example of automatically generating an address



1. Specify the address format prefix (for example, 3ffe:0501:0811:ff01::).
2. The interface ID is automatically generated based on the type of medium used (for example, 0200:87ff:fed0:3090).
3. The generated interface ID and the specified format prefix are combined to create an address.

If an IPv6 address other than a link-local address is assigned to an interface and the interface does not have a link-local address, the Switch automatically generates and assigns a link-local unicast address to the interface. In addition, the Switch can automatically generate and assign only link-local unicast addresses to interfaces.

(4) Specifiable prefix lengths

The Switch automatically generates interface IDs when they are not specified. Because the length of an interface ID is fixed at 64 bits, if you specify a prefix of 65 or more, the Switch does not automatically generate an interface ID. Instead, it treats the entered prefix as an address and leaves the interface ID portion as 0s. This means you cannot specify a prefix that causes all of the lower 64 bits to be set to 0. The following table describes the specifiable prefix lengths.

Table 17-3: Specifiable prefix lengths

Prefix length range	Specifiable	Description
3ffe:501::/1 to 3ffe:501::/31	Y	If the specified prefix length is within this range, the prefix length is shorter than the allowed prefix length. Because not all of the bits in the interface ID portion are set to 0, the prefix length can be specified.
3ffe:501::/32 to 3ffe:501::/63	N	If the specified prefix length is within this range, the prefix length is longer than the allowed prefix length. Because all the bits in the interface ID portion are set to 0, the prefix length cannot be specified.
3ffe:501::/64 or 3ffe:501::	Y	If the specified prefix length is 64 or no prefix length is specified and the interface ID is omitted, the interface ID is automatically generated by the Switch. Therefore, this prefix length can be specified.

Legend: Y: The prefix length can be specified. N: The prefix length cannot be specified.

17.1.7 Stateless address autoconfiguration

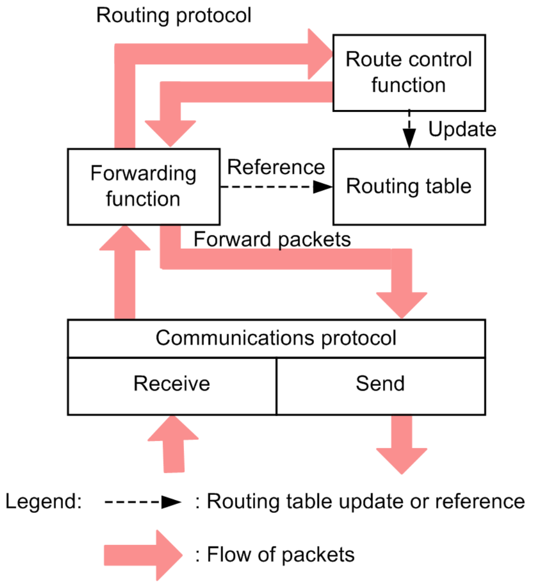
Auto-configuration automatically generates IPv6 link-local addresses and reports necessary information to hosts when hosts automatically generate IPv6 addresses. The Switch supports IPv6 stateless address auto-configuration as defined in RFC 2462.

17.2 Functions in the IPv6 layer

17.2.1 Forwarding function

The Switch forwards received IPv6 packets based on routing tables. This forwarding processing can be roughly divided into the three functions described below. The following figure provides an overview of these functions.

Figure 17-17: Overview of IPv6 routing



- Communication protocols

  Communication protocols are used to send and receive IP packets at the IPv6 layer.

- Forwarding function

  The forwarding function forwards IPv6 packets based on routing tables.

- Route control function

  The route control function exchanges routing information with other devices, determines routes, and creates routing tables.

In AX3650S series switches, you can disable forwarding for a particular VLAN interface and use that VLAN interface to manage remote operation terminals.

17.2.2 Items assigned IPv6 addresses

In the Switch, you can assign IPv6 addresses to VLANs. In IPv6, you can assign multiple IPv6 addresses to an interface. An IPv6 link-local address is automatically assigned to a VLAN assigned with an IPv6 address except when link-local addresses are specified in the configuration.

17.3 Communication protocols

This section describes the communication protocols used to forward IPv6 packets. The following IPv6 communication protocols are available:

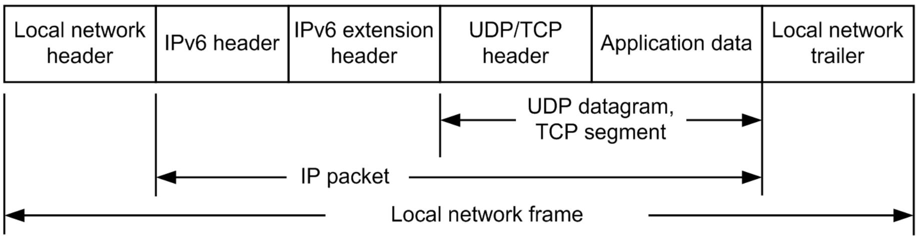
- IPv6
- ICMPv6
- NDP

17.3.1 Internet Protocol version 6 (IPv6)

(1) Format of an IPv6 packet

The format and settings of IPv6 packets sent by the Switch conforms to RFC 2460. The following figure describes the format of an IPv6 packet.

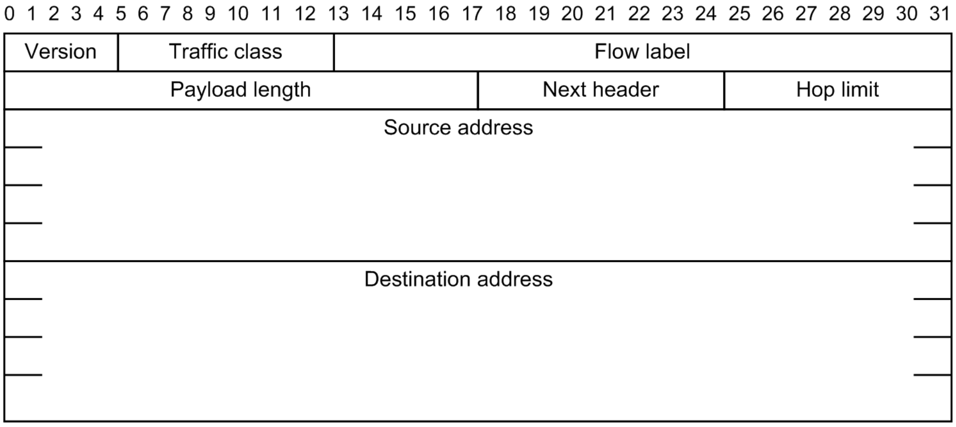
Figure 17-18: Format of an IPv6 packet



(2) Checking the validity of IPv6 packet headers

The 40-octet IPv6 header contains eight fields and two addresses. The following figure shows the format of an IPv6 header.

Figure 17-19: IPv6 header format



▪ Version (4 bits)	: IP version
▪ Traffic class (8 bits)	: Class or priority
▪ Flow label (20 bits)	: The number of the flow containing the packet
▪ Payload length (16 bits)	: The size of the payload in octets
▪ Next header (8 bits)	: The type of the header following the IPv6 header
▪ Hop limit (8 bits)	: The number of hops that are allowed
▪ Source address (128 bits)	: The source address of the packet
▪ Destination address (128 bits)	: The destination address of the packet

The Switch checks the validity of the IPv6 headers in IPv6 packets when IP packets are received. The following table describes the IPv6 packet header items that are checked.

Table 17-4: Items to be checked in an IPv6 packet header

IPv6 packet header field	Check item	Packet process if error detected	Whether to send an ICMPv6 message for a discarded packet
Version	The version must be 6.	Discarded	Do not send
Traffic class	Not checked.	--	--
Flow label	Not checked.	--	--
Payload length	Compare with the packet length. If the payload length is greater than the packet length, the packet is discarded.	Discarded	Do not send
	Compare with the packet length. The payload length must be less than the packet length.	The payload length is removed from the end of the packet.	Do not send
Next header	Not checked.	--	--
Hop limit	The hop limit of the received packets directed to the local device is not checked.	--	--
	For the hop limit of the packets to be forwarded: *Hop limit* - 1 > 0	Discarded	Send#
Source address	The following conditions must be satisfied: 1. The address is not a link-local address. 2. The address is not a multicast address.	Discarded	Do not send
Destination address	The following conditions must be satisfied: 1. The address is not the loopback address. 2. The interface ID is not 0 (except for the unspecified address).	Discarded	Do not send

Legend: --: Not applicable.

#: An ICMPv6 Time Exceeded message is sent.

(3) Supported IPv6 extension headers

The following table describes the IPv6 extension headers supported by the Switch.

Table 17-5: Supported IPv6 extension headers

IPv6 extension header	Type of IPv6 packet		
	Packet sent from the Switch	Packet sent to the Switch#1	Packet forwarded by the Switch
Hop-by-Hop Options Header	Y	Y	Y#2
Routing Header	Y	Y	--
Fragment Header	Y	Y	--

IPv6 extension header	Type of IPv6 packet		
	Packet sent from the Switch	Packet sent to the Switch#1	Packet forwarded by the Switch
Authentication Header	N	N	--
Encapsulating Security Payload Header	N	N	--
Destination Options Header	Y	Y	--

Legend: Y: Supported, N: Not supported, --: Not applicable

#1

If a packet that reaches the Switch satisfies the following conditions, the packet is discarded:

- Nine or more extension headers are set.

- Nine or more options are set in one extension header.

#2

If the packet to be forwarded by the Switch satisfies the following condition, the packet is discarded:

- Nine or more options are set in the Hop-by-Hop Options header.

17.3.2 ICMPv6

The format and settings of ICMPv6 messages sent by the Switch conforms to RFC 2463. The following table describes the supported ICMPv6 message types.

Table 17-6: Supported ICMPv6 message types

ICMPv6 message				Supported
Type	Code (decimal)	Detailed type	Code (decimal)	
Destination Unreachable	1	no route to destination	0	Y
		communication with destination administratively prohibited	1	Y
		beyond scope of source address	2	Y
		address unreachable	3	Y
		port unreachable	4	Y
Packet Too Big	2	--	0	Y
Time Exceeded	3	hop limit exceeded in transit	0	Y
		fragment reassembly time exceeded	1	Y
Parameter Problem	4	erroneous header field encountered	0	Y
		unrecognized Next Header type encountered	1	Y
		unrecognized IPv6 option encountered	2	Y
Echo Request	128	--	0	Y
Echo Reply	129	--	0	Y

ICMPv6 message				Supported
Type	Code (decimal)	Detailed type	Code (decimal)	
Multicast Listener Query	130	--	0	Y
Multicast Listener Report	131	--	0	Y
Multicast Listener Done	132	--	0	Y
Router Solicitation	133	--	0	Y
Router Advertisement	134	--	0	Y
Neighbor Solicitation	135	--	0	Y
Neighbor Advertisement	136	--	0	Y
Redirect	137	--	0	Y

Legend: Y: Supported, --: Not applicable

(1) Sending an ICMPv6 Redirect message

If the receiving interface and the sending interface are the same for a forwarded packet, the hardware determines that a decision about whether an ICMPv6 Redirect message can be sent is required, and the software determines whether an ICMPv6 Redirect message can be sent. The software sends an ICMPv6 Redirect message when the following conditions are satisfied:

- The packet source and the next hop router belong to the same link.
- The received IPv6 packet is not an ICMPv6 packet.

(2) Sending an ICMPv6 Time Exceeded message

The Switch sends an ICMPv6 Time Exceeded message if the following conditions are satisfied:

- The hop limit is 1 in the received IPv6 packet to be forwarded.
- The received IPv6 packet is not an ICMPv6 packet.

17.3.3 NDP

The format and settings of NDP frames sent by the Switch conforms to RFC 2461.

(1) ProxyNDP

You can run Proxy NDP on all interfaces in the Switch connected to Ethernet networks. When the Switch receives a Neighbor Solicitation message that satisfies all of the following conditions, the Switch sends a Neighbor Advertisement message on behalf of the destination protocol indicated by the destination protocol address:

- The destination protocol address in the Neighbor Solicitation message is not a multicast address or anycast address.
- The network number of the source protocol address and that of the destination protocol address in the Neighbor Solicitation message are the same.
- The destination protocol address in the Neighbor Solicitation message exists in the routing table and is reachable.

(2) Conditions for deleting NDP entries

If an NDP entry satisfies any of the condition below, that entry is deleted. (Note, however, that static NDP entries specified in the configuration are not deleted.)

- The communication with the IPv6 address indicated in the NDP entry has stopped and 10 minutes have elapsed.
- When the communication with the IPv6 address indicated in the NDP entry in stale status is resumed, the neighbor is no longer reachable.
- All NDP entries indicate neighbors that exist on interfaces that are down.

(3) Configuring static NDP information

To connect the Switch to a product that does not use NDP, use the `ipv6 neighbor` configuration command to associate Ethernet MAC addresses and IPv6 addresses (static NDP information).

(4) Referencing NDP information

You can execute the `show ipv6 neighbors` command on an operation terminal to check NDP information. By checking NDP information, you can determine the association between the IPv6 address and the MAC address for a specific interface.

17.4 Forwarding function

The forwarding function transfers the received packets to the next router or host based on routing tables.

17.4.1 Routing table contents

A routing table contains multiple entries, each of which consists of the items described below. To check the contents of the routing tables maintained by the Switch, execute the `show ipv6 route` command.

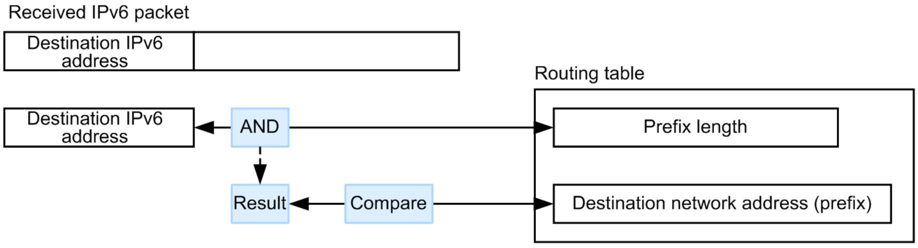
- Destination:

  The destination network address (that is, prefix) and the prefix. The prefix length is the mask that is to be applied to the destination address in a received IPv6 packet when routing tables are searched. This field indicates 128 when the IP packet is forwarded based on the host address.
- Next Hop: IPv6 address of the next router to which the IP packet is forwarded
- Interface: Name of the VLAN containing the next hop router
- Metric: The distance to the destination network
- Protocol: The routing protocol used on the node that previously forwarded the packet
- Age: The time, in seconds, since the route was confirmed or changed

17.4.2 Searching a routing table

The forwarding function searches the routing table for the entry that matches the destination address in the received IPv6 packet. The entry that matches contains the destination network address (that is, prefix). This entry is found when the Switch applies the prefix length (mask) in each entry in an AND operation with the destination address in the received IPv6 packet starting from the high-order bits. The following figure shows how a routing table is searched to determine the destination network address.

Figure 17-20: Searching a routing table



17.5 Notes on using IPv6

(1) Changing the MTU size for IPv6 forward paths

The minimum packet length for IPv6 is defined as 1280 bytes (RFC 2460). Therefore, if the MTU size is less than 1280 bytes, IPv6 communication is not possible. Specify at least 1280 bytes for the MTU size for interfaces that perform IPv6 communication.

(2) Assigning multiple global addresses to an interface

When you assign multiple global addresses to an interface, if terminals that are connected on the same link as the interface use different global addresses to communicate with one another, IPv6 forwarding might occur via the Switch.

If forwarding does occur, the hardware forwards packets to the software so that the software can determine whether an ICMPv6 Redirect message can be sent. Because this process significantly increases the load on the CPU of the Switch, keep the following in mind:

- Use the same prefix for all the terminals that are connected on the same link by using IPv6 address auto-configuration that has been enabled by Router Advertisement messages or other methods.
- If you need to use different prefixes for the terminals that are connected on the same link for security reasons, we recommend that you avoid high CPU load by using the appropriate configuration command to stop the hardware from making a determination regarding whether ICMPv6 Redirect messages can be sent.

(3) Duplicate IPv6 addresses

The IPv6 Neighbor Discovery Protocol (NDP) provides Duplicate Address Detection (DAD) as defined in RFC 2462. If DAD detects a duplicate IPv6 address, that address cannot be used for communication. If you execute the `show ipv6 interface` command or `show ip-dual interface` command and duplicated is indicated next to an IPv6 address in the list, that IPv6 address is already being used by another device. For each case, perform the following procedure:

- If the IPv6 address assigned to the other device is incorrect

  First correct the IPv6 address of the other device, and then delete and set again the IPv6 address of the Switch. Alternatively, restart the Switch.
- If the IPv6 address assigned to the Switch is incorrect

  Delete the duplicate IPv6 address assigned to the Switch in the configuration, and assign a correct IPv6 address.
- If duplicate IPv6 addresses are automatically generated

  A loop occurred on a VLAN interface or a terminal is illegally using the IPv6 address of the Switch. Eliminate the problem and execute the `no ipv6 enable` command. Next, execute the `ipv6 enable` command.

(4) Static NDP entries

When you specify the IPv6 address that is assigned to an interface in the Switch as a static NDP entry, the operation of the Switch becomes unstable and communication might become impossible. To prevent this problem, when you configure an interface, the Switch checks whether the IPv6 address assigned to the interface is specified as a static NDP entry. However, the Switch does not check the following IPv6 addresses for duplication:

- Link-local addresses (automatically and manually generated)
- Global addresses that are automatically generated when interface IDs are omitted

Therefore, do not specify these IPv6 addresses assigned to interfaces as static NDP entries. If you have mistakenly specified such a static NDP entry, delete it and restart the VLAN for the applicable interface.

(5) Forwarding Layer 3 packets with IPv6 extension headers

- If you perform Layer 3 forwarding for the packets with the Hop-by-Hop Options header, the software performs forwarding.
- If the receiving side uses QoS control, the software performs Layer 3 forwarding for the TCP packets with the Routing header or Destination Options header.

Chapter

18. Settings and Operation for IPv6, NDP, and ICMPv6

This chapter describes how to configure and check the status of an IPv6 network.

18.1 Configuration

18.1.1 List of configuration commands

The following table describes the configuration commands for IPv6.

Table 18-1: List of configuration commands

Command name	Description
ipv6 address	Sets the IPv6 address.
ipv6 enable	Enables IPv6 on an interface. This command automatically creates a link-local address.
ipv6 icmp error-interval	Specifies the sending interval of ICMPv6 error messages.
ipv6 icmp nodeinfo-query	Responds to queries from terminals.
ipv6 redirects	Specifies whether ICMPv6 Redirect messages can be sent.
ip redirects (global)#	Specifies whether ICMP and ICMPv6 Redirect messages can be sent for the entire Switch.
ip routing#	`no ip routing` invalidates IPv4 and IPv6 forwarding.

#

See *2. IPv4, ARP, and ICMP* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

18.1.2 Preparation before configuring IPv6

Points to note

In the Switch, hardware resources are configured only for IPv4 by default. Before you can configure IPv6, you need to switch to the mode that allows you to configure hardware resources for IPv6. If hardware resources are already configured for IPv6, this configuration process is unnecessary.

Command examples

1. `(config)# swrt_table_resource l3switch-2`

   Switches to the mode in which you can configure hardware resources for IPv6.

18.1.3 Configuring an interface

Points to note

The example below shows how to set an IPv6 address for a VLAN. A maximum of seven addresses can be specified for one interface. Use the `ipv6 enable` command to enable IPv6. If the `ipv6 enable` command is not set, IPv6 configuration is not enabled.

Command examples

1. `(config)# interface vlan 100`

   Switches to interface configuration mode for VLAN ID 100.

2. `(config-if)# ipv6 enable`

   Enables the use of an IPv6 address for VLAN ID 100.

3. `(config-if)# ipv6 address 2001:100::1/64`

   Sets IPv6 address 2001:100::1 for VLAN ID 100, and specifies a prefix length of 64.

4. `(config-if)# ipv6 address 2001:200::1/64`

   Sets IPv6 address 2001:200::1 for VLAN ID 100, and specifies a prefix length of 64.

18.1.4 Configuring a link-local address manually

Points to note

The Switch automatically generates a link-local address when you execute the `ipv6 enable` configuration command. Each interface can have one link-local address, which, if you prefer, can be set manually.

Command examples

1. `(config)# interface vlan 100`

   Switches to interface configuration mode for VLAN ID 100.

2. `(config-if)# ipv6 enable`

   Enables the use of an IPv6 address for VLAN ID 100. At this time, a link-local address is automatically generated.

3. `(config-if)# ipv6 address fe80::1 link-local`

   Changes the link-local address automatically set for VLAN ID 100 to fe80::1.

18.1.5 Configuring the loopback interface

Points to note

The example below shows how to set the IPv6 address for identifying the Switch. Only 0 can be specified as the interface number, and only one address can be specified.

Command examples

1. `(config)# interface loopback 0`

   Switches to interface configuration mode for the loopback interface.

2. `(config-if)# ipv6 address 2001::1`

   Sets IPv6 address 2001::1 for the Switch.

18.1.6 Configuring static NDP

Points to note

The example below shows how to configure the static NDP of a neighboring node in the NDP table of the Switch.

Command examples

1. `(config)# ipv6 neighbor 2001:100::2 interface vlan 100 0012.e240.0a00`

   Sets the next hop IPv6 address 2001:100::2 and sets the destination MAC address 0012.e240.0a00 for VLAN 100 to configure static NDP.

18.2 Operation

18.2.1 List of operation commands

The following table describes the operation commands for IPv6, NDP, and ICMPv6.

Table 18-2: List of operation commands

Command name	Description
show ip-dual interface	Shows the status of IPv4 and IPv6 interfaces.
show ipv6 interface	Shows the status of IPv6 interfaces.
show ipv6 neighbors	Shows NDP information.
clear ipv6 neighbors	Clears dynamic NDP information.
show netstat (netstat)	Shows the status of a network.
clear netstat	Clears the network statistics counter.
clear tcp	Ends a TCP connection.
ping ipv6	Performs an ICMPv6 Echo test.
traceroute ipv6	Shows an IPv6 route.

18.2.2 Checking the up/down states for an IPv6 interface

After you set an IPv6 address for a line or a port on a line of the Switch connected to an IPv6 network, execute the `show ipv6 interface` command to confirm that the status of the IPv6 interface is `UP`.

Figure 18-1: Example of displaying the IPv6 interface status

```
> show ipv6 interface summary
vlan100: UP  2001::1/64
vlan200: UP  2002::1/64
>
```

18.2.3 Checking the reachability to the destination

Execute the `ping ipv6` command for an interface in the Switch connected to the IPv6 network to determine whether the destination device is reachable.

Figure 18-2: Results of executing the ping ipv6 command (when the destination is reachable)

```
> ping ipv6 2001::2
PING6 (56=40+8+8 Bytes) 2001::1 -->2001::2
16 bytes from 2001::2, icmp_seq=0 ttl=255 time=0.286 ms
16 bytes from 2001::2, icmp_seq=1 ttl=255 time=0.271 ms
16 bytes from 2001::2, icmp_seq=2 ttl=255 time=0.266 ms
^C
--- 2001::2 ping6 statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.266/0.274/0.286 ms
>
```

Figure 18-3: Results of executing the ping ipv6 command (when the destination is not reachable)

```
> ping ipv6 2001::2
PING6 (56=40+8+8 bytes) 2001::1 --> 2001::2
^C
```

```
--- 2001::2 ping6 statistics ---
12 packets transmitted, 0 packets received, 100% packet loss
>
```

18.2.4 Checking the route to the destination

Execute the `traceroute ipv6` command to check the routers between the interface of the Switch connected to the IPv6 network and the destination.

Figure 18-4: Results of executing the traceroute ipv6 command

```
> traceroute ipv6 2003::1 numeric
traceroute6 to 2003::1 (2003::1), 30 hops max, 40 byte packets
1  2001::1  0.612 ms  0.541 ms  0.532 ms
2  2002::1  0.905 ms  0.816 ms  0.807 ms
3  2003::1  1.325 ms  1.236 ms  1.227 ms
>
```

18.2.5 Checking NDP information

After setting an IPv6 address for a line or a port on a line in the Switch connected to the IPv6 network, execute the `show ipv6 neighbors` command to check whether addresses are resolved between the Switch and the neighboring devices (that is, whether NDP entries exist).

Figure 18-5: Results of executing the show ipv6 neighbors command

```
> show ipv6 neighbors interface vlan 100
Date 20XX/10/25 14:00 UTC
Total: 3 entries
Neighbor             Linklayer Address Netif      Expire    S Flgs P
2001::1              0012.e222.f298    VLAN0100   7s        R
2002::1              0012.e26b.8e1b    VLAN0100   24s       R
fe80::1%VLAN0100     0012.e240.3f90    VLAN0100   2s        R R
```

Chapter

19. Null Interface (IPv6)

This chapter describes the null interface for an IPv6 network and how to use it.

19.1 Description

IPv6 supports the null interface. For details about the null interface behavior, see *3. Null Interface (IPv4)*.

For details about IPv6 static routing and route control, see *24. Static Routing (IPv6)* through *28. BGP4+ [OS-L3SA]*.

19.2 Configuration

19.2.1 List of configuration commands

The following table describes the configuration commands for the null interface (IPv6).

Table 19-1: List of configuration commands

Command name	Description
interface null	Accesses the null interface.
ipv6 route#	Generates an IPv6 static route.

#

See *24. Static Routing (IPv6)* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

19.2.2 Configuring a route to the null interface

Points to note

The example below shows how to set the null interface to discard the packets sent to a specific network or terminal via the Switch.

Command examples

1. `(config)# interface null 0`

   Sets the null interface.

2. `(config)# ipv6 route 2001:db8:ffff:1::/64 null 0`

   Specifies the null interface as the next hop of static route 2001:db8:ffff:1::/64. When the packets sent to this network pass through the Switch, the packets are not forwarded and instead sent to the null interface for discarding.

19.3 Operation

19.3.1 List of operation commands

The following table describes the operation command for the null interface (IPv6).

Table 19-2: List of operation commands

Command name	Description
show ipv6 route[#]	Shows routing information stored in the routing table.

#

See *13. IPv6 Routing Protocols* in the manual *Operation Command Reference Vol. 2 For Version 11.10*.

19.3.2 Checking the null interface

You can check the following information when you use the null interface in the Switch.

(1) Check after configuration

(a) Checking the routing information

Execute the `show ipv6 route` command to check whether the routing information you set by using the `static` configuration command is correct.

Figure 19-1: Displaying the routing information related to the null interface

```
> show ipv6 route static
Total: 1 routes
Destination              Next Hop   Interface      Metric   Protocol  Age
3ffe:501:811:ffcc::/64   ----       null0          0/0      Static    16s
>
```

Chapter

20. Router Advertisements

This chapter describes router advertisements (RAs).

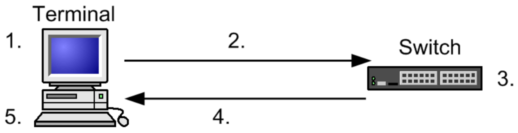
20.1 Description

20.1.1 Overview

Routers use router advertisements (RAs) to distribute to terminals the information that they need to generate IPv6 addresses as well as the default routes.

Routers regularly distribute only the prefix of their addresses in RAs. When a terminal receives an RA, it generates its address by combining its own interface ID and the prefix in the RA. In a sense, RAs provide terminals with a simple method of obtaining the prefixes of links without the need for servers. Note that RAs trigger address auto-configuration only on terminals. If a router receives an RA, it does not automatically configure an address.

Figure 20-1: Address auto-configuration triggered by an RA



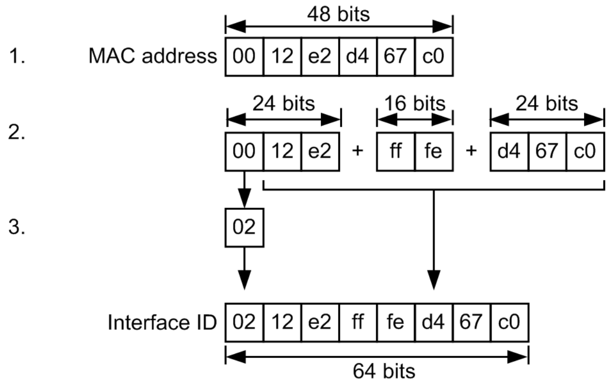
1. Generate an interface ID.
2. Request a prefix.
3. Configure the prefix to be provided in a Router Advertisement.
4. Send the prefix.
5. Combine the provided prefix and the interface ID to generate and configure an address.

20.1.2 Distributing information

There are two types of address distribution by using RAs: periodic distribution by routers and distribution in response to requests from terminals. Both types of address distribution are performed by using ICMPv6 RAs (the type field in the messages indicates 134). When a terminal wants to find a router, it sends an ICMPv6 Router Solicitation message (the value of type field in the packet is 133).

When a terminal receives an RA, it generates a global address by combining the given prefix and the 64-bit interface ID it owns (the interface ID is usually generated based on the 48-bit MAC address). Then the terminal sets the global address for the interface that received the RA. At the same time, the terminal sets the source address of the RA (link-local address of the router interface that sent the RA) as its default gateway address. The following figure shows how an interface ID is generated based on a MAC address.

Figure 20-2: Generating an interface ID from a MAC address



1. The MAC address is divided into two parts, each part of which is 24 bits.
2. Insert a fixed value (ff fe) in the middle.
3. Set the 7th bit in the first group of 8 bits to 1 (02 in decimal notation).

The prefix sent from the router to the terminal is usually the prefix of the address set for the interface that sends RAs. However, routers can advertise other prefixes as well. You can configure the maximum and minimum intervals for sending RAs from routers for each interface. The following table describes the information distributed in an RA.

Table 20-1: Information distributed in an RA

Distributed information	Description	Specifiable value	Default
Managed address configuration flag (ManagedFlag)	This flag tells terminals to perform IPv6 address auto-configuration upon reception of the RA using a method other than RAs (such as DHCPv6). Regardless of the value set for this flag, address configuration triggered by RAs is always performed. This flag is usually set to off.	ON/OFF	OFF
Other stateful configuration flag (OtherConfigFlag)	This flag tells terminals to perform auto-configuration of information (such as the address of a DNS server) other than IPv6 addresses upon reception of the RA using a method other than RAs (such as DHCPv6). This flag is usually set to off.	ON/OFF	OFF
Link MTU (LinkMTU)	The MTU size to be used by terminals in actual communication. The MTU size usually used is that of the interface that received an RA. If you do not want the terminals to send or receive the packets of the MTU size set for an interface, specify a value smaller than the MTU size in this field. Routers cannot provide an MTU size that is larger than the MTU set for the interface.	0 (do not distribute this information), or 1280 to interface MTU	MTU for the interface
Reachable time (ReachableTime)	In IPv6, terminals use ICMPv6 messages to check the reachability of neighboring nodes. This field indicates the length of time terminals assume a neighbor is reachable after they have received a reachability confirmation. If the reachable time is not specified or 0 is specified, the default value determined for each terminal is used. When a value other than 0 is set in this field, the value is also used as the base for the reachable state transition time for neighbor entries that is learned on the applicable interface on the Switch.	0 to 4294967295 (milliseconds)	0

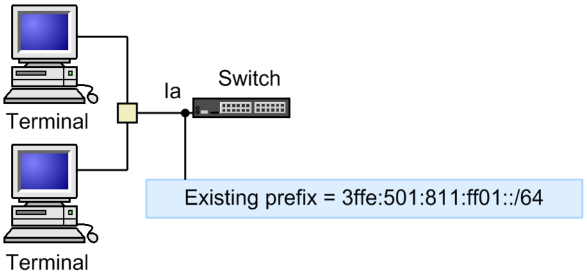
Distributed information	Description	Specifiable value	Default
Retransmission timer (RetransTimer)	In IPv6, terminals use ICMPv6 messages to check the reachability of neighboring nodes. This field specifies the interval of sending ICMPv6 packets to terminals. If the retransmission timer is not specified or 0 is specified, the default value determined for each terminal is used. When a value other than 0 is specified, the value is also used as the interval for resending Neighbor Solicitation messages for address resolution that use neighbor entries and neighbor unreachability detection leaned on the applicable interface of the Switch.	0, or 1000 to 4294967295 (milliseconds)	0
Current hop limit (CurHopLimit)	This field indicates the value to be specified in the Hop Limit field in the IPv6 header of packets sent by terminals. The value in the Hop Limit field indicates how many hops packets are forwarded.	0 to 255	64
Router lifetime (DefaultLifetime)	Expiration time of the default router confirmed by terminals by receiving the RA. When this field is set to 0, terminals do not regard the source of the received RA as the default gateway.	0, or maximum RA sending interval to 9000 (seconds)	1800 (seconds)
Link layer option (SourceLink-layerAddressOption)	The link-layer address associated with the IPv6 address of the source of the RA. For the Switch, this field contains the MAC address of the port on the interface that sent the RA only when the interface is Ethernet or Gigabit Ethernet. When load balancing is performed by using link-layer addresses, this field is set to 0, allowing each terminal to resolve the link-layer address of the default gateway.	ON/OFF	ON
Router preference (DefaultRouterPreference)	This field indicates the priority level of an RA. When a terminal receives RAs from multiple routers, it uses this information to determine which RA it accepts.	high, medium, low	medium
Prefix (PrefixList)	A prefix advertised in an RA. When no prefix is specified in this field, routers advertise the prefix set for the advertising interface (except for the prefix in a link-local address). This field is also used to advertise prefixes other than the prefix assigned to the advertising interface or to set a time limit for the prefix assigned to an interface.	Prefix in a global or site-local address	Prefix in an address assigned to an interface other than link-local addresses
Autonomous address-configuration flag (AutonomousFlag)	Prefixes are not provided to terminals when this flag is set to off. This flag is always set to on except for testing RAs.	ON/OFF	ON

Distributed information	Description	Specifiable value	Default
On-link flag (OnLinkFlag)	Routers do not send ICMPv6 Redirect messages to terminals when this flag is set to off. This flag is always set to on except for testing RAs.	ON/OFF	ON
Preferred lifetime (PreferredLifetime)	The length of time during that terminals are allowed to use the prefix provided in the RA in the source address of packets sent by terminals. If a terminal does not receive a new RA after the preferred lifetime is exceeded, the terminal tries to use an address that does not contain the previously provided prefix as the source address for communication. However, if the terminal does not have another appropriate prefix, it uses the prefix for communication even if the prefix has exceeded the preferred lifetime.	0, or maximum RA sending interval to 4294967295 (seconds)	604800 (seconds)
Valid lifetime (ValidLifetime)	The expiration time of the prefix provided by an RA. If a terminal does not receive a new RA after the valid lifetime is exceeded, the terminal deletes the address containing the expired prefix.	0, or maximum RA sending interval to 4294967295 (seconds)	2592000 (seconds)

20.1.3 Timing for changing prefixes

In a system that uses RAs to distribute terminals with prefixes, if a prefix is changed, addresses are suddenly changed and communication might stop. To avoid this problem, previous prefixes remain for 604800 seconds (7 days) by default. To delete an old prefix, routers need to advertise both the old and new prefixes and gradually shorten the life time of the old prefix. The following figure shows how RAs are used to do this.

Figure 20-3: Example of using RAs



1. Configure Ethernet interface `Ia` to send an RA to the network.
    - Prefix of `Ia` = 3ffe:501:811:ff01::/64
2. Change the prefix of `Ia` from 3ffe:501:811:ff01::/64 to 3ffe:501:811:ff22::/64 as follows:
    - Set a short advertisement interval for the new prefix 3ffe:501:811:ff22::/64 to be advertised on `Ia`, and start advertising.
    - Set a short preferred lifetime and valid lifetime for the old prefix 3ffe:501:811:ff01::/64 of `Ia`, which is going to become unavailable, and advertise it.
    - Restore the advertisement interval of the new prefix 3ffe:501:811:ff22::/64 of `Ia` to the default value.
    - Stop advertising the old prefix 3ffe:501:811:ff01::/64.

20.2 Configuration

When you execute the `ipv6 address` or `ipv6 enable` configuration command, interfaces enabled with IPv6 automatically start sending RAs.

If you want to suppress sending RAs or change the attributes of RAs, do so on each interface.

20.2.1 List of configuration commands

The following table describes the configuration commands for RAs.

Table 20-2: List of configuration commands

Command name	Description
ipv6 hop-limit	Specifies the initial value for the hop limit provided in RAs used by terminals when they send packets.
ipv6 nd link-mtu	Specifies the MTU size set for the MTU option in RAs.
ipv6 nd managed-config-flag	Sets a flag in RAs that tells terminals to perform address auto-configuration by using a method other than RAs (such as DHCPv6).
ipv6 nd no-advertise-link-address	Prohibits the link-layer addresses associated with the IP addresses of routers from being included in RAs.
ipv6 nd ns-interval	Sets the resending interval of Neighbor Solicitation messages provided in RAs to terminals. Terminals use Neighbor Solicitation messages to check the reachability of the neighboring nodes.
ipv6 nd other-config-flag	Sets a flag in RAs that allows terminals to automatically obtain information other than IPv6 addresses by using methods other than RAs.
ipv6 nd prefix	Specifies the IPv6 prefix or information related to the prefix to be sent in RAs.
ipv6 nd ra-interval	Specifies the minimum and maximum intervals of sending RAs.
ipv6 nd ra-lifetime	Specifies the value set in the Router Lifetime field in RAs. The value indicates the effective time of the router used as the default router by terminals.
ipv6 nd reachable-time	Specifies the value set in the Reachable Time field in RAs. The configured time enables terminals to determine how long they should consider that a neighbor node is reachable after they have received reachability confirmation.
ipv6 nd router-preference	Specifies the level of priority of an RA. When a terminal receives multiple RAs, it uses this information to determine which RA it should use.
ipv6 nd suppress-ra	Stops sending RAs.

20.2.2 Configuring settings to stop the transmission of router advertisements

Prohibit an interface from sending RAs.

Points to note

The example below shows how to use the `ipv6 nd suppress-ra` command to stop sending RAs.

Command examples

1.
```
(config)# interface vlan 10
(config-if)# ipv6 nd suppress-ra
```
   Prohibits VLAN 10 from sending RAs.

2.
```
(config-if)# ipv6 address 2001:db8:1:1::1/64
```
   Sets IPv6 address 2001:db8:1:1::1/64 for interface vlan 10.

20.2.3 Configuring information to be distributed in router advertisements

Configure the information to be distributed in RAs.

Points to note

To configure the information to be distributed in RAs, switch to interface mode. In the following example, we use the `ipv6 nd other-config-flag` command to set the Other Stateful Configuration flag and use the `ipv6 nd router-preference` command to set the Default Router Preference.

Command examples

1.
```
(config)# interface vlan 10
(config-if)# ipv6 nd other-config-flag
```
   Sets the other stateful configuration flag (OtherConfigFlag) in RAs sent from VLAN 10. When terminals receive an RA, they obtain information other than the address of the VLAN by using a method other than RAs such as DHCPv6.

2.
```
(config-if)# ipv6 nd router-preference high
```
   Sets high (highest priority) in the Default Router Preference field in RAs sent from VLAN 10.

20.2.4 Configuring the router advertisement sending interval

Configure the interval for sending RAs.

Points to note

The example below shows how to use the `ipv6 nd ra-interval` command to configure the interval for sending RAs.

Command examples

1.
```
(config)# interface vlan 10
(config-if)# ipv6 nd ra-interval 600 1200
```
   Sets a variable interval from 10 to 20 minutes for sending RAs.

20.3 Operation

20.3.1 List of operation commands

The following table describes the operation commands for RAs.

Table 20-3: List of operation commands

Command name	Description
show ipv6 routers	Shows information regarding RAs.
show ipv6 interface#	Shows the status of the IPv6 interface.
show netstat (netstat) (IPv6)#	Shows the network status and statistics.

#

See *9. IPv6, NDP, and ICMPv6* in the manual *Operation Command Reference Vol. 2 For Version 11.10*.

20.3.2 Checking summaries

Display a list of interfaces that sent RAs.

Figure 20-4: List of interfaces that sent RAs

```
> show ipv6 routers global
Date 20XX/07/14 12:00:00 UTC
#Index Name            Prefix
#2     VLAN0010        2001:db8:1:1::/64
#3     VLAN0020        2001:db8:1:2::/64
#4     VLAN0030        2001:db8:1:3::/64
```

20.3.3 Checking details

Display the details about an interface that sent RAs.

Figure 20-5: Details about an interface that sent RAs

```
> show ipv6 routers interface vlan 10
Date 20XX/07/14 12:00:00 UTC
Index: 3, Name: VLAN0010
Statistics:
  RSin(wait): 5(0), RAout: 10, RAin(invalid): 0(0)
Intervals:
  RA Interval: 600-1200s (next=219s later), RA Lifetime: 1800s
  Reachable Time: ---, NS Interval: ---
Managed Config Flag: off, Other Config Flag: on, Hop Limit: 64
No Advertised Link Address: off, Link MTU: 1500

Prefix                           ValidLife[s] PrefLife[s] OnLink Autoconfig
2001:db8:1:1::/64                2592000      604800      on     on
```

Chapter

21. IPv6 DHCP Relays [OP-DH6R]

This chapter describes an IPv6 DHCP relay agent, how to configure it, and how to check it. An IPv6 DHCP relay agent forwards IPv6 DHCP packets when an IPv6 DHCP server and IPv6 DHCP clients are on different network segments. The IPv6 DHCP relay agent is abbreviated hereafter to IPv6 DHCP relay.

21.1 Description

21.1.1 Overview

An IPv6 DHCP relay forwards IPv6 DHCP packets to an IPv6 DHCP server when the IPv6 DHCP server and IPv6 DHCP clients are on different network segments. Forwarding is enabled by setting the destination specified in the configuration (IP address of the IPv6 DHCP server or IP address of the IPv6 DHCP relay that can forward packets to the network segment containing the IPv6 DHCP server) as the destination address of IPv6 DHCP packets.

An IPv6 DHCP relay running on the Switch can handle the two types of devices below for clients. These clients are generically called IPv6 DHCP clients.

- IPv6 DHCP-PD (prefix delegation) clients
- IPv6 DHCP clients that request IPv6 addresses

Similarly, IPv6 DHCP-PD servers and IPv6 DHCP servers that distribute IPv6 addresses are generically called IPv6 DHCP servers.

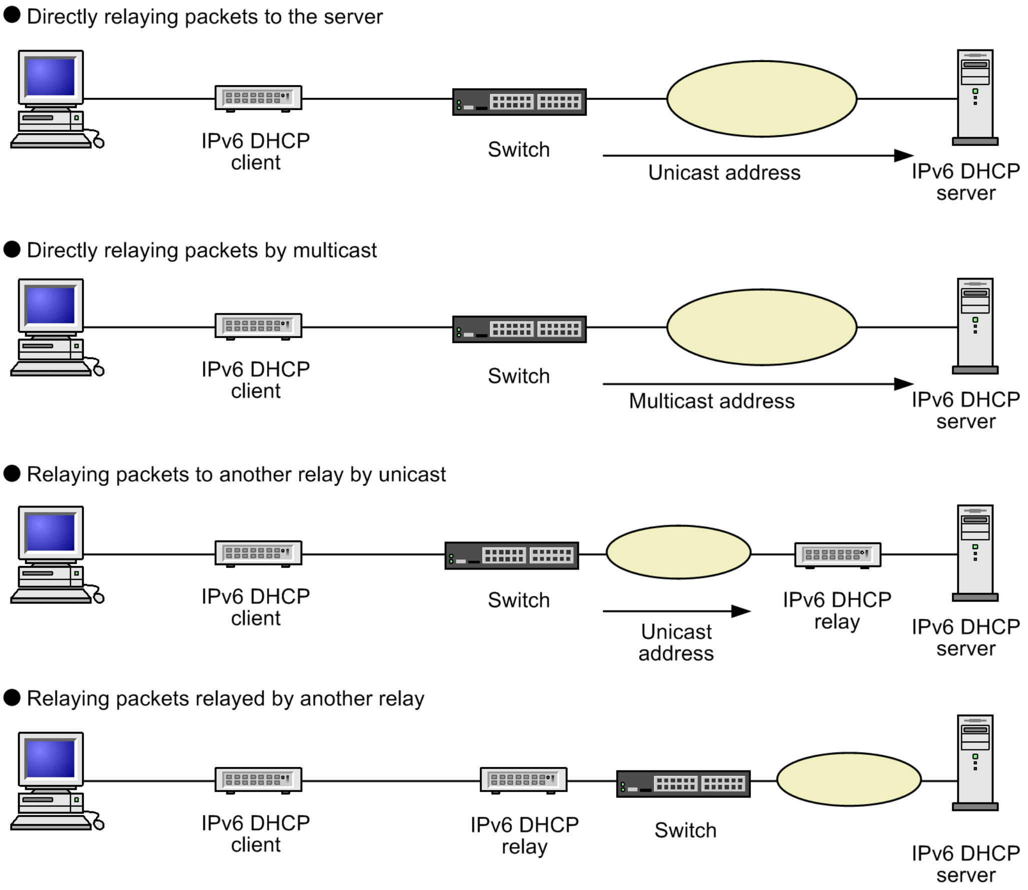
IPv6 DHCP clients use link-local multicasting to use the services provided by an IPv6 DHCP server. Therefore, the IPv6 DHCP server needs to exist on the same network segment as the IPv6 DHCP clients. However, when you use an IPv6 DHCP relay on the Switch, the IPv6 DHCP relay can forward the IPv6 DHCP packets sent by the IPv6 DHCP clients to a different network segment. This way, the IPv6 DHCP server in one network segment can provide services to the IPv6 DHCP clients on another network segment.

Running an IPv6 DHCP relay on the Switch requires an OP-DH6R optional license.

Note that in this chapter, the configuration diagrams use IPv6 DHCP-PD for the examples. IPv6 DHCP clients that request IPv6 addresses operate the same way with the exception of some functions such as automatic route generation linked with prefix distribution and the display of binding information by prefix distribution management.

The following figure shows different connection configurations with an IPv6 DHCP relay.

Figure 21-1: Connection configurations with an IPv6 DHCP relay



21.1.2 Supported specifications

The following table describes the specifications of an IPv6 DHCP relay running on the Switch.

Table 21-1: Specifications of an IPv6 DHCP relay

Item	Specifications	Supported
Connection (client side)	An IPv6 DHCP relay is directly connected to IPv6 DHCP clients.	Y
	Via IPv6 DHCP relays	Y
Connection (server side)	Unicast to a server	Y
	Unicast to a relay	Y
	Multicast to all servers	Y
	Multicast to all servers and all relays	N
	Local	Y
Interface	Ethernet	Y
	Link Aggregation	Y
Functionality	Packet length (UDP payload)	Y#
	Simultaneous operation with an IPv6 DHCP server	N

Item	Specifications	Supporte d
	Automatically generating routes to the clients to which prefixes are distributed	Y

Legend: Y: Supported, N: Not supported.

#: Supports up to 977 octets.

21.1.3 Forwarding

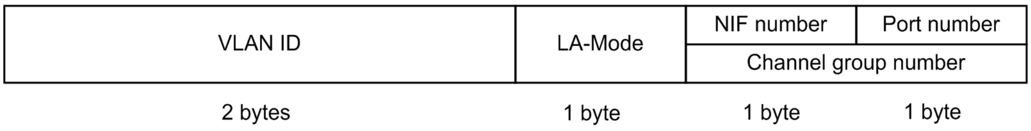
This subsection describes how an IPv6 DHCP relay on the Switch performs forwarding.

(1) Attaching an interface ID

The IPv6 DHCP relay attaches an interface ID to the IPv6 DHCP packets to be forwarded to the IPv6 DHCP server. The interface ID is used to identify the destination when the IPv6 DHCP server sends a reply packet to IPv6 DHCP clients.

The following figure describes the format of an interface ID that is set to the `Interface-ID Option`.

Figure 21-2: Format of an interface ID



VLAN ID	LA-Mode	NIF number	Port number
		Channel group number	
2 bytes	1 byte	1 byte	1 byte

LA-Mode: 0 when the NIF number or port number is specified.
1 when the channel group number is specified.

(2) When the device address (loopback 0 interface) is configured

When the IPv6 DHCP relay forwards an IPv6 DHCP packet to the IPv6 DHCP server and the device address has been configured by using the `interface loopback 0` or `ipv6 address` configuration command on the Switch, the IPv6 DHCP relay sets the device address as the source address.

When the device address is not configured, the IPv6 DHCP relay uses the address of the output interface on the IPv6 DHCP server side as the source address. If a failure occurs on an interface on the priority route between the IPv6 DHCP server and the IPv6 DHCP relay, even when redundant routes exist between them, the IPv6 DHCP clients cannot receive the IPv6 DHCP relay packet that was forwarded by the IPv6 DHCP relay before the failure.

When the device address is configured, the IPv6 DHCP server sends IPv6 DHCP packets to the address that is unique to the Switch, and not to specific interfaces. Therefore, if an error occurs on an interface on the priority route, the interface on a redundant route can receive the IPv6 DHCP packets.

21.1.4 Automatically generating routes to the clients to which prefixes are distributed

When the Switch is used as the default gateway for IPv6 DHCP clients, it can automatically configure routes to the clients to which prefixes are distributed. If an IPv6 DHCP client does not have a function for advertising routing information, use the `ipv6 dhcp relay static-route-setting` configuration command to enable automatic route generation. When the command is used, the Switch automatically adds routes to the clients to which prefixes are distributed.

The distance of a route is fixed at 250 when configured by the automatic generation of routes to the clients to which prefixes are distributed.

21.1.5 Information regarding distributed prefixes

The management information regarding assigned prefixes (hereafter lease information) is maintained by an IPv6 DHCP relay and used to configure routes. To check lease information, execute the `show ipv6 dhcp relay binding` operation command.

Note that distributed prefixes are managed by monitoring the PD option contained in IPv6 DHCP-PD. This means that distribution information is not monitored and routes are not automatically generated for packets to distribute IPv6 addresses because such packets do not include the PD option.

(1) Events that prompt the change of lease information

The following table describes the events that prompt addition or deletion of lease information.

Table 21-2: Events that prompt the addition or deletion of lease information

Description	Trigger
Addition	A distributed prefix is forwarded.
Deletion	A prefix release request is forwarded.#1
	A prefix lease has expired.
	The configuration specified by using the `ipv6 dhcp relay destination` configuration command (which is used to forward the assigned prefix) is deleted.#2
	The `copy` <*file*> `running-config` operation command is executed.#3
	Lease information is deleted by using the `clear ipv6 dhcp relay binding` operation command.

#1

Lease information is not deleted when the IPv6 DHCP relay receives a release request from an interface different from the one that received the distributed prefix. The release request itself is forwarded.

#2

The lease information regarding the prefix that is forwarded to the destination specified by using the `ipv6 dhcp relay destination` command is deleted.

#3

All lease information is deleted.

(2) Passing information related to lease information

The following table describes whether the information related to lease information is passed. The related information that is created based on lease information is the information for automatically generating routes.

Table 21-3: Whether information related to lease information is passed

Information related to lease information	When IPv6 DHCP relay restarted	When the Switch restarted
Automatically generated routes	Y#1, #2	N

Legend: Y: Passing of information is guaranteed, N: Information is deleted

#1

When you perform either of the following operations while the IPv6 DHCP relay program is being restarted, lease information might become invalid. If lease information becomes invalid, it will be deleted.

- The packet destination specified by using the `ipv6 dhcp relay destination` configuration command is deleted.

- The `copy` <*file*> `running-config` operation command is executed.

#2

When the `core-file` parameter is specified in the `restart ipv6-dhcp relay` operation command, passing of information is not guaranteed.

21.1.6 Notes on using an IPv6 DHCP relay

This subsection provides notes on using an IPv6 DHCP relay.

(1) Using together with an IPv6 DHCP server

You cannot run both an IPv6 DHCP relay and the IPv6 DHCP server functionality simultaneously on the Switch. If you run an IPv6 DHCP relay, use the `no service ipv6 dhcp` configuration command to disable the IPv6 DHCP server functionality beforehand.

(2) Concurrent use with IPv6 multicasting

When you use both IPv6 multicasting and an IPv6 DHCP relay simultaneously on the Switch, we recommend that you specify the unicast address of each IPv6 DHCP server as a relay destination of the IPv6 DHCP relay. If you want to use a multicast to all IPv6 DHCP servers, note the following:

- ■ When you specify the all-servers multicast address as the relay destination of the IPv6 DHCP packets sent from the Switch and use IPv6 multicasting at the same time, the configuration below is required on the partner router. For details about the configuration, see the router documentation.
    - Specify the maximum value within the VLAN as the link-local address of the interface to which the Switch is connected so the Switch becomes the designated router (DR) of the IPv6 multicast routing protocol.
- ■ Configure IPv6 multicasting on the Switch and the partner router so that the partner router becomes the rendezvous point.

(3) Notes on relaying packets from an IPv6 DHCP relay

- ■ When both of the following conditions are met, IPv6 DHCP relays cannot correctly relay reply packets to IPv6 DHCP clients:
    - An IPv6 global address is not set for the interface that is specified as the packet destination by using the `ipv6 dhcp relay destination` configuration command.
    - The interface ID appended by the Switch when it relayed packets is not contained in the reply packets from the IPv6 DHCP server.
- ■ The Switch can forward IP packets when the packet size is 1500 bytes or less and the packets are not fragmented.

(4) Notes on using the loopback address

The configuration below is required on a router when the router is positioned between the Switch and an IPv6 DHCP server, the all-servers multicast address is specified as the destination for IPv6 DHCP packets, and an IPv6 address is set for the loopback interface. For details about the configuration, see the router documentation.

- ■ Configure the router so that it treats the IPv6 address set for the loopback interface of the Switch as the server that is directly connected to the router and the router functions.

21.2 Configuration

21.2.1 List of configuration commands

The following table describes the configuration commands for an IPv6 DHCP relay.

Table 21-4: List of configuration commands

Command name	Description
ipv6 dhcp relay destination	Specifies the relay destination of IPv6 DHCP packets.
ipv6 dhcp relay hop-limit	Specifies the maximum hop count for relayed IPv6 DHCP packets.
ipv6 dhcp relay static-route-setting	Automatically adds a distributed prefix to the routing information table of the Switch by specifying the routing information option of an IPv6 DHCP relay.
service ipv6 dhcp relay	Enables or disables IPv6 DHCP relaying.

21.2.2 Overview of configuration

(1) Configuring unicast transmission via one IPv6 DHCP relay

1. Use the `swrt_table_resource` command to enable IPv6.
2. Set an interface for a client.
3. Set interfaces for relaying packets to IPv6 DHCP servers.
4. Disable the IPv6 DHCP server on the Switch.
5. Use the `ipv6 dhcp relay destination` command to set the addresses of the relay destination IPv6 DHCP servers.
6. Set the maximum hop count.
7. Configure the automatic generation of routes to the clients to which prefixes are distributed.

(2) Configuring multicast transmission via one IPv6 DHCP relay

1. Use the `swrt_table_resource` command to enable IPv6.
2. Set an interface for a client.
3. Set interfaces for relaying packets to IPv6 DHCP servers.
4. Disable the IPv6 DHCP server on the Switch.
5. Use the `ipv6 dhcp relay destination` command to set the interface for the destination IPv6 DHCP servers.
6. Set the maximum hop count.
7. Configure the automatic generation of routes to the clients to which prefixes are distributed.

(3) Configuring transmission via multiple IPv6 DHCP relays

(a) Configuration on Switch A

1. Use the `swrt_table_resource` command to enable IPv6.
2. Set an interface for a client.
3. Set interfaces for relaying packets to IPv6 DHCP servers.
4. Disable the IPv6 DHCP server on the Switch.

5. Use the `ipv6 dhcp relay destination` command to set the address of the relay destination Switch B.
6. Set the maximum hop count.
7. Configure the automatic generation of routes to the clients to which prefixes are distributed.

(b) Configuration on Switch B

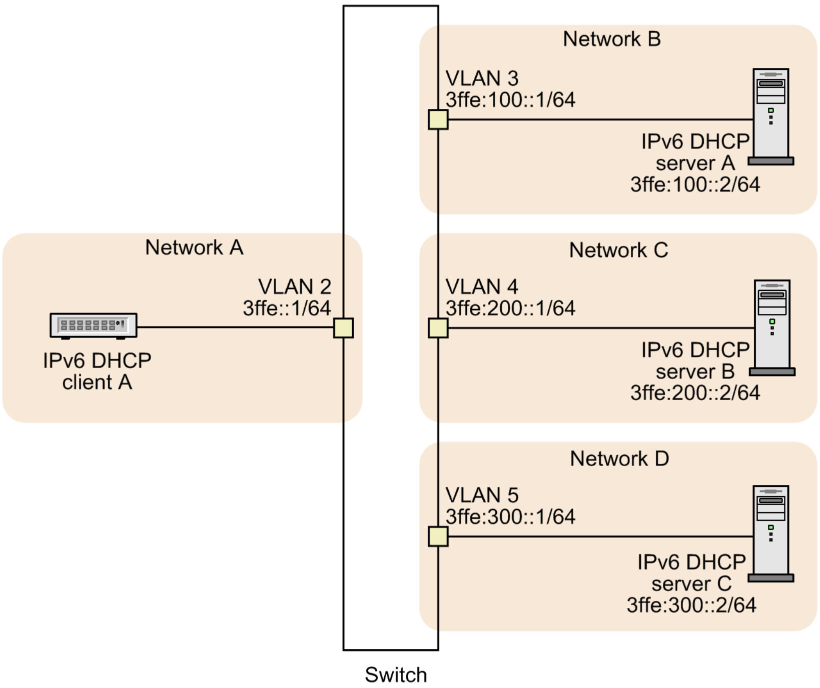
1. Use the `swrt_table_resource` command to enable IPv6.
2. Set an interface for a client.
3. Set interfaces for relaying packets to IPv6 DHCP servers.
4. Disable the IPv6 DHCP server on the Switch.
5. Use the `ipv6 dhcp relay destination` command to set the addresses of the relay destination IPv6 DHCP servers.
6. Set the maximum hop count.

21.2.3 Unicast transmission via one IPv6 DHCP relay

Points to note

The example below shows how to use the `ipv6 dhcp relay destination` command to specify the IPv6 addresses of the relay destination IPv6 DHCP servers.

Figure 21-3: Configuration for unicast transmissions via one IPv6 DHCP relay



Command examples

1. `(config)# no service ipv6 dhcp`

   Disables the IPv6 DHCP server on the Switch.

2. `(config)# service ipv6 dhcp relay`

   Enables the IPv6 DHCP relay on the Switch.

3.
```
(config)# interface vlan 2
(config-if)# ipv6 dhcp relay destination 3ffe:100::2
3ffe:200::2 3ffe:300::2
```

Sets 3ffe:100::2, 3ffe:200::2, and 3ffe:300::2 as relay destinations.

4.
```
(config-if)# ipv6 dhcp relay hop-limit 0
(config-if)# exit
```

Sets 0 as the maximum hop count.

5.
```
(config)# ipv6 dhcp relay static-route-setting
```

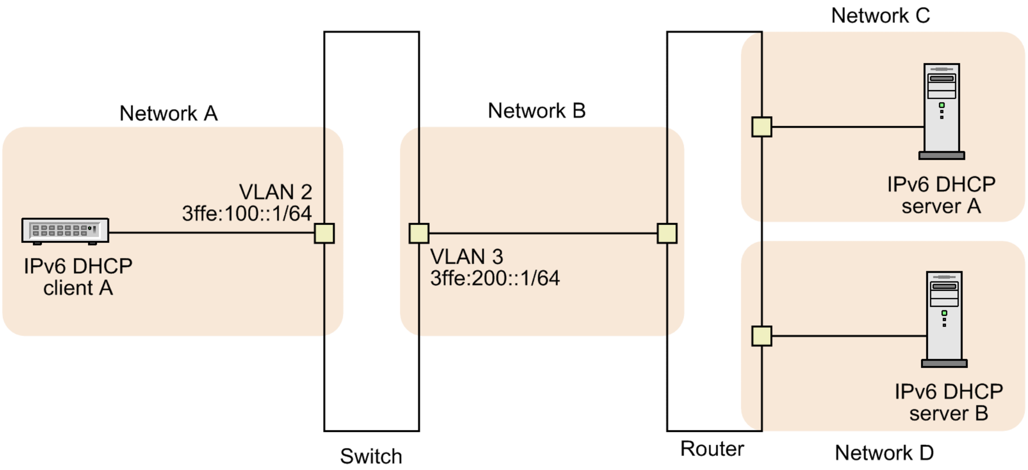
On the Switch, enables the automatic generation of routes from the DHCP servers to the client to which prefixes are distributed. Note that the routes are not configured until the prefixes have been completely distributed.

21.2.4 Multicast transmission via one IPv6 DHCP relay

Points to note

The example below shows how to use the `ipv6 dhcp relay destination` command to specify the interface for relay destination IPv6 DHCP servers.

Figure 21-4: Configuration for multicast transmissions via one IPv6 DHCP relay



Command examples

1.
```
(config)# no service ipv6 dhcp
```

Disables the IPv6 DHCP server on the Switch.

2.
```
(config)# service ipv6 dhcp relay
```

Enables the IPv6 DHCP relay on the Switch.

3.
```
(config)# interface vlan 2
(config-if)# ipv6 dhcp relay destination all-servers vlan 3
```

Sets VLAN 3 as the relay destination.

4.
```
(config-if)# ipv6 dhcp relay hop-limit 0
(config-if)# exit
```
Sets 0 as the maximum hop count.

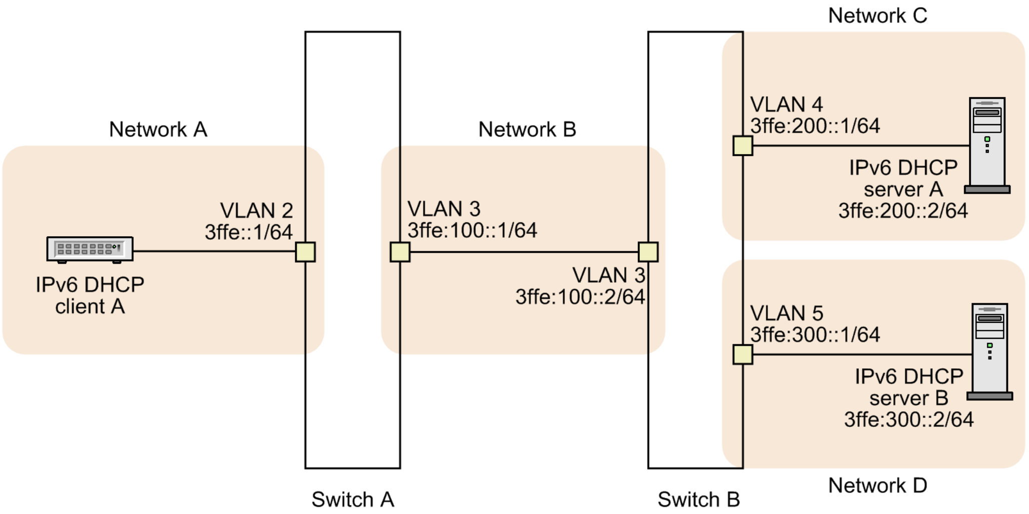
5.
```
(config)# ipv6 dhcp relay static-route-setting
```
On the Switch, enables the automatic generation of routes from the DHCP servers to the client to which prefixes are distributed. Note that the routes are not configured until the prefixes have been completely distributed.

21.2.5 Transmission via multiple IPv6 DHCP relays

Points to note

The example below shows how to use the `ipv6 dhcp relay destination` command to specify the IPv6 addresses of the relay destinations.

Figure 21-5: Configuration for transmissions via multiple IPv6 DHCP relays



Command examples

- Configuration on Switch A

1.
```
(config)# no service ipv6 dhcp
```
Disables the IPv6 DHCP server on the Switch.

2.
```
(config)# service ipv6 dhcp relay
```
Enables the IPv6 DHCP relay on the Switch.

3.
```
(config)# interface vlan 2
(config-if)# ipv6 dhcp relay destination 3ffe:100::2
```
Sets 3ffe:100::2 as the relay destination.

4. `(config-if)# ipv6 dhcp relay hop-limit 0`

   `(config-if)# exit`

   Sets 0 as the maximum hop count.

5. `(config)# ipv6 dhcp relay static-route-setting`

   On the Switch, enables the automatic generation of routes from the DHCP servers to the client to which prefixes are distributed. Note that the routes are not configured until the prefixes have been completely distributed.

- Configuration on Switch B

1. `(config)# no service ipv6 dhcp`

   Disables the IPv6 DHCP server on the Switch.

2. `(config)# service ipv6 dhcp relay`

   Enables the IPv6 DHCP relay on the Switch.

3. `(config)# interface vlan 3`

   ```
   (config-if)# ipv6 dhcp relay destination 3ffe:200::2
   3ffe:300::2
   ```

   Sets 3ffe:200::2 and 3ffe:300::2 as relay destinations.

4. `(config-if)# ipv6 dhcp relay hop-limit 1`

   `(config-if)# exit`

   Sets 1 as the maximum hop count.

21.3 Operation

21.3.1 List of operation commands

The following table describes the operation commands for an IPv6 DHCP relay.

Table 21-5: List of operation commands

Command name	Description
show ipv6 dhcp traffic	Shows IPv6 DHCP relay statistics.
clear ipv6 dhcp traffic	Deletes IPv6 DHCP relay statistics.
show ipv6 dhcp relay binding	Shows lease information for the IPv6 DHCP relay.
clear ipv6 dhcp relay binding	Deletes lease information from the IPv6 DHCP relay database.
restart ipv6-dhcp relay	Restarts the IPv6 DHCP relay program.
dump protocols ipv6-dhcp relay	Outputs relay log data collected by the IPv6 DHCP relay program to a file.

21.3.2 Checking the distributed prefixes

To check the prefixes that are assigned by an IPv6 DHCP server to clients, execute the `show ipv6 dhcp binding` command. The command displays the prefixes whose lease has not expired yet.

Figure 21-6: Example of displaying distributed prefixes

```
> show ipv6 dhcp relay binding
Date 20XX/04/09 12:00:00 UTC
Total: 2 prefixes
<Interface>      <Prefix>                                    <Lease expires>
vlan 10          3ffe:1234:5678::/48                         XX/04/10 11:11:11
vlan 20          3ffe:aaaa:1234::/48                         XX/04/10 12:12:12
>
```

21.3.3 Checking routing information associated with distributed prefixes

When routes are automatically generated for clients to which prefixes are distributed, execute the `show ipv6 route` command to check the routing information. The routes are registered as static routes in the routing information.

Figure 21-7: Example of displaying routing information

```
> show ipv6 route -s static
Date 20XX/04/09 12:00:00 UTC
Total: 5 routes
Destination                                 Next Hop
Interface      Metric   Protocol  Age
3ffe:abcd:1234::/64                         fe80::203:ffff:fe20:9982%VLAN0030
VLAN0030       0/0      Static     4m 33s , <Active Gateway Dhcp>
3ffe:aaaa:1234::/64                         fe80::203:ffff:fe20:9982%VLAN0020
VLAN0020       0/0      Static     4m 33s , <Active Gateway Dhcp>
3ffe:1234:5678::/64                         fe80::203:ffff:fe20:9982%VLAN0010
VLAN0010       0/0      Static     4m 33s , <Active Gateway Dhcp>
>
```

Chapter

22. IPv6 DHCP Server Functionality

The IPv6 DHCP server functionality dynamically assigns prefixes, DNS server addresses, and other information to IPv6 DHCP clients. Assignment of a prefix to an IPv6 DHCP client by the IPv6 DHCP server functionality is called prefix delegation.

This chapter describes the IPv6 DHCP server functionality and how to configure it.

22.1 Description

The IPv6 DHCP server functionality dynamically assigns prefixes, DNS server addresses, and other information to IPv6 DHCP clients.

22.1.1 Supported specifications

The table below describes the specifications of the IPv6 DHCP server functionality for the Switch. An IPv6 DHCP server and IPv6 DHCP clients need to be directly connected via the same network.

Table 22-1: Support specifications of the IPv6 DHCP server functionality

Item	Specifications
Connection	Directly connected to IPv6 DHCP clients.
	Via IPv6 DHCP relays
IPv4/IPv6 dual stack (for IPv6)	Supported

22.1.2 Supported DHCP options

The following table describes the IPv6 DHCP options supported by the Switch.

Table 22-2: IPv6 DHCP options supported by the Switches

Option code	Option name	Meaning	Method of specifying values
1	Client Identifier	The `Client Identifier` option is used to carry a DUID# identifying a client between a client and a server.	C
2	Server Identifier	The `Server Identifier` option is used to carry a DUID identifying a server between a client and a server.	A
3	Identity Association option	The `Identity Association` (IA) option is used to carry an identity association, the parameters associated with the IA, and the addresses associated with the IA.	N
4	Identity Association for Temporary Addresses option	The `Identity Association for Temporary Addresses (IA_TA)` option is used to carry an IA, the parameters associated with the IA, and the addresses associated with the IA. As defined in RFC 3041, all the addresses in this option are used by clients as temporary addresses.	N
5	IA Address option	The IA Address option is used to specify IPv6 addresses associated with an IA. The IA Address option must be encapsulated in the Options field of the Identity Association option. The `Options` field encapsulates the options that are specific to this address.	N
6	Option Request	The `Option Request` option is used to identify a list of options in a message between a client and a server.	A
7	Preference	The `Preference` option is sent by a server to a client to affect the selection of a server by the client.	A

Option code	Option name	Meaning	Method of specifying values
8	Elapsed Time option	A client must include the `Elapsed Time` option in messages to indicate how long the client has been trying to complete an IPv6 DHCP message exchange. The elapsed time is measured from the time at which the client sent the first message in the message exchange, and the elapsed-time field of the first message in the message exchange is set to 0. For example, the `Elapsed Time` option allows a secondary IPv6 DHCP server to respond to a request when a primary server has not answered in a reasonable time.	N
9	Relay Message option	The `Relay Message` option carries an IPv6 DHCP message in a relay-forward or relay-reply message.	A
11	Authentication option	The `Authentication` option carries authentication information to authenticate the identity and contents of IPv6 DHCP messages.	N
12	Server unicast option	The server sends this option to a client to indicate to the client that it is allowed to send unicast messages to the server.	N
13	Status Code	This option returns a status indication related to the IPv6 DHCP message or option in which it appears.	A
14	Rapid Commit	The `Rapid Commit` option is used to signal the use of two message exchanges for address assignment.	A
15	User Class option	The `User Class` option is used by a client to identify the type or category of user or applications it represents.	N
16	Vendor Class Option	This option is used by a client to identify the vendor that manufactured the hardware on which the client is running. The information contained in the data area of this option is contained in one or more opaque fields that identify details of the hardware configuration.	N
17	Vendor-specific Information option	This option is used by clients and servers to exchange vendor-specific information.	N
18	Interface-Id Option	The relay agent may send the `Interface-Id` option to identify the interface on which the client message was received. If a relay agent receives a relay-reply message with an `Interface-Id` option, the relay agent relays the message to the client through the interface identified by the option.	N
19	Reconfigure Message option	A server includes the `Reconfigure Message` option in a reconfigure message to indicate to the client whether the client is to respond with a renew message or an information-request message.	N
20	Reconfigure Nonce option	If a server uses the `Reconfigure Nonce` option to provide security for reconfigure messages, the server maintains a value of `nonce` for each client. The server initially informs the client of the `nonce` value and then includes the `nonce` value in any reconfigure messages sent to the client.	N
21	SIP Servers Domain Name List	The domain names of the SIP outbound proxy servers that the client is to use.	M
22	SIP Servers IPv6 Address List	This option specifies a list of IPv6 addresses indicating SIP outbound proxy servers available to the client.	M
23	DNS Recursive Name Server	This option is specified when a server passes a list of DNS server addresses to a client.	M

Option code	Option name	Meaning	Method of specifying values
24	Domain Search List	When a client receives this option, it searches the given domain list whenever it resolves host names by using DNS. This option must not be used for any purposes other than host name resolution.	M
25	Identify Association for Prefix Delegation Option	This option is used to carry a prefix delegation identity association, the parameters associated with `IA_PD`, and the prefixes associated with it.	M
26	IA_PD Prefix Option	This option is used to specify IPv6 address prefixes associated with `IA_PD`.	M
31	Network Time Protocol (NTP) Servers	This option is used by a server to notify a client of a list of NTP server addresses.	M

Legend:

M: Manually specified in the configuration, A: Automatically configured

C: Client-configured values are used, N: Not supported (ignored)

#: DUID stands for DHCP unique identifier.

22.1.3 Routing information associated with distributed prefixes

The Switch provides two methods for configuring routes to the receivers (clients) of distributed prefixes when the Switch is used as a gateway for clients:

- When clients do not have a function for advertising routing information

  By enabling the automatic generation of routes to clients to which prefixes are distributed in the IPv6 DHCP server configuration on the Switch, routes to the distribution destinations are automatically added to the Switch.

  The distance of the routes configured at this time is fixed at 250.

- When clients have a function for advertising routing information

  Routing information is automatically exchanged between the Switch and clients, and routes are automatically generated. Therefore, disable the automatic generation of routes to clients to which prefixes are distributed in the IPv6 DHCP server configuration on the Switch.

22.1.4 Notes on using the IPv6 DHCP server functionality

The following provides notes on using the IPv6 DHCP server functionality.

(1) DHCP unique identifiers (DUIDs)

DUIDs are used to identify IPv6 DHCP devices. The Switch generates a DUID the first time an IPv6 DHCP server functionality is used. The generated DUIDs are statically stored in memory inside the Switch. You can check the value of a DUID in the `Server DUID` section of the list that is displayed when you execute the `show ipv6 dhcp server statistics` command. When you replace the Switch, the previous DUID is no longer used. If you want to use the previous DUID value, use the `set ipv6-dhcp server duid` command to re-specify it.

(2) Restrictions when the Switch is restarted

If either of the events described in the table below occur, some restrictions apply to the Switch. The table describes whether information is preserved when the indicated events occur.

Table 22-3: Maintainability of information

Prefix-related information to be retained	DHCP server functionality restart		The Switch is restart ed
	Caused by execution of the restart ipv6-dhcpserver command	Caused by a server failure	
Routes to clients	Y	S	N
Prefixes distributed to clients	Y	S	N

Legend:

Y: Information is preserved.

S: Most information is preserved. Prefixes that are distributed immediately before a restart might not be preserved.

N: No information is preserved (the information is initialized).

(3) Notes on using the automatic generation of routes to clients to which prefixes are distributed

In situations such as when clients that do not have a function for advertising routing information, the Switch automatically configures routing information. However, the Switch should use the routing information advertised by clients when multiple paths exist or routes are dynamically changed.

The Switch should also use the routing information advertised by clients when other devices are positioned between clients and the Switch because the routing information is not advertised to intermediate devices.

(4) Notes on using an IPv6 DHCP server and an IPv6 PIM on the same interface

When you run an IPv6 DHCP server on an interface that is enabled with an IPv6 PIM, direct the DHCP control packets from IPv6 DHCP to the global unicast addresses of the Switch, not to the all-servers multicast address (FF05::1:3).

22.2 Configuration

22.2.1 List of configuration commands

The following table describes the configuration commands for an IPv6 DHCP server.

Table 22-4: List of configuration commands

Command name	Description
dns-server	Sets the DNS server address used by the IPv6 DHCP server. The DNS server address is distributed to IPv6 DHCP clients upon request.
domain-name	Sets the IPv6 DHCP server domain name. The domain name is distributed to IPv6 DHCP clients upon request.
ipv6 dhcp pool	Sets information about an IPv6 DHCP address pool. As described in *22.2.3 Configuring a static prefix for each client*, configuration DHCP mode is entered to allow the configuration of static prefixes.
ipv6 dhcp server	Sets the distribution of prefixes. *22.2.5 Configuring priorities for distributing prefixes to clients* provides a method for assigning priority levels to an IPv6 DHCP address pool for the distribution of prefixes in the pool in order of priority.
ipv6 dhcp static-route-setting	Automatically adds routing information to a client in a routing information table maintained on the Switch. A prefix is distributed to this client by the IPv6 DHCP server. For details, see *22.2.6 Configuring the automatic generation of routes to clients to which prefixes are distributed*.
ipv6 local pool	Sets a prefix to be assigned dynamically. For details, see *22.2.4 Configuring a range of dynamically distributed prefixes*.
prefix-delegation	Sets a static IPv6 prefix, and the IAID and lifetime associated with the prefix to be stored in the specified IPv6 DHCP address pool. For details, see *22.2.3 Configuring a static prefix for each client*.
prefix-delegation pool	Sets the IAID and lifetime for the range of IPv6 prefixes specified in IPv6 DHCP address local pool setting. For details, see *22.2.4 Configuring a range of dynamically distributed prefixes*.
service ipv6 dhcp	Enables or disables an IPv6 DHCP server.
sip-domain-name	Sets the SIP domain name provided by an IPv6 DHCP server. The DHCP server distributes the SIP domain name when it receives a request from an IPv6 DHCP client.
sip-server	Sets the IPv6 address of the SIP server provided by an IPv6 DHCP server. The DHCP server distributes the IPv6 address of the SIP server when it receives a request from an IPv6 DHCP client.
sntp-server	Sets the address of the SNTP server provided by the IPv6 DHCP server. The SNTP server address can be distributed to IPv6 DHCP clients upon request.

22.2.2 Sequence for configuring an IPv6 DHCP server

(1) Configuration for distributing static prefixes to clients

1. Use the `swrt_table_resource` command to enable IPv6.
2. Use the `interface` command to configure a VLAN interface.
3. Use the `ipv6 address` command to set an IPv6 address for the VLAN interface.
4. Use the `ipv6 enable` command to enable IPv6 on the VLAN interface and automatically

generate an IPv6 address for the VLAN interface.

5. Configure a static prefix to each client.
6. Configure the distribution of prefixes to clients.
7. Configure the automatic generation of routes to clients to which prefixes are distributed.

(2) Configuration for distributing dynamic prefixes to clients

1. Use the `swrt_table_resource` command to enable IPv6.
2. Use the `interface` command to configure a VLAN interface.
3. Use the `ipv6 address` command to set an IPv6 address for the VLAN interface.
4. Use the `ipv6 enable` command to enable IPv6 on the VLAN interface and automatically generate an IPv6 address for the VLAN interface.
5. Configure a range for dynamically distributing prefixes.
6. Configure the distribution of prefixes to clients.
7. Configure the automatic generation of routes to clients to which prefixes are distributed.

(3) Configuration for distributing only options to clients

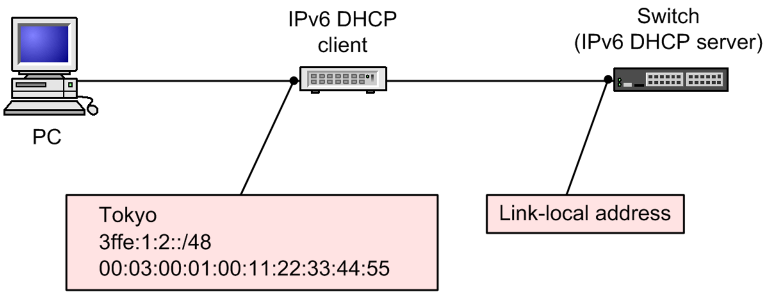
1. Use the `swrt_table_resource` command to enable IPv6.
2. Use the `interface` command to configure a VLAN interface.
3. Use the `ipv6 address` command to set an IPv6 address for the VLAN interface.
4. Use the `ipv6 enable` command to enable IPv6 on the VLAN interface and automatically generate an IPv6 address for the VLAN interface.
5. Configure the distribution of options to clients.

22.2.3 Configuring a static prefix for each client

Points to note

The example below shows how to configure an IPv6 DHCP address pool, and set prefixes and client IDs (DUIDs) in configuration DHCP mode.

Figure 22-1: Configuration for distributing a static prefix to a client



Command examples

1. `(config)# ipv6 dhcp pool Group1`

   Configures an IPv6 DHCP address pool. Switches to configuration DHCP mode.

2.
```
(config-dhcp)# prefix-delegation 3ffe:1:2::/48
00:03:00:01:00:11:22:33:44:55
(config-dhcp)# exit
```

Sets a prefix and a client ID (DUID).

If you want to distribute static prefixes to multiple clients, repeat this step.

3.
```
(config)# interface vlan 10
(config-if)# ipv6 dhcp server Group1
(config-if)# exit
```

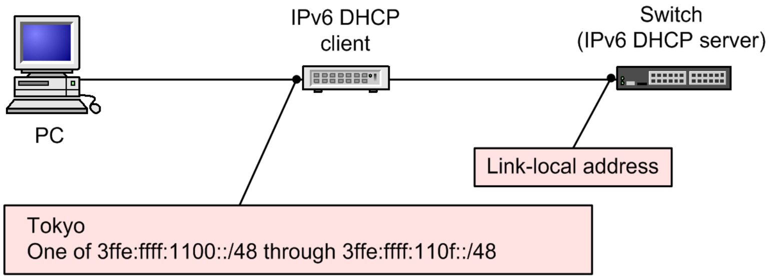
Sets the name of the IPv6 DHCP address pool for the VLAN interface.

22.2.4 Configuring a range of dynamically distributed prefixes

Points to note

After you configure an IPv6 DHCP address pool, configure an IPv6 DHCP address local pool, from which prefixes are dynamically assigned to clients, and in configuration DHCP mode, specify a range of prefixes to be dynamically distributed.

Figure 22-2: Configuration for dynamically distributing prefixes


IPv6 DHCP client
Switch (IPv6 DHCP server)
PC
Link-local address
Tokyo
One of 3ffe:ffff:1100::/48 through 3ffe:ffff:110f::/48

Command examples

1.
```
(config)# ipv6 local pool Group1Local 3ffe:ffff:1100::/44 48
```

Sets the prefix to be dynamically distributed.

2.
```
(config)# ipv6 dhcp pool Group1
```

Configures an IPv6 DHCP address pool.

3.
```
(config-dhcp)# prefix-delegation pool Group1Local
(config-dhcp)# exit
```

Sets the name of the IPv6 DHCP address local pool set in the IPv6 DHCP address local pool configuration.

4.
```
(config)# interface vlan 10
(config-if)# ipv6 dhcp server Group1
(config-if)# exit
```

Sets the name of the IPv6 DHCP address pool for the VLAN interface.

22.2.5 Configuring priorities for distributing prefixes to clients

Points to note

The example below shows how to set the priority of servers that distribute prefixes.

Command examples

1.
```
(config)# ipv6 local pool Group1Local 3ffe:ffff:1100::/44 48
(config)# ipv6 dhcp pool Group1
(config-dhcp)# prefix-delegation pool Group1Local
(config-dhcp)# exit
```
Configures an IPv6 DHCP address pool and sets the prefix to be dynamically distributed.

2.
```
(config)# interface vlan 10
(config-if)# ipv6 dhcp server Group1 preference 255
```
Sets 255 as the priority level for the IPv6 DHCP address pool from which prefixes are distributed in order of priority.

22.2.6 Configuring the automatic generation of routes to clients to which prefixes are distributed

Points to note

The example below shows how to configure the automatic generation of routes to clients to which prefixes are distributed.

Command examples

1.
```
(config)# ipv6 local pool Group1Local 3ffe:ffff:1100::/44 48
(config)# ipv6 dhcp pool Group1
(config-dhcp)# prefix-delegation pool Group1Local
(config-dhcp)# exit
```
Configures an IPv6 DHCP address pool and sets the prefix to be dynamically distributed.

2.
```
(config)# interface vlan 10
(config-if)# ipv6 dhcp server Group1
(config-if)# exit
```
Sets the name of the IPv6 DHCP address pool for the VLAN interface.

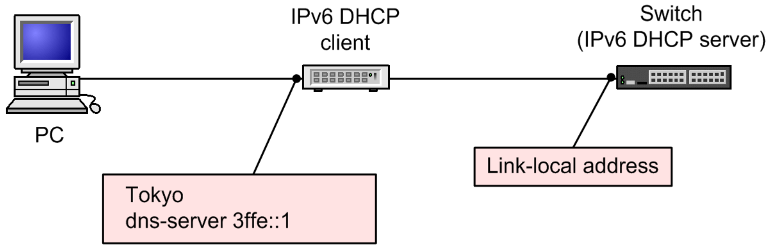
3.
```
(config)# ipv6 dhcp static-route-setting
```
Enables the automatic generation of routes from the DHCP servers to clients to which prefixes are distributed on the Switch. Note that the routes are not configured until the prefixes have been completely distributed.

22.2.7 Configuring the distribution of only options to clients

Points to note

The example below shows how to configure the distribution of options, such as the DNS server option, to clients that do not require prefixes.

Figure 22-3: Configuration for distributing options to clients



Command examples

1.
```
(config)# ipv6 local pool Group1Local 3ffe:ffff:1100::/44 48
```
Sets the prefix to be dynamically distributed.

2.
```
(config)# ipv6 dhcp pool Group1
```
Configures an IPv6 DHCP address pool.

Switches to configuration DHCP mode.

3.
```
(config-dhcp)# prefix-delegation pool Group1Local
```
Sets the name of the IPv6 DHCP address local pool set in the IPv6 DHCP address local pool configuration.

4.
```
(config-dhcp)# dns-server 3ffe::1
(config-dhcp)# exit
```
Configures the DNS server option.

5.
```
(config)# interface vlan 10
(config-if)# ipv6 dhcp server Group1
(config-if)# exit
```
Sets the name of the IPv6 DHCP address pool for the VLAN interface.

22.3 Operation

22.3.1 List of operation commands

The following table describes the operation commands for the IPv6 DHCP server.

Table 22-5: List of operation commands

Command name	Description
show ipv6 dhcp binding	Shows the binding information on the IPv6 DHCP server.
clear ipv6 dhcp binding	Deletes the binding information on the IPv6 DHCP server. Note that the IPv6 DHCP clients that were using the deleted prefix are no longer able to communicate after this operation.
show ipv6 dhcp server statistics	Shows statistics about the IPv6 DHCP server.
clear ipv6 dhcp server statistics	Resets statistics on the IPv6 DHCP server.
restart ipv6-dhcp server	Restarts the IPv6 DHCP server daemon process.
dump protocols ipv6-dhcp server	Outputs the server logs and packet transmission and reception logs recorded by an IPv6 DHCP server to a file.
ipv6-dhcp server monitor	Starts recording transmission and reception logs for the packets that are sent from or received by an IPv6 DHCP server.
no ipv6-dhcp server monitor	Stops obtaining the transmission and reception logs of the packets that are sent and received by an IPv6 DHCP server.
set ipv6-dhcp server duid	Creates a DUID file for an IPv6 DHCP server
show ipv6-dhcp server duid	Shows the contents of a DUID file maintained by an IPv6 DHCP server.
erase ipv6-dhcp server duid	Deletes a DUID file maintained by an IPv6 DHCP server.

22.3.2 Checking the number of prefixes that can be assigned

To check the number of prefixes that can be assigned to clients, execute the `show ipv6 dhcp server statistics` command to display a list and check the `prefix pools` section. Make sure that the displayed number is greater than the number of clients to which you want to distribute prefixes.

Figure 22-4: Results of executing the show ipv6 dhcp server statistics command

```
> show ipv6 dhcp server statistics
Date 20XX/10/15 12:00:00 UTC
   < DHCP Server use statistics >
     prefix pools            :20
     automatic prefixes      :50
     manual prefixes         :4
     expired prefixes        :3
     over pools requests     :0
     discard packets         :0
   < Receive Packets >
     SOLICIT                 :54
     REQUEST                 :54
     RENEW                   :54
     REBIND                  :0
     INFORMATION-REQUEST     :0
     CONFIRM                 :0
     RELEASE                 :0
     DECLINE                 :0
     RELAY-FORW              :0
   < Send Packets >
     ADVERTISE               :54
     REPLY                   :108
     RELAY-REPL              :0
   < Server DUID >
     00:01:00:01:3e:00:2e:22:11:22:33:44:55:01
>
```

22.3.3 Checking the distributed prefixes

To check the prefixes that are actually distributed, execute the `show ipv6 dhcp binding` command. The command displays the prefixes whose lease has not expired yet.

Figure 22-5: Results of executing the show ipv6 dhcp binding command

```
> show ipv6 dhcp binding
Date 20XX/10/15 12:00:00 UTC
Total: 2 prefixes
<Prefix>                 <Lease expiration>   <Type>
3ffe:1:2::/48            XX/10/16 11:15:00    Manual
3ffe:ffff:1101::/48      XX/10/16 11:29:00    Automatic
>
```

Chapter
23. IPv6 Routing Protocol Overview

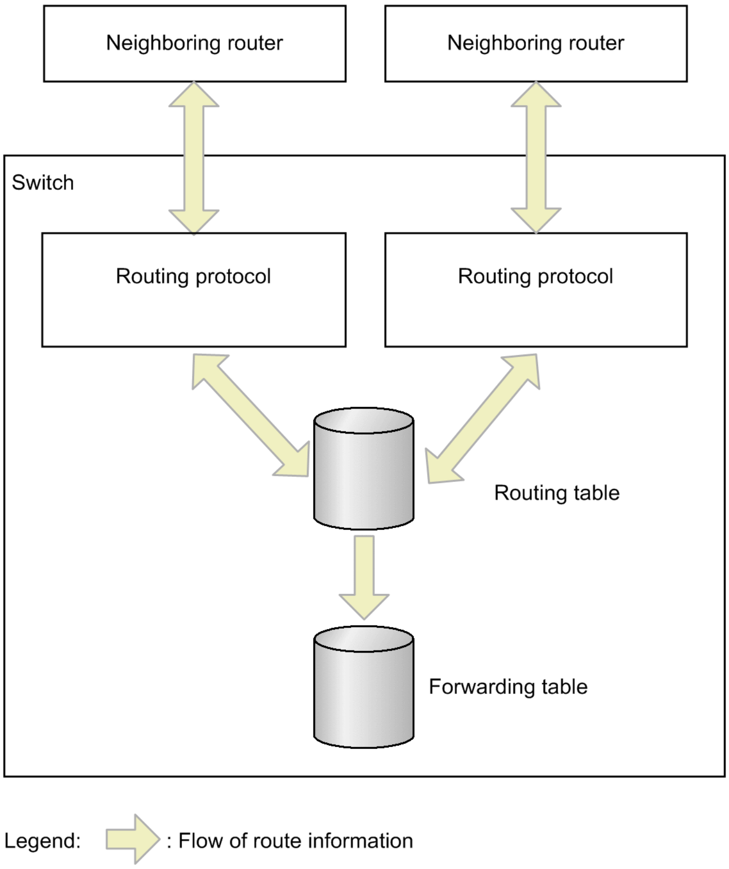
This chapter provides an overview of IPv6 routing protocols.

23.1 Description of IPv6 routing

23.1.1 Overview of routing

Routing protocols define how routers exchange routing information with each other. A router stores the routing information it learns from each routing protocol in a routing table, and then registers preferred routes for packet forwarding in a forwarding table. Packet relay takes place based on the contents of the forwarding table.

Figure 23-1: Routing overview



23.1.2 Static routing and dynamic routing

A switch must create a routing table before it can relay packets. The switch uses static routing and dynamic routing to create routing tables.

- Static routing

  A user sets routing information manually using configuration commands.

- Dynamic routing

  The Switch determines how to relay packets based on routing information it receives from other routers in the network. The Switch supports the RIPng, OSPFv3, and BGP4+ protocols.

23.1.3 Routing information

The table below describes the routing information handled by the Switch (the address types subject to routing). The Switch handles a site-local address the same way as a global address.

Table 23-1: Routing information

Type of routing information		Description
Standard routes	Default route	A route that matches every network destination (destination prefix: ::/0)
	Global route whose prefix length is 1 to 127 bits	A route that summarizes a global route to a specific network and global routes to multiple networks
	Host route	A route to a specific host (global route whose prefix length is 128 bits).
Route not subject to routing	Link local route	(Prefix: fe80::% <*Interface name*> /64)
	Multicast address	(Prefix: ff00::/8)
	IPv4 reserved address	(Prefix: ::/8)

Legend: --: Not applicable

23.1.4 Scope of individual routing protocols

The following table provides an overview of the routing information and functionality offered by the Switch for each supported routing protocol.

Table 23-2: Scope of individual routing protocols

Routing information		Static	Dynamic		
			RIPng	OSPFv3	BGP4+
Routing informat ion	Default route	Y	Y	Y	Y
	Global route	Y	Y	Y	Y
	Host route	Y	Y	Y	Y
	Multipath	Y	N	Y	Y
Route selection		--	Metric (hop count)	Cost (hop count and line speed)	AS path attribute
Routing loop prevention		--	Split horizon	Y	Y
Authentication functionality		--	N	N	Y

Legend: Y: Supported, N: Not supported, --: Not applicable

23.1.5 Concurrent use of routing protocols

You can implement various static and dynamic routing protocols concurrently on the Switch.

(1) Determining priority for learned routes

In an environment running more than one routing protocol concurrently, each protocol uses its own route selection algorithm to select the best route to a given destination. Both summarized routes and directly connected routes are treated as a protocol route as with routes learned by routing protocols. This can result in the Switch learning multiple different routes to the same destination. In this case, the switch compares the distance of each route and applies the routing information with the highest priority.

In the Switch, you can use configuration commands to set the default distance (priority) that applies to each static route and to each piece of routing information generated by dynamic routing protocols (such as RIPng). A route with a smaller distance has a higher priority. The following table describes the default distances for each protocol.

Table 23-3: Default distances

Route	Default distance
Directly connected route	0 (fixed value)
Static route	2
BGP4+ route learned from external peer	20
OSPFv3 (internal AS route)	110
OSPFv3 (external AS route)	110
RIPng route	120
Summarized route	130
BGP4+ route learned from internal peer	200
BGP4+ route learned from inter-AS peer	200
Route imported from another VRF or the global network	210

(2) Route advertisement

In an environment running more than one routing protocol, each routing protocol only advertises the routing information learned by that protocol. It does not advertise routing information it learns from other routing protocols.

In the Switch, you can use route filtering to configure a routing protocol to advertise routing information it learns from other routing protocols, or to exclude specific routes from advertisement. Information of inactive routes cannot be advertised in other routing protocols.

For details, see *29. Route Filtering (IPv6)*.

(a) RIPng route advertisement

RIPng operates as one routing protocol.

(b) OSPFv3 route advertisement

The OSPFv3 routing protocol operates independently in each OSPFv3 domain. For this reason, a number of internal and external AS routes taken from different OSPFv3 domains might exist for a given destination address. When OSPFv3 routes share the same distance, the route with the smaller domain number takes priority. You can change the default distances for internal and external AS OSPFv3 routes (intra-area and inter-area routes).

Under normal circumstances, routes are not advertised between the various OSPFv3 domains configured on the Switch. However, you can configure route filtering to enable advertisement of internal and external OSPFv3 routes to other OSPFv3 domains.

(c) BGP4+ route advertisement

When route filtering is disabled, the switch advertises the BGP4 routes it learns from a given AS to other ASs. In this case, the switch advertises the best BGP4+ route as selected by the BGP4+ routing protocol even if another routing protocol defines a route to the same destination.

If route filtering is enabled, the switch advertises the routing information that represents the route with the highest priority, selected according to its distance.

23.1.6 Considerations when setting or changing routing protocol configurations

When you set or change the configuration of a unicast routing protocol, the protocol re-evaluates every one of its routes according to the new configuration. During this re-evaluation process, operation commands that apply to the unicast routing protocol might take a long time to execute, and MIB information might take a long time to gather via SNMP.

23.2 General IPv6 routing operations

23.2.1 List of operation commands

The following table describes the operation commands for IPv6 routing.

Table 23-4: List of operation commands

Command name	Description
show ipv6 route	Shows routing information stored in the routing table.
clear ipv6 route	Clears the IPv6 forwarding entries stored in the Switch and re-registers the routing entries.
show ipv6 interface ipv6-unicast	Shows information about the IPv6 interfaces on the Switch recognized by the unicast routing program.
debug ipv6	Shows the packets being routed by IPv6 routing protocols in real time.
show system#1	Shows operating status.
show processes cpu unicast#2	Shows the CPU usage of a unicast routing program.
restart unicast#2	Restarts the unicast routing program.
debug protocols unicast#2	Starts the operation message display for event log information output by a unicast routing program.
no debug protocols unicast#2	Stops the operation message display for event log information output by a unicast routing program.
dump protocols unicast#2	Outputs the trace information and control table information collected by the unicast routing program to a file.
erase protocol-dump unicast#2	Deletes the file of trace information and control table information generated by the unicast routing program.
show ipv6 interface#3	Shows the status of the IPv6 interface.
show netstat (netstat) (IPv6)#3	Shows the network status and statistics.
ping ipv6#3	Tests connectivity by sending a test packet to an IPv6 address associated with a specific device.
traceroute ipv6#3	Shows the route an IPv6 datagram travels to reach a destination host.

#1

See *9. Checking Software Versions and Device Statuses* in the manual *Operation Command Reference Vol.1 For Version 11.10.*

#2

See *8. Routing Protocol Common to IPv4 and IPv6* in the manual *Operation Command Reference Vol. 2 For Version 11.10.*

#3

See *9. IPv6, NDP, and ICMPv6* in the manual *Operation Command Reference Vol. 2 For Version 11.10.*

23.2.2 Checking routes to destination addresses

If you configure IPv6 unicast routing information in the Switch, use the `show ipv6 route` command to check whether a route to a specific destination address exists.

Figure 23-2: Results of executing the show ipv6 route command

```
> show ipv6 route
Date 20XX/07/14 12:00:00 UTC
Total: 11 routes
Destination                                 Next Hop
      Interface      Metric   Protocol  Age
4000:110:1:1::/64                           4000:110:1:1::1
      VLAN0010       0/0      Connected 22m 53s
cafe:1001::/64                              4000:110:1:1::200         ...1
      VLAN0010       0/0      Static    41s
      :
      :
>
```

1. Check whether the command output contains a route corresponding to the destination address.

23.3 Network design considerations

This section describes considerations for the design of an IPv6 network.

23.3.1 Address design

When you perform IPv6 addressing, you can achieve a relatively simple network design and avoid associated pitfalls by keeping the following in mind:

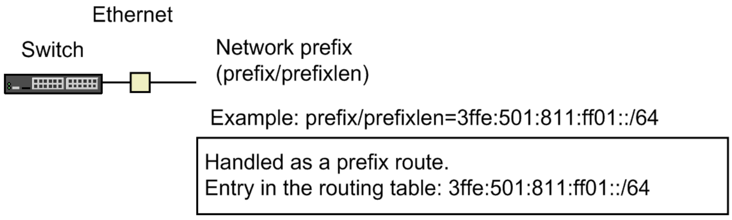
- Divide an NLA and SLA according to the hierarchical configuration of the network topology.

23.3.2 Handling of direct connections

The Switch uses a broadcast interface.

The Switch uses a combination of network prefix (`prefix`) and prefix length (`prefixlen`) for the broadcast interface. The following figure illustrates the handling of broadcast-type direct connections.

Figure 23-3: Handling direct connections (broadcast type)


Ethernet
Switch
Network prefix
(prefix/prefixlen)
Example: prefix/prefixlen=3ffe:501:811:ff01::/64
Handled as a prefix route.
Entry in the routing table: 3ffe:501:811:ff01::/64

23.4 Description of load balancing

23.4.1 Overview of load balancing

Load balancing is the practice of distributing increased traffic loads among multipaths to the same destination. For details about load balancing, see *7.4.1 Overview of load balancing*.

23.4.2 Load balancing specifications

The following table describes the specifications for multipath routing on the Switch.

Table 23-5: IPv6 multipath specifications

Item	Specifications	Remarks
Number of multipaths for a single destination network	2 to 16	--
Maximum number of multipaths that can be specified in the configuration	1 to 16 (Multipaths are not generated if 1 is specified.)	The maximum number of multipaths is specified for each routing protocol.
Maximum number of multipath routes	For AX3800S series switches: 256, 512, or 1024 For AX3650S series switches: 128, 256, 512, or 1024	The value differs depending on the maximum number of multipaths handled by the switch. For details, see *Table 23-6: Maximum number of multipath routes*.
Routing protocols capable of generating multipaths	• Static (IPv6) • OSPFv3 • BGP4+	--
Number of multipaths in a default configuration	• Static (IPv6): 6 • OSPFv3: 4 • BGP4+: 1 (multipath routes are not generated)	--
Prefix length of the routes used	0 to 127	If the prefix length of the route is 128, multiple paths (next hop data) are registered with the forwarding table as a single path. The next hop address used at this point is the same as the address displayed at top when the `show ipv6 route` operation command is executed.
Connection method	Can be used with any combination of line type and interface type. Concurrent use of different types of lines or interfaces is possible.	Multipaths between VRFs are not supported.

Legend: --: Not applicable

Table 23-6: Maximum number of multipath routes

Model	Maximum number of multipaths specified in switch configuration#1	Maximum number of multipaths handled by the switch#2	Maximum number of multipath routes the switch can handle#2, #3
AX3800 S	1 to 4	4	1024#4
	5 to 8	8	512
	9 to 16, or multipath disabled#5	16	256
AX3650 S	1 to 2	2	1024#4
	3 to 4	4	512
	5 to 8	8	256
	9 to 16, or multipath disabled#5	16	128

#1

The maximum number of multipaths specified for static routing (IPv4/IPv6), OSPF/OSPFv3, or BGP4/BGP4+, whichever is largest. For example, if the device configuration specified a maximum number of multipaths of 6 for static routing and 3 for OSPFv3, the larger value of 6 applies.

Each routing protocol can generate a number of multipaths no greater than the maximum number defined for that protocol in the switch configuration. If a change is made to the maximum number of multipaths that affects how many multipaths the switch can handle, the changes will not take effect until the switch is restarted.

#2

The maximum value is determined at startup. If you change the maximum number of multipaths for a unicast protocol after the switch starts, it remains unchanged from the value determined at startup. To change the maximum multipath count, restart the switch after making the appropriate changes in the device configuration.

#3

The maximum number of multipath routes applies to the combined total of IPv4 and IPv6 routes.

#4

For a single path, the maximum number of paths is determined by the capacity limit for the number of table entries. For multipath, the values in the table apply.

#5

When not using static routing (IPv4/IPv6), OSPF/OSPFv3, or BGP4/BGP4+, the switch does not handle multipath routes. In this case, the maximum values in the table apply.

Using the static routing setting for AX3650S switches as an example, the following table describes how maximum multipath values change when you change the device configuration or restart the device.

Table 23-7: Changes in maximum multipath values (for static routing)

Seq.	Status	Maximum number of multipaths for static routing	Maximum number of multipaths handled by the switch	Maximum number of multipath routes the switch can handle
1	The switch starts without static routing configured.	--	16	128
2	You add a static route.	6[#1]	16	128
3	The switch restarts.	6	8	256
4	You set the maximum multipath count for static routing to 3.	3	8	256
5	The switch restarts.	3	4	512
6	You set the maximum multipath count for static routing to 5.	4[#2]	4	512
7	The switch restarts.	5[#2]	8	256

Legend: --: Not applicable

#1

If you configure static routes without specifying a maximum multipath value, the default value applies to the number of static multipaths. For details, see *23.4.3 Notes on using load balancing*.

#2

You cannot generate statically routed multipaths that exceed the maximum number of multipaths the switch can handle. However, after you restart the switch, the maximum number of multipaths handled by the switch changes, and the value specified for static routing multipaths takes effect.

The following table describes the load balancing specifications implemented by the Switch.

Table 23-8: IPv6 load balancing specifications

Item	Specifications	Remarks
Path selection in multipaths	A value (hash value) is calculated and allocated to the designated output path. This hash value is derived from the following four fields: • Source IPv6 address • Destination IPv6 address • Source TCP/UDP port number • Destination TCP/UDP port number For packets with the same hash value, the same output path is selected. This guarantees the transmission order.	--
Multipath information in routing table	Hash values are assigned on a substantially equal basis to the output interfaces in the routing table.	See *23.4.3 Notes on using load balancing*.
Weighting of paths	Not available	

Item	Specifications	Remarks
Processing of packets exceeding output bandwidth	Not reallocated to another path. The packets are retained within the switch as long as bandwidth continues to be exceeded. However, any overflow will be discarded.	

Legend: --: Not applicable

23.4.3 Notes on using load balancing

1. The switch selects one of 16 paths based on the packet's hash value. For this reason, packets are not necessarily distributed equally among the paths to a given destination network.
2. Because the switch applies no weighting to individual paths, line speed is not taken into account when distributing packets. However, you can give greater weighting to faster lines by using a multihomed connection. In this case, you must use a redundant configuration to guard against failures.
3. If the switch attempts to send packets in a manner that continuously exceeds the bandwidth of a path selected on the basis of hash values, these packets will be discarded without being re-allocated to another path.
4. Note the following when using the `traceroute` (IPv6) command to check the paths selected for load balancing:
   - Although the response to the `traceroute` (IPv6) command bears the IPv6 address of the interface that received the command in its source field, the response does not necessarily originate from that interface.
   - If the `traceroute` (IPv6) command is received by a multihomed interface, it cannot determine the address to which the neighboring device sent the command. The interface responds by using one of the addresses in the multihomed environment as the source IPv6 address.
5. When using load balancing, relay performance may decline dramatically if traffic is concentrated on a specific route (gateway). In such a case, configure a static NDP entry for all gateways.
6. When multipaths for a BGP4+ route include a null interface as a result of next hop resolution performed on the basis of an IGP route that specifies a null interface, the switch is unable to use this route to relay packets. In such a case, use the BGP configuration command `bgp nexthop` to ensure that the IGP route that specifies the null interface is not used for next hop resolution of the BGP4+ route.

   If a next hop to which the Switch is not directly connected is included in the multipath static route, and that next hop is resolved in a route in which a null interface is used as a next hop, that route cannot be used for forwarding.
7. If you configure a unicast routing protocol without specifying a maximum number of multipaths, the following default values apply:
   - Static (IPv6): 6
   - OSPFv3: 4
   - BGP4+:1 (does not generate multipaths)
8. You cannot change the maximum number of multipath routes after the Switch has begun operation. To change this setting, modify the maximum number of multipaths in the configuration for each unicast routing protocol (static routing, OSPFv3, BGP4+), and then restart the switch.

9. Although you can configure a static route in which multiple VRFs exist in the next hop, the multipath of the created route consists of only one VRF.

   The path is selected from among the next hops in the same VRF as the next hop with the highest weight value. **[OS-L3SA]**

23.5 Load balancing configuration

23.5.1 List of configuration commands

The following table describes the configuration commands for load balancing.

Table 23-9: List of configuration commands

Command name	Description
maximum-paths (BGP4+)[#1]	Generates multipaths using the specified value as the maximum number of paths when multiple routing information entries of equal cost to a given destination exist.
ipv6 route static maximum-paths[#2]	Specifies the maximum number of paths (maximum number of next hops) that the routing protocol will generate for a static IPv6 route.
maximum-paths (OSPFv3)[#3]	Specifies the maximum number of paths in the route when multipaths (next hop) of equal cost exist for an OSPFv3-generated route.

#1

See *13. BGP4 [OS-L3SA]* in the manual *Configuration Command Reference Vol. 2 For Version 11.10.*

#2

See *24. Static Routing (IPv6)* in the manual *Configuration Command Reference Vol. 2 For Version 11.10.*

#3

See *26. OSPFv3 [OS-L3SA]* in the manual *Configuration Command Reference Vol. 2 For Version 11.10.*

23.5.2 Configuring the maximum number of multipaths handled by the Switch

The maximum number of multipaths used for each protocol in the Switch determines the maximum number of multipaths that the Switch handles and the maximum number of multipath routes that the Switch can handle.

The maximum number of multipaths is determined at startup, and any changes made by configuration commands do not take effect until the switch is restarted. If you use a configuration command in a manner that affects the maximum number of multipaths, a warning-level operation message appears prompting you to restart the switch. The new values for the maximum number of multipaths and multipath routes will take effect after the Switch restarts.

Points to note

By default, the switch handles a maximum of 16 multipaths. For multipath routes, AX3800S series switches handle a maximum of 256 and AX3650S series switches handle a maximum of 128. To change the maximum number of multipaths that the switch handles, you must restart the Switch after setting the maximum number of multipaths in the configuration of a unicast routing protocol. For this reason, we recommend that you set the maximum number of multipaths when you first deploy the switch.

The configuration below uses an example of IPv6 static routing.

Command examples

1.
```
(config)# ipv6 route static maximum-paths 2
```

In configuration mode, sets 2 as the maximum number of multipaths for IPv6 static routes.

2.
```
(config)# ipv6 route 2001:db8:ffff:2::/64 2001:db8:2:1::2
noresolve
(config)# ipv6 route 2001:db8:ffff:2::/64 2001:db8:2:2::2
noresolve
```
In configuration mode, establishes a static IPv6 multipath route (2001:db8:ffff:2::/64).

3.
```
(config)# save
(config)# exit
```
Saves the settings and switches from configuration mode to administrator mode.

4.
```
# reload
```
Restarts the Switch.

23.5.3 Load balancing using static routes

See *24.2.4 Configuring a multipath route*.

23.5.4 Load balancing using OSPFv3 [OS-L3SA]

See *26.2.6 Configuring multipath*.

23.5.5 Load balancing using BGP4+ [OS-L3SA]

See *28.5.3 Configuring BGP4+ multipath*.

23.6 Load balancing operation

23.6.1 Checking the maximum number of multipaths handled by the Switch

Use the `show system` command to check the maximum number of multipaths handled by the Switch.

Figure 23-4: Checking the maximum number of multipaths handled by the Switch

```
>show system
           :
           :
    Device resources
        Current selected swrt_table_resource: l3switch-2
        Current selected swrt_multicast_table: On
        Current selected unicast multipath number: 8
           :
           :
>
```

23.6.2 Checking selected paths

(1) Checking the routing information

Execute the `show ipv6 route` command to see whether the multipath route settings have been applied correctly.

Figure 23-5: Displaying multipath routing information

```
> show ipv6 route
Date 20XX/07/14 12:00:00 UTC
Total: 11 routes
Destination                                 Next Hop
      Interface      Metric   Protocol  Age
4000:110:1:1::/64                           4000:110:1:1::1
      VLAN0010       0/0      Connected 22m 53s
4000:110:1:1::1/128                         ::1
      localhost      0/0      Connected 22m 53s
4000:120:1:1::/64                           4000:120:1:1::1
      VLAN0020       0/0      Connected 22m 53s
4000:120:1:1::1/128                         ::1
      localhost      0/0      Connected 22m 53s
4000:130:1:1::/64                           4000:130:1:1::1
      VLAN0030       0/0      Connected 22m 53s
4000:130:1:1::1/128                         ::1
      localhost      0/0      Connected 22m 53s
4000:210:1:1::/64                           4000:110:1:1::200
      VLAN0010       0/0      Static     6s
                                            4000:120:1:1::200
      VLAN0020        -        -         -
                                            4000:130:1:1::200
      VLAN0030        -        -         -
      :
      :
>
```

(2) Checking reachability of specific destination addresses

Use the `ping ipv6` <*IPv6 Address*> `specific-route source` <*Source Address*> command at each interface of the Switch used for load balancing to confirm that the interface can communicate with the remote system. As <*Source Address*> of the `ping ipv6` command, specify the local IPv6 address on the Switch of the interface performing load balancing.

23.7 Description of route summarization

23.7.1 Overview

Route summarization is a method of representing routing information from one or more source routes by generating routing information that is shorter than the network masks that include information about the corresponding paths. If the information for multiple routes is condensed into one set of routing information containing the information for the multiple routes, and then that summarized information is reported to (for example) neighboring routers, the number of items of routing information on the network is reduced. For example, the switch can generate the summarized route 2001:db8:1:ff00::/56 after learning the routing information for addresses 2001:db8:1:ff01::/64 and 2001:db8:1:ff02::/64.

You must explicitly enable route summarization by executing the `ipv6 summary-address` configuration command. You can assign a distance to a summarized route. If you do not specify a distance, the default of 130 is used. The switch cannot generate a summarized route without first learning the routing information that serves as its source.

23.7.2 Transferring summarized routes

Summarized routes are configured as reject routes. Packets that lack a higher-priority route are discarded.

Summarized routes are configured in this way to prevent routing loops. When the Switch advertises a summarized route, packets destined for that route are relayed to the Switch. If the Switch then transfers a packet destined to a route that is not a summarization source route via the designated next-best route (such as the default route), a routing loop can occur between the Switch and the destination system of the default route. Summarized routes are designated as reject routes to prevent this scenario.

However, summarized routes configured with the `noinstall` parameter specified do not discard packets. If there is a default route or other next-best route, packets will be forwarded via that route. Use the `noinstall` parameter when you wish to advertise a summarized route but would rather it forward packets via the next-best route than discard them.

23.7.3 Summarizing AS_PATH attributes

When a summarized route includes a BGP4+ route among its source routes, the summarized route is tagged with the path attribute of the BGP4+ route. If the source routes include more than one BGP4+ route, the path attributes are summarized among those routes. The `AS_PATH` and `COMMUNITIES` attributes of summarized routes are edited as follows:

(1) AS_PATH attribute

The initial AS path segment within the `AS_SEQUENCE` type of the `AS_PATH` attribute which is common to the summarization source routes is assigned as the `AS_SEQUENCE` type of the `AS_PATH` attribute of the summarized route. All other AS paths in the `AS_SEQUENCE` type and those outside the `AS_SEQUENCE` type appear in the `AS_SET` type of the `AS_PATH` attribute of the summarized route only if you execute the `ipv6 summary-address` configuration command with the `as_set` parameter specified.

(2) COMMUNITIES attribute

Every community in the summarization source BGP4+ routes is set to the `COMMUNITIES` attribute of the summarized route.

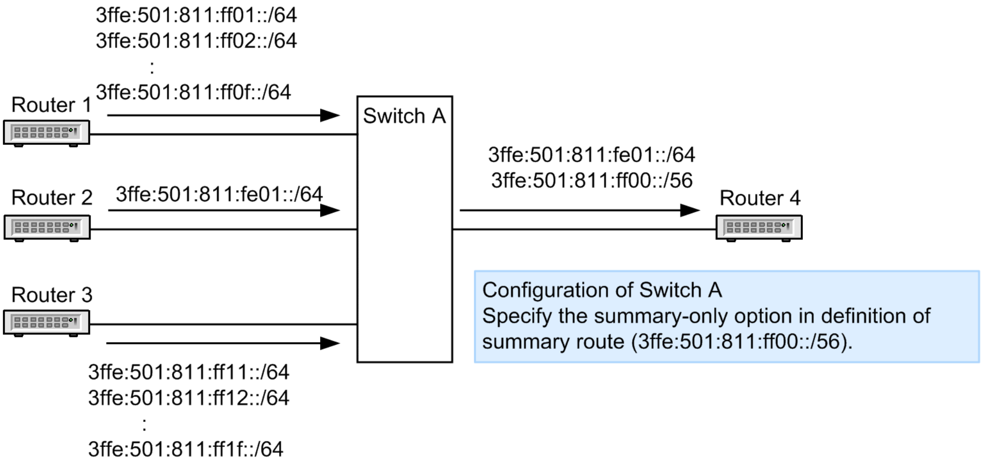
23.7.4 Suppressing advertisement of summarization source route

After you summarize a route, it is possible to advertise the summarized route but exclude its source routes from being advertised. For example, you can advertise the RIPng routes that are not the source nodes of a summarized route, and suppress advertisement of the RIPng routes that are the source nodes of a summarized route.

You can suppress the advertisement of source routes on an individual basis or for all summarized routes. To suppress advertisement for an individual summarized route, specify the `summary-only` parameter when you execute the `ipv6 summary-address` configuration command.

The following figure shows an example in which advertisement is suppressed for a summarized route.

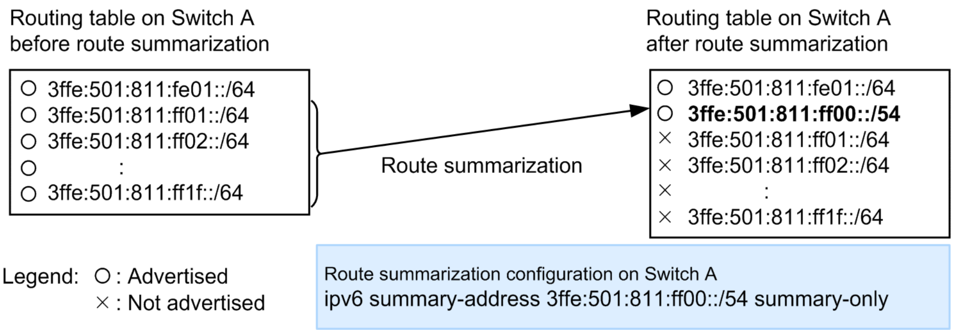
Figure 23-6: Example of suppressing summarization source route advertisement



Configuration of Switch A
Specify the summary-only option in definition of summary route (3ffe:501:811:ff00::/56).

Switch A receives addresses in the range from 3ffe:501:811:ff01::/64 to 3ffe:501:811:ff0f::/64 from Router 1 and 3ffe:501:811:fe01::/64 from Router 2, and learns routes in the range from 3ffe:501:811:ff11::/64 to 3ffe:501:811:ff1f::/64 from Router 3. Switch A then configures the advertised route filter to advertise the summarized route 3ffe:501:811:ff00::/56 and learned route 3ffe:501:811:fe01::/64 to Router 4. At this point, if you specified the `summary-only` parameter when you configured the switch to generate the summarized route 3ffe:501:811:ff00::/56 from learned routes, you do not need to configure the advertised route filter to prevent the switch from advertising the summary source routes. The following figure shows an example configuration for route summarization, and the entries in the routing table before and after summarization.

Figure 23-7: Example of route summarization configuration and routing entries before and after summarization



Route summarization configuration on Switch A
ipv6 summary-address 3ffe:501:811:ff00::/54 summary-only

23.8 Route summarization configuration

23.8.1 List of configuration commands

The following table describes the configuration commands for route summarization.

Table 23-10: List of configuration commands

Command name	Description
ipv6 summary-address	Generates a summarized IPv6 route.
redistribute (BGP4+)#	Sets the protocol types of routes advertised from BGP4+.
redistribute (OSPFv3)#	Sets the protocol types of routes advertised from OSPFv3.
redistribute (RIPng)#	Sets the protocol types of routes advertised from RIPng.

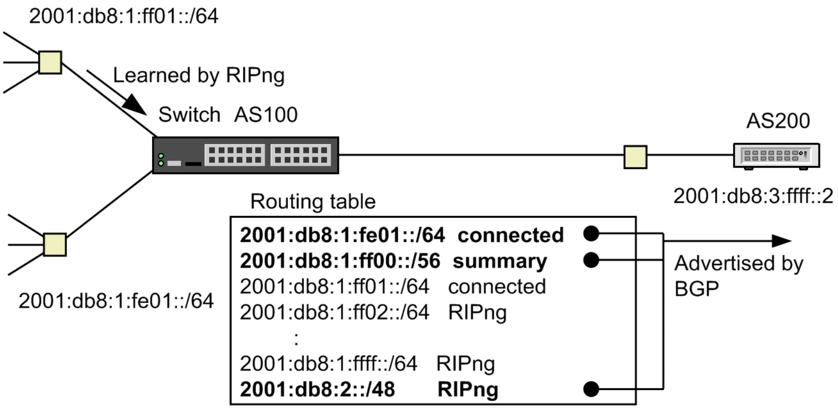
#

See *14. Route Filters (IPv4 and IPv6)* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

23.8.2 Configuring route summarization and advertisement of summarized routes

Configure summarization of directly connected routes and RIPng routes as summarization source routes. Also, configure the switch to re-advertise the summarized routes and directly connected routes by BGP4+ without re-advertising the directly connected routes and RIPng routes that have become summarization source routes.

Figure 23-8: Configuration for advertising summarized routes by BGP4+



2001:db8:1:fe01::/64	**connected**
2001:db8:1:ff00::/56	**summary**
2001:db8:1:ff01::/64	connected
2001:db8:1:ff02::/64	RIPng
:	
2001:db8:1:ffff::/64	RIPng
2001:db8:2::/48	**RIPng**

Points to note

To generate a summarized route, use the `ipv6 summary-address` command. Use the `redistribute summary` command to configure BGP4+ to advertise the summarized route.

Command examples

1.
```
(config)# interface vlan 10
(config-if)# ipv6 address 2001:db8:1:fe01::1/64
```

Sets the IPv6 address 2001:db8:1:fe01::1/64 for interface vlan 10.

2.
```
(config-if)# exit
(config)# interface vlan 20
(config-if)# ipv6 address 2001:db8:1:ff01::1/64
```
Sets the IPv6 address 2001:db8:1:ff01::1/64 for interface vlan 20.

3.
```
(config-if)# ipv6 rip enable
```
Enables RIPng packet transmission and reception on interface vlan 20.

4.
```
(config-if)# exit
(config)# ipv6 summary-address 2001:db8:1:ff00::/56
summary-only
```
Configures the switch to generate the summarized route 2001:db8:1:ff00::/56. By specifying `summary-only`, you suppress re-advertisement of the summarization source routes.

5.
```
(config)# router bgp 100
(config-router-af)# neighbor 2001:db8:3:ffff::2 remote-as 200
```
Establishes a BGP4 connection with neighboring router 2001:db8:3:ffff::2.

6.
```
(config-router)# address-family ipv6
(config-router-af)# redistribute summary
```
Uses BGP4+ to re-advertise the summarized route.

7.
```
(config-router-af)# redistribute connected
```
Uses BGP4+ to re-advertise the directly connected route.

8.
```
(config-router-af)# redistribute rip
```
Uses BGP4+ to re-advertise the RIPng route.

9.
```
(config-router-af)# neighbor 2001:db8:3:ffff::2 activate
```
Enables exchanging routes with the neighboring router 2001:db8:3:ffff::2.

23.9 Route summarization operation

23.9.1 List of operation commands

The following table describes the operation commands for route summarization.

Table 23-11: List of operation commands

Command name	Description
show ipv6 route	Shows routing information stored in the routing table.
show ipv6 rip	Shows information about the RIPng protocol.
show ipv6 ospf	Shows information about the OSPFv3 protocol.
show ipv6 bgp	Shows information about the BGP4+ protocol.
restart unicast#	Restarts the unicast routing program.
dump protocols unicast#	Outputs the trace information and control table information collected by the unicast routing program to a file.
erase protocol-dump unicast#	Deletes the file of trace information and control table information generated by the unicast routing program.

#

See *8. Routing Protocol Common to IPv4 and IPv6* in the manual *Operation Command Reference Vol. 2 For Version 11.10.*

23.9.2 Checking summarized routes

You can display information about summarized routes entered in the routing table. The following figure shows an example of displaying a summarized route.

Figure 23-9: Example of displaying a summarized route

```
> show ipv6 route brief summary_routes
Date 20XX/07/14 12:00:00 UTC
Total: 1 routes
Destination                    Next Hop                                Protocol
2001:db8:1:ff00::/56           ----                                    Summary
```

You can also display information about the active routes in a specific network (2001:db8:1:ff00::/56). The following figure shows an example of displaying active routes.

Figure 23-10: Example of displaying active routes

```
> show ipv6 route brief 2001:db8:1:ff00::/56 longer-prefixes
Date 20XX/07/14 12:00:00 UTC
Total: 256 routes
Destination                    Next Hop                             Protocol
2001:db8:1:ff00::/56           ----                                 Summary
2001:db8:1:ff01::/64           2001:db8:1:ff01::1                   Connected
2001:db8:1:ff02::/64           2001:db8:1:ff01::2                   RIPng
2001:db8:1:ff03::/64           2001:db8:1:ff01::2                   RIPng
        :                             :                              :
2001:db8:1:ffff::/64           2001:db8:1:ff01::2                   RIPng
```

23.10 Route deletion delay functionality

For details about the route deletion delay functionality, see *7.10 Route deletion delay functionality*.

23.11 Description of VRF [OS-L3SA]

Virtual Routing and Forwarding (VRF) is a technology that logically partitions routing space. VRF allows a router to retain multiple instances of routing tables to perform concurrent transmission according to each routing table.

Because VRF separates the IPv6 address space, each VRF instance can use the same IPv6 address more than once. Routing protocols operate independently for each VRF.

23.11.1 Scope of support

The following table describes the IPv6 routing protocol functionality supported in VRFs.

Table 23-12: Functionality supported in VRFs

Functionality		Supported
Static routing		Y
Dynamic routing	RIPng	Y
	OSPFv3	Y
	BGP4+	Y
Multipaths and load balancing		Y#1
Route summarization		Y
Route deletion delay functionality		Y
Graceful restart		Y#2
Limiting the number of routes		Y
Extranet	Route exchange between VRFs	Y
	Static routing across VRFs	Y

Legend: Y: Supported

#1: Multipaths across VRFs are not supported.

#2: Only OSPFv3 helper router and BG4P+ receiving router functionality are supported.

23.11.2 Limiting the number of routes

You can limit the number of routes specified for each VRF.

(1) Suppressing the addition of routes

When the number of routes per VRF exceeds the specified maximum number of routes, the Switch suppresses subsequently learned routes to be added to the forwarding table.

The suppressed routes are retained in the routing table and will be added sequentially when the routes that have been added are deleted, freeing up space in the forwarding table.

(2) Outputting warning messages

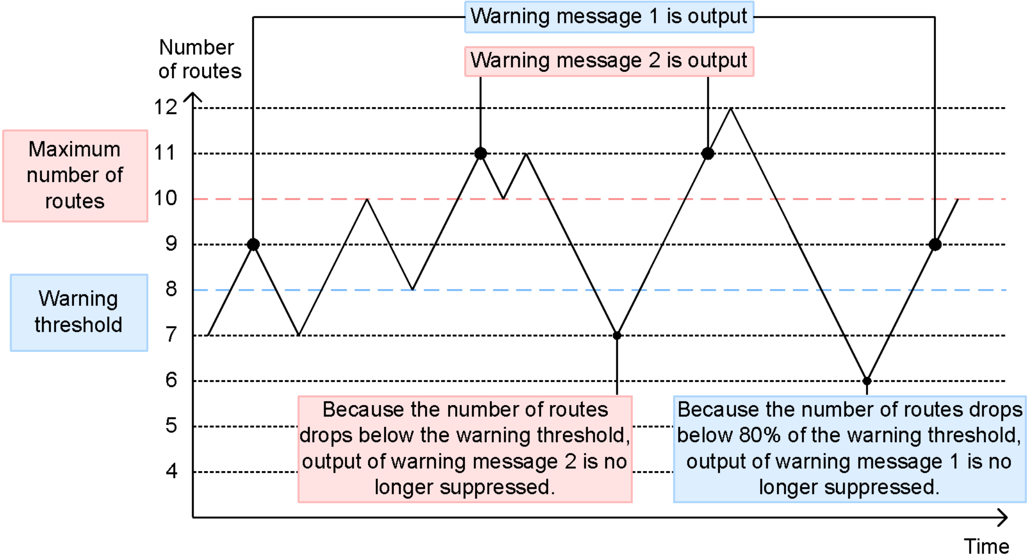
When the number of routes per VRF exceeds the specified warning threshold or the maximum number of routes, a warning message is output.

Additional output of warning message 1, which is output when the warning threshold is exceeded, is suppressed until the number of routes drops below 80% of the warning threshold.

Additional output of warning message 2, which is output when the maximum number of routes is exceeded, is suppressed until the number of routes drops below the warning threshold.

The following figure shows the relationship between the number of routes and the output of warning messages.

Figure 23-11: Relationship between the number of routes and the output of warning messages



(3) Notes

If you change the maximum number of routes in the configuration to a value smaller than the number of routes registered in the forwarding table, the number of routes registered in the forwarding table is not immediately reduced to the new maximum number of routes.

To forcibly reduce the number of routes registered in the forwarding table to the maximum number of routes that you specified, execute the `clear ipv6 route` operation command.

23.11.3 Extranet

The following two methods are available for implementing an extranet:

- Route exchange between VRFs
- Static routing across VRFs

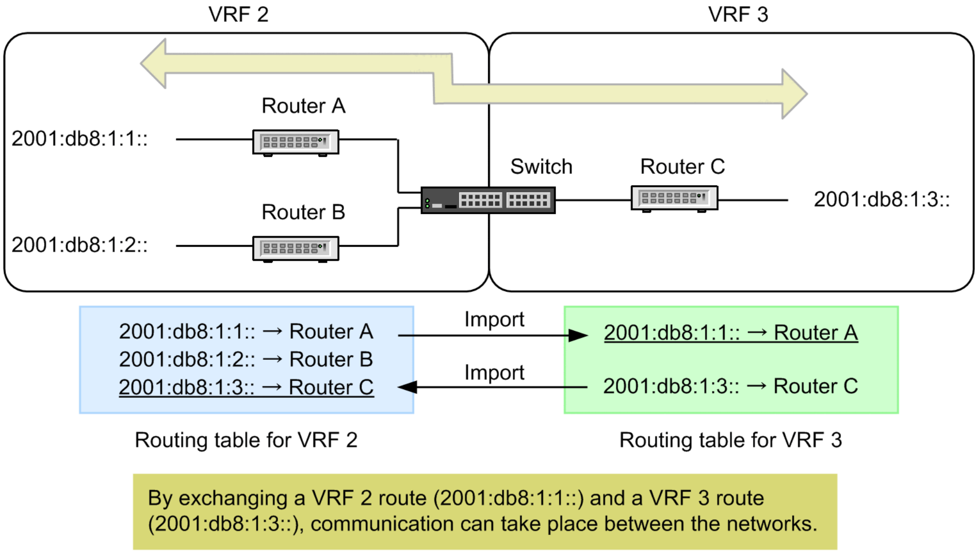
The following describes the exchange of routes between VRFs that uses a routing table and static routing across VRFs. Routes that can be imported between VRFs are also described.

(1) Route exchange between VRFs

An extranet can be implemented by exchanging routing information held by VRFs with other VRFs.

The following figure shows an example of exchanging routes between VRFs.

Figure 23-12: Route exchange between VRFs



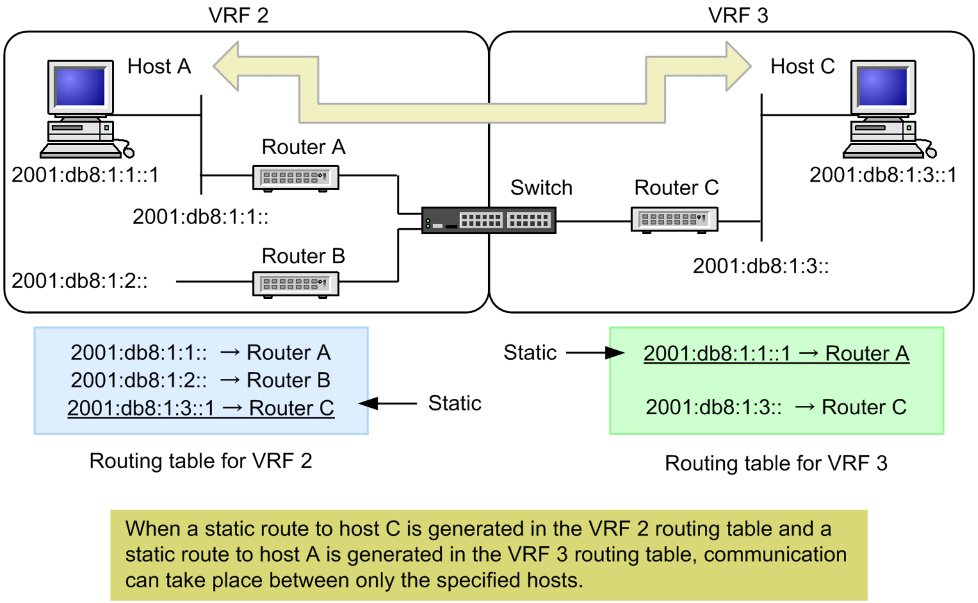
By exchanging a VRF 2 route (2001:db8:1:1::) and a VRF 3 route (2001:db8:1:3::), communication can take place between the networks.

(2) Static routing across VRFs

An extranet can be implemented by creating a static route that uses a gateway for another VRF as the next hop.

The following figure shows static routing across VRFs.

Figure 23-13: Static routing across VRFs



When a static route to host C is generated in the VRF 2 routing table and a static route to host A is generated in the VRF 3 routing table, communication can take place between only the specified hosts.

(3) Routes that can be imported between VRFs

The following table describes the types of routes that can be imported from another VRF or the global network.

Table 23-13: Types of routes that can be imported from another VRF or the global network

Route type	Importable?
Inactive route	N

Route type	Importable?
Route whose deletion is pending	N
Route imported for an extranet	N
Summarized route	Y
Route of an IPv6 device address specified for a loopback interface	Y
Route directly connected to a VLAN interface (global address)	Y
Route directly connected to a VLAN interface (link-local address)	N
Route whose output interface is a VLAN interface	Y
Route whose output interface is a loopback interface	Y
Route whose output interface is a null interface	Y

Legend: Y: Imported, N: Not imported

If the routes to be imported match multiple route types, import is possible only when all the matching route types can be imported.

23.12 VRF configuration [OS-L3SA]

23.12.1 List of configuration commands

The following table describes the configuration commands for VRFs.

Table 23-14: List of configuration commands

Command name	Description
match vrf[#1]	Configures `route-map` to use a VRF as filter conditions.
route-map[#1]	Configures `route-map`.
ipv6 route[#2]	Generates an IPv6 static route.
ipv6 import inter-vrf[#3]	Applies a filter to control which routes are imported from another VRF or the global network.
ipv6 maximum routes[#3]	Sets the maximum number of routes for the VRF and the threshold for output of a warning message.
vrf definition[#3]	Configures a VRF.

#1

See *14. Route Filters (IPv4 and IPv6)* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

#2

See *24. Static Routing (IPv6)* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

#3

See *30. VRF [OS-L3SA]* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

23.12.2 Configuring the maximum number of routes

Set the maximum number of routes for the VRF and the threshold for output of a warning message.

Points to note

The example below shows how to use the `ipv6 maximum routes` command to set the maximum number of routes and the threshold for output of a warning message.

Command examples

1.
```
(config)# vrf definition 2
```
Switches to VRF 2 configuration mode.

2.
```
(config-vrf)# ipv6 maximum routes 1000 80
```
Sets 1000 as the maximum number of routes of IPv6 that are handled by VRF2. Also, sets the threshold for outputting a warning message to 80%.

23.12.3 Configuring extranets

For details about how to configure extranets, see *24.2.7 Configuring a static route between VRFs*

[OS-L3SA] and *29.2.8 Extranet [OS-L3SA]*.

23.13 VRF operation [OS-L3SA]

23.13.1 List of operation commands

The following table describes the operation commands for VRFs.

Table 23-15: List of operation commands

Command name	Description
show ipv6 route	Shows routing information stored in the routing table.
show ipv6 static	Shows information related to a static route.
show ipv6 vrf	Shows the IPv6 information of a VRF.

23.13.2 Checking the maximum number of routes

Use the `show ipv6 vrf` command to display the current number of routes registered in the VRF forwarding table and the maximum number of routes that can be specified.

Figure 23-14: Results of executing show ipv6 vrf

```
> show ipv6 vrf 2
Date 20XX/12/20 12:00:00 UTC
VRF              Routes        Neighbor
2                12/100        7/50         ...1
>
```

1. The numerator represents the current number of routes, and the denominator represents the maximum number of routes.

Figure 23-15: Results of executing show ipv6 vrf detail

```
> show ipv6 vrf 2 detail
Date 20XX/12/20 12:00:00 UTC
VRF 2
  Maximum routes: 100, Warn threshold: 70%, Current routes: 12    ...1
  Maximum Neighbor entries: 50, Current Neighbor entries: 7
  Import inter-vrf: -
Interface
Name           Address                                    Status
VLAN0009       3ffe:501:ffff:2::200/64                    Up
VLAN0009       fe80::1001:201a:1%VLAN0009/64              Up
localhost      ::1/128                                    Up
localhost      fe80::1%localhost/64                       Up
>
```

1. Information is displayed in the following order: the maximum number of routes, the threshold for the output of a warning message, and the current number of routes.

23.13.3 Checking extranets

For details about how to check extranets, see *29.3.9 Checking extranet [OS-L3SA]*.

Chapter

24. Static Routing (IPv6)

This chapter describes IPv6 static routing.

24.1 Description

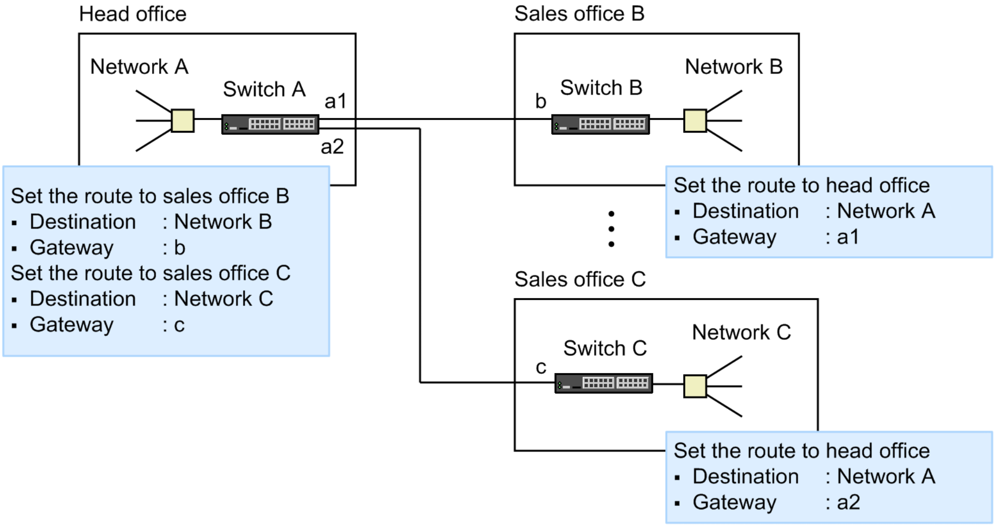
24.1.1 Overview

In static routing, packets are forwarded according to the routing information (static routes) set by using configuration commands.

In the Switch, you can set multiple forwarding routes, including a default route, to a particular destination subnetwork or host.

The following figure shows an example of a network configuration that uses static routing. Set the static route from the head office to each sales office, and then from the sales offices to the head office. In this example, the sales offices cannot communicate with each other.

Figure 24-1: Example of network configuration using static routing



Set the route to sales office B
- Destination : Network B
- Gateway : b

Set the route to sales office C
- Destination : Network C
- Gateway : c

Set the route to head office
- Destination : Network A
- Gateway : a1

Set the route to head office
- Destination : Network A
- Gateway : a2

24.1.2 Route selection conditions

In static routing, routes to the same destination network are grouped by distance, and a route from the group that has the smallest distance is selected.

When the maximum number of multipath routes is set to 2 or greater, the routes are configured according to the priorities given in the following table. When the maximum number of multipath routes is set to 1, the route with the highest priority is selected.

The default maximum number of multipaths is 6. You can change this value by using the `ipv6 route static maximum-paths` configuration command.

Table 24-1: Route selection priority

Priority	Description
High	Selects the route with the greatest weight.
Low	Selects the route with the smallest next-hop address.

24.1.3 Specifying a forwarding route for static routing

For the forwarding route (gateway), you can specify a directly connected neighboring gateway or a remote gateway that is not directly connected to the switch. With a neighboring gateway, the status of the connected interface governs path generation and deletion. With a remote gateway, the presence or absence of a route to that gateway controls path generation and deletion. The Switch uses remote gateways as the default gateway type. When you set up a neighboring gateway using the `ipv6 route` configuration command, specify the `noresolve` parameter.

Two additional parameters can be specified for the path to a neighboring or remote gateway specified in the command. Both parameters stop packets from being sent via that gateway. Packet transmission will also be disabled if you have specified a null interface for the gateway.

- `noinstall` parameter

  If you specify the `noinstall` parameter for a static route, that route will not be used for packet transfer. If there is a default route or other next-best route, packets will be forwarded via that route. Use the `noinstall` parameter when you want to set a static advertising route but have packets forwarded through a different route.

- `reject` parameter

  If you specify the `reject` parameter for a static route, that route becomes a reject route. Any packets that match the route are discarded. An ICMP Unreachable message notifies the source device that the packet has been discarded. Use the `reject` parameter when you want to set a static advertising route but want to discard packets for which a route with a higher priority has not been set in the Switch. You can also use this parameter to prevent packets destined for a particular address or destination from being forwarded via this route.

- Null interface

  If you specify only a null interface for a static route without specifying a gateway, all packets on that static route will be discarded. Unlike packets discarded by the `reject` parameter, no ICMP message is sent to the source device. Specify a null interface when you want to set the same behavior as the `reject` parameter but do not want ICMP packets to be returned. For details on null interface behavior, see *19. Null Interface (IPv6)*.

24.1.4 Dynamic monitoring

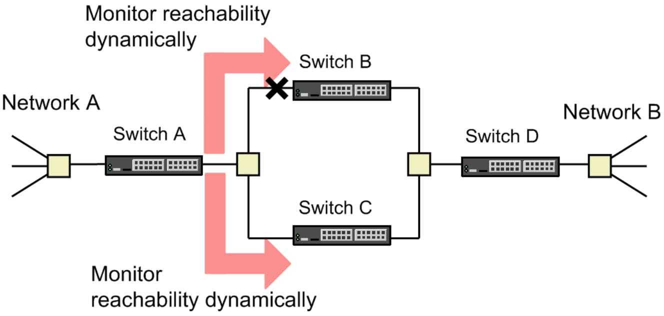
The generation and deletion of static routes is governed according to the status of the interface directly connected to the gateway, or the presence or absence of a route to that gateway. Consequently, there is no guarantee that packets will reach the gateway even if a route has been created. The Switch provides functionality for dynamically monitoring packet delivery by polling gateways at regular intervals using ICMPv6 Echo Request and Echo Reply messages. By using this monitoring functionality, you can control static route generation so that a route is created only when the route generation and deletion conditions described in *24.1.3 Specifying a forwarding route for static routing* are met and, moreover, the packets are reliably deliverable.

Even if a gateway that was unreachable becomes reachable, the route is not generated at that time. Rather, the route is generated after the reachability of the gateway is monitored for a set period and stability is confirmed.

(1) Path switching by dynamic monitoring of static routes

The following figure shows an example of monitoring static routes dynamically.

Figure 24-2: Example of dynamic monitoring of static routes


Monitor reachability dynamically
Switch B
Network A
Switch A
Network B
Switch D
Switch C
Monitor reachability dynamically

In this example, two static routes to network B have been set up in Switch A: a preferred route via Switch B and a next-best route via Switch C. Without dynamic monitoring, if an error occurs in the Switch B interface, the static route via Switch B would not be deleted because the interface on the Switch A side is still working normally. As a result, the path is not switched to the static route via Switch C, so communication between Switch A and network B ceases.

With dynamic monitoring, however, the monitoring functionality of Switch A detects that Switch B is unreachable although the interface on the Switch A side is normal. The static route through Switch B is deleted, and the path is switched to the static route via Switch C, assuring normal communication between Switch A and network B.

(2) Timing of static route generation, deletion, and restoration when using dynamic monitoring

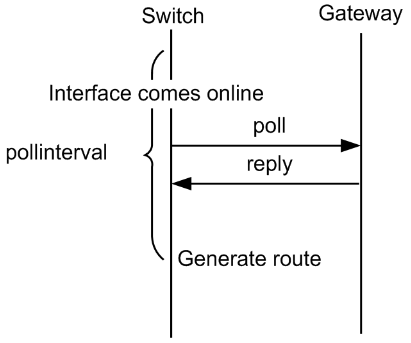
The timing for generating, deleting, or restoring a static route by dynamic monitoring depends on the values set in the `ipv6 route static poll-interval` and `ipv6 route static poll-multiplier` configuration commands.

In the following description, the relevant settings are `pollinterval` in `ipv6 route static poll-interval`, and `invalidcount` and `restorecount` in `ipv6 route static poll-multiplier`.

(a) Timing for generating a route

The Switch polls a gateway when its interface comes online or some other triggering event occurs. If a response is received, a static route through that gateway is created when the next polling time arrives (`pollinterval`). The following figure shows an example of route generation by dynamically monitoring static routes.

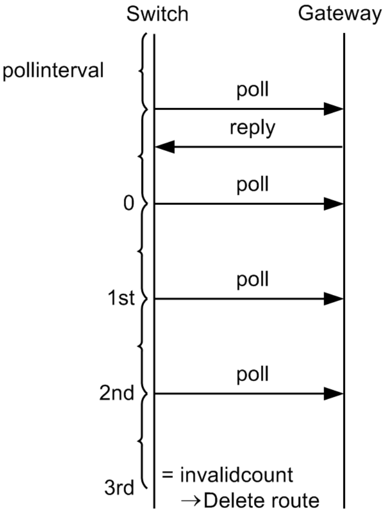
Figure 24-3: Example of generating a route by dynamic monitoring of static routes



(b) Timing for deleting a route

A static route through a gateway that is polled at regular intervals (`pollinterval`) will be deleted if no response is received after the number of consecutive poll attempts set in the `invalidcount` parameter. For example, if the `invalidcount` setting is 3, the route will be deleted if no response is received after three consecutive polls. If the interface goes down or some other trigger for route generation is lost, the static route will be deleted just as if the gateway were not being polled (`poll` parameter not specified). The following figure shows an example of route deletion by dynamically monitoring static routes.

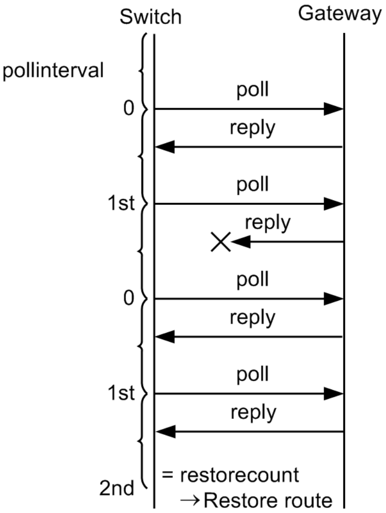
Figure 24-4: Example of deleting a static route by dynamic monitoring (invalidcount = 3)



(c) Timing for restoring a route

A static route deleted as a result of dynamic monitoring will be restored when the number of consecutive responses received from the polled gateway matches the number set in the `restorecount` parameter. For example, if the `restorecount` setting is 2, the route will be restored when responses are received for two consecutive polls. The following figure shows an example of restoring a static route as a result of dynamic monitoring.

Figure 24-5: Example of restoring a static route by dynamic monitoring (restorecount =2)



24.2 Configuration

24.2.1 List of configuration commands

The following table describes the configuration commands for static routing (IPv6).

Table 24-2: List of configuration commands

Command name	Description
ipv6 route	Generates an IPv6 static route.
ipv6 route static poll-interval	Specifies the interval for polling the gateway.
ipv6 route static poll-multiplier	Specifies the number of consecutive poll attempts and responses.

24.2.2 Configuring the default route

Set the default static route.

Points to note

Use the `ipv6 route` command for setting a static route. To set a default static route, specify `::/0` as the prefix.

Command examples

1.
```
(config)# ipv6 route ::/0 2001:db8:1:1::2
```
Specifies the remote gateway 2001:db8:1:1::2 as the next hop of the default route.

24.2.3 Configuring single-path routes

Set single-path static routes. The distances will determine path priority.

Points to note

For the static route you are setting as the alternate route, specify a distance larger than that of the preferred route.

Command examples

1.
```
(config)# ipv6 route 2001:db8:ffff:1::/64 2001:db8:1:2::2 100
```
Specifies the remote gateway 2001:db8:1:2::2 as the next hop of the static route 2001:db8:ffff:1::/64. Also, specify 100 as the distance.

2.
```
(config)# ipv6 route 2001:db8:ffff:1::/64 fe80::2 vlan 10 200
noresolve
```
Specifies the link-local address fe80::2 for interface vlan 10 for the neighboring gateway 172.16.1.100 as the next hop of static route 2001:db8:ffff:1::/64. Also, specify 200 as the distance. The Switch will use this path as the alternate route if the route to gateway 2001:db8:1:2::2 becomes invalid.

24.2.4 Configuring a multipath route

Set a multipath static route.

Points to note

Use the `ipv6 route` command to configure a multipath route by omitting the distance, or by specifying the same distance, for static routes to the same destination.

Command examples

1.
```
(config)# ipv6 route 2001:db8:ffff:2::/64 2001:db8:2:1::2
noresolve
```

   Specifies neighboring gateway 2001:db8:2:2::1 as the next hop of the static route 2001:db8:ffff:2::/64.

2.
```
(config)# ipv6 route 2001:db8:ffff:2::/64 2001:db8:2:2::2
noresolve
```

   Specifies neighboring gateway 2001:db8:2:2::2 as the next hop of the static route 2001:db8:ffff:2::/64. The static route 2001:db8:ffff:2::/64 is therefore configured as a multipath route through neighboring gateways 2001:db8:2:1::2 and 2001:db8:2:2::2.

24.2.5 Applying the dynamic monitoring functionality

Before you can apply the dynamic monitoring functionality to static routes, you must set an appropriate interval for polling the gateway and adjust the timing for route deletion and generation.

Points to note

To set the polling interval and the number of consecutive poll failures or responses, use the `ipv6 route static poll-interval` command and the `ipv6 route static poll-multiplier` command. To apply dynamic monitoring to a static route, set the `poll` parameter in the `ipv6 route` command.

Command examples

1.
```
(config)# ipv6 route static poll-interval 10
```

   Specifies 10 seconds as the polling interval for dynamic monitoring.

2.
```
(config)# ipv6 route static poll-multiplier 4 2
```

   Specifies 4 as the number of consecutive failures (`invalidcount`) and 2 as the number of consecutive responses (`restorecount`) for dynamic monitoring.

3.
```
(config)# ipv6 route 2001:db8:ffff:3::/64 2001:db8:3:1::2 poll
(config)# ipv6 route 2001:db8:ffff:4::/64 2001:db8:3:1::3 poll
```

   Applies dynamic monitoring to static routes 2001:db8:ffff:3::/64 and 2001:db8:ffff:4::/64.

24.2.6 Configuring a static route for a VRF [OS-L3SA]

Set a static route for a VRF.

Points to note

Specifies a VRF with the `vrf` parameter of the `ipv6 route` command.

Command examples

1.
```
(config)# ipv6 route vrf 2 2001:db8:ffff:20::/64
2001:db8:20:1::2 noresolve
```

   Generates the static route 2001:db8:ffff:20::/64 for VRF2. For the next hop, specifies the neighboring gateway 2001:db8:20:1::2.

24.2.7 Configuring a static route between VRFs [OS-L3SA]

Set a static route between VRFs to configure an extranet between specific hosts.

Points to note

Specifies the partner VRF by using the `vrf` parameter of the `ipv6 route` command (after the next hop address).

Command examples

1.
```
(config)# ipv6 route vrf 2 2001:db8:ffff:31::1/128
2001:db8:30:1::2 vrf 3 noresolve
```

   Generates the static route 2001:db8:ffff:31::1/128 for VRF 2. For the next hop, specifies the neighboring gateway 2001:db8:30:1::2 for VRF 3.

2.
```
(config)# ipv6 route vrf 3 2001:db8:ffff:21::1/128
2001:db8:20:1::2 vrf 2 noresolve
```

   Generates the static route 2001:db8:ffff:21::1/128 for VRF 3. For the next hop, specifies neighboring gateway 2001:db8:20:1::2 for VRF 2.

24.2.8 Configuring a static route across VRFs by using an IPv6 link-local address as the next hop [OS-L3SA]

Implement an extranet between specified hosts by configuring a static route across VRFs by using an IPv6 link-local address as the next hop.

Points to note

The example below shows how to specify the IPv6 link-local address as the next hop address in the `ipv6 route` command, and specify the interface in the subsequent interface parameter. If the VRF for the static route is different from the VRF for the interface, a static route between VRFs is created.

Command examples

1.
```
(config)# interface vlan 2
(config-if)# vrf forwarding 2
(config-if)# ipv6 enable
(config-if)# ipv6 address 2001:db8:ffff:ffff::1/64
```

   Specifies VRF 2 and an IPv6 address for VLAN ID 2.

2.
```
(config)# interface vlan 3
(config-if)# vrf forwarding 3
(config-if)# ipv6 enable
(config-if)# ipv6 address 2001:db8:ffff:fff0::1/64
```

   Specifies VRF 3 and an IPv6 address for VLAN ID 3.

3.
```
(config)# ipv6 route vrf 2 2001:db8:ffff:41::1/128 fe80::3 vlan
3 noresolve
```

Generates the static route 2001:db8:ffff:41::1/128 for VRF 2. For the next hop, specifies the link-local address fe80::3 for the VLAN 3 interface for the neighboring gateway.

4.
```
(config)# ipv6 route vrf 3 2001:db8:ffff:51::1/128 fe80::4 vlan
2 noresolve
```

Generates the static route 2001:db8:ffff:51::1/128 for VRF 3. For the next hop, specifies the link-local address fe80::4 for the VLAN 2 interface for the neighboring gateway.

24.3 Operation

24.3.1 List of operation commands

The following table describes the operation commands for static routing (IPv6).

Table 24-3: List of operation commands

Command name	Description
show ipv6 route	Shows routing information stored in the routing table.
clear ipv6 route	Clears the IPv6 forwarding entries stored in the Switch and re-registers the routing entries.
show ipv6 static	Shows information related to a static route.
clear ipv6 static-gateway	Performs polling for the gateways on the routes that were disabled by dynamic monitoring of static routes, and generates routes if the gateway responded.
show ipv6 vrf	Shows the IPv6 information of a VRF.
show ipv6 interface ipv6-unicast	Shows information about the IPv6 interfaces on the Switch recognized by the unicast routing program.
restart unicast#	Restarts the unicast routing program.
dump protocols unicast#	Outputs the trace information and control table information collected by the unicast routing program to a file.
erase protocol-dump unicast#	Deletes the file of trace information and control table information generated by the unicast routing program.

#

See *8. Routing Protocol Common to IPv4 and IPv6* in the manual *Operation Command Reference Vol. 2 For Version 11.10*.

24.3.2 Checking routing information

Check information about static routes.

Figure 24-6: Results of executing the show ipv6 static route command

```
>show ipv6 static route
Date 20XX/07/14 12:00:00 UTC
Status Codes: * valid, > active, r RIB failure
   Destination                                 Next hop
         Distance Weight Status      Flag
   ::/0                                        2001:db8:1:1::2
         2        0      IFdown      -
*> 2001:db8:ffff:1::/64                        2001:db8:1:2::2
         100      0      Act         -
*  2001:db8:ffff:1::/64                        fe80::2%VLAN0010
         200      0      Act         NoResolve
*> 2001:db8:ffff:2::/64                        2001:db8:2:1::2
         2        0      Act         NoResolve
                                               2001:db8:2:2::2
         2        0      Act         NoResolve
*> 2001:db8:ffff:3::/64                        2001:db8:3:1::2
         2        0      Act Reach   Poll
   2001:db8:ffff:4::/64                        2001:db8:3:1::3
         2        0      Unreach     Poll
```

Notes

1. When a route is registered in the routing table, an asterisk and angle bracket (`*>`) appear in the leftmost `Status Codes` field.
2. For an alternate route not registered in the routing table, the angle bracket (`>`) is omitted but the asterisk (`*`) appears in the `Status Codes` field if the route is valid.
3. For an invalid route, neither the asterisk nor angle bracket appears in the `Status Codes` field, and a problem of some sort is indicated in the `Status` field. `IFdown` means that the route is invalid due to a failure in an interface. `UnReach` means that the dynamic monitoring functionality has not confirmed that the route is reachable.

24.3.3 Checking gateway information

Check information about a static route gateway.

Figure 24-7: Results of executing the show ipv6 static gateway command

```
>show ipv6 static gateway
Date 20XX/07/14 12:00:00 UTC
Gateway                              Status  Success     Failure  Transition
2001:db8:1:1::2                      IFDown  -           -           -
2001:db8:1:2::2                      -       -           -           -
2001:db8:2:1::2                      -       -           -           -
2001:db8:2:2::2                      -       -           -           -
2001:db8:3:1::2                      Reach   -           0/4         13m 39s
2001:db8:3:2::2                      UnReach 1/2         -           21s
fe80::3%VLAN0010                     -       -           -           -
```

Notes

1. The `Status` field indicates whether a dynamically monitored gateway is reachable. `Reach` means that the route is confirmed to be reachable, and `UnReach` means that the route has not been confirmed as reachable.
2. If the dynamic monitoring functionality has not yet confirmed that the route is reachable (`UnReach` appears in the `Status` field), check the gateway's monitoring status from the `Success` counter. In the above results, gateway 2001:db8:3:2::2 has a Success counter of 1/2. This means that two consecutive responses are required to confirm reachability, and at present one successful response has been received.

Chapter

25. RIPng

This chapter describes the IPv6 routing Information Protocol (RIPng).

25.1 Description

25.1.1 Overview

The Routing Information Protocol (RIPng) is used between the routers connected on a network. By using RIPng, each router generates its own routing information by exchanging information about the networks it can reach and the number of hops (a metric) to those networks. The Switches support RIPng version 1 (compliant with RFC 2080).

(1) Message type

RIPng uses two types of messages: request messages and response messages. A request message is used to request routing information from another router. A response message is used to reply to a request from another router. Response messages are also used to notify other routers of Switch's routing information, either periodically or when there is a change in the network topology.

(2) Message processing

At startup, the Switch sends all of its request messages to neighboring routers, and requests neighboring routers to send all of their routing information. After startup, the Switch sends responses for the following purposes:

- To send relevant routing information in response to a request from a neighboring router
- To periodically report routing information. At 30-second intervals the Switch sends all of its routing information as responses to neighboring routers.
- To report a detected route change. The Switch sends routing information regarding a changed route as a response to neighboring routers.

On receipt of a response from a neighboring router, the Switch updates its routing information if a route change has been detected. Responses are also used to check the transmission status to a neighboring router. If no response is received for 180 seconds or longer, the neighboring router is deemed unreachable and the routing table is updated with an alternate route, if available. Otherwise, the unreachable route is deleted.

(3) Preventing routing loops

The Switch use split horizon logic to prevent loops in forwarding routes. Split horizon processing stops received information from being forwarded to the interface from which the information originated.

(4) Functional differences between RIPng (IPv6) and RIP (IPv4)

The following table describes the functional differences between RIPng (IPv6) and RIP (IPv4).

Table 25-1: Functional differences between RIPng (IPv6) and RIP (IPv4)

Functionality	RIPng(IPv6)	RIP(IPv4)
triggered update	Y	Y
Split horizon	Y	Y
Route poisoning	Y	Y
Poison reverse	N	N
Hold-downs	N	N
Route tagging	Y	Y
Read specified next hop	Y	Y
Plain-text password authentication	N	Y

Functionality	RIPng(IPv6)	RIP(IPv4)
Cryptographic authentication (Keyed-MD5)	N	Y
If a route having the same metric as that of the existing route is received from a different gateway when the expired aging time of the existing route is equal to or more than half a second of the timer value, the route is changed to the most recently learned route.	N	Y

Legend: Y: Supported, N: Not supported

25.1.2 Route selection conditions

From information learned by each protocol about routes to the same destination, the Switch selects the best route according to the independent route selection procedures associated with each protocol. If there are a number of alternative routes in the generated information, their distances are compared and the routing information that has the highest priority is selected.

Under RIPng, priority rules govern the selection of the best route to a given destination among the routes learned and advertised by different routers. The following table explains the priority rules for selecting the best route.

Table 25-2: Route selection priority

Priority	Description
High	Selects the route with the lowest metric.
↑	Selects the route with the smallest next-hop address.
↓	Selects the route whose next-hop address in the routing information matches the address of the gateway from which the information originated.#2
Low	In all other cases, ignores the most recently learned route.

#: This condition applies when routing information that contains the same next-hop address is learned from different neighboring routers on the network.

When the routes to a given destination learned by each protocol (OSPFv3, BGP4+, and static) result in multiple entries, their distances are compared and the route that has the highest priority is set in the routing table.

(1) Generating secondary routes

Using the `generate-secondary-route` configuration command, you can generate up to two route entries (primary route and secondary route) to the same destination learned by two different neighboring routers. The following table describes the conditions for generating a secondary route.

Table 25-3: Secondary route generation conditions

Conditions		Secondary route generation
Specification of the generate-secondary-route configuration command	Distance	
N	--	Not generated
Y	Different values for primary route and secondary route	Not generated
Y	Same values for primary route and secondary route	Generated

Legend: Y: Command entered, N: Command not entered, --: Not applicable

When secondary route generation is enabled, route priority to a given destination is determined as follows:

Table 25-4: Route selection priority when registration of a secondary route is specified

Priority	Description
High	Selects the route with the next lowest metric.
↑	Selects the route with the next smallest next-hop address[#1]
	Selects the route whose next-hop address in the routing information matches the address of the gateway from which the information originated.[#2]
↓	Selects the route that has been the primary route until now.
Low	In all other cases, ignores the most recently learned route.

Note:

If routes have the same next-hop address, only a primary route is generated.

#1

This condition does not apply to a newly learned route if a secondary route has already been registered.

#2

This condition applies if routing information containing the same next-hop address is learned by different neighboring routers on the network.

25.1.3 Route advertisements

(1) Advertised routes

(a) Learning protocol

If filtering of advertised routes has not been specified, learned RIPng routes and directly-connected routes within the RIPng network are advertised. If filtering is specified, the advertising behavior is governed by the filter conditions. The following table describes the learning protocol for route advertising via RIPng.

Table 25-5: Learning protocol and advertising behavior

Learning protocol		Advertising behavior without route filtering	Order in which the advertising metrics are applied[#5]
Directly connected route[#1]	Interface on which RIPng is running	Advertised	1. Setting for advertised route filtering 2. Default (metric: 1)
	Interface other than the above	Not advertised	
Summarized route		Not advertised	
Static route		Not advertised	1. Setting for advertised route filtering 2. Setting by `default-metric` 3. Default (metric: 1)
RIPng[#2]		Advertised	1. Setting for advertised route filtering 2. Metric in the routing table

Learning protocol	Advertising behavior without route filtering	Order in which the advertising metrics are applied#5
OSPFv3	Not advertised	1. Setting for advertised route filtering 2. Metric in the routing table if `inherit-metric` is set#3 3. Setting by `default-metric`#4
BGP4+	Not advertised	
Route imported from another VRF or the global network	Not advertised	

#1

The secondary address is also advertised.

#2

Split horizon is applied.

#3

The route is not advertised if the metric in the routing table is 16 or higher.

#4

The route is not advertised if advertised route filtering is unspecified or if no metric is specified by the `inherit-metric` or `default-metric` command.

#5

If set, the `metric-offset out` setting is added to the selected metric. If this results in a metric of 16 or higher, the route is not advertised.

(b) Address types

The following table describes the types of addresses that can be advertised via RIPng.

Table 25-6: Types of routing information

Type of routing information	Definition	Example	Advertised
Default routing information	Routing information about all destination networks	::/0	Y
Network routing information	Global routing information about a specific network destination	2001:db8:1:1::/64 2001:db8:1::/56	Y#
Host routing information	Global routing information about a specific destination host	2001:db8:1:1::1/128	Y#

Legend: Y: Can be advertised

#: Only global addresses and local site addresses can be advertised.

(2) Destination of route advertisements

In RIPng, route advertisements are sent to all neighboring routers (including routers on a network to which the interface's secondary address belongs) connected to the interface specified by the `ipv6 rip enable` configuration command.

(3) Timing of route advertisements

The following table describes the functionality related to the timing of route advertisements distributed via RIPng.

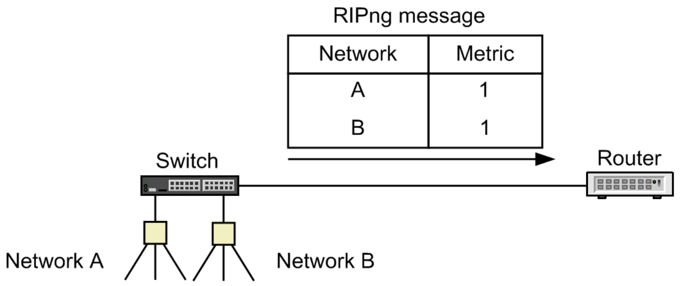
Table 25-7: Timing of route advertisements

Functionality	Description
Periodic route advertisement	Neighboring routers are reported periodically about routing information held by the switch.
triggered update	Any change in the routing information held by the switch is reported immediately without waiting for a periodic advertisement.
Response to a request from a neighboring router	The neighboring router that sent the request packet is notified.
Route poisoning	Neighboring routers are notified for a set duration of any deleted routing information.

(a) Periodic route advertisement

RIPng periodically advertises all routing information held by the local router to neighboring routers. The following figure shows an example of a periodic route advertisement.

Figure 25-1: Periodic route advertisement



RIPng message

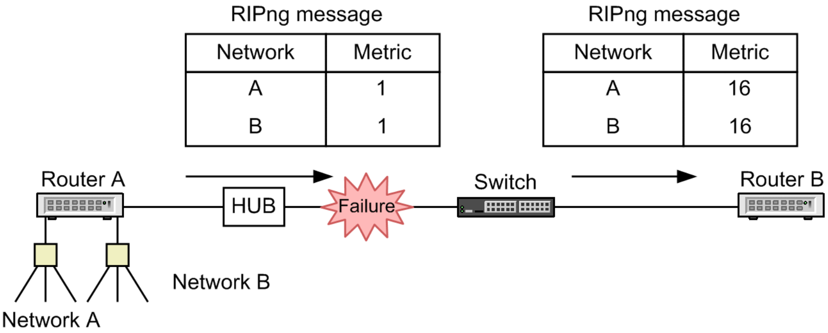
Network	Metric
A	1
B	1

The Switch advertises routing information about networks A and B to the router at 30-second intervals (advertisement timer).

(b) triggered update

Changes in the routing information held by the switch are distributed immediately without waiting for the periodic distribution cycle. The following figure shows a route advertisement distributed as a triggered update.

Figure 25-2: Route advertisement sent as a triggered update



RIPng message

Network	Metric
A	1
B	1

RIPng message

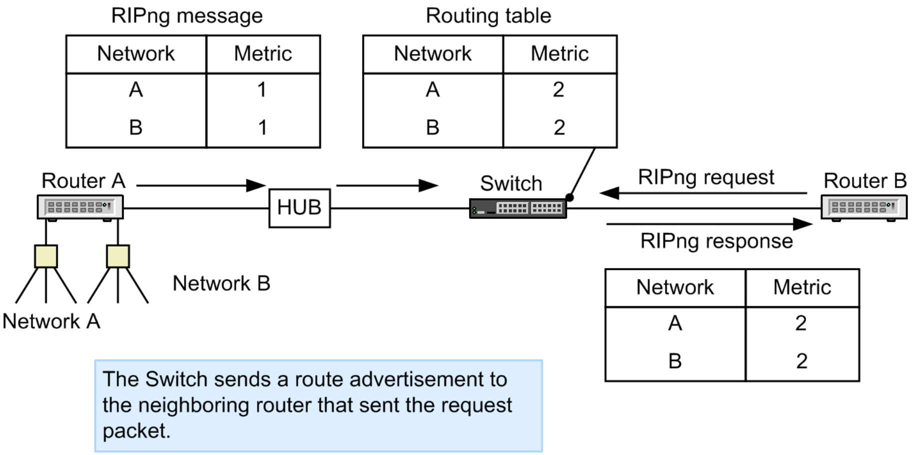
Network	Metric
A	16
B	16

If the Switch detects a failure between itself and the hub, the Switch deletes routing information about networks A and B from its routing table.
At the same time, the Switch sends a route advertisement to router B, giving networks A and B a metric of 16 (unreachable).

(c) Response to request packets

On receipt of a request packet, the Switch sends the requested information to the neighboring router that sent the packet. The following figure shows how routing information is advertised when a request packet is received.

Figure 25-3: Route advertisement sent on receipt of a request packet



RIPng message

Network	Metric
A	1
B	1

Routing table

Network	Metric
A	2
B	2

RIPng response

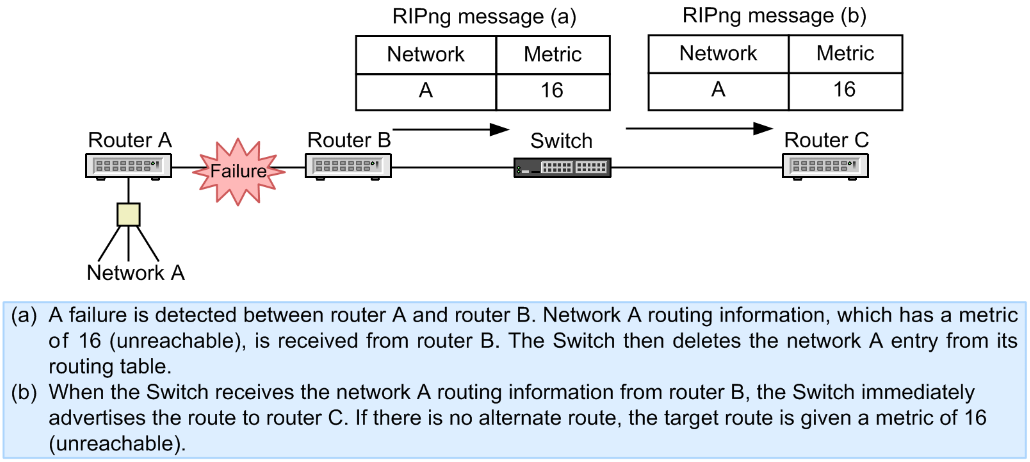
Network	Metric
A	2
B	2

The Switch sends a route advertisement to the neighboring router that sent the request packet.

(d) Route poisoning

When a route changes from reachable to unreachable status (on receipt of a metric-16 route advertisement or on deletion of a route learned from an interface that has since failed), the router advertises the metric-16 (unreachable) route to its neighboring routers for a set period (60 seconds: garbage collection timer). The following figure shows route poisoning.

Figure 25-4: Route poisoning



RIPng message (a)

Network	Metric
A	16

RIPng message (b)

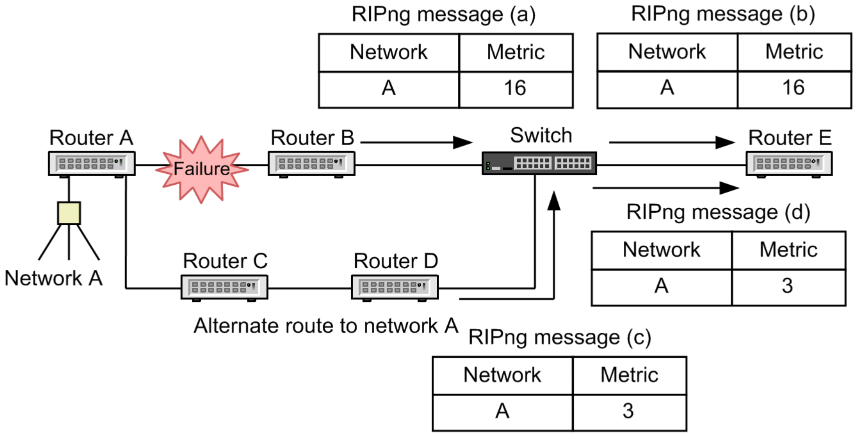
Network	Metric
A	16

(a) A failure is detected between router A and router B. Network A routing information, which has a metric of 16 (unreachable), is received from router B. The Switch then deletes the network A entry from its routing table.

(b) When the Switch receives the network A routing information from router B, the Switch immediately advertises the route to router C. If there is no alternate route, the target route is given a metric of 16 (unreachable).

If a new route to the affected destination is learned while route poisoning is in effect, the Switch advertises the new routing information. This is illustrated in the following figure.

Figure 25-5: Learning a new route during route poisoning



RIPng message (a)

Network	Metric
A	16

RIPng message (b)

Network	Metric
A	16

RIPng message (d)

Network	Metric
A	3

RIPng message (c)

Network	Metric
A	3

(a) A failure is detected between router A and router B. Network A routing information, which has a metric of 16 (unreachable), is received from router B. The Switch then deletes the network A entry from its routing table.
(b) At the same time, the Switch advertises the network A routing information, which has a metric of 16 (unreachable), to router E.
(c) The Switch receives the network A routing information in a periodic advertisement from router D. This information is added to its routing table (update timing is dependent on the periodic advertisement timer of router D).
(d) The Switch advertises the network A routing information to router E.

25.1.4 Learning routing information

(1) Origin of learned routing information

In RIPng, routing information can be learned from all neighboring routers connected to the interface specified by the `ipv6 rip enable` configuration command. This includes routers on a network to which the secondary address of the interface belongs.

(2) Timing of route learning and updating

The following table describes the functionality related to the timing of route updates learned via RIPng.

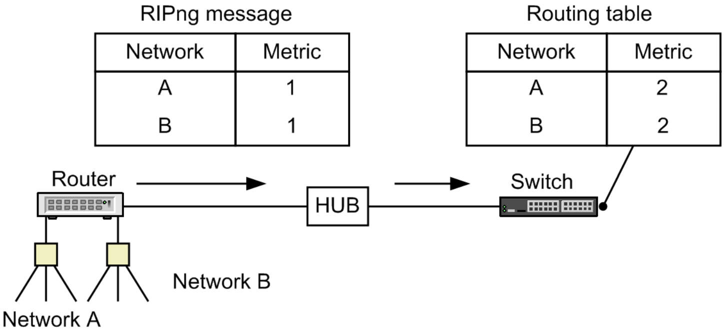
Table 25-8: Timing of route learning and updating

Functionality	Description
Response packets received from neighboring routers	Adds, changes, or deletes routing information as reported by neighboring routers.
Aging timeout	Deletes routing information if there is no periodic notification from a neighboring router for a set time.
Recognition of interface failures	Deletes routing information learned from a RIPng-enabled interface on which a failure has been discovered.

(a) Receiving response packets

Under RIPng, the routing information in the response packets received from neighboring routers is written to the Switch's routing table. The following figure shows the generation of routing information from a received response packet.

Figure 25-6: Generation of routing information from a received response packet

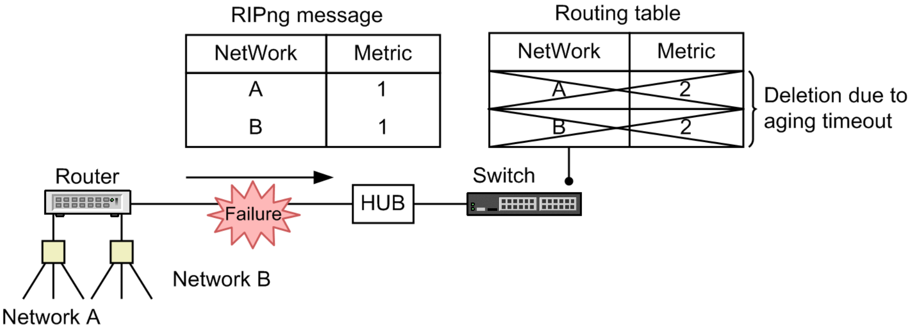


The Switch adds the routing information for networks A and B, which were learned from the neighboring router, to its routing table.

(b) Aging timeout

If routing information generated from a received response packet is the best route, the routing information is written to the routing table of the Switch, and then is monitored by an aging timer. The aging timer is reset (cleared) by a periodic advertisement from the neighboring router. If the router fails to generate an advertisement for the route being monitored for 180 seconds (aging timeout value) due to a hardware failure or a line error between the router and switch, the affected entries are deleted from the switch's routing table. The following figure shows deletion of routing information by aging timeout.

Figure 25-7: Deletion of routing information by aging timeout

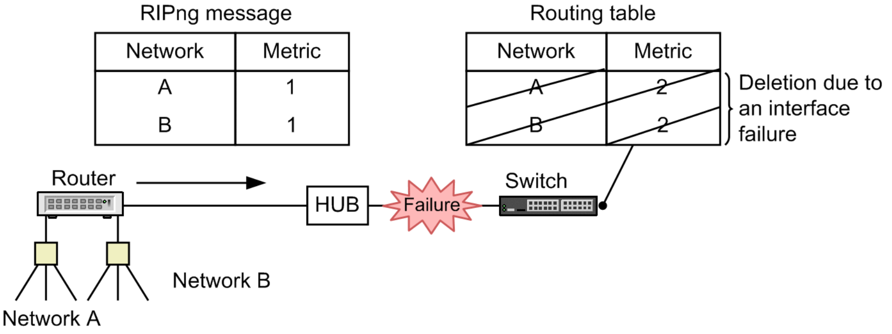


If a failure occurs between the router and hub, routing information for networks A and B is not advertised to the Switch.
The Switch deletes the affected routing information from its routing table if no advertisement is received for 180 seconds (aging timeout).

(c) Recognizing interface failures

On recognizing that the interface that connects the switch to a neighboring router has failed, the Switch immediately deletes all routing information learned from that interface. The following figure shows the deletion of routing information due to an interface failure.

Figure 25-8: Deletion of routing information due to an interface failure



RIPng message

Network	Metric
A	1
B	1

Routing table

Network	Metric
~~A~~	~~2~~
~~B~~	~~2~~

If the Switch detects a failure in the interface that connects it to neighboring router, the Switch deletes all routing information learned from that interface from its routing table.

25.1.5 RIPng features

RIPng sets the prefix length of advertised routes in the routing information, allowing variable-length prefix lengths to be handled. The following describes the RIPng features.

(1) Authentication functionality

The Switch does not support the authentication functionality.

(2) Route tagging

In the Switch, route tags are written to routing tables if set in the routing information reported in a response message. The route tag of the corresponding entry in the routing table is set as the route tag in the routing information in the response message sent by the Switch. The valid range is 1 to 65535 (in decimal).

RIPng does not support filtering by using route tags in the import filter or changing of route tags in the export filter (distribution of routes to RIPng from another protocol).

(3) Prefix

In the Switch, the prefix length of the routing information reported in a response message is written to the routing table. The prefix length of the corresponding entry in the routing table is set as the prefix length in the routing information in the response message sent by the Switch.

(4) Next hop

In the Switch, next-hop information is written to the routing tables if set in the routing information reported in a response message. If no next-hop information is set, the originating gateway is regarded as the next hop.

The next-hop information in the routing information is not set in a response message sent from the Switch. Accordingly, a router that received a route from the Switch via RIPng uses the source interface address as the next hop.

(5) Link-local multicasting

The Switch supports link-local multicasting to reduce the unnecessary load on the hosts that do not receive RIPng messages. The link-local multicast address used for transmitting RIPng messages is the multicasting address for all RIPng routers (ff02::9).

25.1.6 Notes

Note the following restrictions when configuring a network that uses RIPng.

(1) Differences in implementation from the RFCs

The Switch complies with RFC 2080 (RIPng version 1). However, there are some differences due to the functional limitations of the software. Differences are described in the table below.

Table 25-9: Differences from the RFC 2080

RFC		Switch
must be zero field	Nothing is specified for processing.	The Switch does not check the value of the `must be zero` field. This field is set to 0 when sending information.
Network prefix	Nothing is specified for the status of the address field after the prefix length.	If the address field after the prefix length in routing information in the received RIPng packet is not cleared to zero, the addresses after the prefix length are cleared to zero.
Triggered update	After a triggered update is sent, a timer for a random interval between one and five seconds should be set. Even if the setting is changed to send an update before a timeout occurs, the update is sent when a timeout occurs.	After a triggered update is sent, a timer for a random interval between one and five seconds is not set. Instead, the Switch sends a triggered update whenever routing information is changed.
	If a regular update occurs while the timer for a random interval between one and five seconds is running after a triggered update is sent, triggered updates might be suppressed.	Triggered updates are not suppressed.
Split horizon	The split horizon setting must be capable of being changed for an interface.	The Switch does not support changing of the split horizon setting for an interface.
Specification of next-hop information for routes	Next-hop information for a route can be explicitly specified.	RIPng packets sent by the Switch does not contain next-hop information. If the Switch receives an RIPng packet with explicitly specified next-hop information, the specified value is used as the next hop.
Destination of a response packet	If ff02::9 is not an appropriate destination (for example, NBMA network), the destination depends on the implementation.	The Switch does not support RIPng operations on the NBMA network.
Authentication	An IPv6 authentication header and encrypted header are used to authenticate a packet.	The Switch does not support packet authentication that uses an IPv6 authentication header and encrypted header.
Sending response packets when request packets are received via unicast from any port other than source port 521	Response packets can be directly sent to the source address.	The Switch sends response packets only for request packets for which a link-local address is specified as the source address.

25.2 Configuration

25.2.1 List of configuration commands

The following table describes the configuration commands for RIPng.

Table 25-10: List of configuration commands

Command name	Description
default-metric	Specifies the metric to be used when routing information learned by another protocol is advertised by RIPng.
disable	Disables RIPng.
distance	Specifies the distance for routing information learned by RIPng.
generate-secondary-route	Registers a secondary route in the routing table.
inherit-metric	Specifies that the metric is to be inherited when routing information learned by another routing protocol is advertised by RIPng.
ipv6 rip enable	Sends and receives RIPng packets via a specific interface.
ipv6 rip metric-offset	Specifies the metric increment when RIPng packets are sent or received via a specific interface.
ipv6 router rip	Configures RIPng-related operation information.
passive-interface	Disables a specified interface from sending routing information in RIPng packets.
timers basic	Specifies the values of the various RIPng timers.
distribute-list in (RIPng)#	Applies a filter to control whether routes learned by RIPng are entered in a routing table.
distribute-list out (RIPng)#	Applies a filter to control which routes are advertised by RIPng.
ipv6 prefix-list#	Configures an IPv6 prefix list.
redistribute (RIPng)#	Specifies the protocol for routes advertised by RIPng.
route-map#	Configures `route-map`.

#

See *14. Route Filters (IPv4 and IPv6)* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

25.2.2 Applying RIPng

Set the interface for sending and receiving RIPng packets.

Points to note

The example below shows how to use the `ipv6 rip enable` command to apply RIPng.

Command examples

1.
```
(config)# interface vlan 10
(config-if)# ipv6 address 2001:db8:1:1::1/64
(config-if)# ipv6 enable
```
Sets IPv6 address 2001:db8:1:1::1/64 for interface vlan 10.

2. `(config-if)# ipv6 rip enable`

   Enables RIPng packet transmission and reception on interface vlan 10.

3.
   ```
   (config-if)# exit
   (config)# interface vlan 20
   (config-if)# ipv6 address 2001:db8:1:2::1/64
   (config-if)# ipv6 enable
   ```

   Sets IPv6 address 2001:db8:1:2::1/64 for interface vlan 20.

4. `(config-if)# ipv6 rip enable`

   Enables RIPng packet transmission and reception on interface vlan 20.

5. `(config-if)# exit`

25.2.3 Configuring metrics

(1) Setting the metric for advertising non-RIPng routing information

Set the metric to be used when routing information learned by another protocol is advertised by RIPng.

Points to note

To advertise OSPFv3 routes or BGP4+ routes via RIPng, you must set the metric by using the `default-metric` command.

Command examples

1.
   ```
   (config)# ipv6 router rip
   (config-rtr-rip)# default-metric 10
   ```

   Sets 10 as the metric to be used when routing information learned by another protocol is advertised by RIPng.

2. `(config-rtr-rip)# redistribute static`

   Specifies that static routes are to be advertised by RIPng.

3. `(config-rtr-rip)# redistribute ospf`

   Specifies that OSPFv3 routes are to be advertised by RIPng.

(2) Setting the metric increment for packet transmission

Set the value by which to increment the metric for sending and receiving RIPng packets.

Points to note

The example below shows how to use the `ipv6 rip metric-offset` command to set the value to be added to the metric of an incoming or outgoing route via a specific interface.

Command examples

1.
```
(config)# interface vlan 10
(config-if)# ipv6 rip metric-offset 2 out
```
Adds 2 to the metric of RIPng packets sent from interface vlan 10.

25.2.4 Adjusting the timers

Adjust the value of the RIPng periodic advertisement timer, the value of the aging timer, and the wait time before entries are deleted from the routing tables.

To reduce the convergence time when a route is changed, set a value smaller than the default for the periodic advertisement timer and aging timer. To reduce RIPng advertisement traffic, set a value greater than the default for the periodic advertisement timer.

If you change a RIPng timer value, apply the same timer value to all routers on the RIPng network.

Points to note

The example below shows how to use the `timers basic` command to change the value of the RIPng timers.

Command examples

1.
```
(config)# ipv6 router rip
(config-rtr-rip)# timers basic 40 200 100
```
Sets 40 seconds for the RIPng periodic advertisement timer, 200 seconds for the aging timer, and 100 seconds before entries are deleted from the routing tables.

25.2.5 Applying RIPng for a VRF [OS-L3SA]

Enable RIPng for VRF.

Points to note

The example below shows how to use the `ipv6 rip enable` command to enable RIPng packet transmission and reception on the VRF interface.

Command examples

1.
```
(config)# interface vlan 10
(config-if)# vrf forwarding 2
(config-if)# ipv6 address 2001:db8:1:1::1/64
(config-if)# ipv6 enable
(config-if)# ipv6 rip enable
(config-if)# exit
```
Sets the address 2001:db8:1:2::1/64 to the interface VLAN 10 of VRF 2 to enable RIPng packet transmission and reception.

2.
```
(config)# interface vlan 20
(config-if)# vrf forwarding 2
(config-if)# ipv6 address 2001:db8:1:2::1/64
(config-if)# ipv6 enable
(config-if)# ipv6 rip enable
```

```
(config-if)# exit
```

Sets the address 2001:db8:1:2::1/64 to the interface VLAN 20 of VRF 2 to enable RIPng packet transmission and reception.

3.
```
(config)# ipv6 router rip vrf 2
```

Switches to `config-rtr-rip` mode, and then specifies information about the RIPng that runs in VRF 2.

4.
```
(config-rtr-rip)# default-metric 10
(config-rtr-rip)# redistribute static
(config-rtr-rip)# exit
```

Sets 10 as the metric for when using RIPng to advertise routing information learned by another protocol. Also, specifies that static routes are to be advertised by using RIPng.

25.3 Operation

25.3.1 List of operation commands

The following table describes the operation commands for checking RIPng information.

Table 25-11: List of operation commands

Command name	Description
show ipv6 route	Shows routing information stored in the routing table.
clear ipv6 route	Clears the IPv6 forwarding entries stored in the Switch and re-registers the routing entries.
show ipv6 rip	Shows information about the RIPng protocol.
clear counters rip ipv6-unicast	Clears information about the RIPng protocol.
show ipv6 vrf	Shows the IPv6 information of a VRF.
show ipv6 interface ipv6-unicast	Shows information about the IPv6 interfaces on the Switch recognized by the unicast routing program.
debug ipv6	Shows the packets being routed by IPv6 routing protocols in real time.
show processes cpu unicast#	Shows the CPU usage of a unicast routing program.
debug protocols unicast#	Starts the operation message display for event log information output by a unicast routing program.
no debug protocols unicast#	Stops the operation message display for event log information output by a unicast routing program.
dump protocols unicast#	Outputs the trace information and control table information collected by the unicast routing program to a file.
erase protocol-dump unicast#	Deletes the file of trace information and control table information generated by the unicast routing program.

#

See *8. Routing Protocol Common to IPv4 and IPv6* in the manual *Operation Command Reference Vol. 2 For Version 11.10*.

25.3.2 Checking the RIPng operating status

Show information about the RIPng protocol.

Figure 25-9: Results of executing the show ipv6 rip command

```
> show ipv6 rip
Date 20XX/07/14 12:00:00 UTC
RIPng Flags: <ON>
Default Metric: 10, Distance: 120
Timers (seconds)
  Update             : 40
  Aging              : 200
  Garbage-Collection : 100
```

25.3.3 Checking destination information

Show the information at the RIPng transmission destination.

Figure 25-10: Results of executing the show ipv6 rip target command

```
> show ipv6 rip target
Date 20XX/07/14 12:00:00 UTC
Source Address                            Destination    Flags
fe80::4048:47ff:fe10:1%VLAN0010           VLAN0010       <Multicast>
fe80::4048:47ff:fe10:1%VLAN0020           VLAN0020       <Multicast>
```

25.3.4 Checking learned routing information

(1) Per-network check

Show routing information learned by RIPng in the specified network and stored in the routing tables.

Figure 25-11: Results of executing the show ipv6 rip route command

```
> show ipv6 rip route brief 4001::/16
Date 20XX/07/14 12:00:00 UTC
Status Codes: * valid, > active, r RIB failure
   Destination                              Interface      Metric Tag   Timer
*> 4001:21f7:2910:3029::/64                 VLAN0010       3      0      4s
*> 4001:64b9:4ba6:dd65::/64                 VLAN0020       4      0     10s
*> 4001:652c:7a78:c37::/64                  VLAN0020       3      0      9s
*> 4001:ddd9:158:9a2f::/64                  VLAN0010       5      0      4s
```

(2) Per-gateway check

Show per gateway the routing information received by RIPng in the specified network and stored in the routing tables.

Figure 25-12: Results of executing the show ipv6 rip received-routes command

```
> show ipv6 rip received-routes brief 4001::/16
Date 20XX/07/14 12:00:00 UTC
Status Codes: * valid, > active, r RIB failure
Neighbor Address: fe80::4048:47ff:fe10:10%VLAN0010
   Destination                              Interface      Metric Tag   Timer
*> 4001:21f7:2910:3029::/64                 VLAN0010       3      0      9s
*> 4001:ddd9:158:9a2f::/64                  VLAN0010       5      0      9s
Neighbor Address: fe80::4048:47ff:fe10:20%VLAN0020
   Destination                              Interface      Metric Tag   Timer
*> 4001:64b9:4ba6:dd65::/64                 VLAN0020       4      0     15s
*> 4001:652c:7a78:c37::/64                  VLAN0020       3      0     14s
```

25.3.5 Checking advertised routing information

Show the routing information sent to the specified interface.

Figure 25-13: Results of executing the show ipv6 rip advertised-routes command

```
> show ipv6 rip advertised-routes brief interface vlan 10
Date 20XX/07/14 12:00:00 UTC
Target Interface: VLAN0010
Destination                                 Interface      Metric Tag  Age
*> 2001:db8:1:2::/64                        VLAN0020       1      0    22m 37s
*> 4001:64b9:4ba6:dd65::/64                 VLAN0020       5      0    21s
*> 4001:652c:7a78:c37::/64                  VLAN0020       4      0    20s
```

Chapter
26. OSPFv3 [OS-L3SA]

This chapter describes the OSPFv3 routing protocol, which applies mainly to intranets.

26.1 Description of basic OSPFv3 functionality

OSPFv3 (Open Shortest Path First version 3) is a routing protocol for IPv6 that uses Dijkstra's algorithm to calculate the shortest path to known destinations, based on a topology map constructed from information about the state of links between routers.

26.1.1 Features of OSPFv3

OSPFv3 is typically used to route packets within a single autonomous system (AS). OSPFv3 maintains a network topology constructed from information about link states within the AS in a database on each router and uses this database to calculate shortest routes. OSPFv3 has the following advantages over RIPng:

- Less routing traffic

  OSPFv3 sends updates to other routers only when there is a change to the link state between routers. This generates far less traffic than routing protocols such as RIPng which exchange the entire routing table at fixed time intervals. In OSPFv3, each router distributes information about its own link state every 30 minutes.

- Elimination of routing loops

  Each router using OSPFv3 maintains an identical database, which it uses to select suitable routes. Therefore, unlike RIPng, OSPFv3 does not produce routing loops.

- Cost-based route selection

  When there is more than one route to a given destination, OSPFv3 selects the route with the lowest overall cost. Unlike RIPng, OSPFv3 allows route costs to be set in a flexible manner. This guarantees that the most desirable route is selected regardless of the number of hops.

- Operation of large-scale networks

  OSPFv3 can process routes whose total cost is 16777214 or less. Therefore, in contrast to RIPng which offers metrics in a range from 1 to 15, OSPFv3 is well suited for use in larger-scale networks where each route might traverse a large number of routers.

26.1.2 OSPFv3 functionality

Although OSPFv3 and OSPF are similar protocols, they operate independently of each other.

(1) Functional differences from OSPF

The following table describes the functional differences between OSPFv3 (IPv6) and OSPF (IPv4).

Table 26-1: Functional differences between OSPFv3 (IPv6) and OSPF (IPv4)

Functionality	OSPFv3(IPv6)	OSPF(IPv4)
Forwarding addresses for external AS routes	N	Y
Not-so-stubby areas (NSSA)	N	Y
Authentication	N	Y
Non-broadcast (NBMA) networks	N	Y
Equal-cost multipath	Y#	Y
Virtual links	Y	Y
Multi-backbone	Y	Y
Helper functionality for graceful restarts	Y	Y

Functionality	OSPFv3(IPv6)	OSPF(IPv4)
Restart functionality for graceful restarts	N	N
Stub router	Y	Y

Legend: Y: Supported, N: Not supported

#

The route selection method differs between OSPF (IPv4) and OSPFv3 (IPv6). In equal-cost mode, the smallest next-hop address is selected in OSPF (IPv4), while the next-hop address with the smallest router ID is selected in OSPFv3 (IPv6). If multiple next-hop addresses have the same router ID, the next-hop address for which the smallest interface ID has been advertised in a hello packet is selected.

(2) Domains

The Switch allows the division of an AS into up to four OSPFv3 networks, and can exchange, calculate, and generate routing information for OSPFv3 networks individually. This functionality is called OSPFv3 multi-backbone. Each independent OSPFv3 network is called an OSPFv3 domain.

OSPFv3 must be configured for each OSPFv3 domain.

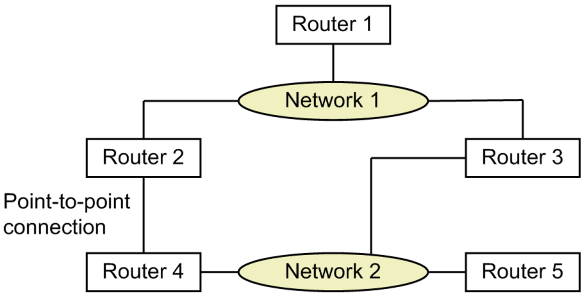
When OSPFv3 is used with VRFs, each VRF can be partitioned into a maximum of four domains.

26.1.3 Route selection algorithm

OSPFv3 uses the SPF (Shortest Path First) algorithm for route selection. Each router that runs OSPFv3 maintains a database containing information on all OSPFv3-enabled routers and the connections between them, as well as the connections between routers and networks. From this database, the protocol constructs a network topology that has routers and networks at its vertices, and connections between routers and between routers and networks at its edges. By applying the SPF algorithm to this topology, the OSPFv3 protocol computes a shortest-path tree that it uses to determine the routes to each vertex and address.

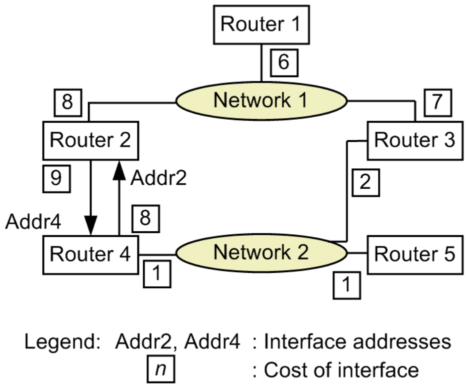
The following figure shows an example network configuration.

Figure 26-1: Example network configuration



The following figure shows an example of setting a topology and assigning costs when OSPFv3 is used on the network shown above.

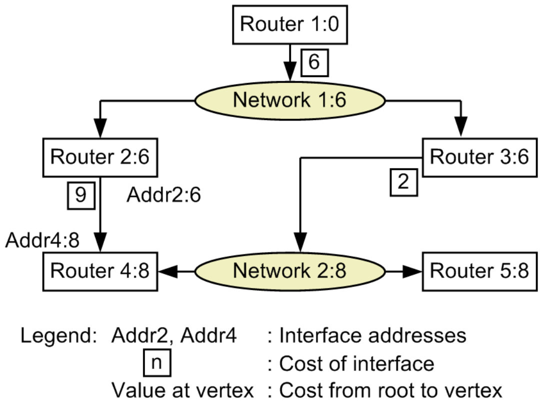
Figure 26-2: Example of setting topology and assigning costs



The cost value can be different according to packet transmission direction. For the point-to-point connection between Routers 2 and 4 in *Figure 26-2: Example of setting topology and assigning costs*, the cost of the route from Router 2 to Router 4 is 9, and the cost of the route from Router 4 to Router 2 is 8. For connections between routers and networks, you can only assign a cost to the connection from the router to the network. Paths leading from networks to routers always have a cost of zero.

Based on the topology in *Figure 26-2: Example of setting topology and assigning costs*, the following figure shows a shortest path tree generated with Router 1 as its root node. The cost to a given destination reflects the total transmission cost of the interfaces that the route traverses. For example, the cost of the route from Router 1 to Network 2 is 8 (6 (Router 1 to Network 1) + 0 (Network 1 to Router 3) + 2 (Router 3 to Network 2) = 8).

Figure 26-3: Shortest path tree with Router 1 as the root node



26.1.4 Advertising link states

(1) LSA types

Routing updates for OSPFv3 are called link state advertisements (LSA).

There are three main LSA types:

(a) Intra-area routing information

Announces the status of links to routers and networks for use by the SPF algorithm.

(b) Inter-area routing information

Announces the status of routes to other areas.

(c) External AS routing information

When an OSPFv3 router learns routing information for an external AS, the router can use OSPFv3 to distribute these routes to all other OSPFv3 routers. Any router that injects external AS routing information into an OSPFv3 domain is called an AS boundary router.

(2) External AS route

If you configure route redistribution filtering in the switch configuration, the switch will advertise external AS routes. An AS boundary router adds the following information to LSAs that advertise external routes:

- Metric

  The metric is used by the learning router for path selection among LSAs.

  Use the `default-metric` command to set the default metric.

- Metric types for external AS routes

  There are two types of metric, type 1 and type 2. The route priority and the calculation method when using the metric for route selection differ between the types. The default metric type is 2.

- Forwarding address (destination)

  Not used in the Switch.

- Tag

  Additional information can be advertised as a tag.

(3) Advertising external AS routes between domains

The multiple OSPFv3 domains connected via a single router operate as independent OSPFv3 networks. For this reason, unless there are configuration settings that specify route redistribution, the router does not distribute the routes of one OSPFv3 domain to the others. If you configure redistribution filtering in the configuration for OSPFv3 routes learned from other domains, routes from other domains are advertised as external AS routes. The following table describes the default filter attributes.

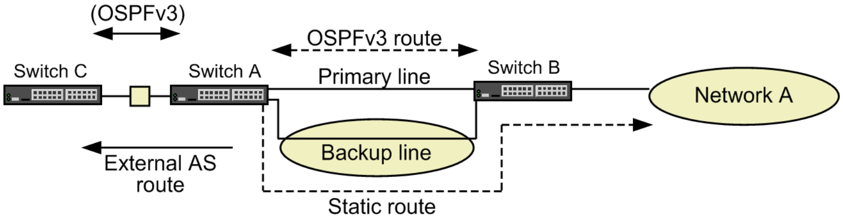
Table 26-2: Filter attributes when redistributing routes to other domains

Attribute	Default value	
	External AS route	Intra-area/inter-area route
Metric	The value specified by the `default-metric` command. If no value is specified, 20 is used.	The value specified by the `default-metric` command. If no value is specified, 20 is used.
Metric type	Type 2 for external AS routes	
Tag value	Inherits the tag value of the route.	The tag value is not advertised.

26.1.5 Example of injecting external AS routes

The following figure shows an example of injecting external AS routes in a configuration that includes a backup line.

Figure 26-4: External AS route injection in configuration using backup line



In OSPFv3, packets are exchanged at regular intervals to detect neighboring routers. If a backup line is incorporated into the OSPFv3 topology, the backup line stays active because it is called on consistently to carry these packets. However, if you want the backup line to enter an idle state when it is not required for communication, perform the configuration described below.

On Switch A, enable OSPFv3 on the primary line. On the backup line, configure a static route to Network A. As the distance of the static route, assign a larger value (lower priority) than the distance for an internal OSPFv3 route. This results in the internal AS route learned by the OSPFv3 protocol being selected as the route to Network A. In the event of a primary line failure, the relevant internal AS route is deleted from Switch A, and the static route is re-selected. However, the routing information to Network A will no longer exist in Switch C. To advertise an external AS route to Switch C that describes the static route to Network A, you need to configure route redistribution at Switch A. By doing so, information about a valid route to Network A can be injected into OSPFv3 should the primary line fail, without hello packets being exchanged over the backup line.

26.1.6 Criteria for route selection

The OSPFv3 protocol performs SPF calculation whenever LSAs are updated by LSA generation or learning. In SPF calculation, routes are selected based on the SPF algorithm. When a destination becomes unreachable by the SPF algorithm, the protocol deletes the route to that destination.

An area border router performs route selection separately via the SPF algorithm for each area it serves.

The following table describes the priority of route selection in OSPFv3. You cannot change this priority.

Table 26-3: Route selection priority

Priorit y	Selected item	Description
High	Type of routing information	An OSPFv3 internal AS route (intra-area or inter-area route) has priority over an external AS route.
↑	Learning source domain	If routes exist in more than one domain, the protocol selects the route with the smallest distance. When the distances are equal, the protocol selects the route with the smallest OSPFv3 domain number.
	Route destination type	• Internal AS route: An intra-area route has priority over an inter-area route. • External AS route: A route advertised by an AS boundary router within the same area has priority over a route advertised by a router in another area.
	External AS route type	An external AS route with metric type 1 has priority over an external AS route with metric type 2.
	Areas traversed by external AS route	For a router located at an area border, when a destination AS boundary router connects to more than one area, the area with the smallest cost to the AS boundary router is selected. When the cost values are equal, the area with the largest area ID is selected.
	Cost	• Internal AS route: The route with the smallest cost to the destination has priority. • Type 1 external AS route: The route for which the sum of the metric of the external AS route and the cost to the AS boundary router is smallest is selected. • Type 2 external AS route: The route with the smallest external AS route metric is selected. If the metrics are equal, the route with the smallest cost to the AS boundary router is selected.

Priorit y	Selected item	Description
↓	Router ID	The protocol selects the route with the smallest router ID of the router used as the next hop.
Low	Interface ID	The protocol selects the interface that learned the smallest interface ID in the hello packet from the router used as the next hop.

(1) Distance

When the Switch learns more than one route to the same destination via a number of protocols, it compares the distance of each route and applies the route with the highest priority.

In OSPFv3, you can set default distances on a per-domain basis. You can assign different distances to external AS routes, intra-area routes, and inter-area routes.

26.1.7 Equal-cost multipath

When an equal-cost multipath exists from the local router to a given destination in a topology that includes more than one forwarding destination router, the OSPFv3 protocol can balance the load to that destination by distributing packets among multiple next hops.

For internal AS routes, the Switch selects multiple paths which share the same learning source domain, destination type (inter-area or intra-area route), and cost. Similarly, for external AS routes, the switch selects multiple paths which share the same learning source domain, external AS route type, cost, and metric.

You can use the `maximum-paths` command to change the maximum number of paths. The default is 4.

26.1.8 Notes

(1) Notes on router IDs

The OSPFv3 protocol uses a router ID to identify routers when constructing the network topology.

The protocol cannot select paths correctly if the same router ID is set for multiple routers in the network design. To prevent problems, assign a unique router ID to each router in the network design.

When one router connects to more than one OSPFv3 domain, using the same router ID in every domain does not present a problem.

(2) Notes on route redistribution filter and learning filter

The OSPFv3 protocol advertises every LSA it learns from neighboring routers to its other neighbors.

You cannot use redistribution filtering to prevent the advertisement of routes learned by OSPFv3 within the same domain. Also, when using the route summarization functionality (the `ipv6 summary-address` command) to summarize OSPFv3 routes, you cannot prevent LSA within the same domain even if you configure route filtering to exclude summarization source routes from advertisement.

You can use the `distribute-list in` command to suppress learning of external AS routes that match the filter conditions. However, you cannot prevent OSPFv3 from learning and advertising LSAs. Therefore, OSPFv3 will also advertise the routes it has not itself learned.

(3) Notes on using the multi-backbone functionality

(a) Notes on using multi-backbone

In an environment that divides the network into more than one OSPFv3 domain, the advantages of OSPFv3 such as cost-based route selection and the avoidance of routing loops are lost when the selection and distribution of routes crosses OSPFv3 domain boundaries. When building a new network, if there is no need to divide the network into more than one OSPFv3 domain, we recommend that you design it to operate as a single OSPFv3 network.

(b) Notes on configuring multiple domains

If you need to advertise the switch address to more than one OSPFv3 domain, advertise it as an OSPFv3 external AS route. You cannot assign an interface to more than one OSPFv3 domain in the switch configuration.

(4) Restrictions on OSPFv3

The Switch conforms to RFC 2740 (OSPF for IPv6) but does not support the following functions due to software limitations:

- Route selection based on forwarding addresses for external AS routes
- Non-broadcast (NBMA) network

26.2 Configuration of basic OSPFv3 functionality

26.2.1 List of configuration commands

The following tables describe the configuration commands for basic OSPFv3 functionality.

Table 26-4: List of configuration commands related to enabling OSPFv3

Command name	Description
disable	Disables OSPFv3.
ipv6 ospf area	Defines the domain number and area ID of the area in which OSPFv3 operates.
router-id	Assigns a router ID (to identify a specific router).

Table 26-5: List of configuration command relating to external AS route advertisement

Command name	Description
default-metric	Sets a fixed value as the metric to a destination.
distribute-list out (OSPFv3)#	Configures redistribution filtering to control which routes are advertised.
redistribute (OSPFv3)#	Configures redistribution filtering to advertise external AS routes.

#

See *14. Route Filters (IPv4 and IPv6)* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

Table 26-6: List of configuration commands relating to route selection and learning

Command name	Description
distance ospf	Sets the distance for OSPFv3 routes.
ipv6 ospf cost	Sets the cost value for OSPFv3 routes.
maximum-paths	Sets the maximum number of equal-cost paths to a given destination.
timers spf	Configures the delay time between when OSPFv3 generates or learns an LSA and when it starts an SPF calculation, and the minimum time between consecutive SPF calculations.
distribute-list in (OSPFv3)#	Suppresses learning of external AS routes.

#

See *14. Route Filters (IPv4 and IPv6)* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

26.2.2 Overview of configuration

(1) Configuring basic OSPFv3 functionality

1. Use the `swrt_table_resource` command to set the IPv6 resource.

   This step is required to perform IPv6 routing.

2. Configure the IPv6 interfaces in advance.

3. Enable OSPFv3.

   Assign a unique router ID to each router.

If IPv4 interfaces exist, router IDs can be selected automatically.

4. Configure advertisement of external AS routes.

   This step is required to advertise the routes of other protocols in OSPFv3.

   It is also required if you intend to redistribute routes between domains using multi-backbone functionality.

5. Configure route selection.

   If you need to assign weighting to routes that travel over specific interfaces, use the `ipv6 ospf cost` command to assign cost values.

26.2.3 Configuring OSPFv3

Points to note

- Exchange of LSAs with neighboring routers is enabled on an interface specified by the `ipv6 ospf area` command.
- If you are not using multiple areas, make sure that the same area ID is set on all OSPFv3 routers.

Command examples

1.
```
(config)# ipv6 router ospf 1
```
   Places the router in OSPFv3 mode. Assigns 1 as the domain number.

2.
```
(config-rtr)# router-id 100.1.1.1
(config-rtr)# exit
```
   Assigns 100.1.1.1 as the router ID.

3.
```
(config)# interface vlan 1
(config-if)# ipv6 ospf 1 area 0
```
   Specifies that OSPFv3 runs in area 0 of domain 1.

26.2.4 Configuring advertisement of external AS routes

Points to note

- The example below shows how to use the `redistribute` command to specify what information (such as the metric, tags, and metric type) the protocol adds to redistributed routes. If you omit the metric in the `redistribute` command, the value specified by the `default-metric` command applies.
- You cannot control redistribution in the same domain for routes learned by OSPFv3.

Command examples

1.
```
(config)# ipv6 router ospf 1
(config-rtr)# default-metric 10
```
   Sets 10 as the default metric.

2.
```
(config-rtr)# redistribute static
```
   Advertises static routes using the default metric mentioned above.

26.2.5 Configuring route selection

Points to note

The example below shows how to use the `ipv6 ospf cost` command to specify the cost of sending a packet on a specific interface.

If you specify 1 in the `maximum-paths` command, the protocol does not establish an equal-cost multipath even when routes have an equal-cost value.

This subsection provides an example of configuring route selection when using single-path routes.

Command examples

1.
```
(config)# ipv6 router ospf 1
(config-rtr)# maximum-paths 1
(config-rtr)# exit
```
Sets 1 as the maximum number of OSPFv3 paths.

2.
```
(config)# interface vlan 1
(config-if)# ipv6 ospf 1 area 0
(config-if)# ipv6 ospf cost 10
(config-if)# exit
```
Sets the cost to 10.

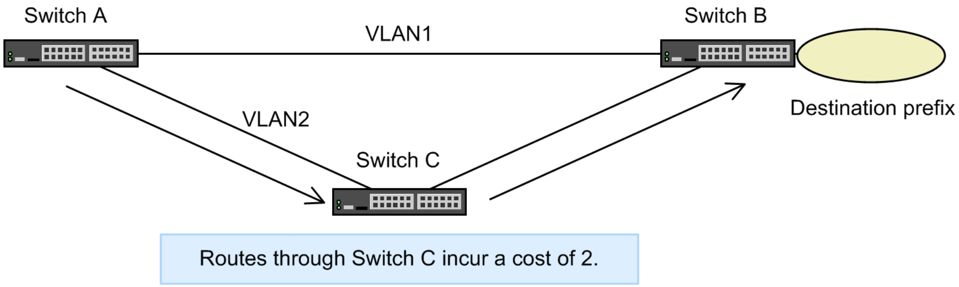
3.
```
(config)# interface vlan 2
(config-if)# ipv6 ospf 1 area 0
(config-if)# ipv6 ospf cost 2
```
Sets the cost to 2. The route through VLAN 2 has priority because it has a smaller cost value than VLAN 1.

26.2.6 Configuring multipath

Points to note

By adjusting cost values, you can establish an equal-cost multipath to a destination regardless of the number of routers through which the route passes.

Figure 26-5: Multipath configuration



This subsection provides an example of establishing an equal-cost multipath on Switch A.

Command examples

1.
```
(config)# interface vlan 2
(config-if)# ipv6 ospf 1 area 0
(config-if)# exit
```

2.
```
(config)# interface vlan 1
(config-if)# ipv6 ospf 1 area 0
(config-if)# ipv6 ospf cost 2
```
Sets the cost value of VLAN 1 to 2. This gives routes through VLAN 1 the same cost as routes through VLAN 2.

26.2.7 Applying OSPFv3 for a VRF

Points to note

The example below shows how to use the `ipv6 ospf area` command.

Command examples

This procedure uses the loopback address as the router ID and enables OSPFv3 for VRF 2.

1.
```
(config)# interface loopback 2
```
Switches to interface mode and specifies information about loopback 2.

2.
```
(config-if)# vrf forwarding 2
(config-if)# ip address 100.1.1.1
(config-if)# exit
```
Specifies VRF 2 and sets the IP address to 100.1.1.1.

3.
```
(config)# interface vlan 1
```
Switches to interface mode and specifies information about VLAN 1.

4.
```
(config-if)# vrf forwarding 2
(config-if)# ipv6 address 2001:db8:1:1::1/64
(config-if)# ipv6 enable
```
Specifies VRF 2 and sets the IPv6 address to 2001:db8:1:1::1/64.

5.
```
(config-if)# ipv6 ospf 1 area 0
```
Specifies that OSPFv3 runs in area 0 of domain 1.

26.3 Description of interfaces

26.3.1 OSPFv3 interface types

OSPFv3 classifies the interfaces that connect routers into the following three types, in terms of how packets are sent and received:

- Broadcast

  A broadcast type network in which multiple neighboring routers are managed in an integrated manner and addressed by multicast packets

- Non-broadcast (NBMA), which is not supported

  A broadcast type network in which multiple neighboring routers are managed in an integrated manner, but which has no broadcast or multicast capability

- Point-to-point

  An interface that directly connects to a single neighboring router. On a virtual link, this interface type operates as a virtual point-to-point interface.

(1) Notes on configuring OSPFv3 interfaces

OSPFv3 sometimes transmits packets that are equal in length to the Maximum Transfer Unit (MTU) value assigned to the interface. If the packet is longer than the Maximum Receive Unit (MRU, typically equal to the MTU) set on the interface that receives it, a situation might arise in which routers are unable to communicate with each other. This would not occur for normal traffic. For this reason, we recommend that you make sure the MTU of every network and every router that connects them is set to a value equal to or less than the MRU of all other interfaces when using OSPFv3.

26.3.2 Connecting with neighboring routers

(1) Hello packets

Routers running OSPFv3 send hello packets on each interface to verify that the links between routers are active. A hello packet is the means by which a router recognizes other routers running OSPFv3.

(2) Conditions for connecting routers

For each network that provides a direct connection between routers, the interface parameters below must be consistent among all connected routers. Routers that do not share the same parameters are not considered to be connected.

(a) Area ID

To establish a direct connection between routers, the interfaces on both routers must be assigned the same area ID.

(b) Hello interval and dead interval

In OSPFv3, a router sends hello packets so that directly connected routers will detect it. The hello interval is the sending interval for hello packets. The dead interval is the number of seconds that a router's hello packets must not have been seen before its neighbors declare the connection down. For a router to properly judge when a connection is lost, these two parameters must have the same values on the interfaces of the directly connected routers.

(c) Area configuration

The information reported to an area differs for stub areas and other area types. To allow OSPFv3 to determine that two routers are directly connected, the areas to which the interfaces belong must share the same stub configuration.

(d) Instance ID

OSPFv3 advertises an instance ID as a group ID to define groups of connected routers. The interfaces of routers that exchange routing information with each other must share the same instance ID.

26.3.3 Designated routers in broadcast networks

On a broadcast network, OSPFv3 selects a designated router and backup designated router to manage the connections between the network at the vertex of the topology and the routers directly connected to the network. To limit disruption to routing behavior, the backup designated router takes over the role of the designated router immediately if the designated router fails.

(1) Selecting the designated router and backup designated router

Each router advertises its priority to become a designated router on an interface in its hello packets.

If there is no designated router or backup designated router on an interface, OSPFv3 selects the router with the highest priority as the designated router. If the interface has a designated router but lacks a backup designated router, OSPFv3 selects the router with the next highest priority as the backup designated router. The designated router and backup designated router do not give up their roles if a router comes online with a higher priority.

When an interface of a given router has a priority of 0, that router will never be selected as the designated router or backup designated router in an area where the interface is connected.

When multiple routers exist in a broadcast network that is used to forward traffic, an interface of at least one router that connects to the network must have a priority of 1 or higher.

26.3.4 Transmitting LSAs

In OSPFv3, neighboring routers exchange link state advertisements (LSAs) to fill the gaps in their routing tables. When a router generates or receives a new LSA, it sends the LSA to all its neighboring routers. This ensures that an identical database is maintained by the Switch and its neighbors. The relationships maintained to create a synchronized database by exchanging LSAs with other routers are called adjacencies.

The Switch uses an LSA synchronization process to send its LSAs to all of its neighboring routers. In turn, the neighboring router sends the LSAs of the Switch to all of its neighbors. Those neighbors then send the LSAs to all of their neighbors. By this process, the LSAs of the Switch are distributed to every router in the area.

(1) LSA age

The age of an LSA is the length of time since it was generated. An LSA remains valid until its age reaches 3600 seconds or it is deleted by the originating router. The switch adds a delay time (set by the `ipv6 ospf transmit-delay` command) to the value of the `Age` field in valid LSAs it transmits.

26.3.5 Passive interfaces

An interface that has no neighboring OSPFv3 routers can be configured as a passive interface. When you apply OSPFv3 to a loopback interface, it becomes a passive interface.

A passive interface does not transmit or receive OSPFv3 packets.

Routes that directly connect to a passive interface are advertised as inter-area or intra-area routes.

26.4 Interface configuration

26.4.1 List of configuration commands

The following table describes the configuration commands for an OSPFv3 interface.

Table 26-7: List of configuration commands

Command name	Description
ipv6 ospf dead-interval	Sets the length of time that the router maintains adjacency after receiving no hello packets from a neighboring router.
ipv6 ospf hello-interval	Sets the sending interval for hello packets.
ipv6 ospf network	Sets the interface type (broadcast or point-to-point).
ipv6 ospf priority	Sets the router priority which determines the designated router for the network.
ipv6 ospf retransmit-interval	Sets the time between link-state advertisement (LSA) retransmissions.
ipv6 ospf transmit-delay	Sets the time required to send an OSPFv3 packet.
passive-interface (OSPFv3)	Sets the interface to passive mode.

The following table describes the configuration commands related to OSPFv3 operation.

Table 26-8: List of configuration commands (commands related to OSPFv3 operation)

Command name	Description
system mtu#1	Sets the MTU of the switch.
ip mtu#2	Specifies the MTU length of IP packets sent on the interface.
interface loopback#3	Sets the loopback interface (used as a passive interface in OSPFv3).

#1

See *10. Ethernet* in the manual *Configuration Command Reference Vol. 1 For Version 11.10*.

#2

See *2. IPv4, ARP, and ICMP* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

#3

See *3. Loopback Interface (IPv4)* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

26.4.2 Changing interface parameters

On interfaces with OSPFv3 enabled, behavior such as hello packet transmission conforms to the default values specified in the configuration. You can change how the interface behaves by using the `priority` parameter and `passive-interface` command, among others.

(1) Priority for designated router selection

Networks that contain a large number of routers place a heavy load on the designated router. When a router connects to multiple such networks, we recommend that you set its priority such that it does not become the designated router for more than one network.

Points to note

The greater the value of the `priority` parameter, the higher the priority of the router.

Command examples

1.
```
(config)# interface vlan 1
(config-if)# ipv6 ospf 1 area 0
(config-if)# ipv6 ospf priority 10
```
Sets the interface priority to 10.

(2) Passive interfaces

Points to note

The example below shows how to use the `passive-interface` command to configure a passive interface. If you specify the `ipv6 ospf cost` command, the interface assigns the specified cost value to the directly connected routes it advertises.

Command examples

1.
```
(config)# interface vlan 2
(config-if)# ipv6 ospf 1 area 0
(config-if)# ipv6 ospf cost 10
(config-if)# exit
```
Enables OSPFv3.

2.
```
(config)# ipv6 router ospf 1
(config-rtr)# passive-interface vlan 2
```
Sets VLAN 2 as a passive interface.

26.5 OSPFv3 operation

26.5.1 List of operation commands

The following table describes the operation commands for OSPFv3.

Table 26-9: List of operation commands

Command name	Description
show ipv6 route	Shows the entries in the routing table.
clear ipv6 route	Clears the IPv6 forwarding entries stored in the Switch and re-registers the routing entries.
show ipv6 ospf	Shows OSPFv3 information such as the domain number, information about neighboring routers, interface information, and LSAs.
clear ipv6 ospf	Clears information about the OSPFv3 protocol.
show ipv6 vrf	Shows the IPv6 information of a VRF.
show ipv6 interface ipv6-unicast	Shows information about the IPv6 interfaces on the Switch recognized by the unicast routing program.
debug ipv6	Shows the packets being routed by IPv6 routing protocols in real time.
show processes cpu unicast#	Shows the CPU usage of a unicast routing program.
restart unicast#	Restarts the unicast routing program.
debug protocols unicast#	Starts the operation message display for event log information output by a unicast routing program.
no debug protocols unicast#	Stops the operation message display for event log information output by a unicast routing program.
dump protocols unicast#	Outputs the trace information and control table information collected by the unicast routing program to a file.
erase protocol-dump unicast#	Deletes the file of trace information and control table information generated by the unicast routing program.

#

See *8. Routing Protocol Common to IPv4 and IPv6* in the manual *Operation Command Reference Vol. 2 For Version 11.10.*

26.5.2 Checking the domain

While OSPFv3 is active, you can use the `show ipv6 ospf` operation command to check relevant settings such as router IDs and distances.

Figure 26-6: Results of executing the show ipv6 ospf command

```
>show ipv6 ospf
Date 20XX/07/14 12:00:00 UTC
OSPFv3 protocol: ON

Domain: 1
Router ID: 172.16.1.1
Distance:
  Intra Area: 10, Inter Area: 10, External: 150
Flags: <AreaBorder ASBoundary>
SPF Interval: 7s, SPF Delay: 3s
Area: 1, Interfaces: 1
    Network Range                                State
    -                                            -
```

26.5.3 Checking information about neighboring routers

You can use the `show ipv6 ospf neighbor` operation command to check the link-local address (`Neighbor Address`), neighbor state (`State`), ID (`Router ID`), and priority of neighboring routers.

On an OSPFv3 interface, the designated router (DR) forms adjacencies with the other routers. You can monitor its progress by checking the `State` field in the command output.

`Full` appears as the status of routers that have formed an adjacency. Otherwise, the router is in the process of forming an adjacency with the designated router and cannot learn OSPFv3 routes at its interfaces.

You can examine the neighboring routers in more detail by executing the `show ipv6 ospf interface` and `show ipv6 ospf neighbor detail` operation commands. The command output includes the interface state, network type, and connectivity with neighboring routers.

- Make sure that the OSPFv3 network type of the interface is the same as that of its neighboring routers.
- When `DR` or `P to P` appears as the interface state, make sure that the status of each neighboring router in the neighbor list is `Full`.
    - A value other than `Full` indicates that the interface has not formed an adjacency with the neighboring router. Check the neighboring routers.
- When `BackupDR` or `DR Other` appears as the interface state, check whether the neighbor list contains a router that is eligible to be elected the DR.
    - If a DR exists in the neighbor list but its neighboring routers have a status other than `Full`, this indicates that the interface has not formed an adjacency with the neighboring router. Check the neighboring routers.
    - If no DR exists, this might indicate that no priority has been assigned to the Switch or its neighbors. Check the priority setting for the switch and neighboring routers.

Figure 26-7: Results of executing the show ipv6 ospf neighbor command

```
>show ipv6 ospf neighbor
Date 20XX/07/14 12:00:00 UTC
Domain: 1
Area: 0
Neighbor Address          State               Router ID    Priority Interface
fe80::1000:00ff:fe00:2002 Full/BackupDR       172.16.10.12        1 VLAN0003
fe80::1000:00ff:fe00:2003 Full/DR Other       172.16.10.13        1 VLAN0003
fe80::1000:00ff:fe00:2004 Exch Start/DR Other 172.126.10.14       1 VLAN0003
```

Figure 26-8: Results of executing the show ipv6 ospf interface command (with individual interface specified)

```
>show ipv6 ospf interface vlan 10
```

```
Date 20XX/05/30 12:00:00 UTC
Domain: 1
Area: 0
Interface ID: 2,Link Local Address : fe80::1000:00ff:fe00:0001%VLAN0010
    IPv6 Address: -
    MTU: 1460, DDinPacket: 71, LSRinPacket: 120, ACKinPacket:72
    Router ID: 172.16.1.1, Network Type: P to P, State: P to P
    DR: none, Backup DR: none
    Priority: 0, Cost: 1, Instance: 0
    Transmit Delay: 1s
    Intervals:
        Hello: 10s, Dead: 40s, Retransmit: 5s

  Neighbor List (1):
    Address                     State        Router ID         Priority
    fe80::1000:00ff:fe00:2002   Full         172.16.1.2           0
>
```

Figure 26-9: Results of executing the show ipv6 ospf neighbor command (detail)

```
>show ipv6 ospf neighbor detail
Date 20XX/05/30 12:00:00 UTC
Domain: 1
Area: 0
Interface: VLAN0020, Interface State: Backup DR
    Neighbor Address: fe80::1000:00ff:fe00:2002, State: Full/DR
    Neighbor Router ID: 172.16.10.11, Priority: 1
    Neighbor Interface ID: 2
    DR: 172.16.10.11, Backup DR: 172.16.10.10
    Last Hello: 6s, Last Exchange: 45d 12h
    DS: 0, LSR: 0, Retrans: 0, <Master>
    Neighbor Address: fe80::1000:00ff:fe00:2001, State: Full/DR Other
    Neighbor Router ID: 172.16.10.12, Priority: 1
    Neighbor Interface ID: 404
    DR: 172.16.10.11, Backup DR: 172.16.10.10
    Last Hello:  3s, Last Exchange:  1m  8s
    DS: 0, LSR: 0, Retrans: 1, <>
>
```

26.5.4 Checking interface information

You can use the `show ipv6 ospf interface` operation command to display information about interfaces running OSPFv3, including the interface name (`Interface`), state (`State`), priority (`Priority`), cost value (`Cost`), and the number of neighboring routers (`Neighbor`).

The command does not display information for IPv6 interfaces that are down.

Figure 26-10: Results of executing the show ipv6 ospf interface command

```
>show ipv6 ospf interface
Date 20XX/07/14 12:00:00 UTC
Domain: 1
Area: 0
  Interface         State     Priority  Cost   Neighbor
  VLAN0003          DR           1      1      1

Area: 1
  Interface         State     Priority  Cost   Neighbor
  VLAN0004          BackupDR     10     20     10
```

26.5.5 Checking LSA information

(1) Checking the number of LSAs

You can use the `show ipv6 ospf database database-summary` operation command to check how many LSAs are in the OSPFv3 database.

Figure 26-11: Results of executing the show ipv6 ospf database database-summary command

```
>show ipv6 ospf database database-summary
Date 20XX/07/14 12:00:00 UTC
Domain: 1
Local Router ID: 172.16.251.141
Area: 0
  [Linklocal scope]
    Link             :     1
    Opaque-Link      :     1
    Grace            :     0
    ------------------------
    Total                  2
  [Area scope]
    Router           :     2
    Network          :     0
    Inter-Area-Prefix:     0
    Inter-Area-Router:     1
    Intra-Area-Prefix:     1
    ------------------------
    Total                  4

[AS scope]
    External:     1
>
```

(2) Checking advertisement information of LSAs

You can use the `show ipv6 ospf database` command to display a list of LSAs. You can also check the LSID and ages for each LSA type. Each LSA can be identified by its advertising router ID (`Advertising Router`) and LSID.

Confirm that LSA is configured as follows on the Switch:

1. The switch advertises Router-LSA.

   `LSID` in the command output indicates the LSA ID, which is set to 0 for Router-LSA advertised by the Switch.

2. If there is an interface on which the Switch is a designated router, the Switch advertises Network-LSA.

   `LSID` in the command output indicates the interface ID that has the same value as the `LSID` indicated for Link-LSA.

3. Link-LSA is advertised to each interface.

   `LSID` in the command output indicates the interface ID.

4. If the Switch is an AS border router, it advertises its routes as AS External Link LSAs.

   If you want to check the destination of advertised routes, use the `show ipv6 ospf database external` command to display detailed information.

   Figure 26-12: Results of executing the show ipv6 ospf database command

```
>show ipv6 ospf database
Date 20XX/07/14 12:00:00 UTC
Domain: 1
Local Router ID: 172.16.251.141
Area: 0
  LS Database: Router-LSA
    Advertising Router  LSID      Age   Sequence  Checksum  Length
    10.0.1.3            00000000  221   8000000b  0dad      40
    172.16.251.141      00000000  275   80000002  6d7a      24
  LS Database: Network-LSA
    Advertising Router  LSID      Age   Sequence  Checksum  Length
    10.0.1.3            00000000  221   8000000b  0dad      40
    172.16.251.141      00000002  226   80000002  94f6      32
  LS Database: Inter-Area-Prefix-LSA
    Advertising Router  LSID      Age   Sequence  Checksum  Length
    10.0.1.3            00000001  210   80000002  7d89      32
    255.255.255.255     00000001  210   80000003  7d89      32
  LS Database: Inter-Area-Router-LSA
    Advertising Router  LSID      Age   Sequence  Checksum  Length
    172.16.251.141      0301000a  262   80000002  4e74      32
    172.16.251.143      0301000a  262   80000002  4e74      32
  LS Database: Link-LSA
  Interface: VLAN0003
    Advertising Router  LSID      Age   Sequence  Checksum  Length
    10.0.1.3            00000001  336   80000001  87f0      44
    172.16.251.141      00000001  399   80000002  7e8d      44
  Interface: VLAN0004
    Advertising Router  LSID      Age   Sequence  Checksum  Length
    172.16.251.141      00000002  399   80000002  7e8d      44
  LS Database: Intra-Area-Prefix-LSA
    Advertising Router  LSID      Age   Sequence  Checksum  Length
    172.16.251.141      00000001  275   80000002  0d9a      52
```

```
AS:
  LS Database: AS-external-LSA
    Advertising Router  LSID      Age   Sequence  Checksum  Length
    172.16.251.141      00000001  275   80000002  0d9a      52
```

You can use the `show ipv6 ospf database external` command to check information such as the destination prefix and metric of the external AS routes.

Figure 26-13: Results of executing the show ipv6 ospf database external command

```
> show ipv6 ospf database external
Date 20XX/07/14 12:00:00 UTC
Domain: 1
Local Router ID: 100.1.1.1
LS Database: AS-external-LSA
Advertising Router: 100.1.1.1
    LSID: 00000000, Age: 6, Length: 44
    Sequence: 80000001, Checksum: 5373
    Prefix: 3ffe:4:1::1/128
       Prefix Options: <LocalAddress>
       Type: 2, Metric: 20, Tag: ----
```

Chapter
27. Extended OSPFv3 Functionality [OS-L3SA]

This chapter describes the extended functionality of the OSPFv3 protocol.

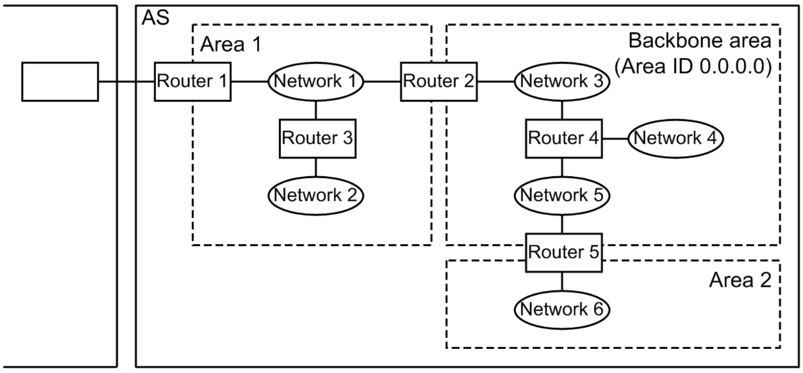
27.1 Description of areas and area division functionality

27.1.1 Area border

The OSPFv3 protocol allows you to divide an autonomous system (AS) into a number of areas to minimize routing traffic and cut down the processing time required by the SPF algorithm.

The following figure shows an example of an OSPFv3 network topology in which an AS is divided into several areas.

Figure 27-1: Example OSPFv3 network topology using area division



A router that attaches to multiple areas is called an area border router. For example, Router 2 and Router 5 in the figure above are area border routers.

Information about the connection status of a given area is not reported outside that area. A router contains no information about the connection status of areas to which it is not connected.

(1) Backbone area

An area with an area ID of 0.0.0.0 is called backbone. The backbone has a special role in a split-area topology. If an AS is divided into areas, one of the areas must be designated as the backbone area. Be careful to avoid creating a configuration in which an AS has multiple backbones. In such a configuration, information about inter-area routes is spread across multiple backbones. This can result in unreachable routes being generated or the protocol failing to select the best route.

An area border router uses the backbone to disseminate routing information to other areas. For this reason, an area border router must connect to the backbone area.

(2) Notes on area division

Although dividing an AS into areas reduces the load on the routers and minimizes routing traffic, it also adds a level of complexity to the OSPFv3 algorithm. In particular, you can run into difficulties putting the appropriate fault handling in place. We recommend that you do not use area division unless you have a specific need to reduce the load on routers or networks.

(3) Notes on area border routers

- An area border router runs an iteration of the SPF algorithm for each area it serves. It provides functionality that summarizes the topology information for the areas to which it is connected for distribution to other areas in the AS. Thus, an area border router that is associated with a large number of areas is subject to heavy loads. We recommend a network configuration that limits the number of areas each area border router serves.
- When an area has only one area border router, a fault in that router can partition the area from the backbone, resulting in a loss of connectivity with other areas. When building a network, we recommend that you place more than one area border router in areas with servers and AS

boundary routers that support mission-critical functions and connections, so that there are sufficient alternate routes available if a router fails.

27.1.2 Route control for divided areas

Each area border router summarizes the routing information for each non-backbone area to which it belongs, and then sends it to all other routers in the backbone area. The summarized routing information for the backbone area and the summarized routing information sent from other areas to the backbone area are reported to the routers in non-backbone areas.

When a router uses this summarized routing information to determine the route to an address, the route will traverse the area border router that is the source of the summarized routing information. Therefore, a route between different areas always passes through the backbone area.

When advertising routing information to other areas, the area border router summarizes topology information derived from link states between routers and networks and the costs of those links. This information describes the cost between the area border router and specific routers and networks. The summarized information is called inter-area routing information. The information that describes routes to networks is advertised in Type 3 LSAs, and the information that describes routes to AS boundary routers is advertised in Type 4 LSAs.

(1) Route summarization in area border routers

The following table describes the summarization and suppression of routing information, and the summary information that the area border router reports to other areas.

Table 27-1: Route summarization and suppression, and summaries sent to external areas

Network addresses in area	Summarization and suppression	Summary reported to other areas
3ffe:501:811:10::/60 3ffe:501:811:20::/61 3ffe:501:811:28::/61 3ffe:501:811:30::/60	None	3ffe:501:811:10::/60 3ffe:501:811:20::/61 3ffe:501:811:28::/61 3ffe:501:811:30::/60
3ffe:501:811:10::/60 3ffe:501:811:20::/61 3ffe:501:811:28::/61 3ffe:501:811:30::/60	3ffe:501:811::/59 3ffe:501:811::20::/60	3ffe:501:811::/59 3ffe:501:811:20::/60 3ffe:501:811:30::/60
3ffe:501:811:10::/60 3ffe:501:811:20::/61 3ffe:501:811:28::/61 3ffe:501:811:30::/60 3ffe:501:811:ff00::/58	3ffe:501:811::/58 (suppressed) 3ffe:501:811:ff00::/56	3ffe:501:811:ff00::/56

You can use configuration commands to specify an address range that the area border router condenses into a single summarized route when summarizing the topology information for a given area. To specify an address range, use the `area range` command to specify a prefix and the prefix length. You can also specify a parameter that suppresses advertisement of the address range.

If the area contains even one network that matches the prefix range specified in the configuration command, the area border router summarizes all networks within the range into routing information with the prefix as its destination, and then reports it to other areas. The router does not report information for each network in the range outside area boundaries. The summarized routing information will adopt the highest cost among its constituent routes.

If you suppress advertisement, no summary LSAs for networks in the address range are advertised outside area boundaries, nor are the routes that are summarized under the prefix advertised to other areas. As a result, routes to addresses within the specified address range remain hidden from other areas.

27.1.3 Stub areas

You can configure an area as a stub area if it is not a backbone area and does not contain an AS boundary router. To do so, use the `area stub` configuration command.

The area border router does not employ external AS routes in stub areas. This reduces the amount of routing traffic in stub areas and minimizes the demand that route updates and route selection place on router resources. The area border router employs a default route in place of the external routes in the stub area.

If you execute the `area stub` command with the `no-summary` parameter specified, the area border router does not advertise routes from other areas (inter-area routes) into the stub. Routers in the stub area must use the default route to send any traffic outside the area.

27.1.4 Virtual links

In OSPFv3, you can simulate a point-to-point connection between two area border routers. Both routers must be located in a non-backbone area that is not configured as a stub area. The line can be used as an interface to the backbone area. This virtual line is called a virtual link. The area that carries its routes is called the transit area.

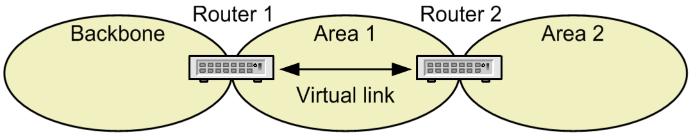
The use of virtual links is described based on the following three examples:

- Providing virtual connections for areas not physically connected to the backbone
- Joining multiple backbones
- Adding redundant connectivity to prevent faults from partitioning the backbone

(a) Providing virtual connections for areas not physically connected to the backbone

In the figure below, Area 2 is not connected to the backbone. If you configure a virtual link between Router 1 and Router 2 that uses Area 1 as its transit area, Router 2 considers itself an area border router because it has a connection to the backbone. Area 2 can now connect to the backbone through Router 2.

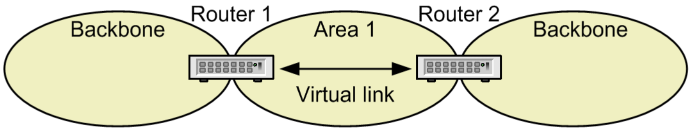
Figure 27-2: Connecting an area to the backbone



(b) Joining multiple backbones

The figure below shows an AS containing two backbone areas. This partition in the backbone can cause some destinations to become unreachable. You can avoid this problem by configuring a virtual link between Router 1 and Router 2 that uses Area 1 as its transit area to join the backbones.

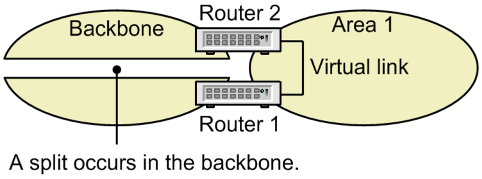
Figure 27-3: Providing a connection between backbones



(c) Adding redundant connectivity to prevent faults from partitioning the backbone

In the figure below, the connection between Router 1 and Router 2 is lost when a network fault occurs in the backbone, causing the backbone to be partitioned. By configuring a virtual link between Router 1 and Router 2 that uses Area 1 as its transit area, you can provide a secondary route (or a primary route if the cost of the virtual link is sufficiently lower than that between Router 1 and Router 2 on the backbone) that acts as an alternate route if the backbone is partitioned.

Figure 27-4: Providing an alternate route in case the backbone is partitioned



27.1.5 Operation of virtual links

A virtual link must be configured in both routers that serve as the endpoints. The routers at each end of the virtual link send and receive OSPFv3 packets with neighboring routers over the virtual link by using IPv6 global or IPv6 site-local addresses. The routers use IPv6 addresses set for an interface that connects to the transit area.

Note the following when operating a virtual link:

- An IPv6 global address or IPv6 site-local address must be set for an interface that connects to the transit area. A virtual link cannot work with a neighboring router that has not advertised any IPv6 global or IPv6 site-local address.
- The cost of a virtual link is the cost of the route between the routers at the two endpoints over the transit area.
- The route for normal traffic might differ from that of routing information traffic over the virtual link if the route between the two endpoints over the transit area is an equal-cost multipath.

(1) Connections with neighboring routers

When the virtual link is active, hello packets are sent to the neighboring routers on the virtual link to detect connectivity. A virtual link becomes active when the transit area contains a path to the router at the other end of the virtual link.

A hello packet is the means by which a router recognizes other routers running OSPFv3.

Use the `area virtual-link` command to configure hello packets in the context of the virtual link. The `dead-interval` parameter must be longer than any interval value (as set by the `ipv6 ospf dead-interval` command) set for an interface in any network that forms the route between the end points of the virtual link in the transit area. If you specify a shorter value and a network failure occurs that affects routes in the transit area, the virtual link might be shut down before the routers have time to switch to alternate routes.

The resending interval for LSAs (as set by the `retransmit-interval` parameter of the `area virtual-link` command) must comfortably exceed the expected round-trip time for packets between the endpoint routers of the virtual link.

27.2 Area configuration

27.2.1 List of configuration commands

The following tables describe the configuration commands used to designate a stub area and to operate the Switch as an area border router.

For details about the commands for the functionality described in *26. OSPFv3 [OS-L3SA]*, see *Table 26-5: List of configuration command relating to external AS route advertisement*, *Table 26-6: List of configuration commands relating to route selection and learning*, and *Table 26-7: List of configuration commands*.

Table 27-2: List of configuration commands related to areas

Command name	Description
area default-cost	Sets the cost of the default route advertised to stub areas.
area range	Summarizes inter-area routes on an area border router under a single prefix for advertisement to other areas.
area stub	Configures an area as a stub area.
area virtual-link	Establishes a virtual link.

Table 27-3: List of configuration commands related to enabling OSPFv3

Command name	Description
disable	Disables OSPFv3.
ipv6 ospf area	Defines the domain number and area ID of the area in which OSPFv3 operates.
router-id	Assigns a router ID (to identify a specific router).

27.2.2 Overview of configuration

(1) Configuring a stub area when the switch is not an area border router

1. Use the `swrt_table_resource` command to set the IPv6 resource.

   This step is required to perform IPv6 routing.
2. Configure the IPv6 interfaces in advance.
3. Configure the area to operate as a stub area.
4. Enable OSPFv3.

(2) Configuring an area border router

1. Use the `swrt_table_resource` command to set the IPv6 resource.

   This step is required to perform IPv6 routing.
2. Configure the IPv6 interfaces in advance.
3. Configure the area to operate as a stub area.
4. Configure route summarization.
5. Enable OSPFv3.

   Configure more than one area. At this time, you must configure an interface that connects to area 0 (the backbone), or establish a virtual link.
6. Establish virtual links.

27.2.3 Configuring a stub area

Points to note

An area border router advertises a default route into areas specified by the `area stub` command.

A stub area must be configured on every router in the same area.

Command examples

1.
```
(config)# ipv6 router ospf 1
```
   Places the router in OSPFv3 mode. Assigns 1 as the domain number.

2.
```
(config-rtr)# area 1 stub
```
   Designates area 1 as a stub area.

3.
```
(config-rtr)# router-id 100.1.1.1
(config-rtr)# exit
```
   Assigns 100.1.1.1 as the router ID.

4.
```
(config)# interface vlan 2
(config-if)# ipv6 ospf 1 area 1
```
   Specifies that OSPFv3 runs in area 1 of domain 1.

27.2.4 Configuring an area border router

Points to note

You can use the `area range` command with the `not-advertise` parameter specified to prevent the networks in the prefix range from being advertised outside the area.

For a given area, you can specify more than one address range for route summarization and for route advertisement suppression. A router or network in the area can be assigned an address not included in any of the address ranges you specify. However, when building a network, by allocating addresses to suit the topology first and then configuring summarization for address ranges in a manner appropriate for the topology, you can efficiently reduce the amount of OSPFv3 routing traffic without impairing the ability of the protocol to select the best routes.

This subsection provides an example of configuring route summarization for an area border router associated with areas 0 and 1.

Command examples

1.
```
(config)# ipv6 router ospf 1
(config-rtr)# area 0 range 3ffe:501:811::/59
```
   Having learned routes in the range of prefix 3ffe:501:811::/59 in area 0, the router advertises a summarized route to area 1.

2.
```
(config-rtr)# area 1 range 3ffe:501:811::20::/60
(config-rtr)# exit
```

Having learned routes in the range of prefix 3ffe:501:811::20::/60 in area 1, the router advertises a summarized route to area 0.

3.
```
(config)# interface vlan 3
(config-if)# ipv6 ospf 1 area 0
(config-if)# exit
(config)# interface vlan 1
(config-if)# ipv6 ospf 1 area 1
```

Sets up the interfaces on which OSPFv3 is enabled to configure the area border router for areas 0 and 1.

27.2.5 Configuring virtual links

Points to note

The example below shows how to use the `area virtual-link` command to specify the router ID of the virtual link neighbor.

Command examples

1.
```
(config)# interface vlan 1
(config-if)# ipv6 ospf 1 area 1
(config-if)# exit
```

Enables OSPFv3.

2.
```
(config)# ipv6 router ospf 1
(config-rtr)# area 1 virtual-link 10.0.0.1
(config-rtr)# area 1 virtual-link 10.0.0.2
```

Sets the virtual link neighbor in transit area 1.

27.3 Description of graceful restart

27.3.1 Overview

In OSPFv3, a device that uses a graceful restart to restart the OSPFv3 protocol is called a restarting router. The functionality that implements this graceful restart is called the graceful restart functionality. A neighboring router that assists the restarting router in this process is called a helper router. The functionality by which the helper router assists the graceful-restart process is called the helper functionality.

The Switch supports an implementation of the helper functionality.

27.3.2 Helper functionality

When acting as a helper router, the Switch preserves the routes associated with the restarting router for the duration of the graceful-restart process.

(1) Operating conditions for the helper functionality

The switch can operate as a helper router as long as the following conditions are met:

- It is not already acting as a helper for another restarting router in the same domain.

  In a given domain, the Switch can only act as a helper for one router at a time. If only one router is restarting, the switch provides the helper functionality on every interface that connects to the restarting router.

- The local router has not executed the graceful-restart process as the restarting router.
- It is not waiting to receive an ACK for an OSPFv3 update packet sent to the restarting router.

(2) Scenarios where the helper functionality fails

The switch continues to operate as a helper router until the restarting router re-establishes adjacency or a notification is received indicating that the restarting router has exited the graceful-restart process.

To avoid potential conflicts with the routes maintained by the restarting router, the switch stops its helper functionality and recalculates its OSPFv3 routes if any of the following events occurs:

- A new LSA (excluding periodic updates) is learned from a neighbor and advertised to the restarting router.
- The OSPFv3 interface goes down.
- The switch loses or establishes adjacency with a router other than the restarting router, thus generating an LSA update.
- More than one router is restarting in the same OSPFv3 domain.
- You use the `graceful-restart mode` command to disable helper mode in the switch configuration.

27.4 Graceful restart configuration

27.4.1 List of configuration commands

Configure the OSPFv3 helper functionality on the Switch if you intend to use the OSPFv3 restart functionality on any of its neighboring routers.

The following table describes the configuration commands for the graceful restart functionality.

Table 27-4: List of configuration commands

Command name	Description
graceful-restart mode	Enables the helper functionality.
graceful-restart strict-lsa-checking	Allows the helper router to terminate the graceful-restart process if it detects a discrepancy between its own LSA database and that of the restarting router.

27.4.2 Helper functionality

Points to note

The example below shows how to enable the helper functionality on the switch. If you omit this configuration, the switch will not act as a helper router.

Command examples

1.
```
(config)# ipv6 router ospf 1
(config-rtr)# graceful-restart mode helper
```
Enables the helper functionality on the switch.

27.5 Description of stub routers

27.5.1 Overview

If the connection with neighboring router is incomplete or unstable, the entire routing might become unstable. This situation might occur when starting, restarting or adding a router to the network. In such a situation, the OSPFv3 protocol allows a router to disseminate routing information that persuades nearby devices to use alternate routes. A router that uses this behavior is called a stub router in OSPFv3. This functionality allows you to prevent instability in a specific device from destabilizing the rest of the network.

(1) Max metric

A stub router advertises the maximum cost of 65535 for all of its connected OSPFv3 interfaces. This action gives OSPFv3 routes through the stub router a higher cost than their alternatives.

However, if the interface is a stub network (having no neighboring routers), the stub router will advertise the cost specified in the configuration. The routes advertised by the stub router might take priority over the stub network or external AS routes.

The maximum metric advertised for routes through the stub router causes nearby routers to give priority to alternate routes. You can also use the address of the stub router for Telnet management access or to exchange BGP4+ routing information.

27.5.2 Stub router operation

You can use the `max-metric router-lsa` configuration command to enable stub router functionality at the domain level. You can also specify whether the router functions as a stub router on a permanent basis, or only at startup.

(1) Permanently operating as a stub router

The router advertises maximum cost for its routes on a permanent basis. It continues to operate as a stub router until the user deletes the setting.

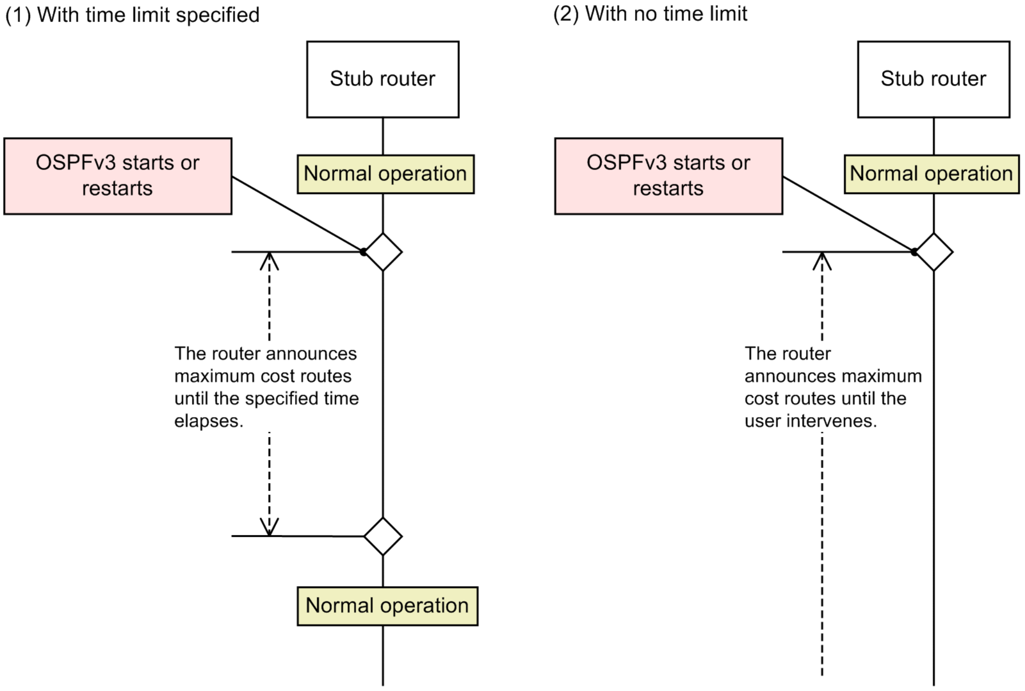
(2) Operating as a stub router at startup

The router advertises maximum cost for its routes when either of the events below occurs. It continues to operate as a stub router until the timer specified in the configuration expires.

- The routing program restarts.
- The switch starts.

You can stop the router from operating as a stub router by executing the `clear ipv6 ospf stub-router` operation command or by removing the setting from the configuration. The following figure shows the operation of a stub router:

Figure 27-5: Stub router operation


(1) With time limit specified
Stub router
OSPFv3 starts or restarts
Normal operation
The router announces maximum cost routes until the specified time elapses.
Normal operation
(2) With no time limit
Stub router
OSPFv3 starts or restarts
Normal operation
The router announces maximum cost routes until the user intervenes.

(3) Notes

1. Do not change the configuration of a stub router that is participating as a helper router in a graceful restart. The stub router might begin or cease operation causing disruption to the helper process.
2. If a router is set up to always operate as a stub router and you then change the settings so that it operates as a stub router after startup, it will immediately cease to operate as a stub router.
3. You cannot configure a virtual link through a stub router.

   If the cost in the transit area is higher than 65535, the virtual link neighbor will mark the virtual link as unreachable.

27.6 Stub router configuration

27.6.1 List of configuration commands

You can configure the Switch as a stub router to de-prioritize the routes that pass through it. This process increases the metric of routes that traverse the stub router.

The following table describes the configuration command for a stub router.

Table 27-5: List of configuration commands

Command name	Description
max-metric router-lsa	Configures the switch to operate as a stub router.

27.6.2 Stub router functionality

Points to note

The example below shows how to specify that the switch is to operate as a stub router. If you omit the `on-startup` parameter, the switch operates as a stub router on a permanent basis.

Command examples

1.
```
(config)# ipv6 router ospf 1
(config-rtr)# max-metric router-lsa
```
Enables stub router functionality.

27.7 Operation for extended OSPFv3 functionality

27.7.1 List of operation commands

The following table describes the operation commands for the extended functionality of the OSPFv3 protocol.

Table 27-6: List of operation commands

Command name	Description
show ipv6 ospf	Shows domain information (including the status of area borders and graceful restarts) and the area configuration.
clear ipv6 ospf	Clears information about the OSPFv3 protocol. You can stop the switch from acting as a stub router by executing this command with the `stub-router` parameter.

27.7.2 Checking area borders

On an area border router, you can use the `show ipv6 ospf` operation command to make sure that `AreaBorder` appears as the router type in the `Flags` section of the command output.

You can also check whether inter-area route summarization has been applied correctly.

Figure 27-6: Results of executing the show ipv6 ospf command

```
>show ipv6 ospf
Date 20XX/07/14 12:00:00 UTC
OSPFv3 protocol: ON

Domain: 1
Router ID: 172.16.1.1
Distance:
  Intra Area: 10, Inter Area: 10, External: 150
Flags: <AreaBorder ASBoundary>
SPF Interval: 7s, SPF Delay: 3s
Graceful Restart: Helper
    Helper Status : Finished  20XX/07/08 14:12:22
Area: 0, Interfaces: 2
    Network Range                                 State
    3ffe:501:ffff:100::/64                        DoNotAdvertise
    3ffe:501:ffff:200::/64                        Advertise
Area: 1, Interfaces: 1
    Network Range                                 State
    -                                             -
```

27.7.3 Checking areas

You can check whether the areas you set in the switch configuration have been applied correctly. To display a list of areas, execute the `show ipv6 ospf` command with the `area` parameter specified.

Figure 27-7: Results of executing the show ipv6 ospf command (area parameter)

```
>show ipv6 ospf area
Date 20XX/07/14 12:00:00 UTC
Domain: 1
ID               Neighbor  SPFcount  Flags
0                3         14        <ASBoundary>
10               2         8         <Stub>
>
```

27.7.4 Checking graceful restart

You can check the status of the graceful restart functionality by executing the `show ipv6 ospf` operation command.

Figure 27-8: Results of executing the show ipv6 ospf command

```
>show ipv6 ospf
Date 20XX/07/14 12:00:00 UTC
OSPFv3 protocol: ON

Domain: 1
Router ID: 172.16.1.1
Distance:
  Intra Area: 10, Inter Area: 10, External: 150
Flags: <AreaBorder ASBoundary>
SPF Interval: 7s, SPF Delay: 3s
Graceful Restart: Helper
    Helper Status : Finished  20XX/07/08 14:12:22
Area: 0, Interfaces: 2
    Network Range                                 State
    3ffe:501:ffff:100::/64                        DoNotAdvertise
Area: 1, Interfaces: 1
    Network Range                                 State
    -                                             -
```

Chapter

28. BGP4+ [OS-L3SA]

This chapter describes the specifications of BGP4+ and provides notes on using it.

28.1 Description of basic functionality

28.1.1 Overview

The BGP4 routing protocol used at the interface backbone level has been extended to BGP4+ (Multiprotocol Extensions for Border Gateway Protocol 4) to make it available to protocols other than IPv4. BGP4+ can handle all the routing information used on the Internet.

The following table describes the functional differences between BGP4+ (IPv6) and BGP4 (IPv4).

Table 28-1: Functional differences between BGP4+ (IPv6) and BGP4 (IPv4)

Functionality	BGP4+ (IPv6)	BGP4 (IPv4)
EBGP, IBGP peering, and route distribution	Y	Y
Route filtering and BGP attribute manipulation	Y	Y
Community	Y	Y
Route reflection	Y	Y
Confederations	Y	Y
Capability negotiation	Y	Y
Route refresh capability	Y	Y
Multipathing	Y	Y
Peer groups#1	Y	Y
Route flap dampening#2	Y	Y
BGP4 MIB#2	N	Y
TCP MD5 authentication	Y	Y
Graceful restart	Y#3	Y#3
Maximum number of learned routes	Y	Y

Legend: Y: Supported, N: Not supported

#1: A peer group is a grouping of external peers and member AS peers, or a grouping of internal peers.

#2: Not supported in VRFs

#3: Only the receiving router functionality is supported.

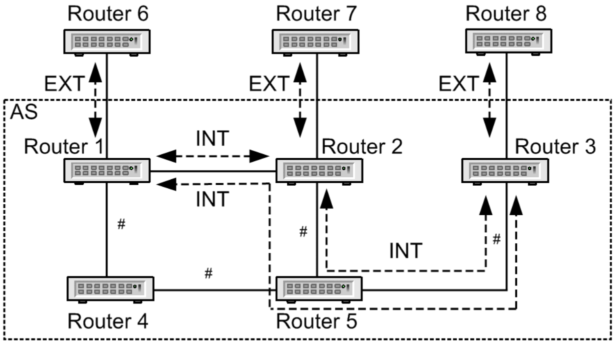
28.1.2 Peer types and connectivity

Because BGP4+ operates between ASs, the routing information it handles consists of AS path information to a destination network (the series of ASs that a packet traverses to reach the destination network). A router running BGP4+ is known as a BGP4+ speaker. This BGP4+ speaker forms a peering relationship with another BGP4+ speaker in order to exchange routing information.

The Switch uses two types of peers: external peers and internal peers. In addition to these two types, a third category called a member AS peer is used when configuring a confederation. For details about member AS peers, see *28.4.10 Confederations*.

Use the appropriate peer type for your network configuration. The following figure shows internal and external peers.

Figure 28-1: Internal and external peers



Legend: Routers 1, 2, and 3: Internal BGP4+ speaker
Routers 6, 7, and 8: External BGP4+ speaker
Routers 4 and 5: Internal non-BGP4+ speaker
INT: Internal peer
EXT: External peer
#: IGP is enabled.

(1) External peer

External peer relations are formed between BGP4+ speakers that belong to different ASs. The link-local or global interface address of the directly connected interface is used as the IP address for peering.

The peers can be connected by a different address (for example, the device address) by using the `neighbor ebgp-multihop` configuration command.

In *Figure 28-1: Internal and external peers*, Routers 1 and 6, Routers 2 and 7, and Routers 3 and 8 are in external peer relationships.

(2) Internal peer

Internal peer relations are formed between BGP4+ speakers in the same AS. BGP4+ uses TCP (port 179) to establish connections between peers. Therefore, although there is no requirement for all BGP4+ speakers to be physically fully meshed, internal peers must be logically fully meshed with each BGP4+ speaker within the AS. This is because an internal peer does not announce received routing information to other internal peers. When route reflection or confederation is used, this condition is relaxed.

In *Figure 28-1: Internal and external peers*, Routers 1 and 2, Routers 1 and 3, and Routers 2 and 3 are in internal peer relationships.

(3) Peering using the device address

In the Switch, an IPv6 address can be assigned to a device. This is called a device address. Using the device address as the IPv6 address of an external or internal peer can eliminate the effects of a particular physical interface (TCP connection) on peering.

For example, suppose that the interface IPv6 address is used for connectivity between the internal peers Router 1 and Router 2 in *Figure 28-1: Internal and external peers*. If the interface between these two routers fails, a peering relationship cannot be established. In contrast, by using the device address as an internal peer IPv6 address, internal peering relations can be established between Routers 1 and 2 via Routers 4 and 5 even though the interface between Routers 1 and 2 is down.

Notes on using the device address

Before peers can use a device address for interconnectivity, they must learn information about routes to that address via static routing or by an IGP (such as RIPng or OSPFv3). The Switch handles the device address as directly connected routing information.

Notes on internal peering via a router that is not a BGP4+ speaker

When routing information is reported via an internal peer that is not a BGP4+ speaker (for example, via the route from Router 2 to Router 3), the non-BGP4+ speaker must have already learned the routing information via an IGP. This is necessary to prevent IPv6 packets sent from the destination BGP4+ speaker from being discarded by a non-BGP router that has not learned the route to the originating router. For example, in *Figure 28-1: Internal and external peers*, this would prevent an IPv6 packet from Router 3 from being discarded by Router 5.

28.1.3 Route selection

From the routing information to a given destination learned by the various protocols, the Switch selects the most appropriate route in accordance with each protocol's independent route selection criteria. If the various protocols generate more than one route to the same destination, the distances are compared and the route with the highest priority is selected.

In BGP4+, the best route is selected from the multiple routes to the same destination learned using BGP4+, following the steps in the table below. If multiple routes to a given destination are found to exist after route selection by each protocol (RIPng, OSPFv3, and static), their distances are compared and the route with the highest priority is entered in the routing table.

For route selection in a confederation, see *28.4.10 Confederations*.

Table 28-2: Route selection priority

Priority	Description
High	Selects the route with the greatest weight.
↑	Selects the route with the largest `LOCAL_PREF` attribute value.
	Selects the route whose `AS_PATH` attribute has the smallest number of ASs.[#1]
	Selects by the `ORIGIN` attribute value, preferring `IGP`, `EGP`, and `Incomplete`, in that order.
	Selects the route with the smallest `MED` attribute value.[#2]
	Selects a route learned by an external peer in preference to a route learned by an internal peer.
	Selects the route with the closest next hop (the smallest metric of the IGP routes used when resolving the next-hop address).
	Selects the route whose peer has the smallest BGP identifier (router ID). For a route having the `ORIGINATOR_ID` attribute, however, the `ORIGINATOR_ID` attribute values are compared instead of the peers' BGP identifiers.[#3]
↓	Selects the route with the smallest `CLUSTER_LIST` attribute length.[#4]
Low	Selects the route whose learning source peer has the smallest address.[#3]

#1

The `AS_SET` path type of the `AS_PATH` attribute is counted as one AS.

#2

Route selection by `MED` attribute value applies only to redundant routes learned from the same neighboring AS. However, redundant routes learned from different neighboring ASs can also be compared by specifying the `bgp always-compare-med` configuration command.

#3

If the routes received from an external peer have different peer BGP identifiers (router IDs), the route selected at the previous step is used, and route selection based on peer BGP identifier and learning source peer address is skipped. However, routes that have different peer BGP identifiers can also be considered by specifying the `bgp bestpath compare-routerid` configuration command.

#4

When a route does not have the `CLUSTER_LIST` attribute, it is compared assuming a `CLUSTER_LIST` attribute length of 0.

Weights and the BGP attributes involved in route selection (the `LOCAL_PREF`, `AS_PATH`, `ORIGIN`, `MED`, and `MP_REACH_NLRI` attributes) are explained below.

(1) Weight

A weight is a valuation applied to routes per learning source peer. Routes with higher weights are preferred.

Weights in the range from 0 to 255 can be used in the Switch. The default is 0.

(a) Changing a weight

Using the `neighbor weight` configuration command, you can change the weight of routes learned from a peer.

(2) LOCAL_PREF attribute

The `LOCAL_PREF` attribute is reported among routers in the same AS. When there is more than one route to a destination network, the `LOCAL_PREF` attribute indicates the preferred route. Routes with higher `LOCAL_PREF` values are preferred.

`LOCAL_PREF` attribute values in the range from 0 to 65535 can be used in the Switch. The default is 100.

(a) Changing the LOCAL_PREF attribute default

Using the `bgp default local-preference` configuration command, you can change the `LOCAL_PREF` attribute value of routing information imported to the Switch from an external peer.

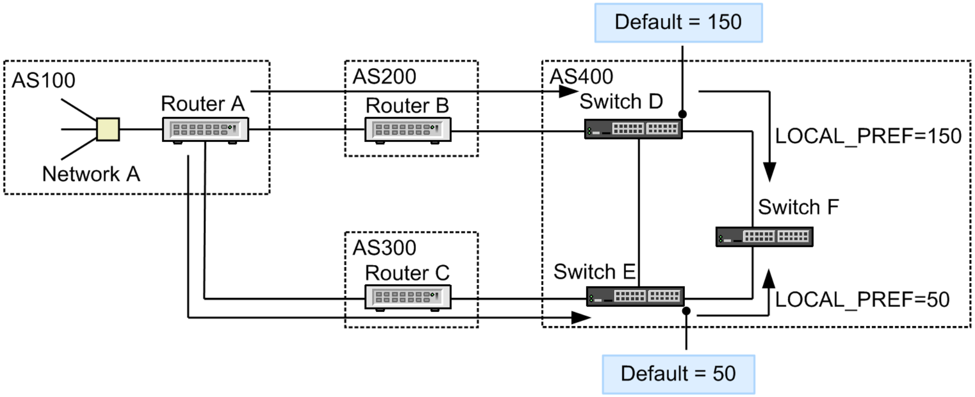
(b) Changing the LOCAL_PREF attribute on a per-filter basis

Using the `set local-preference` configuration command in combination with a learned route filter and advertised route filter, you can change the `LOCAL_PREF` attribute value of routing information imported to the Switch or reported to other switches.

(c) Example of route selection by LOCAL_PREF attribute

The following figure shows route selection using the `LOCAL_PREF` attribute.

Figure 28-2: Route selection by LOCAL_PREF attribute



In the above figure, AS400 receives routing information for network A from AS200 and AS300. Assume that the `LOCAL_PREF` attribute value of Switch D is set to 150 and that the `LOCAL_PREF` attribute value of Switch E is set to 50. Thus, a LOCAL_PREF value of 150 would apply when

Switch D notifies Switch F of the routing information received from AS200, and a LOCAL_PREF value of 50 would apply when Switch E notifies Switch F of the routing information received from AS300. Of the two sets of network-A routing information destined for Switch F, the routing information from Switch D has a higher `LOCAL_PREF` value than the routing information from Switch E. Therefore, the routing information from Switch D via AS200 is selected.

(3) ORIGIN attribute

The `ORIGIN` attribute indicates the source of the routing information. The following table describes the attribute values.

Table 28-3: ORIGIN attribute

ORIGIN attribute	Description
IGP	The route was generated within the AS.
EGP	The route was learned via EGP.
Incomplete	The route was learned by some other means.

When there are multiple paths to a given destination, the `ORIGIN` attribute is selected in the order `IGP`, `EGP`, and then `Incomplete`.

(a) Changing the ORIGIN attribute value

Using the `set origin` configuration command in combination with a route filter, you can change the `ORIGIN` attribute value of routing information imported to the Switch or reported to other switches.

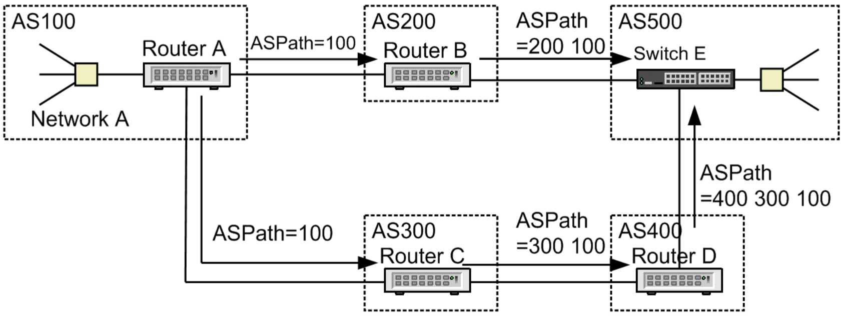
(4) AS_PATH attribute

The `AS_PATH` attribute is a list of AS numbers that a route traverses to reach the destination network. When routing information is advertised to another AS, the local AS number is added to the `AS_PATH` attribute in that routing information. Using learning filter information and advertising filter information in combination with the `set as-path prepend count` configuration command, you can add more than one local AS number to the `AS_PATH` attribute. This is useful for selecting a specific path when there are a number of routes to a given destination network.

(a) Example of route selection by AS_PATH attribute

The following figure shows route selection using the `AS_PATH` attribute.

Figure 28-3: Route selection by the AS_PATH attribute

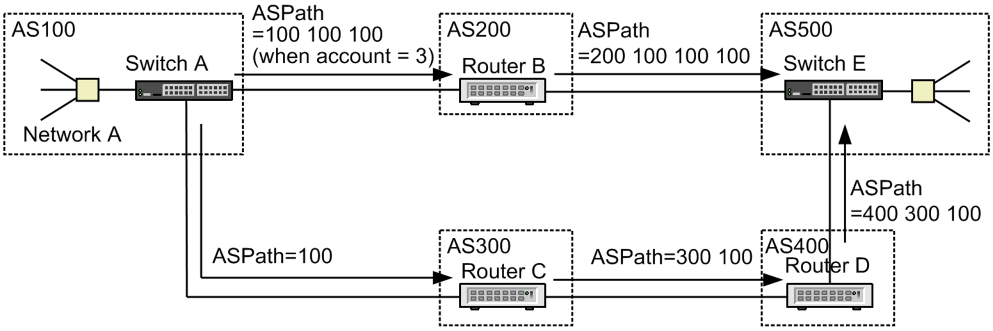


Suppose that Router A reports information about network A in its local AS to Switch E. When the routing information reaches AS500, it will have an `AS_PATH` attribute of "200 100". However, if the routing information is instead directed through AS300 and AS400, the `AS_PATH` attribute when the routing information reaches AS500 will be "400 300 100". Therefore, Switch E determines that the route via AS200 has the smaller `AS_PATH` attribute and selects that route.

(b) Route selection when using the set as-path prepend count command

The following figure shows an example of the `set as-path prepend count` configuration command.

Figure 28-4: Example of using the set as-path prepend count command



In the above figure, to force Switch E to select the route from Router A via AS300 and AS400, you need to add multiple local AS numbers to the `AS_PATH` attribute of the routing information sent from Router A to AS200. For example, if you add three local AS numbers, the `AS_PATH` attribute when the routing information reaches AS500 via AS200 is "200 100 100 100". Therefore, Switch E determines that the route through AS300 and AS400 has the smaller `AS_PATH` attribute and selects that route.

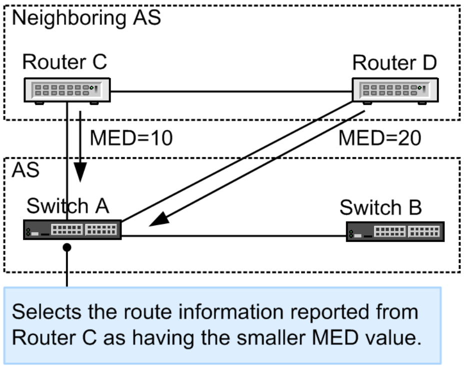
(5) MED attribute

The `MED` attribute determines the priority of multiple BGP4+ routes to a given destination learned from the same neighboring AS. Routes with lower `MED` attribute values are preferred. Using the `bgp always-compare-med` configuration command, you can use `MED` attribute values for priority selection among BGP4+ routes learned from different neighboring ASs.

(a) Example of route selection by MED attribute

The following figure shows route selection using the `MED` attribute.

Figure 28-5: Route selection by MED attribute


Neighboring AS
Router C
Router D
MED=10
MED=20
AS
Switch A
Switch B
Selects the route information reported from Router C as having the smaller MED value.

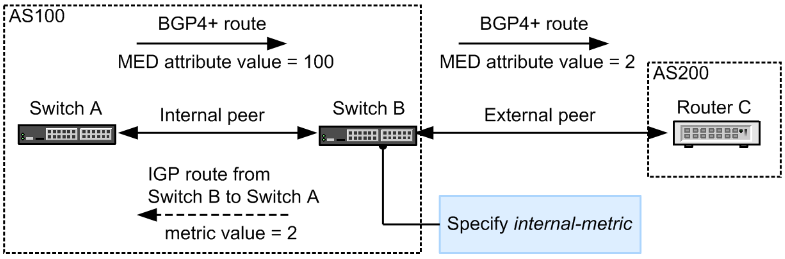
Suppose that Router C advertises routing information for a given destination network with a `MED` attribute value of 10, and Router D advertises routing information to the same network with a `MED` attribute value of 20. Switch A will therefore select the routing information reported from Router C as the preferred route to the destination network.

(b) Changing the MED attribute value

Using learned filter information and advertised filter information in combination with the `set metric` configuration command, you can change the `MED` attribute value of routing information imported to the Switch or reported to other switches.

By specifying `internal` in `set metric-type`, you can set the metric of the IGP route used for next-hop resolution as the `MED` attribute value of the BGP4+ route to be advertised. The following figure shows an example of using `set metric-type internal`.

Figure 28-6: Example of using the set metric-type internal command



In the above figure, Switches A and B are internal peers. Suppose that you want to set the metric 2 of the IGP route from Switch B to Switch A as the `MED` attribute value when the BGP4+ routing information that was reported from Switch A with a `MED` attribute value of 100 is advertised by Switch B to Router C. You can do so by specifying the `set metric` configuration command on Switch B.

(6) Next-hop information in the MP_REACH_NLRI attribute

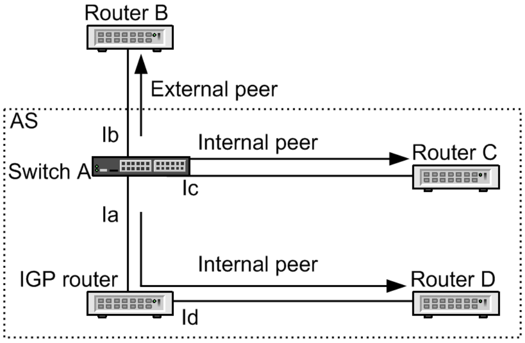
In BGP4+, the value of the `NextHop` attribute received from a BGP4+ peer is ignored. Instead, the next-hop information in the `MP_REACH_NLRI` attribute is adopted as the next hop of the route.

In BGP4+, the global address and link-local address (only for external peers) on the local side of the interface used for peering are set for next-hop information in the `MP_REACH_NLRI` attribute when routing information is reported to a remote BGP4+ speaker. These addresses are set only when IPv6 global addresses are used for peering.

(a) Example of setting the next hop

The following figure shows examples of setting the next hop of the reported routing information when advertising a route learned from a BGP4+ peer.

Figure 28-7: Examples of setting the next hop of the reported routing information when advertising a route learned from a BGP4+ peer



- Routing information destined for external peer Router B

  The global and link-local addresses `Ib` on the Switch A side of the interface between Switch A and Router B are assigned as the next hop in the `MP_REACH_NLRI` attribute. Switch A is not concerned with which address Router B uses as the next hop.

- Routing information from directly connected external peer Router B

  If the next hop information in the `MP_REACH_NLRI` attribute contains either the global address or the link-local address, that address is used as the next hop. If both addresses are present, the link-local address is used.

- Routing information destined for internal peer Router C

  The global address set for the next hop in the MP_REACH_NLRI attribute in the routing information received from Router B is used.

  If the next hop in the routing information received from Router B does not contain the global address, the global address on the Switch side of the interface between Switch A and Router C is set.

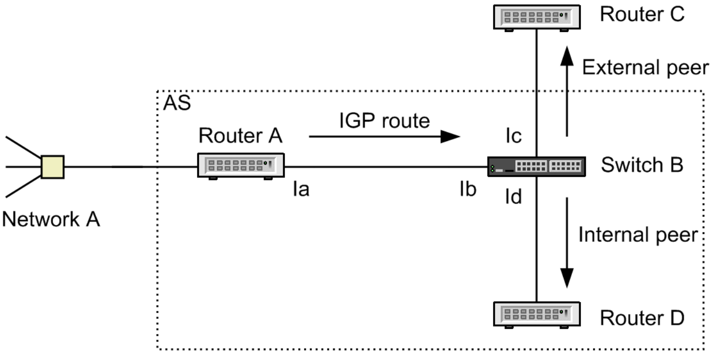
- Routing information destined for internal peer Router D

  The global address set for the next hop in the MP_REACH_NLRI attribute in the routing information received from Router B is used.

  If the next hop in the routing information received from Router B does not contain the global address, the global address on the Switch side of the interface between Switch A and Router D is set.

The following figure shows examples of setting the next hop in the routing information that is reported when advertising an IGP route via BGP4+.

Figure 28-8: Examples of setting the next hop in routing information that is reported when advertising an IGP route via BGP4+



- Routing information destined for external peer Router C

  The global and link-local addresses *Ic* on the Switch B side of the interface between Switch B and Router C are set as the next hop in the MP_REACH_NLRI attribute. Switch B is not concerned with which address Router C uses as the next hop.

- Routing information destined for internal peer Router D

  The interface address *Ia* of Router A, which is the next-hop address to network A specified by the IGP route, is set as the next hop in the MP_REACH_NLRI attribute. Note, however, that if *Ia* is a link-local address, the global address *Id* on the Switch B side of the interface between Switch B and Router D is set as the next hop in the MP_REACH_NLRI attribute.

(b) Modifying the next hop

Use the following configuration commands to modify the next hop in the MP_REACH_NLRI attribute:

- `neighbor next-hop-self` command

  Changes the next hop in the MP_REACH_NLRI attribute used when routing information received from a BGP4+ peer is advertised to BGP4+ peers to the peering address of the local router. This command applies when only the global address is set for the next hop in the MP_REACH_NLRI attribute in the routing information. This command does not apply when using route reflection or when advertising an IGP route to an internal peer by BGP4+.

- `neighbor always-nexthop-self` command

  Changes the next hop in the MP_REACH_NLRI attribute used when advertising routing information to an internal peer to the peering address of the local router. This command also applies when using route reflection or when advertising an IGP route by BGP4+.

- `neighbor set-nexthop-peer` command

  Changes the next hop in the `MP_REACH_NLRI` attribute of learned routing information to the remote peer address used for peering.

(c) Resolving the next hop

When BGP4+ routing information has been learned from an internal peer, the path to reach an address indicated by the next-hop information of the `MP_REACH_NLRI` attribute specified by an IGP route, a static route, or a directly connected route. Among the routes that can reach the next hop of the BGP4+ route, the route having the longest destination mask length is selected and the path of that route is used as the BGP4+ route.

By using the `bgp nexthop` configuration command, you can specify the protocol type and prefix of the route to be used in resolving the next hop.

If the route to which the next hop resolves is a static route and the `noinstall` parameter is specified, the BGP4+ route is suppressed.

28.1.4 BGP4+ functionality for VRFs

(1) Overview

BGP4+ operates independently within networks logically divided by the VRF functionality. Note that a peer connection between different VRFs is not possible.

(2) Notes on using the BGP4+ functionality for a VRF

On the Switch, a route imported from a different VRF or the global network inherits the PATH attribute from the source route. Therefore, if a target route is advertised from the Switch, a route loop might be detected on neighboring devices.

1. Note on when different VRFs or global networks use the same AS number

   Be careful when you import a route between VRFs or global networks that use the same AS number. If the target route is advertised to the import-source VRF or global network, the route is not treated as a valid route because an AS loop is detected on neighboring devices. The Switch provides the `neighbor as-override` configuration command that overwrites the first AS number set for the AS_PATH attribute of the VRF or the global network with the AS number of the Switch. If you use BGP4+ for a connection between VRFs or global networks that have the same AS number, be sure to use this command.

   Note that the `neighbor as-override` configuration command cannot resolve AS loops in neighboring devices when the VRFs or global networks are not directly connected to the Switch and using the same AS number. If the Switch is used as a neighboring device, you can use the `neighbor permit-asloop` configuration command, which treats an AS loop route as a valid route. Use this command when using the same AS number in VRFs or global networks. Note that this command must be used in carefully designed networks because its use can increase the risk of a route loop.

2. Note on when different VRFs or global networks use the same router ID or cluster ID (route reflection)

   When different VRFs or global networks use the same router ID (originator ID) or when the route reflectors in different VRFs or global networks use the same cluster ID, the routes are not treated as valid routes because the route reflectors detect a loop. Note this when designing networks.

28.1.5 Notes on using BGP4+

Note the following restrictions when configuring a network using BGP4+.

(1) BGP4+ restrictions

With some exceptions, the Switch is compliant with RFC 4271 (Border Gateway Protocol 4), RFC 1997 (Communities), RFC 5492 (Capabilities Advertisement), RFC 2918 (Route Refresh Capability), RFC 4456 (Route Reflection), RFC 5065 (Confederations), RFC 4760 (BGP4 Multiprotocol Extensions), and RFC 2545 (Extensions for IPv6 Application to RFC 4760). Differences are due to the functional limitations of the software, and are described in the table below. The Switch supports only BGP version 4.

Table 28-4: Differences with the RFC

RFC No.	RFC		Switch
RFC 4271	Path attribute: NEXT_HOP	If the external peer to which the route is being advertised shares a subnet with one of the interfaces of the announcing BGP speaker, the speaker may use the IP address associated with such an interface in the NEXT_HOP attribute. This is known as a "first party" NEXT_HOP attribute.	The "first party" NEXT_HOP attribute is not supported.
		When sending a message to an external peer, and the peer is multiple IP hops away from the speaker (also known as a "multihop EBGP"), the BGP speaker may be configured to propagate the NEXT_HOP attribute without modification.	The Switch changes the NEXT_HOP attribute to the address of the local router when advertising a route to an external peer.
	Path attribute: MULTI_EXIT_DISC	A BGP speaker must implement a mechanism (based on local configuration) that allows the MULTI_EXIT_DISC attribute to be removed from a route. If a BGP speaker is configured to remove the MULTI_EXIT_DISC attribute from a route, then this removal must be done prior to determining the degree of preference of the route and prior to performing route selection.	A mechanism for removing the MULTI_EXIT_DISC attribute from a route is not implemented.
	Connection collision detection	Upon receipt of an OPEN message, the local system must examine all of its connections that are in the OpenConfirm state. The local system also examines connections in an OpenSent state if it knows the BGP identifier of the peer by some means outside the protocol.	Upon receipt of an OPEN message, the Switch examines all connections that are in the OpenSent state or Connect state.
	BGP FSM: Idle state	For a peer that has previously transitioned to the Idle state because of an error, the time between consecutive generations of Start events (if such events are generated automatically) must increase exponentially. The value of the initial timer is 60 seconds. The time is doubled for each consecutive retry.	The initial value of the timer until the peer changes from the Idle to the Start state is from 16 to 36 seconds.
	BGP FSM: Active state	If the transport protocol connection succeeds, the local system clears the Connect Retry timer, and completes initialization.It then sends an OPEN message to its peer, sets its hold timer, and changes its state to Open Sent. A hold timer value of 4 minutes is suggested.	The hold timer is 180 seconds (3 minutes) by default, or, if specified, the value in the configuration.
	Frequency of route advertisement	Min Route Advertisement Interval determines the minimum interval between route advertisements to a particular destination by a BGP speaker. This rate limit applies on a per-destination basis. However, the value of Min Route Advertisement Interval is set on a per-BGP peer basis.	Min Route Advertisement Interval is not supported.

RFC No.	RFC		Switch
		`Min AS Origination Interval` determines the minimum time that must elapse between successive advertisements of `UPDATE` messages that report changes within the advertising BGP speaker's own ASs.	`Min AS Origination Interval` is not supported.
	Jitter	To minimize the likelihood that the distribution of BGP messages by a given BGP speaker will contain peaks, jitter should be applied to the timers associated with `Min AS Origination Interval`, `Keep Alive`, and `Min Route Advertisement Interval`.	Jitter is not applied.
	Route summarization	Routes that have different `MULTI_EXIT_DISC` attributes must not be aggregated.	Routes that have different `MULTI_EXIT_DISC` attributes are aggregated.
		When aggregating routes that have different `NEXT_HOP` attributes, the `NEXT_HOP` attribute of the aggregated route must identify an interface on the BGP speaker that performs the aggregation.	The `NEXT_HOP` attribute is not set for an aggregated route.
	BGP timers	The suggested value for the `Connect Retry` timer is 120 seconds.	The `Connect Retry` timer value is a variable (from 16 to 148 seconds) that changes according to the `Connect Retry` count.
		The suggested value for the `Hold Time` attribute is 90 seconds.	The default `Hold Time` value is 180 seconds. The value of `Hold Time` in the configuration is used if specified.
		The suggested value for the `Keep Alive` timer is 30 seconds.	The default value of the `Keep Alive` timer is one-third of the `Hold Time` value. The value of `Keep Alive` in the configuration is used if specified.
		Two optional timers (`DelayOpenTimer` and `IdleHoldTimer`) may be supported by BGP.	`DelayOpenTimer` and `IdleHoldTimer` are not supported.
RFC 2545	Only when the next hop to be reported and the destination peer are on the same network is the link-local next hop also reported.		The Switch performs the same processing described in the RFC only when the external peer is connected via a direct network.
	Transport protocol		The TCP connection used for a BGP4+ session is IPv4 or IPv6. The Switch supports reporting of only IPv6 routing information via IPv6 TCP.
	Peering address type		IPv4 or IPv6 addresses are used for BGP4+ peering. The Switch supports only IPv6 addresses. BGP4+ connection to an internal peer by using an IPv6 link-local address is not supported.

RFC No.	RFC	Switch
RFC 5065	All BGP speakers participating as members of a confederation must recognize the `AS_CONFED_SET` and `AS_CONFED_SEQUENCE` path types.	`AS_CONFED_SET` is not supported. Upon receipt of a route containing `AS_CONFED_SET`, the Switch ignores that path type.

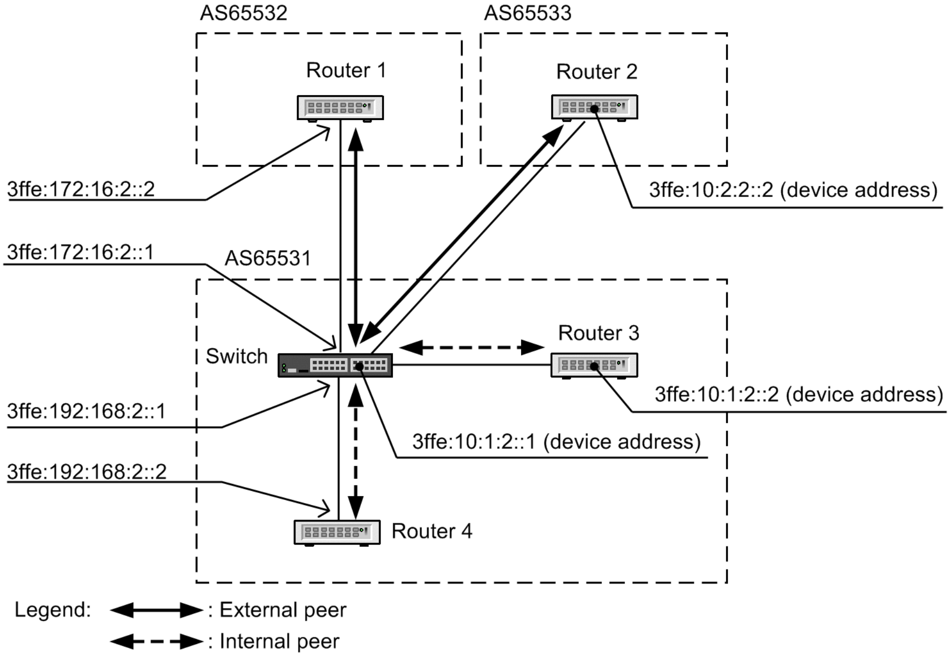
(2) Notes on peering with a directly connected interface

When the Switch uses external peers or member AS peers between BGP speakers on a directly connected interface and the Switch is in a transit area of an OSPFv3 virtual link on that interface, peers cannot be connected. In such cases, you can specify the `neighbor ebgp-multihop` configuration command to establish the peer connection.

28.2 Configuration of basic functionality

This section describes how to configure the basic BGP4+ functionality, based on the following configuration example.

Figure 28-9: Connectivity example



- Between the Switch and Router 1: Interface address
- Between the Switch and Router 2: Device address
- Between the Switch and Router 3: Device address
- Between the Switch and Router 4: Interface address

28.2.1 List of configuration commands

The following tables describe the configuration commands for peer types and BGP4+ connections, and operation commands.

Table 28-5: List of configuration commands

Command name	Description
address-family ipv6	Switches to `config-router-af` (`ipv6`) mode to configure the global network information or `config-router-af` (`ipv6 vrf`) mode to configure VRF information.
bgp always-compare-med	Enables comparison of the `MED` attribute of routes learned from different ASs.
bgp bestpath compare-routerid#1	Enables selection among routes learned from external peers based on the peer's BGP identifier (router ID).
bgp default local-preference	Sets the default of the `LOCAL_PREF` attribute of routes advertised by BGP4+.
bgp nexthop	Specifies the `route-map` to be used in resolving the next hop of a BGP4+ route.
bgp router-id#1	Specifies the identifier of the local router.
default-information originate	Advertises a default route to all peers.

Command name	Description
default-metric	Sets the default of the MED attribute of routes advertised by BGP4+.
disable#1	Disables BGP4/BGP4+.
distance bgp	Sets the distance of a route learned via BGP4+.
neighbor activate	Enables an exchange of routes between an IPv6 address family and peers.
neighbor description	Sets a supplementary description of a peer.
neighbor ebgp-multihop	Allows connections to external peers that are not directly connected on the interface and to member AS peers.
neighbor next-hop-self	Allows the next hop in the MP_REACH_NLRI attribute when a route learned by a BGP4+ peer is advertised to BGP4+ peers to be changed to the address used for peering on the local side.
neighbor password	Uses TCP MD5 authentication for connection with peers.
neighbor remote-as	Configures the BGP4+ peers.
neighbor remove-private-as	Specifies the removal of private AS numbers when advertising to BGP4+ peers.
neighbor shutdown	Disables a peer connection.
neighbor soft-reconfiguration	Stores routes that have been suppressed by the input policy.
neighbor timers	Sets the KEEPALIVE message sending interval and hold timer value used in a peer connection.
neighbor update-source	Sets the device address as the local address to be used for peering.
neighbor weight	Sets the weighting of routes learned from a peer.
router bgp#1	Configures information about the behavior of the BGP4/BGP4+ routing protocol.
timers bgp#1	Sets the KEEPALIVE message sending interval and hold timer value to be applied to all peers.
distribute-list in (BGP4+)#2	Specifies the route filter to be used as BGP4+ learned route filter conditions.
distribute-list out (BGP4+)#2	Specifies the route filter to be used as BGP4+ advertised route filter conditions.
neighbor in (BGP4+)#2	Specifies the route filter to be used as the learned route filter conditions for a specific BGP4+ peer.
neighbor out (BGP4+)#2	Specifies the route filter to be used as the advertised route filter conditions for a specific BGP4+ peer.
redistribute (BGP4+)#2	Specifies the protocol of learned routes advertised by BGP4+.

#1

Common command for BGP4 (IPv4) and BGP4+ (IPv6) peers.

#2

See *14. Route Filters (IPv4 and IPv6)* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

Table 28-6: List of operation commands used in BGP4+ configuration

Command name	Description
clear ipv6 bgp	1. When `* in` is specified in the parameter: - Applies the latest route filtering settings to BGP4+ learned route filtering. - Requests re-advertisement of BGP4+ routes to all BGP4+ peers. 2. When `* out` is specified in the parameter - Applies the latest route filtering settings to BGP4+ advertised route filtering. - Applies the `neighbor remove-private-as` setting to the operation. - Re-advertises BGP4+ routes to all BGP4+ peers. 3. When `* both` is specified in the parameter - Applies the latest route filtering settings to BGP4+ learned route filtering and advertised route filtering. - Applies the `neighbor remove-private-as` setting to the operation. - Requests re-advertisement of BGP4+ routes to all BGP4+ peers and redistributes BGP4+ routes to all BGP4+ peers. 4. When `*` is specified in the parameter Disconnects all BGP4+ peers.

28.2.2 Overview of configuration

1. Execute the `swrt_table_resource` command to enable IPv6 routing in advance.
2. Configure the IPv6 interfaces in advance.
3. Set the address of the local device in the loopback interface in advance.
4. Configure the BGP4+ peers.
5. Set the BGP4+ route learning policy.
6. Set the BGP4+ route advertising policy.
7. Configure learned route filtering.
8. Configure advertised route filtering.
9. Set the learned route filter conditions.
10. Set the advertised route filter conditions.
11. Apply the filters to BGP4+ operation.

Notes

- If you connect a BGP4+ peer, use the `neighbor activate` configuration command to enable the IPv6 address family. If the IPv6 family is disabled, BGP4+ peers cannot be connected.
- If you set up BGP4+ peering without first configuring the route filters, the Switch will begin performing route learning and advertising as soon as the peer relations are established. To prevent unintended route learning and advertising, first set the `disable` configuration command to disable BGP4+ operation, and then set the `neighbor remote-as` configuration command. To start BGP4+ after configuring the route filters, use the `no` form of the `disable` configuration command.

28.2.3 Configuring BGP4+ peers

Command examples

1. `(config)# router bgp 65531`

   Sets BGP/BGP4+ as the routing protocol. Specify the AS number (65531) of the autonomous system to which the local router belongs for the parameter.

2. `(config-router)# bgp router-id 192.168.1.100`

   Sets the identifier of the local router (192.168.1.100).

3. `(config-router)# neighbor 3ffe:172:16:2::2 remote-as 65532`

   Configures an external peer (remote peer address:3ffe:172:16:2::2; AS number: 65532).

4. `(config-router)# neighbor 3ffe:10:2:2::2 remote-as 65533`

   Configures an external peer (remote peer address:3ffe:10:2:2::2; AS number: 65533).

5. `(config-router)# neighbor 3ffe:10:2:2::2 ebgp-multihop`

   Disallows use of the interface address of the interface directly connected to the peer as the peering address.

6.
   ```
   (config-router)# neighbor 3ffe:10:2:2::2 update-source
   loopback 0
   ```

   Sets the device address as the peering address of the local router.

7. `(config-router)# neighbor 3ffe:192:168:2::2 remote-as 65531`

   Configures an internal peer (remote peer address: 3ffe:192.168:2::2).

8. `(config-router)# neighbor 3ffe:10:1:2::2 remote-as 65531`

   Configures an internal peer (remote peer address: 3ffe:10:1:2::2).

9.
   ```
   (config-router)# neighbor 3ffe:10:1:2::2 update-source
   loopback 0
   ```

   Sets the device address as the peering address of the local router.

10. `(config-router)# address-family ipv6`

    Places the router in `config-router-af (ipv6)` mode.

11. `(config-router-af)# neighbor 3ffe:172:16:2::2 activate`

    Enables the IPv6 address family for the external peer (remote peer address: 3ffe:172:16:2::2).

12. `(config-router-af)# neighbor 3ffe:10:2:2::2 activate`

    Enables the IPv6 address family for the external peer (remote peer address: 3ffe:10:2:2::2).

13. `(config-router-af)# neighbor 3ffe:192:168:2::2 activate`

    Enables the IPv6 address family for the internal peer (remote peer address: 3ffe:192.168:2::2).

14. `(config-router-af)# neighbor 3ffe:10:1:2::2 activate`

    Enables the IPv6 address family for the internal peer (remote peer address: 3ffe:10:1:2::2).

28.2.4 Configuring the BGP4+ route learning policy

Points to note

The example below shows how to set learned route preferences on a per-peer basis by setting the weight of each peer.

Command examples

1. `(config-router-af)# bgp always-compare-med`

   For the purpose of route selection, allows comparison of `MED` attribute values in routing information received from different ASs.

2.
   ```
   (config-router-af)# neighbor 3ffe:172:16:2::2 weight 20
   (config-router-af)# neighbor 3ffe:10:2:2::2 weight 20
   (config-router-af)# neighbor 3ffe:10:1:2::2 weight 10
   (config-router-af)# neighbor 3ffe:192:168:2::2 weight 10
   ```

   Specifies a weight for the routes learned from each peer.

   Gives priority to routes learned from an external peer over routes learned from an internal peer.

28.2.5 Configuring the BGP4+ route advertising policy

Points to note

The example below shows how to set the BGP4+ path attributes to be used in route selection at the advertisement destination router.

Command examples

1. `(config-router-af)# default-metric 120`

   Sets 120 as the `MED` attribute value of advertised routes.

2.
   ```
   (config-router-af)# bgp default local-preference 80
   (config-router-af)# exit
   (config-router)# exit
   ```

   Sets 80 as the `LOCAL_PREF` attribute value advertised to internal peers.

28.2.6 Configuring learned route filtering

Points to note

The example below shows how to set the priority of learned BGP4+ routes by using `route-map` filters and specifying the conditions and settings.

Command examples

1.
```
(config)# ipv6 prefix-list EXT_IN seq 10 permit 3ffe:10:10::/64
(config)# route-map SET_LOCPREF_IN permit 10
(config-route-map)# match ipv6 address prefix-list EXT_IN
(config-route-map)# set local-preference 120
(config-route-map)# exit
(config)# route-map SET_LOCPREF_IN permit 20
(config-route-map)# exit
```
   Sets 120 in the LOCAL_PREF attribute when the destination network is 3ffe:10:10::/64.

2.
```
(config)# ip as-path access-list 10 permit "_65529$"
(config)# route-map SET_ASPREPEND_IN permit 10
(config-route-map)# match as-path 10
(config-route-map)# set as-path prepend count 1
(config-route-map)# exit
(config)# route-map SET_ASPREPEND_IN permit 20
(config-route-map)# exit
```
   Prepends one AS to the AS array when the end of the AS array of the AS_PATH attribute is 65529.

3.
```
(config)# ipv6 prefix-list INT_IN_1 seq 10 permit
3ffe:172:20::/64
(config)# route-map SET_ORIGIN_IN permit 10
(config-route-map)# match ipv6 address prefix-list INT_IN_1
(config-route-map)# set origin incomplete
(config-route-map)# exit
(config)# route-map SET_ORIGIN_IN permit 20
(config-route-map)# exit
```
   Sets INCOMPLETE in the ORIGIN attribute when the destination network is 3ffe:172:20::/64.

4.
```
(config)# ipv6 prefix-list INT_IN_2 seq 10 permit
3ffe:172:30::/64
(config)# route-map SET_MED_IN permit 10
(config-route-map)# match ipv6 address prefix-list INT_IN_2
(config-route-map)# set metric 100
(config-route-map)# exit
(config)# route-map SET_MED_IN permit 20
(config-route-map)# exit
```
   Sets 100 in the MED attribute when the destination network is 3ffe:172:30::/64.

28.2.7 Configuring advertised route filtering

Points to note

The example below shows how to set the priority of advertised BGP4+ routes by using `route-map` filters and specifying the conditions and settings.

Command examples

1.

```
(config)# ipv6 prefix-list MY_NET_1 seq 10 permit
3ffe:192:169:10::/64
(config)# ipv6 prefix-list MY_NET_2 seq 10 permit
3ffe:192:169:20::/64
(config)# route-map SET_EXT_OUT permit 10
(config-route-map)# match ipv6 address prefix-list MY_NET_1
(config-route-map)# set metric 120
(config-route-map)# exit
(config)# route-map SET_EXT_OUT permit 20
(config-route-map)# match ipv6 address prefix-list MY_NET_2
(config-route-map)# exit
```

Sets 120 in the `MED` attribute when the destination network is 3ffe:192:169:10::/64.

Allows route advertisements to be sent also to destination network 3ffe:192:169:20::/64.

28.2.8 Configuring the learned route filter conditions

Points to note

The example below shows how to apply learned route filtering on a per-peer basis by specifying the filters to be applied in the `neighbor in` command.

Command examples

1.

```
(config)# router bgp 65531
(config-router)# address-family ipv6
(config-router-af)# neighbor 3ffe:172:16:2::2 route-map
SET_LOCPREF_IN in
```

Sets 120 in the `LOCAL_PREF` attribute of routes to the destination network 3ffe:10:10.::/64 that were learned from the peer (remote peer address: 3ffe:172:16:2::2) and gives such routes higher priority than routes learned from other peers.

2.

```
(config-router-af)# neighbor 3ffe:10:2:2::2 route-map
SET_ASPREPEND_IN in
```

Prepends one AS in the AS array when the end of the AS array of the `AS_PATH` attribute learned from the peer (remote peer address: 3ffe:10:2:2::2) is 65529, and gives such routes lower priority than routes learned from other peers.

3.

```
(config-router-af)# neighbor 3ffe:10:1:2::2 route-map
SET_ORIGIN_IN in
```

Sets `INCOMPLETE` in the `ORIGIN` attribute of routes to the destination network 3ffe:172:20:0::/

64 that were learned from the peer (remote peer address: 3ffe:10:1:2::2) and gives such routes lower priority than routes learned from other peers.

4.
```
(config-router-af)# neighbor 3ffe:192:168:2::2 route-map
SET_MED_IN in
```

Sets 100 in the `MED` attribute of routes to the destination network 3ffe:172:30::/64 that were learned from the peer (remote peer address: 3ffe:192:168:2::2).

28.2.9 Configuring the advertised route filter conditions

Points to note

The example below shows how to apply the same advertised route filter to all peers by specifying the filter in the `distribute-list out` command.

Command examples

1.
```
(config-router-af)# distribute-list route-map SET_EXT_OUT out
(config-router-af)# exit
(config-router)# exit
(config)# exit
```

Advertises routes to the destination networks 3ffe:192:169:10::/64 and 3ffe:192:169:20::/64 to all external peers.

28.2.10 Applying filters

Points to note

The example below shows how to apply the route filters set as the learned route and advertised route filter conditions by using the `clear ipv6 bgp` operation command.

Command examples

1.
```
# clear ipv6 bgp * both
```

Applies the learned route filter and advertised route filter to BGP4+ operation.

Notes

The `clear ipv6 bgp` operation command (with `* in`, `* out`, or `* both` specified) applies the new route filtering settings and implements the route refresh capability (see *28.4.5 Route refresh capability*). If route refresh capability has not been negotiated, no route refresh requests are made in order to relearn changed routes, although the route filter changes are applied.

28.2.11 Configuring BGP4+ for a VRF

Points to note

The example below shows how to configure the BGP4+ functionality for a VRF in `config-router-af` (`ipv6 vrf`) mode.

Command examples

1.
```
(config)# router bgp 64496
```

Specifies the local AS number (64496).

2. `(config-router)# address-family ipv6 vrf 10`

   Changes the mode to `config-router-af` (`ipv6 vrf`) mode for VRF 10.

3. `(config-router-af)# bgp router-id 192.168.1.100`

   Specifies the identifier of the local router (192.168.1.100).

4. `(config-router-af)# neighbor 2001:db8:1::2 remote-as 64511`

   Configures an external peer (remote peer address: 2001:db8:1::2; AS number: 64511).

5. `(config-router-af)# neighbor 2001:db8:2::2 remote-as 64496`

   Configures an internal peer (remote peer address: 2001:db8:2::2; AS number: 64496).

6. `(config-router-af)# neighbor 2001:db8:1::2 activate`

   Enables the IPv6 address family for the external peer (remote peer address: 2001:db8:1::2).

7. `(config-router-af)# neighbor 2001:db8:2::2 activate`

   Enables the IPv6 address family for the internal peer (remote peer address: s2001:db8:2::2).

28.3 Operation for basic functionality

28.3.1 List of operation commands

The following table describes the operation commands for basic BGP4+ functionality.

Table 28-7: List of operation commands

Command name	Description
show ipv6 route	Shows routing information stored in the routing table.
clear ipv6 route	Clears the IPv6 forwarding entries stored in the Switch and re-registers the routing entries.
show ipv6 bgp	Shows information about the BGP4+ protocol.
clear ipv6 bgp	Clears BGP4+ sessions or BGP4+-related information, or filters inbound or outbound routes using new BGP filter information. Also reconnects a BGP4+ session that was terminated because a peer exceeded the maximum number of BGP4+ routes that can be learned from a particular device.
show ipv6 vrf	Displays the IPv6 information of a VRF.
show processes cpu unicast#	Shows the CPU usage of a unicast routing program.
restart unicast#	Restarts the unicast routing program.
dump protocols unicast#	Outputs the trace information and control table information collected by the unicast routing program to a file.
erase protocol-dump unicast#	Deletes the file of trace information and control table information generated by the unicast routing program.

#

See *8. Routing Protocol Common to IPv4 and IPv6* in the manual *Operation Command Reference Vol. 2 For Version 11.10*.

28.3.2 Checking peer type and connectivity

The figures below show the execution results for the connectivity example in *Figure 28-9: Connectivity example*. To view peer connection information, use the `show ipv6 bgp` operation command with the `neighbors` parameter specified. To view detailed information, specify the `neighbors` and `detail` parameters.

Figure 28-10: Results of executing the show ipv6 bgp command (with the neighbors parameter specified)

```
> show ipv6 bgp neighbors
Date 20XX/10/18 22:45:55 UTC
Peer Address      Peer AS  Local Address     Local AS  Type      Status
3ffe:10:1:2::2    65531    3ffe:10:1:2::1    65531     Internal  Established
3ffe:192:168:2::2 65531    3ffe:192:168:2::1 65531     Internal  Established
3ffe:10:2:2::2    65533    3ffe:10:1:2::1    65531     External  Established
3ffe:172:16:2::2  65532    3ffe:172:16:2::1  65531     External  Established
```

Figure 28-11: Results of executing the show ipv6 bgp command (with the neighbors and detail parameters specified)

```
> show ipv6 bgp neighbors detail
Date 20XX/10/17 15:52:14 UTC
BGP4+ Peer: 3ffe:10:1:2::2       , Remote AS: 65531
Remote Router ID: 10.1.2.102
    BGP4+ Status: Established       HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/10/17 15:51:00
    BGP4+ Version: 4                Type: Internal
    Local Address: 3ffe:10:1:2::1
    Local AS: 65531                 Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:52:00  Last Keep Alive Received: 15:52:00
    BGP4+ Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                   0          0          2          4
    BGP4+ Capability Negotiation: <IPv6-Uni Refresh Refresh(v)>
      Send   : <IPv6-Uni Refresh Refresh(v)>
      Receive: <IPv6-Uni Refresh Refresh(v)>
    Password: UnConfigured
BGP4+ Peer: 3ffe:192:168:2::2   , Remote AS: 65531
Remote Router ID: 192.168.1.102
    BGP4+ Status: Established       HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/10/17 15:50:43
    BGP4+ Version: 4                Type: Internal
    Local Address:3ffe:192:168:2::1
    Local AS: 65531                 Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:43  Last Keep Alive Received: 15:51:43
    BGP4+ Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                   0          0          2          4
    BGP4+ Capability Negotiation: <IPv6-Uni Refresh Refresh(v)>
      Send   : <IPv6-Uni Refresh Refresh(v)>
      Receive: <IPv6-Uni Refresh Refresh(v)>
    Password: UnConfigured
BGP4+ Peer: 3ffe:10:2:2::2     , Remote AS: 65533
Remote Router ID: 10.2.2.102
    BGP4+ Status: Established       HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/10/17 15:50:30
    BGP4+ Version: 4                Type: External
    Local Address: 3ffe:10:1:2::1
    Local AS: 65531                 Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:30  Last Keep Alive Received: 15:51:30
    BGP4+ Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                   0          0          2          4
    BGP4+ Capability Negotiation: <IPv6-Uni Refresh>
      Send   : <IPv6-Uni Refresh Refresh(v)>
      Receive: <IPv6-Uni Refresh Refresh(v)>
    Password: UnConfigured
BGP4+ Peer: 3ffe:172:16:2::2   , Remote AS: 65532
Remote Router ID: 172.16.1.102
    BGP4+ Status: Established       HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/10/17 15:49:35
    BGP4+ Version: 4                Type: External
    Local Address:3ffe172:16:2::1
    Local AS: 65531                 Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:35  Last Keep Alive Received: 15:51:35
    BGP4+ Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                   0          0          3          5
    BGP4+ Capability Negotiation: <IPv6-Uni Refresh Refresh(v)>
      Send   : <IPv6-Uni Refresh Refresh(v)>
      Receive: <IPv6-Uni Refresh Refresh(v)>
    Password: UnConfigured
>
```

28.3.3 Checking the BGP4+ route selection result

To check the BGP4+ route selection result, use the `show ipv6 bgp` operation command.

Figure 28-12: Results of executing the show ipv6 bgp command

```
# show ipv6 bgp
Date 20XX/10/18 22:44:23 UTC
Local AS: 65531, Local Router ID: 192.168.1.100
Status Codes: d dampened, * valid, > active, S Stale, r RIB failure
Origin Codes: i - IGP, e - EGP, ? - incomplete
   Network                                     Next Hop
         MED    LocalPref weight Path
*> 3ffe:10:10::/64                            fe80::200:87ff:fe16:90d5%VLAN0005
         -      120        20     65532 65528 i                            ...1
*  3ffe:10:10::/64                            3ffe:10:2:2::2
         -      80         20     65533 65533 65529 i                      ...2
*  3ffe:10:10::/64                            3ffe:10:1:2::2
         -      80         10     65534 i                                  ...3
*> 3ffe:10:20::/64                            fe80::200:87ff:fe16:90d5%VLAN0005
         -      80         20     65532 65528 i                            ...4
*  3ffe:10:20::/64                            3ffe:10:2:2::2
         -      80         20     65533 65533 65529 i                      ...5
*> 3ffe:172:20::/64                           3ffe:10:1:2::2
         -      100        10     65534 i                                  ...6
*  3ffe:172:20::/64                           3ffe:192:168:2::2
         -      100        10     65530 ?                                  ...7
*> 3ffe:172:30::/64                           3ffe:10:1:2::2
         -      100        10     65534 i                                  ...8
*  3ffe:172:30::/64                           3ffe:192:168:2::2
         100    100        10     65530 i                                  ...9
*> 3ffe:192:168:10::/64                       3ffe:10:1:2::2
         -      100        10     65534 i                                 ...10
*  3ffe:192:168:10::/64                       3ffe:192:168:2::2
         -      100        10     65530 i                                 ...11
*> 3ffe:192:169:10::/64                       3ffe:192:168:2::2
         -      100        10     65530 i                                 ...12
*> 3ffe:192:169:20::/64                       3ffe:192:168:2::2
         -      100        10     65530 i                                 ...13
```

1 to 3: Route selection for destination network 3ffe:10:10::/64

Routes 1 and 2 are preferred by comparing the weights of the three routes, and then 1 is selected by comparing the `LOCAL_PREF` values of 1 and 2.

4 to 5: Route selection for destination network 3ffe:10:20::/64

Route 4 is selected by comparing the `AS_PATH` length of each route.

6 to 7: Route selection for destination network 3ffe:172:20::/64

Route 6 is selected by comparing the `ORIGIN` attribute of each route.

8 to 9: Route selection for destination network 3ffe:172:30::/64

Route 8 is selected by comparing the `MED` attribute of each route.

10 to 11: Route selection for destination network 3ffe:192:168:10::/64

Route 10 is selected by comparing the peer BGP identifier of each route.

12 to 13: Route selection for destination networks 3ffe:192:169:10::/64 and 3ffe:192:169:20::/64

Route 12 and route 13 are selected because there are no other routes to the destinations.

28.3.4 Checking the content of BGP4+ route advertisements

To check the path attributes of advertised BGP4+ routes, use the `show ipv6 bgp` command with the `advertised-routes` parameter specified.

Figure 28-13: Results of executing the show ipv6 bgp command (with the advertised-routes parameter specified)

```
> show ipv6 bgp advertised-routes
Date 20XX/10/18 22:44:54 UTC
BGP4+ Peer: 3ffe:10:2:2::2, Remote AS: 65533
Local AS: 65531, Local Router ID: 192.168.1.100
Origin Codes: i - IGP, e - EGP, ? - incomplete
Network                                     Next Hop
      MED    LocalPref Path
3ffe:192:169:10::/64                        3ffe:192:168:2::2
      120    -         65531 i                                      ...1
3ffe:192:169:20::/64                        3ffe:192:168:2::2
      100    -         65531 i
BGP4+ Peer: 3ffe:172:16:2::2, Remote AS: 65532
Local AS: 65531, Local Router ID: 192.168.1.100
Origin Codes: i - IGP, e - EGP, ? - incomplete
Network                                     Next Hop
      MED    LocalPref Path
3ffe:192:169:10::/64                        3ffe:192:168:2::2
      120    -         65531 i                                      ...2
3ffe:192:169:20::/64                        3ffe:192:168:2::2
      100    -         65531 i
```

1 and 2: The `MED` attribute (value: 120) is set for the advertised routes.

28.4 Description of extended functionality

28.4.1 BGP4+ peer groups

The basic operation of BGP4+ (IPv6) peer grouping is the same as for BGP4 (IPv4) peer grouping. For details, see *12.4.1 BGP4 peer groups*.

28.4.2 Community

The basic operation of BGP4+ (IPv6) communities is the same as for BGP4 (IPv4) communities. For details, see *12.4.2 Community*.

28.4.3 BGP4+ multipath

The basic operation for BGP4+ (IPv6) multipathing is the same as for BGP (IPv4) multipathing. For details, see *12.4.3 BGP4 multipath*.

Notes on BGP4+ multipath routing via multiple IGP routes

Only static routes and OSPFv3 routes can be configured as multipath IGP routes in the Switch. For a description of static multipath routing, see *24. Static Routing (IPv6)*, and for a description of OSPF multipath routing, see *26.1.7 Equal-cost multipath*.

28.4.4 Capability negotiation

Capability negotiation is a means of negotiating which capabilities are supported by the peers. This is done by adding capability information to the OPEN message at the establishment of a BGP4+ negotiation session. Functionality that is matched (supported) by the respective peers' advertised capability information can be used by those peers.

In the Switch, the following capability information is always added to the OPEN message: IPv6 unicast routing, route refresh capability (Capability Code: 2), route refresh capability (Capability Code: 128), and the graceful restart functionality (Capability Code: 64). When an OPEN message without any capability information is received from a peer, only IPv6 unicast route advertisements will be sent over the established BGP4+ connection.

The following table describes the capabilities that can be negotiated.

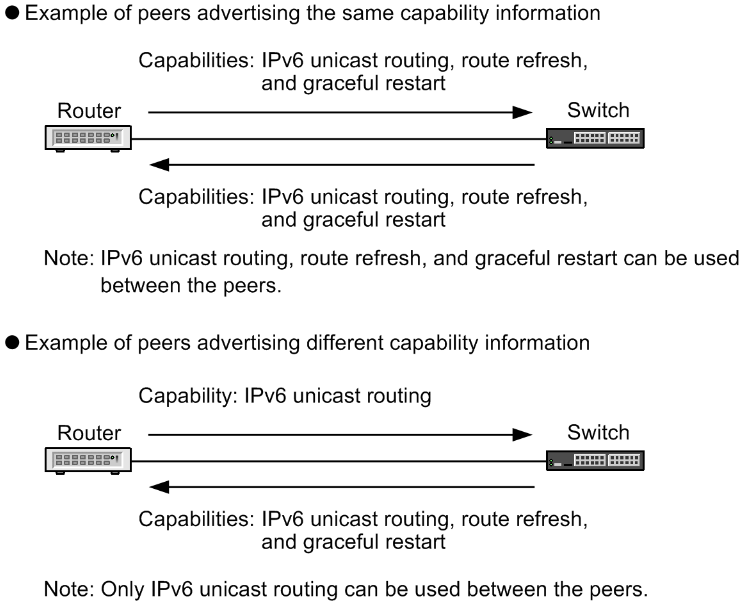
Table 28-8: Negotiable capabilities

Capability	Capability information in OPEN message	Description
IPv6 routing	Capability Code: 1 AFI field of the Capability Value: 2 SAFI field of the Capability Value: 1	IPv6 unicast routes can be sent and received between the peers.
Route refresh capability	Capability Code: 2 AFI field of the Capability Value: 2#	Route refresh is supported on IPv6 routes.
	Capability Code: 128 AFI field of the Capability Value: 2#	
Graceful restart	Capability Code: 64 AFI field of the Capability Value: 1 SAFI field of the Capability Value: 2	Graceful restart is supported.

#: IPv6 route refresh is supported if negotiation of either of these two capabilities is successful.

The following figure illustrates how negotiation operates.

Figure 28-14: Operation of capability negotiation



● Example of peers advertising the same capability information

Capabilities: IPv6 unicast routing, route refresh, and graceful restart
Router
Switch
Capabilities: IPv6 unicast routing, route refresh, and graceful restart

Note: IPv6 unicast routing, route refresh, and graceful restart can be used between the peers.

● Example of peers advertising different capability information

Capability: IPv6 unicast routing
Router
Switch
Capabilities: IPv6 unicast routing, route refresh, and graceful restart

Note: Only IPv6 unicast routing can be used between the peers.

28.4.5 Route refresh capability

A fundamental aspect of BGP4+ is that only route updates are advertised. In contrast, the route refresh capability forcibly re-advertises previously propagated routes, which otherwise would not be advertised in BGP4+.

Route refresh covers the re-advertisement of routes from the local device and from the remote BGP4+ peer. You can select which types of routes to re-advertise. This functionality can be executed by using the `clear ipv6 bgp` command.

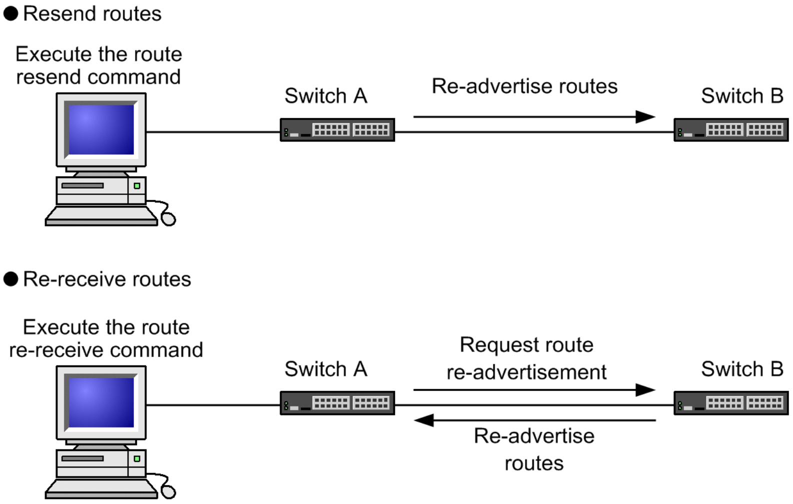
The following table describes the route refresh capability.

Table 28-9: Route refresh capability

Capability	Route type	Direction of re-advertisement
Resend IPv6 unicast routes	IPv6 unicast route	Re-advertised from local peer to remote peer.
Re-receive IPv6 unicast routes		Re-advertised from remote peer to local peer.

The following figure illustrates how the route refresh capability operates.

Figure 28-15: Operation of the route refresh capability



(1) Notes on using the route refresh capability

For a route to be resent from the remote device, both routers in a peering relationship must support the route refresh capability. To use the route refresh capability, its use must be negotiated between the routers at the time of BGP4+ peer establishment.

When the `inbound` parameter is specified in the `neighbor soft-reconfiguration` configuration command, because the routes suppressed by the learned route filter are retained as invalid routes, the remote peer does not send any route refresh requests for route re-advertisement to the local peer.

The route refresh capability of the Switch complies with RFC 2918. The capability codes used in negotiation are the RFC-2918-compliant code (value: 2) and a private code (value: 128). In addition, a private capability code (value: 128 to 255) defined in RFC 2434 may be used by other vendors.

Take care when using the route refresh capability between the Switch and devices from other vendors.

28.4.6 TCP MD5 authentication

The basic operation of TCP MD5 authentication for BGP4+ (IPv6) is the same as for BGP4 (IPv4). For details, see *12.4.6 TCP MD5 authentication*.

28.4.7 BGP4+ advertised route generation

The basic operation of BGP4+ (IPv6) advertised route generation is the same as for BGP4 (IPv4) advertised route generation. For details, see *12.4.7 BGP4 advertised route generation*.

28.4.8 Route flap dampening

The basic operation of BGP4+ (IPv6) route flap dampening is the same as for BGP4 (IPv4) route flap dampening. For details, see *12.4.8 Route flap dampening*.

28.4.9 Route reflection

BGP4+ (IPv6) route reflection is the same as BGP4 (IPv4) route reflection. For details, see *12.4.9 Route reflection*.

28.4.10 Confederations

The basic operation of BGP4+ (IPv6) confederations is the same as for BGP4 (IPv4) confederations. For details, see *12.4.10 Confederations*.

28.4.11 Graceful restart

The basic operation of the BGP4+ (IPv6) graceful restart functionality is the same as that for BGP4 (IPv4). For details, see *12.4.11 Graceful restart*.

28.4.12 Maximum number of learned BGP4+ routes

Basic operations related to the maximum number of BGP4+ (IPv6) learned routes are the same as for the maximum number of BGP4 (IPv4) learned routes. For details, see *12.4.12 Maximum number of learned BGP4 routes*.

28.5 Configuration of extended functionality

28.5.1 Configuring BGP4+ peer groups

(1) List of configuration commands

The following table describes the configuration commands for BGP4+ peer groups.

Table 28-10: List of configuration commands

Command name	Description
neighbor peer-group (assigning members)	Assigns a peer to a peer group.
neighbor peer-group (creating)	Creates a peer group.

(2) Creating a BGP4+ peer group

Points to note

The example below shows how to create a peer group by using the `neighbor peer-group` (creating) command. Settings such as the AS number, optional settings, and advertising filters apply to all peers in the peer group.

Command examples

1.

```
(config)# router bgp 65531
(config-router)# bgp router-id 172.16.2.100
(config-router)# neighbor INTERNAL-GROUP peer-group
```

Creates a peer group (group identifier: `INTERNAL-GROUP`) using the `neighbor peer-group` (creating) command.

2.

```
(config-router)# neighbor INTERNAL-GROUP remote-as 65531
(config-router)# address-family ipv6
(config-router-af)# neighbor INTERNAL-GROUP
soft-reconfiguration inbound
(config-router-af)# exit
(config-router)# neighbor INTERNAL-GROUP timers 30 90
```

Sets the AS number (AS: 65531) and optional settings for the peer group (group identifier: `INTERNAL-GROUP`) .

3.

```
(config-router)# neighbor EXTERNAL-GROUP peer-group
(config-router)# address-family ipv6
(config-router-af)# neighbor EXTERNAL-GROUP activate
(config-router-af)# neighbor EXTERNAL-GROUP send-community
(config-router-af)# top
```

Creates a peer group (group identifier: EXTERNAL-GROUP) using the `neighbor peer-group` (creating) command, and performs optional settings.

4.
```
(config)# route-map SET_COM permit 10
(config-route-map)# set community 1000:1001
(config-route-map)# exit
```
Sets the community value 1000:1001 for the specified `route-map`.

5.
```
(config)# router bgp 65531
(config-router)# address-family ipv6
(config-router-af)# neighbor EXTERNAL-GROUP route-map SET_COM
out
(config-router-af)# exit
```
Sets an advertised route filter for the peer group (group identifier: `EXTERNAL-GROUP`).

(3) Assigning BGP4+ peers to a peer group

Points to note

The example below shows how to assign a peer to a peer group by using the `neighbor peer-group` (assigning members) command.

Command examples

1.
```
(config-router)# neighbor 3ffe:172:16:2::2 peer-group
INTERNAL-GROUP
```
Assigns a peer (remote peer address: 3ffe:172:16:2::2) to the peer group (group identifier: `INTERNAL-GROUP`) by using the `neighbor peer-group` (assigning members) command. The AS number 65531 set for the peer group will be used for the peer's AS number.

2.
```
(config-router)# neighbor 3ffe:172:17:3::3 peer-group
INTERNAL-GROUP
```
Assigns a peer (remote peer address: 3ffe:172:17:3::3) to the peer group (group identifier: `INTERNAL-GROUP`) by using the `neighbor peer-group` (assigning members) command. The AS number 65531 set for the peer group will be used for the peer's AS number.

3.
```
(config-router)# neighbor 3ffe:192:168:4::4 remote-as 65533
(config-router)# neighbor 3ffe:192:168:4::4 peer-group
EXTERNAL-GROUP
```
Creates a peer (remote peer address: 3ffe:192:168:4::4) and assigns it to a peer group (group identifier: `EXTERNAL-GROUP`). The AS number 65533 set for the peer will be used as its AS number.

4.
```
(config-router)# neighbor 3ffe:192:168:5::5 remote-as 65534
(config-router)# neighbor 3ffe:192:168:5::5 peer-group
EXTERNAL-GROUP
```
Creates a peer (remote peer address: 3ffe:192:168:5::5) and assigns it to a peer group (group identifier: `EXTERNAL-GROUP`). The AS number 65534 set for the peer will be used as its AS number.

28.5.2 Configuring communities

(1) List of configuration commands

The following table describes the configuration commands for communities.

Table 28-11: List of configuration commands

Command name	Description
neighbor send-community	Specifies that the COMMUNITIES attribute should not be removed from route advertisements to peers.
distribute-list in (BGP4+)#	Specifies the route filter to be used as BGP4+ learned route filter conditions.
distribute-list out (BGP4+)#	Specifies the route filter to be used as BGP4+ advertised route filter conditions.
neighbor in (BGP4+)#	Specifies in the route-map parameter the route filter to be used as learned route filter conditions for a specific BGP4+ peer only.
neighbor out (BGP4+)#	Specifies in the route-map parameter the route filter to be used as advertised route filter conditions for a specific BGP4+ peer only.
redistribute (BGP4+)#	Specifies the protocol of learned routes advertised by BGP4+.

#

See *14. Route Filters (IPv4 and IPv6)* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

(2) Creating a community

Points to note

The example below shows how to add the COMMUNITIES attribute to advertised BGP4+ routes by setting the neighbor send-community configuration command for the concerned peers.

Command examples

1.
```
(config)# router bgp 65531
(config-router)# bgp router-id 192.168.1.100
(config-router)# neighbor 3ffe:192:168:2::2 remote-as 65531
(config-router)# neighbor 3ffe:172:16:2::2 remote-as 65532
(config-router)# neighbor 3ffe:10:2:2::2 remote-as 65533
```
Configures the BGP4+ peers.

2.
```
(config-router)# address-family ipv6
```
Places the router in config-router-af (ipv6) mode.

3.
```
(config-router-af)# neighbor 3ffe:172:16:2::2 send-community
(config-router-af)# neighbor 3ffe:10:2:2::2 send-community
(config-router-af)# exit
(config-router)# exit
```

Specifies that the COMMUNITIES attribute is to be added to BGP4+ routes advertised to the peers.

4.

```
(config)# ip community-list 10 permit 1000:1002
(config)# ip community-list 20 permit 1000:1003
(config)# route-map SET_LOCPREF permit 10
(config-route-map)# match community 10
(config-route-map)# set local-preference 120
(config-route-map)# exit
(config)# route-map SET_LOCPREF permit 20
(config-route-map)# match community 20
(config-route-map)# set local-preference 80
(config-route-map)# exit
(config)# route-map SET_LOCPREF permit 30
(config-route-map)# exit
```

Sets 120 as the LOCAL_PREF attribute value of routes that have the community value 1000:1002 in their COMMUNITIES attribute, and 80 as the LOCAL_PREF attribute value of routes that have the community value 1000:1003 in their COMMUNITIES attribute.

5.

```
(config)# ipv6 prefix-list MY_NET seq 10 permit 3ffe:192:168::/48 ge 32 le 64
(config)# route-map SET_COM permit 10
(config-route-map)# match ipv6 address prefix-list MY_NET
(config-route-map)# set community 1000:1001
(config-route-map)# exit
```

Sets a COMMUNITIES attribute that has the community value 1000:1001 for routes to destination network 3ffe:192:168::/48 (prefix length 32 to 64).

6.

```
(config)# router bgp 65531
(config-router)# address-family ipv6
(config-router-af)# distribute-list route-map SET_LOCPREF in
(config-router-af)# distribute-list route-map SET_COM out
```

Sets a learned route filter and advertised route filter for all peers.

7.

```
(config-router-af)# neighbor 3ffe:192:168:2::2 activate
(config-router-af)# neighbor 3ffe:172:16:2::2 activate
(config-router-af)# neighbor 3ffe:10:2:2::2 activate
```

Enables the IPv6 address family.

(3) Applying filters

Points to note

To apply route filters as learned route and advertised route filter conditions, use the `clear ipv6 bgp` operation command.

Command examples

1.
```
# clear ipv6 bgp * both
```
Applies the community-based route filters to the network operation.

28.5.3 Configuring BGP4+ multipath

(1) List of configuration commands

The following table describes the configuration commands for BGP4+ multipaths.

Table 28-12: List of configuration commands

Command name	Description
bgp always-compare-med	Enables comparison of the `MED` attribute of routes learned from different ASs. (If this command has not been set, you cannot set the `all-as` parameter in the `maximum-paths` command.)
maximum-paths	Sets the maximum number of paths to a given destination.

(2) Configuring BGP4+ multipath

Points to note

To specify the `all-as` parameter in the `maximum-paths` command, you must first set the `bgp always-compare-med` command.

Command examples

1.
```
(config)# router bgp 65531
(config-router)# bgp router-id 192.168.1.100
(config-router)# neighbor 3ffe:172:16:2::2 remote-as 65532
(config-router)# neighbor 3ffe:172:17:2::2 remote-as 65533
```
Configures the peers that will participate in multipath routing. In this example, the routes learned from AS65532 and AS65533 will be alternative paths in a multipath route.

2.
```
(config-router)# address-family ipv6
```
Places the router in `config-router-af (ipv6)` mode.

3.
```
(config-router-af)# bgp always-compare-med
(config-router-af)# maximum-paths 4 all-as
```
Specifies that a maximum of four paths, including paths learned from different ASs, can be in a multipath.

4.
```
(config-router-af)# neighbor 3ffe:172:16:2::2 activate
(config-router-af)# neighbor 3ffe:172:17:2::2 activate
```

Enables the IPv6 address family.

28.5.4 Configuring TCP MD5 authentication

(1) List of configuration commands

The following table describes the configuration commands for TCP MD5 authentication (BGP4+).

Table 28-13: List of configuration commands

Command name	Description
neighbor password	Specifies that TCP MD5 authentication is to be used for peer connection.

(2) Configuring TCP MD5 authentication

Points to note

The example below shows how to set an authentication key for TCP MD5 authentication by using the `neighbor password` configuration command.

Command examples

1.
```
(config)# router bgp 65531
(config-router)# bgp router-id 192.168.1.100
(config-router)# neighbor 3ffe:172:16:2::2 remote-as 65532
(config-router)# neighbor 3ffe:192:168:2::2 remote-as 65531
```
Configures the BGP4+ peers.

2.
```
(config-router)# neighbor 3ffe:172:16:2::2 password
"authmd5_65532"
```
Sets up TCP MD5 authentication based on the authentication key `authmd5_65532` for the peer whose remote peer address is 3ffe:172:16:2::2.

3.
```
(config-router)# address-family ipv6
```
Places the router in `config-router-af (ipv6)` mode.

4.
```
(config-router-af)# neighbor 3ffe:172:16:2::2 activate
(config-router-af)# neighbor 3ffe:192:168:2::2 activate
```
Enables the IPv6 address family.

28.5.5 Configuring BGP4+ advertised route generation

(1) List of configuration commands

The following table describes the configuration commands used to generate BGP4+ advertised routes.

Table 28-14: List of configuration commands

Command name	Description
network	Specifies the generation of a BGP4+ advertised route.

(2) Configuring BGP4+ advertised route generation

Points to note

To generate BGP4+ advertised routes, use the `network` configuration command. To filter the generated routes, specify `local` in the `match route-type` command in the `route-map`.

Command examples

1.
```
(config)# router bgp 65531
(config-router)# bgp router-id 192.168.1.100
(config-router)# neighbor 3ffe:172:16:2::2 remote-as 65532
(config-router)# neighbor 3ffe:192:168:2::2 remote-as 65531
```
Configures the BGP4+ peers.

2.
```
(config-router)# address-family ipv6
```
Places the router in `config-router-af (ipv6)` mode.

3.
```
(config-router-af)# network 3ffe:192:169:10::/64
(config-router-af)# exit
```
Generates a BGP4+ advertised route for 3ffe:192:169:10::/64 when a route 3ffe:192:169:10::/64 is registered in the routing table.

4.
```
(config)# route-map ADV_NET permit 10
(config-route-map)# match route-type local
(config-route-map)# exit
```
Specifies the generated BGP4+ advertised routes.

5.
```
(config)# route-map ADV_NET deny 20
(config-route-map)# match protocol bgp
(config-route-map)# exit
```
Specifies the BGP protocol.

6.
```
(config)# router bgp 65531
(config-router)# address-family ipv6
(config-router-af)# neighbor 3ffe:172:16:2::2 route-map
ADV_NET out
(config-router-af)# exit
```
Specifies that generated BGP4+ advertised routes only are to be advertised to the peer whose remote peer address is 3ffe:172:16:2::2 (learned BGP4+ routes are not advertised).

7.
```
(config)# route-map DENY_NET deny 10
(config-route-map)# match route-type local
```

```
(config-route-map)# exit
```

Specifies the generated BGP4+ advertised routes.

8.

```
(config)# router bgp 65531
(config-router)# address-family ipv6
(config-router-af)# neighbor 3ffe:192:168:2::2 route-map
DENY_NET out
```

Specifies that the generated BGP4+ advertised routes are not to be advertised to the peer whose remote peer address is 3ffe:192:168:2::2.

9.

```
(config-router-af)# neighbor 3ffe:172:16:2::2 activate
(config-router-af)# neighbor 3ffe:192:168:2::2 activate
```

Enables the IPv6 address family.

(3) Applying filters

Points to note

The example below shows how to use the `clear ipv6 bgp` operation command to apply filter settings to generated BGP4+ advertised routes.

Command examples

1.

```
# clear ipv6 bgp * out
```

Applies the route filter to BGP4+ advertised routes.

28.5.6 Configuring route flap dampening

(1) List of configuration commands

The following table describes the configuration commands for route flap dampening.

Table 28-15: List of configuration commands

Command name	Description
bgp dampening	Temporarily stops the use of an unstable route and reduces the effect of route flapping.#

#: Only specified for the global network.

(2) Configuring route flap dampening

Points to note

The example below shows how to apply route flap dampening to a BGP4+ route by using the `bgp dampening` command in `config-router af (ipv6)` mode.

Command examples

1.

```
(config)# router bgp 65531
(config-router)# bgp router-id 192.168.1.100
(config-router)# neighbor 3ffe:172:16:2::2 remote-as 65532
(config-router)# neighbor 3ffe:172:17:2::2 remote-as 65533
```

Configures the BGP4+ peers.

2. `(config-router)# address-family ipv6`

   Places the router in `config-router-af (ipv6)` mode.

3. `(config-router-af)# bgp dampening`

   Applies route flap dampening.

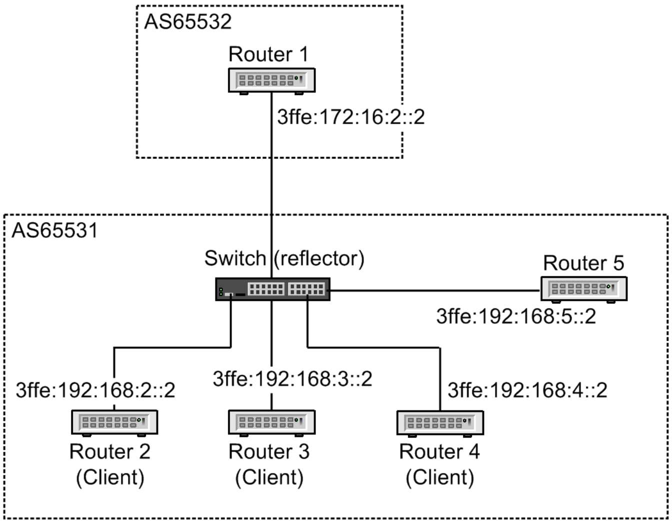
4.
   ```
   (config-router-af)# neighbor 3ffe:172:16:2::2 activate
   (config-router-af)# neighbor 3ffe:172:17:2::2 activate
   ```

   Enables the IPv6 address family.

28.5.7 Configuring route reflection

This section describes how to configure route reflection using the following figure as a reference.

Figure 28-16: Example of configuring route reflection



(1) List of configuration commands

The following table describes the configuration commands for route reflection.

Table 28-16: List of configuration commands

Command name	Description
bgp client-to-client reflection	Specifies that BGP4+ routes are to be reflected between the route reflector and clients.
bgp cluster-id	Specifies the cluster ID to be used in route reflection.
bgp router-id	If `bgp cluster-id` is not set, the router ID is used as the cluster ID for route reflection.
neighbor always-nexthop-self	Specifies that the next hop in the `MP_REACH_NLRI` attribute of routes advertised to an internal peer (including route reflection) is to be forcibly changed to the local address being used for peering with the internal peer.

Command name	Description
neighbor route-reflector-client	Specifies the route reflector client.

(2) Configuring route reflection

Points to note

The `bgp client-to-client reflection` configuration command is enabled by default and does not need to be set. If you do not want BGP4+ routes to be reflected between the route reflector and clients, use the `no bgp client-to-client reflection` command in the `config-router-af (ipv6)` mode or the `config-router-af (ipv6 vrf)` mode.

Command examples

1.

```
(config)# router bgp 65531
(config-router)# bgp router-id 192.168.1.100
(config-router)# neighbor 3ffe:172:16:2::2 remote-as 65532
(config-router)# neighbor 3ffe:192:168:2::2 remote-as 65531
(config-router)# neighbor 3ffe:192:168:3::2 remote-as 65531
(config-router)# neighbor 3ffe:192:168:4::2 remote-as 65531
(config-router)# neighbor 3ffe:192:168:5::2 remote-as 65531
```

Configures the BGP4+ peers, defining Router 1 as an external peer and Routers 2 to 5 as internal peers.

2.

```
(config-router)# bgp cluster-id 10.1.2.1
```

Sets the cluster ID.

3.

```
(config-router)# address-family ipv6
```

Places the router in `config-router-af (ipv6)` mode.

4.

```
(config-router-af)# neighbor 3ffe:192:168:2::2
route-reflector-client
(config-router-af)# neighbor 3ffe:192:168:3::2
route-reflector-client
(config-router-af)# neighbor 3ffe:192:168:4::2
route-reflector-client
```

Defines Routers 2, 3, and 4 as route reflector clients.

5.

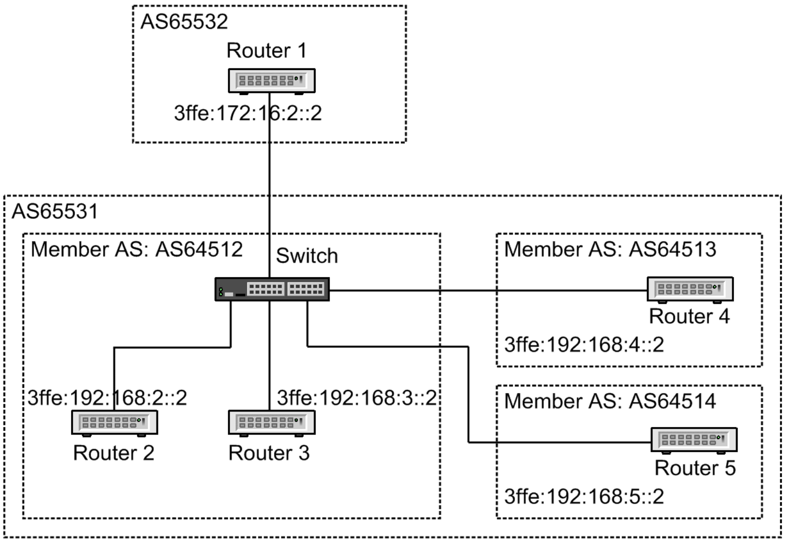
```
(config-router-af)# neighbor 3ffe:172:16:2::2 activate
(config-router-af)# neighbor 3ffe:192:168:2::2 activate
(config-router-af)# neighbor 3ffe:192:168:3::2 activate
(config-router-af)# neighbor 3ffe:192:168:4::2 activate
(config-router-af)# neighbor 3ffe:192:168:5::2 activate
```

Enables the IPv6 address family.

28.5.8 Configuring confederations

This section describes how to configure a confederation using the following figure as a reference.

Figure 28-17: Example of configuring a confederation



(1) List of configuration commands

The following table describes the configuration commands for confederations.

Table 28-17: List of configuration commands

Command name	Description
bgp confederation identifier	Specifies the AS number when configuring a confederation.#
bgp confederation peers	Specifies the member AS numbers of the ASs connected to the local member AS when configuring a confederation.
neighbor remote-as	Configures a BGP4/BGP4+ peer. Sets the local member AS number when configuring a confederation.

#: Commonly specified for both VRFs and the global network.

(2) Configuring a confederation

Points to note

Specify the local member AS number using the `router bgp` command. Then, set the member AS numbers of the other ASs connected to each AS by using the `bgp confederation peers` command in `config-router` mode.

Command examples

1.
```
(config)# router bgp 64512
```
Specifies the local member AS number (64512).

2.
```
(config-router)# bgp router-id 192.168.1.100
```
Specifies the router ID.

3.
```
(config-router)# bgp confederation identifier 65531
```
Specifies the AS number (65531) of the confederation.

4.
```
(config-router)# bgp confederation peers 64513 64514
```
Specifies the numbers of the other member ASs (64513 and 64514) connected to the local member AS.

5.
```
(config-router)# neighbor 3ffe:172:16:2::2 remote-as 65532
(config-router)# neighbor 3ffe:192:168:2::2 remote-as 64512
(config-router)# neighbor 3ffe:192:168:3::2 remote-as 64512
(config-router)# neighbor 3ffe:192:168:4::2 remote-as 64513
(config-router)# neighbor 3ffe:192:168:5::2 remote-as 64514
```
Configures the BGP4+ peers, defining Router 1 as an external peer, Routers 2 and 3 as internal peers, and Routers 4 and 5 as member AS peers.

6.
```
(config-router)# address-family ipv6
```
Places the router in `config-router-af (ipv6)` mode.

7.
```
(config-router-af)# neighbor 3ffe:172:16:2::2 activate
(config-router-af)# neighbor 3ffe:192:168:2::2 activate
(config-router-af)# neighbor 3ffe:192:168:3::2 activate
(config-router-af)# neighbor 3ffe:192:168:4::2 activate
(config-router-af)# neighbor 3ffe:192:168:5::2 activate
```
Enables the IPv6 address family.

28.5.9 Configuring graceful restart

(1) List of configuration commands

The following table describes the configuration commands for the graceful restart functionality.

Table 28-18: List of configuration commands

Command name	Description
bgp graceful-restart mode	Specifies use of the graceful restart functionality.#
bgp graceful-restart restart-time	Specifies the maximum time that a peer will wait to be reconnected after a neighboring router has commenced a graceful restart.#
bgp graceful-restart stalepath-time	Specifies the maximum time that a peer will keep routes received before a graceful restart after the neighboring router has commenced a graceful restart.#

#: Commonly specified for both VRFs and the global network.

(2) Configuring the graceful restart functionality

Points to note

The example below shows how to use the graceful restart functionality by setting the `bgp graceful-restart mode` command in `config-router` mode.

Command examples

1.
```
(config)# router bgp 65531
(config-router)# bgp router-id 192.168.1.100
(config-router)# neighbor 3ffe:172:16:2::2 remote-as 65532
(config-router)# neighbor 3ffe:192:168:2::2 remote-as 65531
```
Configures the BGP4+ peers.

2.
```
(config-router)# bgp graceful-restart mode receive
```
Specifies use of the graceful restart receiving router functionality.

3.
```
(config-router)# address-family ipv6
```
Places the router in `config-router-af (ipv6)` mode.

4.
```
(config-router-af)# neighbor 3ffe:172:16:2::2 activate
(config-router-af)# neighbor 3ffe:192:168:2::2 activate
```
Enables the IPv6 address family.

28.5.10 Configuring the maximum number of learned BGP4+ routes

(1) List of configuration commands

The following table describes the configuration commands for the number of learned BGP4+ routes.

Table 28-19: List of configuration commands

Command name	Description
neighbor maximum-prefix	Limits the number of routes that can be learned from a specified peer.

(2) Configuring the maximum number of learned BGP4+ routes

Points to note

The example below shows how to limit the number of BGP4+ routes that can be learned from a peer by using the `neighbor maximum-prefix` command.

Command examples

1.
```
(config)# router bgp 65531
(config-router)# bgp router-id 192.168.1.100
(config-router)# neighbor 3ffe:172:16:2::2 remote-as 65532
(config-router)# neighbor 3ffe:192:168:2::2 remote-as 65531
```
Configures the BGP4+ peers.

2. `(config-router)# address-family ipv6`

   Places the router in `config-router-af (ipv6)` mode.

3. `(config-router-af)# neighbor 3ffe:172:16:2::2 maximum-prefix 1000 80 restart 60`

   Allows a maximum of 10000 routes to be learned from the external peer (remote peer address: 3ffe:172:16:2::2). Sets a threshold of 80% for output of a warning message, and an interval of 60 minutes before reconnection of any peer that was disconnected because it exceeded the maximum.

4. `(config-router-af)# neighbor 3ffe:192:168:2::2 maximum-prefix 100 warning-only`

   Allows a maximum of 1000 routes to be learned from the internal peer (remote peer address: 3ffe:172:16:2::2), and specifies that any peer exceeding this maximum will not be disconnected.

5. `(config-router-af)# neighbor 3ffe:172:16:2::2 activate`

   `(config-router-af)# neighbor 3ffe:192:168:2::2 activate`

   Enables the IPv6 address family.

28.6 Operation for extended functionality

28.6.1 Checking BGP4+ peer groups

(1) List of operation commands

The following table describes the operation commands for BGP4+ peer groups.

Table 28-20: List of operation commands

Command name	Description
show ipv6 bgp	Shows information about the BGP4+ protocol.

(2) Checking a BGP4+ peer group

To check the peering information about the peers in a peer group, use the `show ipv6 bgp` command with the `peer-group` parameter specified.

Figure 28-18: Results of executing the show ipv6 bgp command (with the peer-group parameter specified)

```
>show ipv6 bgp peer-group INTERNAL-GROUP
Date 20XX/07/17 18:40:00 UTC
Local AS: 65531, Local Router ID: 172.16.2.100
BGP4+ Peer                              AS     Received   Sent
Up/Down            Status
3ffe:172:16:2::2                        65531  36         42
20XX/07/16 18:42:26  Established
3ffe:172:17:3::3                        65531  51         63
20XX/07/16 12:42:31  Established
```

(3) Checking the peer relationships in a BGP4+ peer group

To display information about the peers in a peer group, use the `show ipv6 bgp` command with the `neighbors` parameter and the `peer-group` or `detail` parameter specified.

Figure 28-19: Results of executing the show ipv6 bgp command (with the neighbors and peer-group parameters specified)

```
>show ipv6 bgp neighbors EXTERNAL-GROUP
Date 20XX/07/17 18:45:09 UTC
Peer Address                                Peer AS  Local Address
              Local AS Type      Status
3ffe:192:168:4::4                           65533    3ffe:192:168:4::214
              65531    External  Established
3ffe:192:168:5::5                           65534    3ffe:192:168:5::189
              65531    External  Active
```

(4) Checking the BGP4+ peer group membership of a peer

To check which peer group a peer belongs to, use the `show ipv6 bgp` command with the `neighbors` parameter and a *<Peer Address>* or *<Host name>* argument specified.

Figure 28-20: Results of executing the show ipv6 bgp command (with the neighbors and <Peer Address> parameters specified)

```
>show ipv6 bgp neighbors 3ffe:172:16:2::2
Date 20XX/07/17 18:45:09 UTC
BGP4+ Peer: 3ffe:172:16:2::2, Remote AS: 65531
Remote Router ID: 172.16.2.20, Peer Group: INTERNAL-GROUP                  ...1
    BGP4+ Status:Established        HoldTime: 90   , Keepalive: 30
    Established Transitions: 1      Established Date: 20XX/07/16 18:42:26
    BGP4+ Version: 4                Type: Internal
    Local Address: 3ffe:172:16:2::214
    Local AS: 65531                 Local Router ID: 172.16.2.100
    Next Connect Retry:-,          Connect Retry Timer: -
    Last Keep Alive Sent: 18:42:20, Last Keep Alive Received: 18:42:20
    BGP4+ Message  UpdateIn UpdateOut TotalIn TotalOut
                   12       14        36      42
    BGP4+ Capability Negotiation: <Refresh Refresh(v) IPv6-Uni>
      Send   : <Refresh Refresh(v) IPv6-Uni>
      Receive: <Refresh Refresh(v) IPv6-Uni>
    Password : UnConfigured
```

1. The peer belongs to the peer group `INTERNAL-GROUP`.

28.6.2 Checking communities

(1) List of operation commands

The following table describes the operation commands for communities.

Table 28-21: List of operation commands

Command name	Description
show ipv6 route	Shows routing information stored in the routing table.
show ipv6 bgp	Shows information about the BGP4+ protocol.

(2) Displaying the learned routes in a community

The figures below show information related to *28.5.2 Configuring communities*.

To display routes that have a specific community, use the `show ipv6 bgp` command with the `community` parameter specified.

Figure 28-21: Results of executing the show ipv6 bgp command (with the community parameter specified)

```
> show ipv6 bgp community 1000:1002
Date 20XX/10/20 21:00:00 UTC
Local AS: 65531, Local Router ID: 192.168.1.100
Status Codes: d dampened, * valid, > active, S Stale, r RIB failure
```

```
Origin Codes: i - IGP, e - EGP, ? - incomplete
Network                                     NextHop
         MED      LocalPref  Weight  Path
*> 3ffe:10:10::/64                          fe80::200:87ff:fe16:90d5%VLAN0005
         -        100        0       65532 i
*> 3ffe:10:20::/64                          fe80::200:87ff:fe16:90d5%VLAN0005
         -        100        0       65532 i
```

To display the communities that a route belongs to, use the `show ipv6 bgp` command with the `route` parameter specified.

Figure 28-22: Results of executing the show ipv6 bgp command (with the route parameter specified)

```
> show ipv6 bgp route 3ffe:10:10::/64
Date 20XX/10/20 21:09:12 UTC
BGP4+ Peer: 3ffe:172:16:2::2  , Remote AS: 65532
Local AS: 65531, Local Router ID: 192.168.1.100
Status Codes: d dampened, * valid, > active, S Stale, r RIB failure
Route 3ffe:10:10::/64
*> Next Hop fe80::200:87ff:fe16:90d5%VLAN0005
     MED: -, LocalPref: 100, Weight: 0, Type: External route
     Origin: IGP, IGP Metric: 0
     Path: 65532
     Communities: 1000:1002
```

(3) Displaying the execution result of learned route filtering

To display the result of learned route filtering based on the `COMMUNITIES` attribute, use the `show ipv6 bgp` operation command.

Figure 28-23: Results of executing the show ipv6 bgp command

```
> show ipv6 bgp
Date 20XX/10/20 21:10:09 UTC
Local AS: 65531, Local Router ID: 192.168.1.100
Status Codes: d dampened, * valid, > active, S Stale, r RIB failure
Origin Codes: i - IGP, e - EGP, ? - incomplete
   Network                                  Next Hop
         MED    LocalPref weight Path
*> 3ffe:10:10::/64                          fe80::200:87ff:fe16:90d5%VLAN0005
         -      120       0      65532 i
*  3ffe:10:10::/64                          3ffe:10:2:2::2
         -      80        0      65533 i
*> 3ffe:10:20::/64                          fe80::200:87ff:fe16:90d5%VLAN0005
         -      120       0      65532 i
*  3ffe:10:20::/64                          3ffe:10:2:2::2
         -      80        0      65533 i
*> 3ffe:192:169:10::/64                     3ffe:192:168:2::2
         -      100       0      i
*> 3ffe:192:169:20::/64                     3ffe:192:168:2::2
         -      100       0      i
```

(4) Displaying the advertised routes in a community

To display the `COMMUNITIES` attribute of advertised BGP4+ routes, use the `show ipv6 bgp` operation command with the `advertised-routes` parameter specified.

Figure 28-24: Results of executing the show ipv6 bgp command (with the advertised-routes parameter specified)

```
> show ipv6 bgp advertised-routes 3ffe:192:169:10::/64
Date 20XX/10/18 22:44:54 UTC
BGP4+ Peer: 3ffe:10:2:2::2     , Remote AS: 65533
Local AS: 65531, Local Router ID: 192.168.1.100
Status Codes: * valid, > active
Route 3ffe:192:169:10::/64
*> Next Hop 3ffe:192:168:2::2
```

```
     MED: -, LocalPref: -, Type: Internal route
     Origin: IGP
     Path: 65531
     Next Hop Attribute: 3ffe:10:1:2::1
     Communities: 1000:1001

BGP4+ Peer: 3ffe:172:16:2::2  , Remote AS: 65532
Local AS: 65531, Local Router ID: 192.168.1.100
Status Codes: * valid, > active
Route 3ffe:192:169:10::/64
*> Next Hop 3ffe:192:168:2::2
     MED: -, LocalPref: - , Type: Internal route
     Origin: IGP
     Path: 65531
     Next Hop Attribute: 3ffe:172:16:2::1
                         fe80::200:87ff:fe21:90da
     Communities: 1000:1001
```

28.6.3 Checking BGP4+ multipath

(1) List of operation commands

The following table describes the operation commands for BGP4+ multipath.

Table 28-22: List of operation commands

Command name	Description
show ipv6 route	Shows the routes in a routing table.
show ipv6 bgp	Shows information related to the BGP4+ protocol.

(2) Displaying a BGP4+ multipath

The following figure shows information related to *28.5.3 Configuring BGP4+ multipath*. To display multipath settings, use the `show ipv6 route` operation command.

Figure 28-25: Results of executing the show ipv6 route command

```
> show ipv6 route
Date 20XX/10/28 21:47:11 UTC
Total: 13 routes
Destination           Next Hop           Interface  Metric Protocol  Age
::1/128               ::1                localhost  0/0    Connected 10m 51s
3ffe:10:10::/64       fe80::5%VLAN0005   VLAN0005   -/-    BGP4+      4m 50s...1
                      fe80::6%VLAN0006   VLAN0006    -       -         -
3ffe:10:20::/64       fe80::5%VLAN0005   VLAN0005   -/-    BGP4+      4m 50s...2
                      fe80::6%VLAN0006   VLAN0006   -/-    BGP4+      4m 56s
3ffe:172:16::/64      3ffe:172:16:2::2   VLAN0007   0/0    Connected 10m 49s
3ffe:172:16:2::2/128  ::1                localhost  0/0    Connected 10m 49s
3ffe:172:17::/64      3ffe:172:17:2::2   VLAN0005   0/0    Connected 10m 49s
3ffe:172:17:2::2/128  ::1                localhost  0/0    Connected 10m 49s
3ffe:172:10::/64      fe80::5%VLAN0005   VLAN0005   -/-    BGP4+      4m 50s...3
                      fe80::6%VLAN0006   VLAN0006   -/-    BGP4+      4m 56s
3ffe:172:20::/64      fe80::5%VLAN0005   VLAN0005   -/-    BGP4+      4m 50s...4
                      fe80::6%VLAN0006   VLAN0006    -       -         -
3ffe:192:168:2::/64   3ffe:192:168:2::2  VLAN0006   0/0    Connected 10m 48s
3ffe:192:168:2::2     ::1                localhost  0/0    Connected 10m 48s
```

1 to 4: Multiple routes.

28.6.4 Checking capability negotiation

(1) List of operation commands

The following table describes the operation commands for capability negotiation.

Table 28-23: List of operation commands

Command name	Description
show ipv6 bgp	Shows information about the BGP4+ protocol.

(2) Checking negotiation

To display the result of capability negotiation, use the `show ipv6 bgp` operation command with the `neighbors` and `detail` parameters specified.

Figure 28-26: Results of executing the show ipv6 bgp command (with the neighbors and detail parameters specified)

```
> show ipv6 bgp neighbors detail
Date 20XX/10/17 15:52:14 UTC
BGP4+ Peer: 3ffe:10:1:2::2     , Remote AS: 65531
Remote Router ID: 10.1.2.102
    BGP4+ Status: Established       HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/10/17 15:51:00
    BGP4+ Version: 4                Type: Internal
    Local Address: 3ffe:10:1:2::1
    Local AS: 65531                 Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:52:00  Last Keep Alive Received: 15:52:00
    BGP4+ Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                   0          0          2          4
    BGP4+ Capability Negotiation: <IPv6-Uni Refresh Refresh(v)>            ...1
      Send   : <IPv6-Uni Refresh Refresh(v)>
      Receive: <IPv6-Uni Refresh Refresh(v)>
    Password: UnConfigured
BGP4+ Peer: 3ffe:192:168:2::2    , Remote AS: 65531
Remote Router ID: 192.168.1.102
    BGP4+ Status: Established       HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/10/17 15:50:43
    BGP4+ Version: 4                Type: Internal
    Local Address:3ffe:192:168:2::1
    Local AS: 65531                 Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:43  Last Keep Alive Received: 15:51:43
    BGP4+ Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                   0          0          2          4
    BGP4+ Capability Negotiation: <IPv6-Uni Refresh>                       ...2
      Send   : <IPv6-Uni Refresh Refresh(v)>
      Receive: <IPv6-Uni Refresh >
    Password: UnConfigured
BGP4+ Peer: 3ffe:10:2:2::2        , Remote AS: 65533
Remote Router ID: 10.2.2.102
    BGP4+ Status: Established       HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/10/17 15:50:30
    BGP4+ Version: 4                Type: External
    Local Address: 10.1.2.1
    Local AS: 65531                 Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:30  Last Keep Alive Received: 15:51:30
    BGP4+ Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                   0          0          2          4
    BGP4+ Capability Negotiation: <IPv6-Uni>                               ...3
      Send   : <IPv6-Uni Refresh Refresh(v)>
      Receive: <IPv6-Uni>
    Password: UnConfigured
BGP4+ Peer: 3ffe:172:16:2::2    , Remote AS: 65532
Remote Router ID: 172.16.1.102
    BGP4+ Status: Established       HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/10/17 15:49:35
    BGP4+ Version: 4                Type: External
    Local Address:3ffe:172:16:2::1
    Local AS: 65531                 Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:35  Last Keep Alive Received: 15:51:35
    BGP4+ Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                   0          0          3          5
    BGP4+ Capability Negotiation: <>                                       ...4
      Send    : <IPv6-Uni Refresh Refresh(v)>
      Receive: <>
    Password: UnConfigured
>
```

1. The following capabilities were successfully negotiated: `IPv6-Uni`: IPv6 unicast routing; `Refresh`: Route refresh capability (RFC 2918-compliant); and `Refresh(v)`: Route refresh capability (Capability Code: 128).
2. The following capabilities were successfully negotiated: `IPv6-Uni`: IPv6 unicast routing; and `Refresh`: Route refresh capability (RFC 2918-compliant).
3. The following capability was successfully negotiated: `IPv6-Uni`: IPv6 unicast routing.
4. Capabilities were not successfully negotiated.

28.6.5 Checking the route refresh functionality

(1) List of operation commands

The following table describes the operation commands for the route refresh capability.

Table 28-24: List of operation commands

Command name	Description
clear ipv6 bgp	Clears BGP4+ sessions or BGP4+-related information, or filters inbound or outbound routes using new BGP filter information.
show ipv6 route	Shows routing information stored in the routing table.
show ipv6 bgp	Shows information about the BGP4+ protocol.

(2) Checking negotiation of route refresh capability

First, make sure that route refresh capability has been successfully negotiated with the BGP4+ peer that will be requested to re-advertise its BGP4+ routes. To do so, use the `show ipv6 bgp` operation command with the `neighbors` parameter specified. If route refresh capability has not been negotiated, route refresh requests to relearn changed routes will not be sent to the remote peer.

Figure 28-27: Results of executing the show ipv6 bgp command (with the neighbors parameter specified)

```
> show ipv6 bgp neighbors 3ffe:172:16:2::2
Date 20XX/10/17 16:52:14 UTC
BGP4+ Peer: 3ffe:172:16:2::2     , Remote AS: 65532
Remote Router ID: 172.16.1.102
    BGP4+ Status: Established       HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/10/17 16:49:35
    BGP4+ Version: 4                Type: External
    Local Address: 3ffe:172:16:2::1
    Local AS: 65531                 Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 16:51:35  Last Keep Alive Received: 16:51:35
    BGP4+ Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                   1          1          4          6
    BGP4+ Capability Negotiation: <IPv6-Uni Refresh Refresh(v)>     ...1
      Send    : <IPv6-Uni Refresh Refresh(v)>
      Receive: <IPv6-Uni Refresh Refresh(v)>
    Password: UnConfigured
```

1. Route refresh capability with the remote peer was successfully negotiated.

(3) Requesting BGP4+ route re-advertisement and re-advertising the routes

To request that all BGP4+ peers re-advertise their BGP4+ routes, and to re-advertise the routes to all BGP4+ peers, use the `clear ipv6 bgp` command with the `* both` parameter specified.

Figure 28-28: Results of executing the clear ipv6 bgp command

```
#clear ipv6 bgp * both
```

(4) Checking the relearning and re-advertising of BGP4+ routes

To check that BGP4+ routes have been re-advertised and relearned by using the route refresh functionality, use the `show ipv6 bgp` command with the `neighbors` parameter specified.

Figure 28-29: Results of executing the show ipv6 bgp command (with the neighbors parameter specified)

```
> show ipv6 bgp neighbors 3ffe:172:16:2::2
Date 20XX/10/17 16:52:14 UTC
BGP4+ Peer: 3ffe:172:16:2::2     , Remote AS: 65532
Remote Router ID: 172.16.1.102
    BGP4+ Status: Established       HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/10/17 16:49:35
    BGP4+ Version: 4                Type: External
    Local Address: 3ffe:172:16:2::1
    Local AS: 65531                 Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 16:51:35  Last Keep Alive Received: 16:51:35
    BGP4+ Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                   2          2          11         14                ...1
    BGP4+ Capability Negotiation: <IPv6-Uni Refresh Refresh(v)>
      Send   : <IPv6-Uni Refresh Refresh(v)>
      Receive: <IPv6-Uni Refresh Refresh(v)>
    Password: UnConfigured
```

1. There has been an increase in the numbers of received `UPDATE` messages and sent `UPDATE` messages.

Notes

The `clear ipv6 bgp` operation command (with `* in`, `* out`, or `* both` specified) applies the new route filtering settings and implements the route refresh capability (see *28.4.5 Route refresh capability*). If route refresh capability has not been negotiated, no route refresh requests are made in order to relearn changed routes, although the route filter changes are applied.

28.6.6 Checking TCP MD5 authentication

(1) List of operation commands

The following table describes the operation commands for TCP MD5 authentication (BGP4+).

Table 28-25: List of operation commands

Command name	Description
show ipv6 bgp	Shows information about the BGP4+ protocol.

(2) Checking TCP MD5 authentication

To display whether TCP MD5 authentication has been performed, use the `show ipv6 bgp` command with the `neighbors` and `detail` parameters specified.

Figure 28-30: Results of executing the show ipv6 bgp command (with the neighbors and detail parameters specified)

```
> show ipv6 bgp neighbor detail
Date 20XX/10/07 21:24:24 UTC
BGP4+ Peer: 3ffe:192:168:2::2  , Remote AS: 65531
Remote Router ID: 192.168.2.100
    BGP4+ Status: Established       HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/10/07 21:23:48
    BGP4+ Version: 4                Type: Internal
    Local Address:3ffe:192:168:2::1
    Local AS: 65531                 Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 21:23:48  Last Keep Alive Received: 21:23:48
    BGP4+ Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                   0          0          0          3
    BGP4+ Capability Negotiation: <IPv6-Uni Refresh Refresh(v)>
      Send   : <IPv6-Uni Refresh Refresh(v)>
      Receive: <IPv6-Uni Refresh Refresh(v)>
    Password: UnConfigured                                              ...1

BGP4+ Peer: 3ffe:172:16:2::2   , Remote AS: 65532
Remote Router ID: 172.16.2.100
    BGP4+ Status: Established       HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/10/07 21:23:58
    BGP4+ Version: 4                Type: External
    Local Address:3ffe:172:16:2::1
    Local AS: 65531                 Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 21:23:58  Last Keep Alive Received: 21:23:58
    BGP4+ Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                   0          0          1          3
    BGP4+ Capability Negotiation: <IPv6-Uni Refresh Refresh(v)>>
      Send   : <IPv6-Uni Refresh Refresh(v)>
      Receive: <IPv6-Uni Refresh Refresh(v)>
    Password: Configured                                                ...2
```

1. MD5 authentication was not used for connection with the peer whose peer address is 3ffe:192:168:2::2.
2. MD5 authentication was not used for connection with the peer whose peer address is 3ffe:172:16:2::2.

Notes

A peer relationship is not established if TCP MD5 authentication fails (if the peer's BGP Status is not `Established`). Check the logged messages to check whether TCP MD5 authentication has failed.

28.6.7 Checking BGP4+ advertised route generation

(1) List of operation commands

The following table describes the operation commands for displaying information about generated BGP4+ advertised routes.

Table 28-26: List of operation commands

Command name	Description
show ipv6 bgp	Shows information about the BGP4+ protocol.
show ipv6 route	Shows routing information stored in the routing table.

(2) Checking BGP4+ advertised routes

(a) Displaying generated BGP4+ advertised routes

To display generated BGP4+ advertised routes, use the `show ipv6 bgp` operation command. In the following example, 3ffe:173:16::/48 and 3ffe:192:169:10::/64 are generated BGP4+ advertised routes.

Figure 28-31: Results of executing the show ipv6 bgp command

```
> show ipv6 bgp
Date 20XX/10/20 22:43:26 UTC
Local AS: 65531, Local Router ID: 192.168.1.100
Status Codes: d dampened, * valid, > active, S Stale, r RIB failure
Origin Codes: i - IGP, e - EGP, ? - incomplete
   Network                  Next Hop        MED    LocalPref Weight Path
*  3ffe:173:16::/48         ----            -      100       0      i
*  3ffe:192:169:10::/64     ----            -      100       0      i
```

(b) Checking whether routes have been advertised

To check whether a generated BGP4+ advertised route has been advertised, use the `show ipv6 bgp` operation command with the `advertised-routes` parameter specified.

Figure 28-32: Results of executing the show ipv6 bgp command (with the advertised-routes parameter specified)

```
> show ipv6 bgp advertised-routes 3ffe:173:16::/48
Date 20XX/10/29 18:08:54 UTC
BGP4+ Peer: 3ffe:172:16:2::2, Remote AS: 65532
Local AS: 65531, Local Router ID: 192.168.1.100
Status Codes: * valid, > active
Route 3ffe:173:16::/48
*  Next Hop ----
     MED: -, LocalPref: -,   Type: Internal route
     Origin: IGP
     Path:65531
     Next Hop Attribute: 3ffe:172:16:2::1

> show ipv6 bgp advertised-routes 3ffe:192:169:10::/64
Date 20XX/10/29 18:08:54 UTC
BGP4+ Peer: 3ffe:172:16:2::2, Remote AS: 65532
Local AS: 65531, Local Router ID: 192.168.1.100
Status Codes: * valid, > active
Route 3ffe: 3ffe:192:169:10::/64
*  Next Hop ----
     MED: -, LocalPref: -, Type: Internal route
     Origin: IGP
     Path:65531
     Next Hop Attribute: 3ffe:172:16:2::1
```

28.6.8 Checking route flap dampening

(1) List of operation commands

The following table describes the operation commands for route flap dampening.

Table 28-27: List of operation commands

Command name	Description
show ipv6 route	Shows routing information stored in the routing table.
show ipv6 bgp	Shows information about the BGP4+ protocol.
clear ipv6 bgp	Removes the suppressed status of a suppressed route and clears route flap statistics.

(2) Checking route flap dampening

To display routes suppressed by route flap dampening, use the `show ipv6 bgp` operation command with the `dampened-routes` parameter (only for the global network) specified.

Figure 28-33: Results of executing the show ipv6 bgp command (with the dampened-routes parameter specified)

```
>show ipv6 bgp neighbor 3ffe:172:16:2::2 dampened-routes
Date 20XX/10/29 18:08:54 UTC
Status Codes: d dampened, h history, * valid, > active
   Network                                      Peer Address
     ReUse
 d 3ffe:172:21:211::/64                         3ffe:172:16:2::2
     00:07:11
 d 3ffe:172:21:212::/64                         3ffe:172:16:2::2
     00:19:10
```

To display the flap state of the routes, use the `show ipv6 bgp` operation command with the `flap-statistics` parameter (only for the global network) specified.

Figure 28-34: Results of executing the show ipv6 bgp command (with the flap-statistics parameter specified)

```
>show ipv6 bgp flap-statistics
Date 20XX/10/29 18:08:54 UTC
Status Codes: d dampened, h history, * valid, > active
   Network                                      Peer Address
          Flaps      Duration ReUse    Penalty
 d 3ffe:172:21:211::/64                         3ffe:172:16:2::2
          114        00:12:30 00:07:11 5.0
 d 3ffe:172:21:212::/64                         3ffe:172:16:2::2
          108        00:12:30 00:19:10 4.0
 h 3ffe:172:27:119::/64                         3ffe:192:168:2::2
          2          00:11:20          1.7
 h 3ffe:172:27:191::/64                         3ffe:192:168:2::2
          2          00:11:20          1.7
*> 3ffe:172:30:189::/64                         3ffe:192:168:79:188
          1          00:05:10          0.6
*> 3ffe:172:30:192::/64                         3ffe:192:168:79:188
          3          00:05:10          0.6
>
```

28.6.9 Checking route reflection

(1) List of operation commands

The following table describes the operation commands for route reflection.

Table 28-28: List of operation commands

Command name	Description
show ipv6 route	Shows routing information stored in the routing table.
show ipv6 bgp	Shows information about the BGP4+ protocol.

(2) Checking route reflection

To display the route reflector clients, use the `show ipv6 bgp` operation command with the `neighbors` and `detail` parameters specified.

Figure 28-35: Results of executing the show ipv6 bgp command (with the neighbors and detail parameters specified)

```
> show ipv6 bgp neighbors detail
Date 20XX/01/17 15:52:14 UTC
BGP4+ Peer: 3ffe:192:168:2::2       , Remote AS: 65531
Remote Router ID: 192.168.100.2
    BGP4+ Status: Established       HoldTime: 180   , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/01/17 15:51:00
    BGP4+ Version: 4                Type: Internal RRclient              ...1
    Local Address: 3ffe:192:168:2::1
    Local AS: 65531                 Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:52:00  Last Keep Alive Received: 15:52:00
    BGP4+ Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                   0          0          2          4
    BGP4+ Capability Negotiation: <IPv4-Uni Refresh Refresh(v)>
      Send   : <IPv6-Uni Refresh Refresh(v)>
      Receive: <IPv6-Uni Refresh Refresh(v)>
    Password: UnConfigured

BGP4+ Peer: 3ffe:192:168:3::2    , Remote AS: 65531
Remote Router ID: 192.168.1.103
    BGP4+ Status: Established       HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/01/17 15:50:43
    BGP4+ Version: 4                Type: Internal RRclient              ...1
    Local Address: 3ffe:192:168:3::1
    Local AS: 65531                 Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:43  Last Keep Alive Received: 15:51:43
    BGP4+ Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                   0          0          2          4
    BGP4+ Capability Negotiation: <IPv4-Uni Refresh Refresh(v)>
      Send   : <IPv6-Uni Refresh Refresh(v)>
      Receive: <IPv6-Uni Refresh Refresh(v)>
    Password: UnConfigured

BGP4+ Peer: 3ffe:192:168:4::2    , Remote AS: 65531
Remote Router ID: 192.168.1.104
    BGP4+ Status: Established       HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/01/17 15:50:30
    BGP4+ Version: 4                Type: Internal RRclient              ...1
    Local Address: 3ffe:192:168:4::1
    Local AS: 65531                 Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:30  Last Keep Alive Received: 15:51:30
    BGP4+ Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                   0          0          2          4
    BGP4+ Capability Negotiation: <IPv4-Uni Refresh>
      Send   : <IPv6-Uni Refresh Refresh(v)>
      Receive: <IPv6-Uni Refresh Refresh(v)>
    Password: UnConfigured

BGP4+ Peer: 3ffe:172:16:2::2     , Remote AS: 65532
Remote Router ID: 172.16.1.102
    BGP4+ Status: Established       HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/01/17 15:49:35
    BGP4+ Version: 4                Type: External
    Local Address: 3ffe:172:16:2::1
    Local AS: 65531                 Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:35  Last Keep Alive Received: 15:51:35
    BGP4+ Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                   0          0          3          5
    BGP4+ Capability Negotiation: <IPv4-Uni Refresh Refresh(v)>
      Send   : <IPv6-Uni Refresh Refresh(v)>
      Receive: <IPv6-Uni Refresh Refresh(v)>
    Password: UnConfigured
>
```

1. Specified as a route reflector client.

To display reflected routes, use the `show ipv6 bgp` operation command with the `advertised-routes` parameter specified.

Figure 28-36: Results of executing the show ipv6 bgp command (with the advertised-routes parameter specified)

```
> show ipv6 bgp advertised-routes
Date 20XX/01/18 22:44:54 UTC
BGP4+ Peer: 3ffe:192:168:3::2       , Remote AS: 65531
Local AS: 65531, Local Router ID: 192.168.1.100
Origin Codes: i - IGP, e - EGP, ? - incomplete
Network                                 Next Hop
        MED    LocalPref Path
3ffe:192:169:10::/64                    3ffe:192:168:2::2
         120    100      i
3ffe:192:169:20::/64                    3ffe:192:168:2::2
         100    100      i
BGP4+ Peer: 3ffe:192:168:4::2       , Remote AS: 65531
Local AS: 65531, Local Router ID: 192.168.1.100
Origin Codes: i - IGP, e - EGP, ? - incomplete
Network                                 Next Hop
        MED    LocalPref Path
3ffe:192:169:10::/64                    3ffe:192:168:2::2
         120    100      i
3ffe:192:169:20::/64                    3ffe:192:168:2::2
         100    100      i
```

28.6.10 Checking confederations

(1) List of operation commands

The following table describes the operation commands for confederations.

Table 28-29: List of operation commands

Command name	Description
show ipv6 route	Shows routing information stored in the routing table.
show ipv6 bgp	Shows information about the BGP4+ protocol.

(2) Checking confederations

To display a confederation, use the `show ipv6 bgp` operation command with the `neighbors` and `detail` parameters specified.

Figure 28-37: Results of executing the show ipv6 bgp command (with the neighbors and detail parameters specified)

```
> show ipv6 bgp neighbors detail
Date 20XX/01/17 15:52:14 UTC
BGP4+ Peer: 3ffe:192:168:2::2    , Remote AS: 64512                       ...2
Remote Router ID: 192.168.100.2
    BGP4+ Status: Established       HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/01/17 15:51:00
    BGP4+ Version: 4                Type: Internal
    Local Address: 3ffe:192:168:2::1
    Local AS: 64512                 Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:52:00  Last Keep Alive Received: 15:52:00
    BGP4+ Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                   0          0          2          4
    BGP4+ Capability Negotiation: <IPv6-Uni Refresh Refresh(v)>
      Send    : <IPv6-Uni Refresh Refresh(v)>
      Receive: <IPv6-Uni Refresh Refresh(v)>
    Password: UnConfigured

Confederation ID: 65531, Member AS: 64512                                 ...1
BGP4+ Peer: 3ffe:192:168:4::2    , Remote AS: 64513                       ...2
Remote Router ID: 192.168.1.104
    BGP4+ Status: Established       HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/01/17 15:50:30
    BGP4+ Version: 4                Type: ConfedExt                       ...3
    Local Address: 3ffe:192:168:4::1
    Local AS: 64512                 Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:30  Last Keep Alive Received: 15:51:30
    BGP4+ Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                   0          0          2          4
    BGP4+ Capability Negotiation: <IPv6-Uni Refresh>
      Send   : <IPv6-Uni Refresh Refresh(v)>
      Receive: <IPv6-Uni Refresh Refresh(v)>
    Password: UnConfigured

Confederation ID: 65531, Member AS: 64512                                 ...1
BGP4+ Peer: 3ffe:192:168:5::2    , Remote AS: 64514                       ...2
Remote Router ID: 192.168.1.104
    BGP4+ Status: Established       HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/01/17 15:50:30
    BGP4+ Version: 4                Type: ConfedExt                       ...3
    Local Address: 3ffe:192:168:5::1
    Local AS: 64512                 Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:30  Last Keep Alive Received: 15:51:30
    BGP4+ Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                   0          0          2          4
    BGP4+ Capability Negotiation: <IPv6-Uni Refresh>
      Send   : <IPv6-Uni Refresh Refresh(v)>
      Receive: <IPv6-Uni Refresh Refresh(v)>
    Password: UnConfigured

BGP4+ Peer: 3ffe:172:16:2::2     , Remote AS: 65532
Remote Router ID: 172.16.1.102
    BGP4+ Status: Established       HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/01/17 15:49:35
    BGP4+ Version: 4                Type: External
    Local Address: 3ffe:172:16:2::1
    Local AS: 65531                 Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:35  Last Keep Alive Received: 15:51:35
    BGP4+ Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                   0          0          3          5
    BGP4+ Capability Negotiation: <IPv6-Uni Refresh Refresh(v)>
      Send   : <IPv6-Uni Refresh Refresh(v)>
      Receive: <IPv6-Uni Refresh Refresh(v)>
    Password: UnConfigured
>
```

1. The local router belongs to a member AS of the confederation.
2. Shows the member AS number of a connected peer.
3. The peer type of the connected peer is member AS peer.

28.6.11 Checking graceful restart

(1) List of operation commands

The following table describes the operation commands for a graceful restart.

Table 28-30: List of operation commands

Command name	Description
show ipv6 route	Shows routing information stored in the routing table.
show ipv6 bgp	Shows information about the BGP4+ protocol.

(2) Checking the graceful restart functionality

To display whether the graceful restart functionality is in operation, use the `show ipv6 bgp` operation command with the `neighbors` and `detail` parameters specified.

Figure 28-38: Results of executing the show ipv6 bgp command (with the neighbors and detail parameters specified)

```
> show ipv6 bgp neighbors detail
Date 20XX/01/17 15:52:14 UTC
BGP4+ Peer: 3ffe:192:168:2::2       , Remote AS: 65531
Remote Router ID: 192.168.100.2
    BGP4+ Status: Established       HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/01/17 15:51:00
    BGP4+ Version: 4                Type: Internal
    Local Address: 3ffe:192:168:2::1
    Local AS: 65531                 Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:52:00  Last Keep Alive Received: 15:52:00
    Graceful Restart: Receive                                             ...1
            Receive Status : Finished    20XX/01/16 19:11:12
       StalepathTime: 30

    BGP4+ Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                   0          0          2          4
    BGP4+ Capability Negotiation: <IPv6-Uni Refresh Refresh(v)
GracefulRestart>...2
      Send   : <IPv6-Uni Refresh Refresh(v) GracefulRestart(RestartTime:120s)>
      Receive: <IPv6-Uni Refresh Refresh(v) GracefulRestart(RestartTime:120s)>
    Password: UnConfigured

BGP4+ Peer: 3ffe:172:16:2::2     , Remote AS: 65532
Remote Router ID: 172.16.1.102
    BGP4+ Status: Established       HoldTime: 180  , Keepalive: 60
    Established Transitions: 1      Established Date: 20XX/01/17 15:49:35
    BGP4+ Version: 4                Type: External
    Local Address: 3ffe:172:16:2::1
    Local AS: 65531                 Local Router ID: 192.168.1.100
    Next Connect Retry: -           Connect Retry Timer: -
    Last Keep Alive Sent: 15:51:35  Last Keep Alive Received: 15:51:35
    Graceful Restart: Receive                                             ...1
            Receive Status : Finished    20XX/01/16 19:13:40
       StalepathTime: 30
    BGP4+ Message  UpdateIn   UpdateOut  TotalIn    TotalOut
                   0          0          3          5
    BGP4+ Capability Negotiation: <IPv6-Uni Refresh Refresh(v)
GracefulRestart>...2
      Send   : <IPv6-Uni Refresh Refresh(v) GracefulRestart(RestartTime:120s)>
      Receive: <IPv6-Uni Refresh Refresh(v) GracefulRestart(RestartTime:120s)>
    Password: UnConfigured
```

1. The device is acting as a graceful restart receiving router.
2. The peers successfully negotiated the graceful restart functionality at connection of the BGP4+ session.

To display routes used during a graceful restart of the originating router, use the `show ipv6 bgp` operation command.

Figure 28-39: Results of executing the show ipv6 bgp command

```
> show ipv6 bgp
Date 20XX/01/16 19:12:23 UTC
Local AS: 65531, Local Router ID: 192.168.1.100
Status Codes: d dampened, * valid, > active, S Stale, r RIB failure
Origin Codes: i - IGP, e - EGP, ? - incomplete
   Network                                 Next Hop
      MED   LocalPref Weight Path
S  3ffe:10:10::/48                         3ffe:172:16:2::2
      -     120       20     65532 65528 i                     ...1
S  3ffe:10:20::/48                         3ffe:172:16:2::2
```

```
      -     80        20     65532 65528 i                     ...1
*> 3ffe:172:20::/48                        3ffe:192:168:2::2
      -      100       10     65530    i
*  3ffe:172:30::/48                        3ffe:192:168:2::2
      100    100       10     65530 i
*  3ffe:192:168:10::/64                    3ffe:192:168:2::2
      -      100       10     65530 i
*> 3ffe:192:169:10::/64                    3ffe:192:168:2::2
      -      100       10           i
*> 3ffe:192:169:20::/64                    3ffe:192:168:2::2
      -      100       10           i
```

1. Shows the route used during restart of the originating router.

28.6.12 Checking the maximum number of learned BGP4+ routes

(1) List of operation commands

The following table describes the operation commands for limiting the number of learned BGP4+ routes.

Table 28-31: List of operation commands

Command name	Description
show ipv6 route	Shows routing information stored in the routing table.
show ipv6 bgp	Shows information about the BGP4+ protocol.
clear ipv6 bgp	Reconnects a peer that was disconnected because it exceeded the maximum number of BGP4+ routes that can be learned from a particular device.

(2) Checking the allowable maximum and actual number of BGP4+ routes learned from a peer

To check the allowable maximum and the actual number of BGP4+ routes (sum of the active and inactive paths) learned from a particular peer, use the `show ipv6 bgp` operation command with the `neighbors` parameter and an <*As*>, a <*Peer Address*>, or a <*Host name*> argument or the `detail` parameter specified.

Figure 28-40: Results of executing the show ipv6 bgp command (with the neighbors and detail parameters specified)

```
>show ipv6 bgp neighbors detail
 Date 20XX/01/13 18:45:09
 BGP4+ Peer: 3ffe:172:16:2::2, Remote AS: 65532
 Remote Router ID: 172.16.2.200
     BGP4+ Status:Idle               HoldTime: 90   , Keepalive: 60
     Established Transitions: 1      Established Date: 20XX/01/13 18:42:26...1
     BGP4+ Version: 4                Type: External
     Local Address: 3ffe:172:16:23::214
     Local AS: 65531                 Local Router ID: 172.16.2.100
     Next Connect Retry: 00:32,      Connect Retry Timer: 00:32
     Last Keep Alive Sent: 18:42:20, Last Keep Alive Received: 18:42:20
     NLRI of End-of-RIB Marker: Advertised and Received
     BGP4+ Message  UpdateIn UpdateOut TotalIn TotalOut
                    12       14        36      42
     BGP4+ Peer Last Error: Cease(Over Prefix Limit)                      ...2
     BGP4+ Routes  Accepted     MaximumPrefix RestartTime Threshold       ...3
                   0            1000          60m         80%
     BGP4+ Capability Negotiation: <IPv6-Uni>
       Send   : <IPv6-Uni>
       Receive: <IPv6-Uni>
     Password : Configured
 BGP4+ Peer: 3ffe:192:168:2::1, Remote AS: 65531
 Remote Router ID: 192.168.2.200
     BGP4+ Status:Active             HoldTime: 90   , Keepalive: 60
```

```
Established Transitions: 1      Established Date: 20XX/01/13 18:42:31
BGP4+ Version: 4                Type: Internal
Local Address: 3ffe:192:168:23::214
Local AS: 65531                 Local Router ID: 192.168.2.100
Next Connect Retry: 00:32,      Connect Retry Timer: 00:32
Last Keep Alive Sent: 18:44:31, Last Keep Alive Received: 18:44:31
NLRI of End-of-RIB Marker: Advertised and Received
BGP4+ Message  UpdateIn UpdateOut TotalIn TotalOut
               12       14        36      42
BGP4+ Routes  Accepted     MaximumPrefix RestartTime Threshold       ...4
              94           1000          none        75%
BGP4+ Capability Negotiation: <IPv6-Uni>
  Send    : <IPv6-Uni>
  Receive: <IPv6-Uni>
Password : Configured
```

1. The peer was disconnected at 20*XX*/01/13 18:42:26.
2. The peer was disconnected because it exceeded the maximum number of learned routes.
3. The disconnected peer was reconnected after 60 minutes.
4. Of the allowable maximum of 1000 routes, the peer has learned 942 routes from the specified peer.

(3) Reconnecting a BGP4+ session that was disconnected because the peer exceeded the learned routes limit

To reconnect a BGP4+ session that was disconnected because the peer exceeded the maximum number of learned BGP4+ routes, use the `clear ipv6 bgp` operation command with the `*` parameter specified or with a <*Peer Address*> or <*Host Name*> argument specified.

Reconnection of a BGP4+ session by using the command

1. `# clear ipv6 bgp 3ffe:172:16:2::2`

   Reconnects the BGP4+ session with remote peer address 3ffe:172:16:2::2, which was disconnected after exceeding the learned routes limit.

Chapter

29. Route Filtering (IPv6)

This chapter explains IPv6 route filtering.

29.1 Description of route filtering

29.1.1 Overview of route filtering

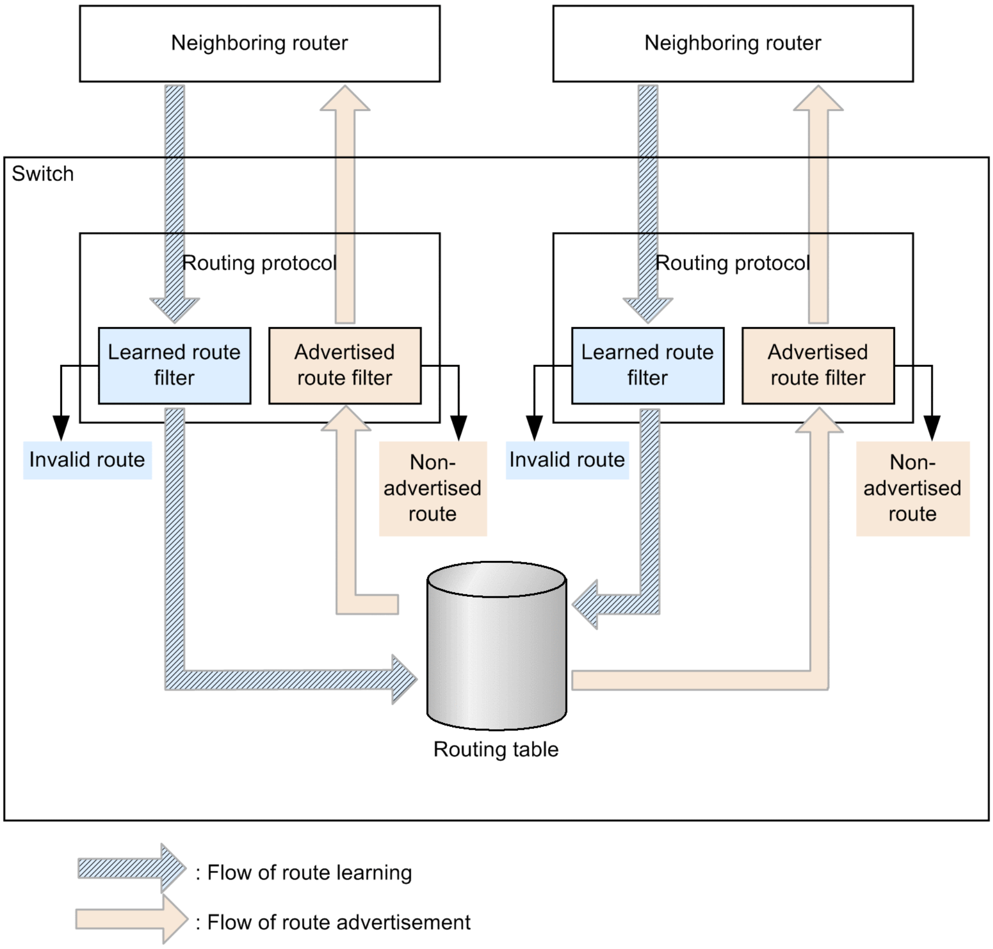
Route filtering allows you to control which routes the switch accepts by passing routing information through a filter.

There are three types of route filtering: learned route filtering, advertised route filtering, and extranet route filtering.

(1) Learned route filtering and advertised route filtering

The following figure shows the concepts of learned route filtering and advertised route filtering.

Figure 29-1: Concept of route filtering



(a) Learned route filtering

Learned route filtering works between the protocol and routing table to filter the routes the protocol learns. This allows you to control which learned routes the protocol should deem valid, or change the attributes of routes that meet certain conditions.

If you do not configure learned route filtering, all learned routes are considered valid.

(b) Advertised route filtering

Advertised route filtering works between the routing table and protocol to filter the routes in the routing table. This allows you to control which routes a protocol advertises, or change the attributes of advertised routes that meet certain conditions.

If you do not configure advertised route filtering, routes are advertised according to the policy of the associated protocol.

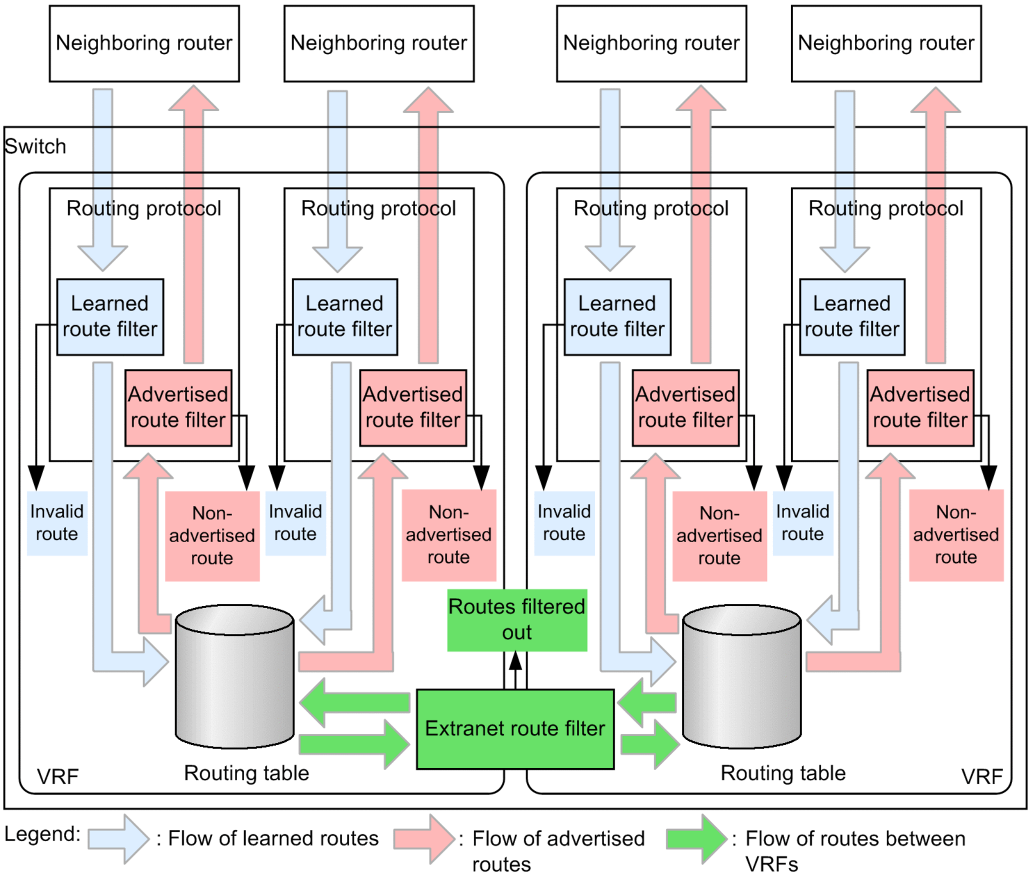
(2) Extranet route filtering [OS-L3SA]

Implementation of an extranet requires technology that allows different VRFs to access each other. For the Switch, one method for implementing an extranet is to exchange routing information between VRF routing tables. When routing information is exchanged between VRF routing tables, extranet route filtering is used to filter routes to be exchanged. By using this filter, you can control the exchange of routing information and change the attributes of routing information to be exchanged.

If extranet route filtering has not been set up, routing information is not exchanged between VRFs.

The following figure shows the concept of extranet route filtering.

Figure 29-2: Concept of extranet route filtering



Route filtering performed between VRFs in extranets is called inter-VRF route filtering.

29.1.2 Filtering methods

A filter is a list of conditions to be fulfilled. You apply a learned route or advertised route filter to routing traffic by specifying a filter ID in the route filtering configuration.

There are two main filter types you can use to filter routes in AX3600S series Switches: `prefix-list`, which filters routing traffic based only on the destination network, and `route-map`, which allows filtering by most key route attributes and allows those attributes to be modified. Other filters include `ipv6 access-list`, which filters routes based on IPv6 addresses, and `ip as-path access-list` and `ip community-list`, which filter routes based on BGP routing attributes. The `ipv6 access-list`, `ip as-path access-list`, and `ip community-list` filters are called from within `route-map`.

To configure a filter, a filter ID, filter conditions, and the action to take when the conditions are met need to be specified. The actions are `permit` or `deny`.

You can assign multiple filters to a single filter ID. When filtering a piece of routing information, the switch evaluates the filters that match the specified ID in the order in which they appear in the configuration, and then adopts the action of the first filter whose conditions match the route. Filters that can be assigned a sequence number are evaluated in sequence number order. Filters that lack a sequence number are evaluated in the order in which they are configured.

If none of the filter conditions associated with the specified ID match, the process ends with a `deny` action. This is called an implicit deny. Filters that specify conditions always end with an implicit deny statement.

Filters that do not specify any conditions end with a `permit` action.

(1) Filtering by destination network

(a) ipv6 prefix-list

The `ipv6 prefix-list` filter specifies a list of prefixes as its conditions. When `ipv6 prefix-list` is used as a route filter, the destination network of the route is compared with the prefixes in the filter.

In addition to prefixes, you can specify minimum and maximum mask lengths in the filter conditions. A route matches the conditions if its destination network is within the address range specified in the filter, and the mask length of the address is within the specified mask length range. If you do not specify a mask length range, a route matches the filter conditions only when the mask length of the prefix matches exactly. The following table describes examples of `ipv6 prefix-list` comparisons:

Table 29-1: Examples of prefix comparison with ipv6 prefix-list

Compared prefix	**Conditions for ipv6 prefix-list**		
	3ffe:5555::/32 **Matches when mask length is 32**	**3ffe:5555::/32 ge 32 le 48** **Matches when mask length is between 32 and 48**	**3ffe:5555::/32 ge 16 le 48** **Matches when mask length is between 16 and 48**
::/0	N	N	N
3ffe::/16	N	N	Y
3fff::/16	N	N	N
3ffe:5555::/32	Y	Y	Y
3ffe:5556::/32	N	N	N
3ffe:5555:feed::/48	N	Y	Y
3ffe:5555:feed:beef::/64	N	N	N

Legend: Y: Matches, N: Does not match

An `ipv6 prefix-list` filter can also be referenced as route destination conditions from the `match ipv6 address` command in `route-map`. The same method of comparison applies as if it were used as a standalone route filter.

A `match ipv6 route-source` command in `route-map` can invoke an `ipv6 prefix-list` filter as conditions for the learning source router. In this case, the conditions are the prefix (the learning source router's IPv6 address with a 128-bit mask applied) and the prefix destination.

(2) route-map

A `route-map` filter is used to specify a number of different conditions. This kind of filter can also change route attributes when certain conditions are met.

Statements in `route-map` each have a sequence number. For each sequence number, you can specify one line of filter conditions for each condition. Multiple filter conditions can be specified in that line. The conditions within a given line are related by an OR condition. Conditions that share the same sequence number but appear on different lines are subject to an AND condition.

The statement represented by a sequence number is considered satisfied when the route matches every one of its filter conditions. When the conditions are satisfied, the action associated with the sequence number is taken, and `route-map` terminates the filter.

If there is even one type of filter conditions for which none of the conditions match, the statement represented by the sequence number is considered not to be satisfied. In this case, the next sequence number in the `route-map` is evaluated.

The tables below list the types of filter conditions you can specify in a `route-map` filter and the attributes the filter can change.

Notes

When the switch applies a series of `route-map` filters to a route in succession, changes to route attributes will affect route filtering by subsequent `route-map` filters.

Suppose you use the RIPng `redistribute` command to apply a `route-map` filter that changes a tag value, and then use the RIPng `distribute-list out` command to apply a `route-map` filter that uses that tag value as a condition. First, the tag is modified by the `redistribute` command, and then a comparison is made using the modified tag value when the `route-map` filter of the `distribute-list out` command is applied.

Table 29-2: Types of route-map filter conditions

Route attribute used as conditions	Description	Configuration commands
Destination network	With the ID of a `prefix-list` or `access-list` filter specified as filter conditions, uses the specified filter to filter the destination network of a route. A match is assumed if the filter action is permit. If the action is deny, the attribute is assumed not to match.	match ipv6 address ipv6 prefix-list ipv6 access-list
Protocol type	Uses the specified routing protocol name as match conditions for the learning source protocol type of the route.	match protocol
Neighboring routers	With the ID of a prefix list or access list specified as filter conditions, uses the specified filter to filter the address of the learning source router. A match is assumed if the filter action is permit. If the action is deny, the attribute is assumed not to match. Only RIPng routes and BGP4+ routes include the address of the learning source router. Other route types cannot match these conditions.	match ipv6 route-source ipv6 access-list ipv6 prefix-list
Interface	Uses interfaces as conditions, and compares the interface with the interface of the next routing network hop. Routes with no next hop do not match the conditions. With BGP4+ learned route filtering, routes do not match any interface.	match interface

Route attribute used as conditions	Description	Configuration commands
Tag value	Uses the specified tag value as match conditions for a tag value of the route. Routes with no tags are assumed to have a tag value of 0.	match tag
AS_PATH attribute	With the ID of `ip as-path access-list` specified as filter conditions, uses the specified `ip as-path access-list` to filter the `AS_PATH` attribute of the route. A match is assumed if the action is permit. If the action is deny, the attribute is assumed not to match. Routes with no `AS_PATH` attribute are assumed to have an `AS_PATH` length of 0.	match as-path ip as-path access-list
COMMUNITIES attribute	With the ID of `ip community-list` specified as filter conditions, uses the specified `ip community-list` to filter the `COMMUNITIES` attribute of the route. A match is assumed if the action is permit. If the action is deny, the attribute is assumed not to match. Routes with no `COMMUNITIES` attribute are assumed to lack community affiliations.	match community ip community-list
ORIGIN attribute	Uses the specified value (`IGP`, `EGP`, or `INCOMPLETE`) as match conditions for the `ORIGIN` attribute of the route. Routes with no `ORIGIN` attribute are assumed to have an IGP origin.	match origin
Route type	Specifies an OSPFv3 route type or local (indicating a route generated by the BGP `network` command) as filter conditions and compares with the route's protocol-dependent route type.	match route-type
VRF ID	Uses the specified VRF ID as a match condition for a VRF ID of the route.	match vrf

Note: If the conditions for an interface condition specify an interface that is not used for IPv4 or IPv6, the interface conditions match any route.

Table 29-3: Route attributes changeable by route-map filter

Changeable attribute	Description	Configuration commands
Distance	Changes the route priority (distance) in the routing table. Valid only for learned route filtering.	set distance
Metric	Changes the metric or `MED` attribute. Values can be added to or subtracted from as well as replaced. For route filtering in BGP4+, the route can inherit the metric of the route to the BGP `NEXT_HOP` attribute.	set metric set metric-type internal (inherits the metric of the route to the `NEXT_HOP` attribute)
MED attribute		
Tag value	Changes the tag value of the route.	set tag
LOCAL_PREF attribute	Changes the `LOCAL_PREF` attribute of the route. Values can be added to or subtracted from as well as replaced. Used with BGP4+ route filtering.	set local-preference

Changeable attribute	Description	Configuration commands
AS_PATH attribute	Changes the AS_PATH attribute of the route. The filter is limited to adding the AS number of the sending peer. Used with route filtering for BGP4+ routes learned and advertised by external peers.	set as-path prepend count
COMMUNITIES attribute	Changes the COMMUNITIES attribute of the route. The filter can replace, add, and delete communities. Used with BGP4+ route filtering.	set community set community-delete
ORIGIN attribute	Changes the ORIGIN attribute of the route. Used with BGP4+ route filtering.	set origin
OSPF metric type	Changes the metric type. Used with OSPFv3 advertised route filtering.	set metric-type

(3) Other filters

In addition to the filters above, you can use the filters below for route filtering. You use the filters below by referencing them as filter conditions in `route-map`.

(a) ipv6 access-list

The main purpose of the `ipv6 access-list` filter is to filter packets. However, the filter can also be used to filter routes.

If the `match ipv6 address` command in `route-map` is used to invoke an `ipv6 access-list` filter as route destination conditions, the route destination network address is compared against the destination address in the conditions. The other conditions, such as the sender address, the type of the upper layer protocol, and the port number, are ignored.

If the `match ipv6 route-source` command in `route-map` is used to invoke an `ipv6 access-list` filter as learning source router conditions, the IPv6 address of the learning source router is compared against the destination address in the conditions. The other conditions, such as the sender address, the type of the upper layer protocol, and the port number, are ignored.

(b) ip as-path access-list

This filter applies exclusively to the `AS_PATH` attribute. It compares the string representation of the `AS_PATH` attribute against conditions specified by a regular expression. You call this filter with the `match as-path` command in `route-map`. For details about regular expressions, see *(e)Regular expressions*.

The `AS_PATH` attribute is a string of decimal AS numbers separated by spaces.

You cannot specify the path type with the `AS_PATH` attribute as a filter condition. Filtering performed on all path types included in the `AS_PATH` attribute for the AS number is specified as a filter condition. In the following example, a route with the following `AS_PATH` attribute is filtered:

AS_PATH attribute

```
AS_SEQ: 100 200 300, AS_SET: 1000 2000 3000, AS_CONFED_SEQUENCE: 65001 65002
```

Display format of the AS_PATH attribute for operation commands

```
100 200 300 {1000 2000 3000} (65001 65002)
```

With the above `AS_PATH` attribute, any of the following AS numbers will match the filter:

- "100 200 300"
- "1000 2000 3000"
- "65001 65002"

- "300 1000"

Note that special characters such as curly brackets ({ }) and parentheses (( )) are used as regular expressions of path type notation for operation commands and cannot be used to specify a path type.

Because the `AS_SET` attributes are sorted in ascending order when receiving a BGP4+ route, the sorting result is filtered.

(c) ip community-list standard

This filter applies exclusively to the `COMMUNITIES` attribute. You can specify multiple communities as filter conditions. The filter matches if the `COMMUNITIES` attribute of the route contains every community you specify. You call this filter with the `match community` command in `route-map`.

(d) ip community-list expanded

This filter applies exclusively to the `COMMUNITIES` attribute. It compares the string representation of the `COMMUNITIES` attribute against conditions specified by a regular expression. You call this filter with the `match community` command in `route-map`. For details about regular expressions, see *(e)Regular expressions*.

The string representation of the `COMMUNITIES` attribute consists of community values converted to character strings and separated by spaces. The values appear in order from smallest to largest. The following table describes the notation used for community values:

Table 29-4: String representations of the COMMUNITIES attribute

Community value	Character string
0xFFFFFF01 (hexadecimal)	no-export
0xFFFFFF02 (hexadecimal)	no-advertise
0xFFFFFF03 (hexadecimal)	local-AS
All other cases	*<AS number>*:*<last 2 octets>* *<AS number>* and *<last 2 octets>* are both written in decimal notation.

(e) Regular expressions

A regular expression is a means of describing a text pattern. You can use regular expressions to represent patterns like repeating strings. Regular expressions can be used to specify the filter conditions for the `AS_PATH` and `COMMUNITIES` attributes.

In regular expressions, you can use simple characters such as numerals, upper and lower case letters, symbols (excluding double-quotation marks), and special characters. Simple characters match their equivalent in the character string, as do special characters if preceded with the \ symbol. Each special character represents a pattern. The following table describes the special characters and the patterns they represent:

Table 29-5: Special characters and patterns

Special charact er	Pattern
.	Represents any single character including spaces.
*	Indicates that the preceding character or set of characters repeats zero or more times.
+	Indicates that the preceding character or set of characters repeats one or more times.
?	Represents 0 or 1 occurrence of the preceding character or set of characters (press **Ctrl** + **V**, and then enter `?` during command entry).

Special character	Pattern
^	Indicates the first character in the string.
$	Indicates the last character in the string.
_	Represents the first or last character of a string, a space, an underscore (_), a comma (,), a left parenthesis (() and a simple character, a right parenthesis ()) and a simple character, a left curly bracket ({), a right curly bracket (}), a left angled bracket (<), or a right angled bracket (>).
[ ]	Represents any single character from the character range inside `[]`. Except for the following, special characters act as simple characters within square brackets. `^`: When a caret is used as the first character in square brackets, the expression matches any character except those in the brackets. `-`: Indicates the beginning and end of a character range. Make sure that the character before the hyphen has a lower character code than the character after it. For details about character codes, see *Table 1-3 List of character codes* in the manual *Configuration Command Reference Vol. 1 For Version 11.10.* Example: `[6-8]` matches any one of 6, 7, or 8. `[^6-8]` matches any single character other than 6, 7, or 8.
( )	Indicates a group of characters. You can specify a maximum of nine character groups in a nested structure.
|	Represents an OR condition.
\	Treats a special character preceded by a backslash as a simple character.

The following table describes the priority of operator characters in regular expressions.

Table 29-6: Operator priority in regular expressions

Priority	Character
High	( )
↑	* + ?
↓	Simple characters, `.`, `[`, `]`, `^`, and `$`
Low	|

When you specify a regular expression in a configuration command or operation command, enclose it in double quotation marks (`"`).

Example 1:

```
> show ipv6 bgp aspath-regexp "^$"
```

Example 2:

```
(config)# ip as-path access-list 10 permit "_100_"
```

29.1.3 RIPng

(1) RIPng learned route filtering

In RIPng, you can filter every route that the protocol has learned. Routes denied by the filter are not added to the routing table.

(a) Method and procedure for applying filters

Learned routes are filtered according to conditions specified by the `distribute-list in` command. You can solely filter routes that the protocol learns from a specific interface by specifying the interface in the command parameters. The table below shows the configuration commands used to filter learned routes in RIPng.

When the switch learns a route, it applies the specified filters in the order shown in the table below. If there are no applicable filters or every filter gives a permit result, the route is entered into the routing table as a valid route. The learned route does not enter the routing table if it is denied by even one filter.

Table 29-7: Configuration commands for RIPng learned route filtering

Command name	Parameter	Filtered routes
distribute-list in (RIPng)	<*Interface*>	Filters RIPng routes learned from the specified IPv6 interface.
	None	Filters all learned RIPng routes.

(b) Route attributes changeable by learned route filtering

The table below describes the attributes that can be changed by RIPng learned route filtering.

The modified metric is used to define route priority in RIPng. The modified distance is used to define the relative priority of routing protocols.

Table 29-8: Route attributes changeable by RIPng learned route filtering

Attribute	Default value
Distance	The value specified by the distance command (RIPng). If no value is specified, 120 is used.
Metric	The attribute of the received route
Tag value	The attribute of the received route

Notes

- We recommend that you do not use methods other than addition to modify the metric. Replacing or subtracting from the metric might cause routing loops and prevent packets from routing correctly.
- You can configure a route filter to change the metric of a route to 16 or greater. However, a RIPng route with a metric of 16 or greater will be deemed invalid.
- Changes to metrics made using the `metric-offset` configuration command take effect after learned route filtering has taken place. You can use the `metric-offset` command to further modify a metric that was changed by a route filter. The route will be deemed invalid if its metric is 16 or greater after modification by the `metric-offset` command.
- The tag can have a maximum value of 4294967295. However, when RIPng advertises the modified route, it uses the lower 16 bits of the binary expression and discards the rest.

(2) RIPng advertised route filtering

The RIPng only advertises the prioritized routes in the routing table. However, it does not advertise routes that are subject to a split horizon.

If you do not configure advertised route filtering, the protocol will advertise RIPng routes and direct routes to RIPng interfaces.

Notes

When advertising OSPFv3 or BGP4+ routes, configure the switch to change the metric in the course of advertised route filtering, or assign an advertised metric. These routes have a default metric of 16 and would otherwise not be advertised.

(a) Route attributes changeable by advertised route filtering

The following table describes the attributes that can be changed by RIPng advertised route filtering.

Table 29-9: Route attributes changeable by RIPng advertised route filtering

Attribute	Learning source protocol	Default value
Metric	Directly connected route Summarized route	1
	Static route	Uses the value specified by default-metric. If no value is specified, 1 is used.
	RIPng route	Inherits the metric of the routing information.
	OSPFv3 route BGP4+ route Route imported from another VRF or the global network	Inherits the metric of the routing information if the `inherit-metric` command is configured. 16 is used if the routing information has no metric. If `inherit-metric` is not configured, the value specified by `default-metric` is used. If neither `inherit-metric` nor `default-metric` are configured, 16 is used as the metric.
Tag value	Common to all protocols	Inherits the tag value of the routing information.

Notes

- When using RIPng to advertise a RIPng route, we recommend that you do not use methods other than addition to modify the metric. Replacing or subtracting from the metric might cause routing loops and prevent packets from routing correctly.
- You can configure a route filter to change the metric of a route to 16 or greater. However, the protocol will not advertise routes with a metric of 16 or greater.
- Metric changes made using the `metric-offset` configuration command take effect after advertised route filtering has taken place. You can use the `metric-offset` command to further modify a metric that was changed by a route filter. A route will not be advertised if its metric is 16 or greater after modification by the `metric-offset` command.
- If you change a tag to a value greater than 65535, the protocol advertises the lower 16 bits of the binary expression and discards the rest.

(3) Method and procedure for applying filters

The application of advertised route filtering involves the following three steps:

1. First, select the routes to be advertised by RIPng. Specify the learning source protocol of the routes you want advertised. To specify the protocol, use the `redistribute` configuration command. By specifying a route type in the `redistribute` command, you can limit advertised routes to those of a certain type. By specifying `route-map`, you can advertise only those routes that the associated filters permit. The `redistribute` command compares the route attributes in the routing table against the conditions.

   RIPng routes and directly connected routes of a RIPng interface are advertised regardless of whether they are specified in the `redistribute` command.

   You can also change the attributes of advertised routes by specifying the new values directly in the `redistribute` command, or by specifying `route-map` in the `redistribute` command that changes the route attributes.

2. The advertised route takes on the default metric of the protocol. If the `redistribute` command changes the metric, the route uses the metric assigned by the command.

   For details about the default metrics for RIPng routes, see *Table 29-9: Route attributes* *changeable by RIPng advertised route filtering.*

3. Use the parameters of the `distribute-list out` command to filter the routes selected by the `redistribute` command. If you specify an interface in the command parameters, the filter is applied only when advertising routes to the specified interface. If you specify a protocol, the filters apply only to routes learned by the specified protocol. The table below describes the configuration command and its parameters.

   When advertising routes to a RIPng interface, applicable filters are selected according to the destination and learning source protocol, and then applied in the order shown in the table. If there are no applicable filters or every filter gives a permit result, the route is advertised to the specified destination. The route is not advertised to the destination if it is denied by even one filter.

   If you specify `route-map` in the `distribute-list out` command, routes are filtered according to the default advertising attributes and the attributes after modification by the `redistribute` command.

   You can also change an attribute of an advertised route by specifying `route-map` in the `distribute-list out` command that performs the desired change.

Table 29-10: Configuration command used for RIPng advertised route filtering

Command name	Parameter	Filtered routes
distribute-list out (RIPng)	<*Interface*>	Filters routes advertised from a specific IPv6 interface.
	<*Protocol*>	Filters routes matching a specific protocol regardless of their advertising destination.
	None	Filters all routes regardless of their advertising destination.

29.1.4 OSPFv3 [OS-L3SA]

(1) OSPFv3 learned route filtering

Of the routes computed by the SPF algorithm, OSPFv3 only allows external AS routes to be filtered. External AS routes that do not pass the filter are entered into the routing table as invalid routes.

Intra-area and inter-area routes are not filtered before entering the routing table.

Even when a route is disabled by the learned route filtering process, corresponding routes are still formed in other routers. This is because the LSA that originates the route is propagated to the other routers within the OSPFv3 domain. Learned route filtering filters the external AS routes generated from the LSA, but not the LSA itself.

(a) Method and procedure for applying filters

The switch filters the external AS routes subject to the filters specified in the `distribute-list in` command. The table below shows the configuration command used to filter learned routes in OSPFv3.

If there are no applicable filters or every filter gives a permit result, the route is entered into the routing table as a valid route. The route is deemed an invalid route if it is denied by a filter.

Table 29-11: Configuration command for OSPFv3 learned route filtering

Command name	Filtered routes
distribute-list in (OSPFv3)	Filters external AS routes calculated in the designated OSPFv3 domain.

(b) Route attributes changeable by learned route filtering

The table below lists the attribute that can be changed by OSPFv3 learned route filtering.

OSPFv3 learned route filtering can only change distances. The modified distance is used to define the relative priority of routing protocols.

Table 29-12: Route attributes changeable by OSPFv3 learned route filtering

Attribute	Default value
Distance	The value specified by distance ospf (OSPFv3). If no value is specified, 110 is used.

(2) OSPFv3 advertised route filtering

In OSPFv3, a directly connected route of an OSPFv3 interface is advertised as an intra-area or inter-area route. This behavior is outside the control of advertised route filtering.

OSPFv3 routes are also propagated to other routers. This also cannot be controlled by route filtering. This is because an LSA that originates a route is propagated unconditionally regardless of what filters are in place.

You can use advertised route filtering to advertise prioritized routes other than the above to OSPFv3. Such routes will be advertised as external AS routes.

If you do not configure advertised route filtering, the OSPFv3 protocol advertises no routes except OSPFv3 routes and directly connected routes of OSPFv3 interfaces.

(a) Route attributes changeable by advertised route filtering

The following table describes the attributes that can be changed by OSPFv3 advertised route filtering.

Table 29-13: External AS route attributes changeable by OSPFv3 advertised route filtering

Attribute	Source protocol	Default value
Metric	Directly connected route	20
	BGP4+ route	The value specified by `default-metric` (OSPFv3). If no value is specified, 1 is used.
	Other	The value specified by `default-metric` (OSPFv3). If no value is specified, 20 is used.
OSPFv3 route type	Common to all protocols	Type 2 external AS
Tag value	Common to all protocols	Inherits the tag value of the routing information.

Notes

You can configure a route filter to change the metric of a route to 16777215 or greater. However, such a route will not be advertised.

(b) Method and procedure for applying filters

The application of advertised route filtering involves the following steps:

1. First, select the routes to be advertised by OSPFv3. Specify the learning source protocol of the routes you want advertised. To specify the protocol, use the `redistribute` configuration command. However, you cannot re-advertise the routes of a given OSPFv3 domain by specifying it in the `redistribute` command.

   By specifying a route type in the `redistribute` command, you can limit advertised routes to those of a certain type. By specifying `route-map`, you can advertise only those routes that the associated filters permit. The `redistribute` command compares the route attributes in the routing table against the conditions in `route-map`.

You can also change the attributes of advertised routes by specifying the new values directly in the `redistribute` command, or by specifying `route-map` in the `redistribute` command that changes the route attributes.

2. The advertised route takes on the default metric and OSPFv3 route type configured for the protocol. If you used the `redistribute` command to change the attribute value, the route retains the attribute value assigned by the command.

   For details about the default metrics for advertised OSPFv3 routes, see *Table 29-13: External AS route attributes changeable by OSPFv3 advertised route filtering*.

3. Use the parameters of the `distribute-list out` command to filter the routes selected by the `redistribute` command. If you specify a protocol in the command parameters, the filter applies only to routes learned by the specified protocol. The table below describes the configuration command and its parameters.

   When advertising routes to an OSPFv3 domain, applicable filters are selected according to the learning source protocol, and then applied in the order shown in the table. The route is advertised if there are no applicable filters or every filter gives a permit result. The route is not advertised if it is denied by even one filter.

   If you specify `route-map` in the `distribute-list out` command, routes are filtered according to the default advertising attributes and the attributes after modification by the `redistribute` command.

   You can also change an attribute of an advertised route by specifying `route-map` in the `distribute-list out` command that performs the desired change.

Notes

If you execute the `match route-type` command while performing advertised route filtering by means of the `distribute-list out` command in step 3, routes will match `external`, and `external 1` or `external 2`. This is because the OSPFv3 route type in the route attribute will already have been rewritten to indicate a type 1 or type 2 external route.

Table 29-14: Configuration command used for OSPFv3 advertised route filtering

Command name	Parameter	Filtered routes
distribute-list out (OSPFv3)	<*Protocol*>	Filters routes matching a specific protocol regardless of their advertising destination.
	None	Filters all routes regardless of their advertising destination.

29.1.5 BGP4+ [OS-L3SA]

(1) BGP4+ learned route filtering

In BGP4+, you can filter every route that the protocol learns. By default, routes that are denied by the filter are not added to the routing table.

Notes

After you specify or change the BGP4+ learned route filtering settings, execute the `clear ipv6 bgp * in` or `clear ipv6 bgp * both` operation command at the appropriate time. Route filtering will operate according to its previous settings until you execute the command.

If you execute `clear ipv6 bgp * in`, the new route filtering settings apply to learned route filtering only. If you execute `clear ipv6 bgp * both`, the new settings apply to learned route filtering and advertised route filtering.

(a) Method and procedure for applying filters

Learned routes are filtered according to the filters specified in the `distribute-list in` and `neighbor in` commands. The filters specified in `neighbor in` apply only to routes learned from specific peers or peers belonging to a specific peer group. The table below describes the configuration commands used in BGP4+ learned route filtering.

When the switch learns a route, it applies the configured filters in the order shown in the table below. If there are no applicable filters or every filter gives a permit result, the route is entered into the routing table as a valid route. The learned route is considered an invalid route if it is denied by even one filter.

Table 29-15: Configuration commands for BGP4+ learned route filtering

Command name	Parameter	Filtered routes
neighbor in (BGP4+) (with `route-map` specified)	<*IPv6*> (peer address)	Filters routes learned from the specified peers.
neighbor in (BGP4+) (with `prefix-list` specified)	<*IPv6*> (peer address)	Filters routes learned from the specified peers.
neighbor in (BGP4+) (with `route-map` specified)	<*Peer-Group*> (peer group)	Filters routes learned from peers belonging to the specified peer groups.
neighbor in (BGP4+) (with `prefix-list` specified)	<*Peer-Group*> (peer group)	Filters routes learned from peers belonging to the specified peer groups.
distribute-list in (BGP4+)	None	Filters all routes learned by BGP4+.

(b) Route attributes changeable by learned route filtering

The table below lists the attributes that can be changed by BGP4+ learned route filtering.

Of the values below, BGP4+ uses values other than the distance to select prioritized routes. The distance defines the relative priority of routing protocols.

Table 29-16: Route attributes changeable by BGP4+ learned route filtering

Attribute	Default value
Distance	The value specified by `distance bgp`. If no value is specified, the following value is used: Internal peer: 200 External peer: 20 Member AS peer: 200
MED attribute	The attribute value of the received route
LOCAL_PREF attribute	Internal peer: The attribute value of the received route. External peer: The value specified by `bgp default local-preference`. If no value is specified, 100 is used. Member AS peer: The attribute value of the received route.
AS_PATH attribute	The attribute value of the received route
COMMUNITIES attribute	The attribute value of the received route
ORIGIN attribute	The attribute value of the received route

Notes

An AS can only be added to the `AS_PATH` attribute of routes learned from an external peer. You cannot add an AS to the `AS_PATH` attribute of a route learned from an internal peer or a member AS peer.

(2) BGP4+ advertised route filtering

In addition to the prioritized route in the routing table, the BGP4+ protocol can advertise BGP4+ routes that have been superseded by higher-prioritized routes from other routing protocols, and BGP4+ routes generated by the `network` command. If a situation arises in which all three types of routes with the same destination network are to be advertised, the protocol selects one route to advertise, in the following order: the prioritized route, the superseded BGP4+ route, and the `network`-generated route.

If you do not configure advertised route filtering, only BGP4+ routes are advertised. However, the protocol cannot advertise a route back to the peer from which it learned the route.

Notes

> After you specify or change the BGP4+ advertised route filtering settings, execute the `clear ipv6 bgp * out` or `clear ipv6 bgp * both` operation command at the appropriate time. Route filtering will operate according to its previous settings until you execute the command.
>
> If you execute `clear ipv6 bgp * out`, the new route filtering settings apply to advertised route filtering only. If you execute `clear ipv6 bgp * both`, the new settings apply to learned route filtering and advertised route filtering.

(a) Route attributes changeable by advertised route filtering

The following table describes the attributes that can be changed by BGP4+ advertised route filtering.

Table 29-17: Route attributes changeable by BGP4+ advertised route filtering

Attribute	Default value
MED attribute	Differs depending on the destination peer type and learning source protocol. When advertising to an internal peer: If a BGP4+ route is used, the metric is inherited. If a non-BGP4+ route is used, the value specified by `default-metric` is inherited. If no value is specified, the route is advertised without a `MED` attribute value. When advertising to an external peer: The value specified by `default-metric` is used. If no value is specified, the route is advertised without a `MED` attribute value. When advertising to a member AS peer: If a BGP4+ route is used, the metric is inherited. If a non-BGP4+ route is used, the value specified by `default-metric` is inherited. If no value is specified, the route is advertised without a `MED` attribute value.
LOCAL_PREF attribute	If a BGP4+ route is used, the `LOCAL_PREF` attribute is inherited. If a non-BGP4+ route is used, the value specified by `bgp default local-preference` is inherited. If no value is specified, 100 is used. If the advertising destination peer is an external peer, the advertisement will not include a `LOCAL_PREF` attribute.
AS_PATH attribute	Inherits the value of the route in the routing table.
ORIGIN attribute	
COMMUNITIES attribute	

Notes

> If you do not configure `neighbor send-community`, advertisements will not include a `COMMUNITIES` attribute.

(b) Method and procedure for applying filters

The application of advertised route filtering involves the following steps:

1. First, select the routes to be advertised by BGP4+. Specify the learning source protocol of the routes you want advertised. To specify the protocol, use the `redistribute` configuration command. By specifying a route type in the `redistribute` command, you can limit

advertised routes to those of a certain type. By specifying `route-map`, you can advertise only those routes that the associated filters permit. The `redistribute` command compares the route attributes in the routing table against the conditions.

BGP4+ routes are advertised regardless of whether you specify the `redistribute` command.

You can also change the attributes of advertised routes by specifying the new values directly in the `redistribute` command, or by specifying `route-map` in the `redistribute` command that changes the route attributes.

2. The `MED` and `LOCAL_PREF` attributes take on the default values determined by the protocol. If you used the `redistribute` command to change the attribute value, the route retains the attribute value assigned by the command.

   For details about the default attribute values for advertised BGP4+ routes, see *Table 29-17: Route attributes changeable by BGP4+ advertised route filtering*.

3. The routes selected by the `redistribute` command are filtered by applying the filters specified in the `distribute-list out` and `neighbor out` commands. The filters specified in `neighbor out` apply only to routes advertised to specific peers or peers belonging to a specific peer group. If you specify a protocol, the filters apply only to routes learned by the specified protocol. The table below describes the configuration commands and the routes to which they apply.

   When advertising a route to a peer, the switch selects the applicable filters according to the advertising destination and learning source protocol, and then applies them in the order shown in the table. If there are no applicable filters or every filter gives a permit result, the route is advertised to the specified peer. The route is not advertised to the peer if it is denied by even one filter.

   If you specify `route-map` in a `neighbor out` or `distribute-list out` command, filtering takes place according to the default advertising attributes and the attributes after modification by the `redistribute` command.

   You can also change an attribute of an advertised route by specifying `route-map` in the `neighbor out` or `distribute-list out` command that performs the desired change.

Table 29-18: Configuration commands used for BGP4+ advertised route filtering

Command name	Parameter	Filtered routes
neighbor out (BGP4+) (with `route-map` specified)	<*IPv6*> (peer address) <*Protocol*>	Filters routes advertised to a specific peer using a specific protocol.
neighbor out (BGP4+) (with `prefix-list` specified)	<*IPv6*> (peer address) <*Protocol*>	
neighbor out (BGP4+) (with `route-map` specified)	<*IPv6*> (peer address)	Filters routes advertised to a specific peer.
`neighbor out` (BGP4+) (with `prefix-list` specified)	<*IPv6*> (peer address)	
neighbor out (BGP4+) (with `route-map` specified)	<*Peer-Group*> (peer group) <*Protocol*>	Filters routes advertised to a member of a specific peer group using a specific protocol.
neighbor out (BGP4+) (with `prefix-list` specified)	<*Peer-Group*> (peer group) <*Protocol*>	
neighbor out (BGP4+) (with `route-map` specified)	<*Peer-Group*> (peer group)	Filters routes advertised to a member of a specific peer group.
neighbor out (BGP4+) (with `prefix-list` specified)	<*Peer-Group*> (peer group)	

Command name	Parameter	Filtered routes
distribute-list out (BGP4+)	*<Protocol>*	Filters routes matching a specific protocol regardless of their advertising destination.
	None	Filters all routes regardless of their advertising destination.

29.1.6 Extranet [OS-L3SA]

(1) Inter-VRF route filtering

Routes between VRFs can be filtered. Routes denied by the filter are not added to the routing table.

(a) Applying filters

The routes between VRFs are filtered by using `ipv6 import inter-vrf`.

The routes permitted by this filtering are added to the routing table. If there are no applicable filters or every filter gives a deny result, the route is not added.

The table below shows the configuration command used to filter the route between VRFs.

Table 29-19: Configuration command used for inter-VRF route filtering

Command name	Filtered routes
ipv6 import inter-vrf	Routes from a VRF specified as `route-map` are filtered.

(b) Route attributes changeable by inter-VRF route filtering

The following table describes the changeable attributes of routes that have been imported from another VRF or the global network.

Table 29-20: Changeable attributes used for inter-VRF route filtering

Attribute	Default
Distance	210
Tag value	Inherits the value of the route in the routing table.
AS_PATH attribute	

(c) Configuring inter-VRF routes

Apply an inter-VRF route filter. Routes imported from another VRF or the global network are added to the local VRF routing table according to the filter conditions. For an imported route, the VRF ID in the source routing table is also used in the destination routing table. Note that the protocol type of imported routes becomes `extra-vrf`.

If the `match vrf` configuration command is specified in the inter-VRF route filter, the VRF ID is compared against the source routing table. If you do not specify the `match vrf` command, the same filter conditions are applied to all other VRFs and global networks.

(d) Advertising inter-VRF routes by using protocols

If an advertised route filter is specified for a protocol, routes are advertised from the routing table of the VRF where the protocol is running. To use routes imported from another VRF or the global network, specify `extra-vrf` as the protocol in the `redistribute` configuration command.

29.2 Configuration

29.2.1 List of configuration commands

The following table describes the configuration commands for route filtering.

Table 29-21: List of configuration commands

Command name	Description
distribute-list in (BGP4+)	Filters which BGP4+-learned routes are added to the routing table.
distribute-list in (OSPFv3)	Filters which OSPFv3-learned routes are added to the routing table.
distribute-list in (RIPng)	Filters which RIPng-learned routes are added to the routing table.
distribute-list out (BGP4+)	Filters which BGP4+ routes are advertised.
distribute-list out (OSPFv3)	Filters which OSPFv3 routes are advertised.
distribute-list out (RIPng)	Filters which RIPng routes are advertised.
ip as-path access-list	Configures an access list to filter routes based on their AS_PATH attribute.
ip community-list	Configures a community list to filter routes based on their COMMUNITIES attribute.
ipv6 prefix-list	Configures an IPv6 prefix list.
match as-path	Configures route-map to use the AS_PATH attribute as filter conditions.
match community	Configures route-map to use the COMMUNITIES attribute as filter conditions.
match interface	Configures route-map to use the interface of the route as filter conditions.
match ipv6 address	Configures route-map to use the IPv6 address prefix as filter conditions.
match ipv6 route-source	Configures route-map to use the source IPv6 address as filter conditions.
match origin	Configures route-map to use the ORIGIN attribute as filter conditions.
match protocol	Configures route-map to use the routing protocol as filter conditions.
match route-type	Configures route-map to use the route type as filter conditions.
match tag	Configures route-map to use the tag value as filter conditions.
match vrf	Configures route-map to use a VRF as filter conditions.
neighbor in (BGP4+)	Specifies the filter to be used for BGP4+ learned route filtering.
neighbor out (BGP4+)	Sets the filters used for BGP4+ advertised route filtering.
redistribute (BGP4+)	Sets the protocol types of routes advertised from BGP4+.
redistribute (OSPFv3)	Sets the protocol types of routes advertised from OSPFv3.
redistribute (RIPng)	Sets the protocol types of routes advertised from RIPng.
route-map	Configures route-map.
set as-path prepend count	Sets the number of AS_PATH numbers added to the routing information.
set community	Replaces the COMMUNITIES attribute of the route.
set community-delete	Deletes the COMMUNITIES attribute of the route.

Command name	Description
set distance	Sets the priority of the routing information.
set local-preference	Sets the `LOCAL_PREF` attribute of the routing information.
set metric	Assigns a metric to the routing information.
set metric-type	Sets the metric type or metric of the routing information.
set origin	Sets the `ORIGIN` attribute of the routing information.
set tag	Sets the tag value of the routing information.
deny (ipv6 access-list)#1	Specifies the conditions by which the IPv6 filter denies access.
ipv6 access-list#1	Configures an access list to serve as an IPv6 filter.
ipv6 access-list resequence#1	Re-sequences the sequence numbers that determine the order in which the IPv6 filter applies filter conditions.
permit (ipv6 access-list)#1	Specifies the conditions by which the IPv6 filter permits access.
router bgp#2	Configures router settings related to the BGP (BGP4 and BGP4+) routing protocol.
ipv6 router rip#3	Configures router settings related to the RIPng routing protocol.
ipv6 router ospf#4	Configures router settings related to the OSPFv3 routing protocol.
ipv6 import inter-vrf#5	Controls routes that are imported from another VRF or the global network according to the filter.

#1

See *19. Access Lists* in the manual *Configuration Command Reference Vol. 1 For Version 11.10*.

#2

See *13. BGP4 [OS-L3SA]* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

#3

See *25. RIPng* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

#4

See *26. OSPFv3 [OS-L3SA]* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

#5

See *30. VRF [OS-L3SA]* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

29.2.2 RIPng learned route filtering

(1) Learning routes to specific destination networks

Configure RIPng to learn RIPng routes destined for only 3ffe:501:811:ff01::/64, but disregard RIP routes destined for all other networks.

Points to note

To apply learned route filtering, configure the `distribute-list in` command. To filter routes by their destination networks, use an `ipv6 prefix-list` filter.

First, configure an `ipv6 prefix-list` filter to permit routes to the 3ffe:501:811:ff01::/64 address range. When this filter is referenced from the `distribute-list in` command, the switch filters learned RIPng routes according to their destination network.

Command examples

1.
```
(config)# ipv6 prefix-list ONLY0811ff01 seq 10 permit
3ffe:501:811:ff01::/64
```

Configures `prefix-list` to permit only routes in the 3ffe:501:811:ff01::/64 range. Because `ONLY0811ff01` has no other conditions, the filter denies routes with any other destination address or mask length.

2.
```
(config)# ipv6 router rip
(config-rtr-rip)# distribute-list prefix-list ONLY0811ff01 in
```

Applies `ONLY0811ff01` to routes learned by the RIPng protocol.

(2) Learning routes from specific interfaces and to specific destination networks

Configure RIPng to learn only those RIPng routes from VLAN 10 that have 3ffe:501:811:ff01::/64 as their destination. Routes learned from interfaces other than VLAN 10 will not be filtered.

Points to note

To apply RIPng learned route filtering on a per-interface basis, specify the <*Interface*> in the parameter of the `distribute-list in` command. First, configure an `ipv6 prefix-list` filter to permit routes to the 3ffe:501:811:ff01::/64 address range. When this filter is referenced from the distribute-list in VLAN 10 command, the switch filters RIPng routes learned from VLAN 10 according to their destination network.

Command examples

1.
```
(config)# ipv6 prefix-list ONLY0811ff01 seq 10 permit
3ffe:501:811:ff01::/64
```

Configures `prefix-list` to permit only routes in the 3ffe:501:811:ff01::/64 range. Because `ONLY0811ff01` has no other conditions, the filter denies routes with any other destination address or mask length.

2.
```
(config)# ipv6 router rip
(config-rtr-rip)# distribute-list prefix-list ONLY0811ff01 in
vlan 10
```

Applies `ONLY0811ff01` to routes learned from VLAN 10.

(3) Filtering learned routes by a combination of tag value and destination network

Configure RIPng not to learn routes that have a destination address in the 3ffe:501::/32 range and a tag value other than 15. All other RIPng routes will be learned.

Points to note

The example below shows how to use `route-map` to filter a route by an attribute other than its destination network, or to modify some of its attributes. You can reference this `route-map` from the `distribute-list in` command.

First, configure a `prefix-list` filter to permit prefixes within the 3ffe:501::/32 range. Next, configure `route-map` to deny any routes that are permitted by this `prefix-list` filter and also have a tag value other than 15.

Finally, by referencing this `route-map` from the `distribute-list in` command, you can configure RIPng learned route filtering based on a combination of tag value and destination network.

Command examples

1.
```
(config)# ipv6 prefix-list LONGER3ffe0501 seq 10 permit
3ffe:501::/32 ge 32 le 128
```
Configures `prefix-list` to permit only routes in the 3ffe:501::/32 range.

2.
```
(config)# route-map TAG permit 10
(config-route-map)# match ipv6 address prefix-list
LONGER3ffe0501
(config-route-map)# match tag 15
(config-route-map)# exit
```
Configures `route-map` to permit routes in the 3ffe:501::/32 range that have a tag value of 15.

3.
```
(config)# route-map TAG deny 20
(config-route-map)# match ipv6 address prefix-list
LONGER3ffe0501
(config-route-map)# exit
```
Configures `route-map` to deny routes in the 3ffe:501::/32 range if they do not match the conditions associated with sequence number 10.

4.
```
(config)# route-map TAG permit 30
(config-route-map)# exit
```
Configures `route-map` to permit any route that does not match the conditions associated with sequence numbers 10 and 20.

5.
```
(config)# ipv6 router rip
(config-rtr-rip)# distribute-list route-map TAG in
```
Applies the above filter to RIPng learned route filtering. This means that RIPng does not learn routes that are in the 3ffe:501::/32 range and have a tag value other than 15.

(4) Changing distances based on destination networks

Assign a distance 50 to RIPng-learned routes whose destination network is in the 3ffe:501::/32 range to give such routes priority over OSPFv3 routes.

Points to note

First, configure a `prefix-list` filter to permit routes in the 3ffe:501::/32 range. Next, configure `route-map` to assign a distance of 50 to routes permitted by the `prefix-list` filter.

Finally, by referencing `route-map` from the `distribute-list in` command, configure RIPng learned route filtering to change a route's distance based on its destination network.

Command examples

1.
```
(config)# ipv6 prefix-list LONGER3ffe0501 seq 10 permit
3ffe:501::/32 ge 32 le 128
```

Configures `prefix-list` to permit only routes in the 3ffe:501::/32 range.

2.
```
(config)# route-map Distance50 permit 10
(config-route-map)# match ipv6 address prefix-list
LONGER3ffe0501
(config-route-map)# set distance 50
(config-route-map)# exit
```

Configures `route-map` to assign a distance of 50 to routes in the 3ffe:501::/32 range, and permits those routes.

3.
```
(config)# route-map Distance50 permit 20
(config-route-map)# exit
```

Configures the `route-map` to permit routes that do not match the conditions associated with sequence number 10, without changing any of their attributes.

4.
```
(config)# ipv6 router rip
(config-rtr-rip)# distribute-list route-map Distance50 in
```

Applies the above filter to RIPng learned route filtering. This means that a distance of 50 is assigned to RIPng-learned routes in the 3ffe:501::/32 range.

29.2.3 RIPng advertised route filtering

(1) Advertising routes associated with specific protocols

Configure the advertisement of static routes and OSPFv3 domain 1 routes by RIPng.

Points to note

The example below shows how to use the `redistribute` command to advertise routes that would not be advertised by default. In the `redistribute` command, specify the protocols that you want advertised.

When configuring advertisement of OSPFv3 routes, you must also specify a metric. OSPFv3 and BGP4+ routes cannot be advertised without a metric.

Command examples

1.
```
(config)# ipv6 router rip
(config-rtr-rip)# redistribute static
```

Advertises static routes into RIPng.

2.
```
(config-rtr-rip)# redistribute ospf 1 metric 2
```

Advertises OSPFv3 domain 1 routes, assigning them a metric of 2.

(2) Advertising routes by specific protocols to specific destination networks

Configure RIPng to advertise static routes, and only those OSPFv3 routes that have a destination network in the 3ffe:501:811:ff01::/64 range.

Points to note

To filter advertised routes based on their learning source protocol, specify `route-map` in the `redistribute` command. Use an `ipv6 prefix-list` filter to supply the destination network conditions for `route-map`.

First, configure an `ipv6 prefix-list` filter to permit only routes in the 3ffe:501:811:ff01::/64 range. Next, configure `route-map` to use this filter as its conditions. Finally, use `redistribute` commands to specify static routes and OSPFv3 routes. In the `redistribute` command for the OSPFv3 routes, specify the `route-map` that you configured.

Command examples

1.
```
(config)# ipv6 prefix-list ONLY0811ff01 seq 10 permit
3ffe:501:811:ff01::/64
```

Configures `prefix-list` to permit only routes in the 3ffe:501:811:ff01::/64 range. Because `ONLY0811ff01` has no other conditions, the filter denies routes with any other destination address or mask length.

2.
```
(config)# route-map ONLY0811ff01 permit 10
(config-route-map)# match ipv6 address prefix-list
ONLY0811ff01
(config-route-map)# exit
```

Configures `route-map` to permit routes whose destination network is in the 3ffe:501:811:ff01::/64 range.

3.
```
(config)# ipv6 router rip
(config-rtr-rip)# redistribute static
```

Configures RIPng to advertise static routes.

4.
```
(config-rtr-rip)# redistribute ospf 1 metric 2 route-map
ONLY0811ff01
```

Configures RIPng to filter OSPFv3 domain 1 routes by `ONLY0811ff01` and advertise permitted routes, assigning them a metric of 2.

(3) Suppressing advertisement of routes to specific destination networks

You can prevent RIPng from advertising routes destined for the 3ffe:501:811:ff01::/64 address range.

Points to note

The example below shows how to use the `distribute-list out` command to filter advertised routes regardless of their learning source protocol.

First, configure an `ipv6 prefix-list` filter to deny routes to the 3ffe:501:811:ff01::/64 address range. By referencing this filter from the `distribute-list out` command, you can configure RIPng to filter learned routes according to their destination network.

Command examples

1. `(config)# ipv6 prefix-list OMIT0811ff01 seq 10 deny 3ffe:501:811:ff01::/64`

   Configures `prefix-list` to deny routes in the 3ffe:501:811:ff01::/64 range.

2. `(config)# ipv6 prefix-list OMIT0811ff01 seq 100 permit ::/0 ge 0 le 128`

   Configures `ipv6 prefix-list` to permit routes with any destination address and mask length. Because `OMIT0811ff01` has no other conditions, the filter denies routes to 3ffe:501:811:ff01::/64 only.

3.
   ```
   (config)# ipv6 router rip
   (config-rtr-rip)# distribute-list prefix-list OMIT0811ff01 out
   ```

   Configures RIPng to apply the `OMIT0811ff01` filter to every route it advertises.

(4) Filtering advertised routes to individual destination interfaces

Configure the switch to use RIPng interface VLAN 10 to advertise routes to 3ffe:501:811:ff01::/64, and RIPng interface VLAN 20 to advertise all other routes. In this scenario, no interface-level filtering is applied to the other RIPng interfaces.

Points to note

To apply route filtering at the level of individual RIPng interfaces, specify the <*Interface*> in the parameters of the `distribute-list out` command.

First, configure an `ipv6 prefix-list` filter to permit routes in the 3ffe:501:811:ff01::/64 range and another to permit any route not in the 3ffe:501:811:ff01::/64 range. Next, specify the `distribute-list out` <*Interface*> command for RIPng interfaces VLAN 10 and VLAN 20. In the `distribute-list out` <*Interface*> command, specify the `prefix-list` filter appropriate to that RIPng interface.

Command examples

1. `(config)# ipv6 prefix-list ONLY0811ff01 seq 10 permit 3ffe:501:811:ff01::/64`

   Configures `prefix-list` to permit only routes in the 3ffe:501:811:ff01::/64 range. Because `ONLY0811ff01` has no other conditions, the filter denies routes with any other destination address or mask length.

2. `(config)# ipv6 prefix-list OMIT0811ff01 seq 10 deny 3ffe:501:811:ff01::/64`

   Configures `prefix-list` to deny routes in the 3ffe:501:811:ff01::/64 range.

3. `(config)# ipv6 prefix-list OMIT0811ff01 seq 100 permit ::/0 ge 0 le 128`

   Configures `prefix-list` to permit routes with any destination address and mask length. Because `OMIT0811ff01` has no other conditions, the filter denies routes to 3ffe:501:811:ff01::/64 only.

4.
   ```
   (config)# ipv6 router rip
   (config-rtr-rip)# distribute-list prefix-list ONLY0811ff01 out
   vlan 10
   ```

   Configures RIPng to apply criterion `ONLY0811ff01` to routes advertised from VLAN 10.

5.
   ```
   (config-rtr-rip)# distribute-list prefix-list OMIT0811ff01 out
   vlan 20
   ```

   Configures RIPng to apply criterion `OMIT0811ff01` to routes advertised from VLAN 20.

(5) Controlling route advertisement based on tag values

Configure the switch to assign a tag value of 210 to any directly connected routes it advertises and advertise static routes only if they have a tag value of 211. You then configure the switch to not advertise routes that have a tag value of 210 or 211 by RIPng. This process prevents RIPng-advertised routes from looping through the Switch.

Points to note

The example below shows how to use `route-map` to filter a route by an attribute other than its destination network, or to change a route attribute other than the metric. You can reference this `route-map` from `redistribute` and `distribute-list out` among other commands.

The commands below configure `route-map` to set the tag value of directly connected routes to 210, `route-map` to permit static routes with a tag value of 211, and `route-map` to deny RIPng routes with tag values of 210 or 211.

Command examples

1.
   ```
   (config)# route-map ConnectedToRIPng permit 10
   (config-route-map)# set tag 210
   (config-route-map)# exit
   ```

   Configures `route-map` to assign a tag value of 210.

2.
   ```
   (config)# route-map StaticToRIPng permit 10
   (config-route-map)# match tag 211
   (config-route-map)# exit
   ```

   Configures `route-map` to permit routes with a tag value of 211.

3.
   ```
   (config)# route-map RIPngToRIPng deny 10
   (config-route-map)# match tag 210 211
   (config-route-map)# exit
   (config)# route-map RIPngToRIPng permit 20
   (config-route-map)# exit
   ```

   Configures `route-map` to deny routes with a tag value of 210 or 211 while permitting all others.

4.
   ```
   (config)# ipv6 router rip
   ```

```
(config-rtr-rip)# redistribute connected route-map
ConnectedToRIPng
```

Advertises direct routes into RIPng. Specify `ConnectedToRIPng` as the advertising conditions.

5. `(config-rtr-rip)# redistribute static route-map StaticToRIPng`

   Advertises static routes into RIPng. Specify `StaticToRIPng` as the advertising conditions.

6. `(config-rtr-rip)# redistribute rip route-map RIPngToRIPng`

   Advertises RIPng routes to RIPng. Specify `RIPngToRIPng` as the advertising conditions.

29.2.4 OSPFv3 learned route filtering [OS-L3SA]

(1) Learning routes to specific destination networks

Configure OSPFv3 to only learn routes to addresses in the 3ffe:501:811:ff01::/64 range.

Points to note

To apply learned route filtering, configure the `distribute-list in` command. To filter routes by their destination networks, use an `ipv6 prefix-list` filter.

First, configure an `ipv6 prefix-list` filter to permit routes to the 3ffe:501:811:ff01::/64 address range. By referencing this filter from the `distribute-list in` command, you can configure OSPFv3 to filter learned routes based on the destination networks.

Command examples

1.
```
(config)# ipv6 prefix-list ONLY0811ff01 seq 10 permit
3ffe:501:811:ff01::/64
```

   Configures `prefix-list` to permit only routes in the 3ffe:501:811:ff01::/64 range. Because `ONLY0811ff01` has no other conditions, the filter denies routes with any other destination address or mask length.

2.
```
(config)# ipv6 router ospf 1
(config-rtr)# distribute-list prefix-list ONLY0811ff01 in
```

   Configures OSPFv3 to apply the `ONLY0811ff01` filter to the OSPFv3 external AS routes it learns.

(2) Filtering learned routes by tag value

Configure the switch not to learn routes with a tag value of 15. Other routes are learned.

Points to note

The example below shows how to use `route-map` to filter a route by an attribute other than its destination network, or to modify some of its attributes. You can reference this `route-map` from the `distribute-list in` command.

First, configure `route-map` to deny routes with a tag value of 15. Next, configure OSPFv3 learned route filtering by tag value by referencing this `route-map` from the `distribute-list in` command.

Command examples

1.
   ```
   (config)# route-map TAG15DENY deny 10
   (config-route-map)# match tag 15
   (config-route-map)# exit
   ```
   Configures `route-map` to deny routes with a tag value of 15.

2.
   ```
   (config)# route-map TAG15DENY permit 20
   (config-route-map)# exit
   ```
   Configures `route-map` to permit routes that do not match the conditions associated with sequence number 10.

3.
   ```
   (config)# ipv6 router ospf 1
   (config-rtr)# distribute-list route-map TAG15DENY in
   ```
   Applies the filter to OSPFv3 learned route filtering. This configures OSPFv3 to not learn external AS routes that have a tag value of 15.

(3) Changing distances based on destination networks

Configure OSPFv3 to assign a distance of 150 to external AS routes whose destination network is in the 3ffe:501::/32 range, thereby giving priority to RIPng routes.

Points to note

The example below shows how to use `route-map` to filter a route by an attribute other than its destination network, or to modify some of its attributes. You can reference this `route-map` from the `distribute-list in` command.

First, configure a `prefix-list` filter to permit routes in the 3ffe:501::/32 range. Next, configure `route-map` to assign a distance of 150 to routes permitted by this `prefix-list` filter.

Finally, by referencing this `route-map` from the `distribute-list in` command, configure the switch to change distances based on destination networks when performing OSPFv3 learned route filtering.

Command examples

1.
   ```
   (config)# ipv6 prefix-list LONGER3ffe0501 seq 10 permit
   3ffe:501::/32 ge 32 le 128
   ```
   Configures `prefix-list` to permit only routes in the 3ffe:501::/32 range.

2.
   ```
   (config)# route-map Distance150 permit 10
   (config-route-map)# match ipv6 address prefix-list
   LONGER3ffe0501
   (config-route-map)# set distance 150
   (config-route-map)# exit
   ```
   Configures `route-map` to assign a distance of 150 to routes in the 3ffe:501::/32 range, and permits those routes.

3.
   ```
   (config)# route-map Distance150 permit 20
   ```

```
(config-route-map)# exit
```

Configures the route-map to permit routes that do not match the conditions associated with sequence number 10, without changing any of their attributes.

4.

```
(config)# ipv6 router ospf 1
(config-rtr)# distribute-list route-map Distance150 in
```

Applies the above filter to OSPFv3 learned route filtering. This configures OSPFv3 to assign a distance of 150 to external AS routes in the 3ffe:501::/32 range.

29.2.5 OSPFv3 advertised route filtering [OS-L3SA]

(1) Advertising routes associated with specific protocols

Configure OSPFv3 to advertise static routes and RIPng routes to OSPFv3 domain 1.

Points to note

The example below shows how to use the redistribute command to advertise routes that would not be advertised by default. In the redistribute command, specify the protocols that you want advertised.

Command examples

1.

```
(config)# ipv6 router ospf 1
(config-rtr)# redistribute static
```

Configures OSPFv3 to advertise static routes.

2.

```
(config-rtr)# redistribute rip
```

Configures OSPFv3 to advertise RIPng routes.

(2) Advertising routes by specific protocols to specific destination networks

You can configure the switch to advertise static routes, and RIPng routes whose destination network is 3ffe:501:811:ff01::/64, into OSPFv3 domain 1.

Points to note

To filter advertised routes based on their learning source protocol, specify route-map in the redistribute command. Configure an ipv6 prefix-list filter to supply the destination network conditions for the route-map, and then call the list from the match ipv6 address command.

First, configure an ipv6 prefix-list filter to permit only routes in the 3ffe:501:811:ff01::/64 range. Next, configure route-map to use this filter as filter conditions. Finally, use the redistribute command to configure the switch to advertise static routes and RIPng routes. In the redistribute command for the RIPng routes, specify the route-map that you configured.

Command examples

1.

```
(config)# ipv6 prefix-list ONLY0811ff01 seq 10 permit
3ffe:501:811:ff01::/64
```

Configures prefix-list to permit only routes in the 3ffe:501:811:ff01::/64 range. Because ONLY0811ff01 has no other conditions, the filter denies routes with any other destination address or mask length.

2.
```
(config)# route-map ONLY0811ff01 permit 10
(config-route-map)# match ipv6 address prefix-list ONLY0811ff01
(config-route-map)# exit
```

Configures `route-map` to permit routes whose destination network is in the 3ffe:501:811:ff01::/64 range.

3.
```
(config)# ipv6 router ospf 1
(config-rtr)# redistribute static
```

Configures OSPFv3 to advertise static routes into OSPFv3 domain 1.

4.
```
(config-rtr)# redistribute rip route-map ONLY0811ff01
```

Configures OSPFv3 to apply the `ONLY0811ff01` filter to RIPng routes and only advertise those routes permitted by the filter.

(3) Suppressing advertisement of routes to specific destination networks

Configure OSPFv3 to advertise static routes and RIPng routes to OSPFv3 domain 1, except for routes destined for the 3ffe:501:811:ff01::/64 address range.

Points to note

The example below shows how to use the `distribute-list out` command to filter advertised routes regardless of their learning source protocol.

First, configure an `ipv6 prefix-list` filter to deny routes to the 3ffe:501:811:ff01::/64 address range. By referencing this filter from the `distribute-list out` command, you can configure the switch to perform advertised route filtering by destination network.

Finally, use the `redistribute` command to configure the switch to advertise static routes and RIPng routes.

Command examples

1.
```
(config)# ipv6 prefix-list OMIT0811ff01 seq 10 deny 3ffe:501:811:ff01::/64
```

Configures `prefix-list` to deny routes in the 3ffe:501:811:ff01::/64 range.

2.
```
(config)# ipv6 prefix-list OMIT0811ff01 seq 100 permit ::/0 ge 0 le 128
```

Configures `prefix-list` to permit routes with any destination address and mask length. Because `OMIT0811ff01` has no other conditions, the filter denies routes to 3ffe:501:811:ff01::/64 only.

3.
```
(config)# ipv6 router ospf 1
(config-rtr)# distribute-list prefix-list OMIT0811ff01 out
```

Configures OSPFv3 to apply the `OMIT0811ff01` filter to advertised routes.

4.
```
(config-rtr)# redistribute static
(config-rtr)# redistribute rip
```
Configures OSPFv3 to advertise static routes and RIPng routes.

(4) Advertising routes between OSPFv3 domains

The procedure below configures the reciprocal exchange of routes between OSPFv3 domain 1 and OSPFv3 domain 2.

Routes associated with OSPFv3 domain 1 are tagged with a value of 1001 and advertised to OSPFv3 domain 2. In turn, the domain 2 does not advertise routes that have a tag value of 1001 to the domain 1. This process prevents routing loops by stopping OSPFv3 domain 2 from advertising OSPFv3 domain 1 routes back to OSPFv3 domain 1.

Similarly, routes associated with OSPFv3 domain 2 are tagged with a value of 1002 and advertised to OSPFv3 domain 1. In turn, the domain 1 does not advertise routes that have a tag value of 1002 to the domain 2.

Points to note

The example below shows how to use `route-map` to filter a route by an attribute other than its destination network, or to change a route attribute other than the metric. You can reference this `route-map` from `redistribute` and `distribute-list out` among other commands.

Configure advertisement to OSPFv3 domain 1 by specifying `route-map` to deny routes with a tag value of 1001 and permits the advertisement of all other routes after assigning them a tag value of 1002. Then, specify this filter in the `redistribute` command that configures the advertisement of the domain 1 routes to the domain 2.

Similarly, configure advertisement to OSPFv3 domain 2 by specifying `route-map` to deny routes with a tag value of 1002 but to permit the advertisement of all other routes after assigning them a tag value of 1001. Then, specify this filter in the `redistribute` command that configures the advertisement of the domain 2 routes to the domain 1.

Command examples

1.
```
(config)# route-map OSPF2to1 deny 10
(config-route-map)# match tag 1001
(config-route-map)# exit
```
Configures the `route-map` filter `OSPF2to1` to deny routes with a tag value of 1001.

2.
```
(config)# route-map OSPF2to1 permit 20
(config-route-map)# set tag 1002
(config-route-map)# exit
```
Configures `route-map` to assign a tag value of 1002 to routes that do not satisfy the above conditions.

3.
```
(config)# ipv6 router ospf 1
(config-rtr)# redistribute ospf 2 route-map OSPF2to1
(config-rtr)# exit
```
Configures OSPFv3 to advertise routes in OSPFv3 domain 2 to OSPFv3 domain 1. Specify

OSPF2to1 as a filter.

4.
```
(config)# route-map OSPF1to2 deny 10
(config-route-map)# match tag 1002
(config-route-map)# exit
(config)# route-map OSPF1to2 permit 20
(config-route-map)# set tag 1001
(config-route-map)# exit
```

   Configures the `route-map` filter `OSPF1to2` to deny routes with a tag value of 1002, and assign a tag value of 1001 to all other routes.

5.
```
(config)# ipv6 router ospf 2
(config-rtr)# redistribute ospf 1 route-map OSPF1to2
(config-rtr)# exit
```

   Configures OSPFv3 to advertise routes in OSPFv3 domain 1 to OSPFv3 domain 2. Specify `OSPF1to2` as a filter.

29.2.6 BGP4+ learned route filtering [OS-L3SA]

(1) Conditional route learning across all peers

Configure the switch to learn BGP4+ routes to any destination network except the routes in the 3ffe:501::/32 range.

Points to note

The example below shows how to use the `distribute-list in` command to apply learned route filtering consistently among all peers. To filter routes by destination network, use an `ipv6 prefix-list` filter.

First, configure an `ipv6 prefix-list` filter to deny routes in the 3ffe:501::/32 range. Then, by referencing this filter from the `distribute-list in` command, configure BGP4+ to filter learned routes by destination network.

Command examples

1.
```
(config)# ipv6 prefix-list LONGER3ffe0501DENY seq 10 deny
3ffe:501::/32 ge 32 le 128
(config)# ipv6 prefix-list LONGER3ffe0501DENY seq 20 permit ::/
0 ge 0 le 128
```

   Configures `prefix-list` to deny prefixes in the 3ffe:501::/32 range but permits all other prefixes.

2.
```
(config)# router bgp 65531
(config-router)# address-family ipv6
(config-router-af)# distribute-list prefix-list
LONGER3ffe0501DENY in
```

   Configures the switch to apply learned route filtering by the specified `prefix-list` filter to all peers.

3.
```
(config-router-af)# end
# clear ipv6 bgp * in
```
Applies the changes to the learned route filtering configuration.

(2) Conditional route learning for individual peers

The following shows how to configure BGP4+ to learn routes received from external peers that have an AS_PATH attribute of 65532 65533 and are not destined for the 3ffe:501::/32 address range. The switch assigns a value of 200 to the LOCAL_PREF attribute of learned routes. Other routes are not learned.

Points to note

The example below shows how to use the neighbor in command to apply learned route filtering to routes learned from individual BGP4+ peers. Use route-map to filter a route by conditions other than its destination network, or to modify some of its attributes.

First, configure a prefix-list filter to permit routes in the 3ffe:501::/32 range, and an ip as-path access-list filter to permit routes with an AS_PATH attribute of 65532 65533. Next, configure route-map to combine these two conditions. Finally, set the neighbor in command for peers that you want to filter by these conditions.

Command examples

1.
```
(config)# ipv6 prefix-list LONGER3ffe0501 seq 10 permit
3ffe:501::/32 ge 32 le 128
```
Configures prefix-list to permit routes whose prefix is within the 3ffe:501::/32 range.

2.
```
(config)# ip as-path access-list 2 permit "^65532_65533$"
```
Configures an ip as-path access-list filter to permit routes with an AS_PATH attribute of 65532 65533.

3.
```
(config)# route-map BGP65532IN deny 10
(config-route-map)# match ipv6 address prefix-list
LONGER3ffe0501
(config-route-map)# exit
```
Configures the route-map filter BGP65502IN to deny routes destined for the 3ffe:501::/32 address range.

4.
```
(config)# route-map BGP65532IN permit 20
(config-route-map)# match as-path 2
(config-route-map)# set local-preference 200
(config-route-map)# exit
```
Configures route-map to assign a value of 200 to the LOCAL_PREF attribute of routes whose AS_PATH attribute matches 65532 65533, and permit those routes. Because BGP65532IN has no other conditions, the filter denies routes that do not match any of the conditions set so far.

5.
```
(config)# router bgp 65531
(config-router)# neighbor 3ffe:502:811:1::1 remote-as 65532
(config-router)# address-family ipv6
(config-router-af)# neighbor 3ffe:502:811:1::1 route-map
BGP65532IN in
```

   Configures BGP4+ to use the `route-map` filter `BGP65532IN` to filter routes received from external peers.

6.
```
(config-router-af)# end
# clear ipv6 bgp * in
```

   Applies the changes to the learned route filtering configuration.

29.2.7 BGP4+ advertised route filtering [OS-L3SA]

(1) Advertising routes of other protocols

Among directly connected and static routes, you can configure BGP4+ to advertise only those routes whose destination network is in the local AS (3ffe:501::/32).

Points to note

The example below shows how to use the `redistribute` command to advertise routes that would not be advertised by default. In the `redistribute` command, specify the protocols that you want advertised.

To define conditions for route advertisement, specify `route-map` in the `redistribute` command. Use a `prefix-list` filter to supply the destination network conditions for the `route-map`.

Command examples

1.
```
(config)# ipv6 prefix-list LONGER3ffe0501 seq 10 permit
3ffe:501::/32 ge 32 le 128
```

   Configures `prefix-list` to permit only routes in the 3ffe:501::/32 range.

2.
```
(config)# route-map LONGER3ffe0501PERMIT permit 10
(config-route-map)# match ipv6 address prefix-list
LONGER3ffe0501
(config-route-map)# exit
```

   Configures `route-map` to permit routes in the 3ffe:501::/32 range.

3.
```
(config)# router bgp 65531
(config-router)# address-family ipv6
(config-router-af)# redistribute connected route-map
LONGER3ffe0501PERMIT
(config-router-af)# redistribute static route-map
LONGER3ffe0501PERMIT
```

Configures the `redistribute` function to only advertise static and directly connected routes permitted by the `route-map` filter `LONGER3ffe0501PERMIT`.

4.
```
(config-router-af)# end
# clear ipv6 bgp * out
```

Applies the changes to the advertised route filtering configuration.

(2) Changing advertised routes for individual peers

You can restrict which routes are advertised to external peers. This example restricts route advertisement to BGP4+ routes received from AS100 that have one AS path, and directly connected routes and static routes destined for networks in the local AS (3ffe:501::/32). When advertising routes, the switch adds two AS numbers to the `AS_PATH` of peer 3ffe:502:812:1::1. Only BGP4+ routes are advertised to internal peers.

Points to note

The example below shows how to use the `neighbor out` command if you need to apply route filtering to individual peers.

Here, configure a total of four `route-map` filters: one to redistribute static and directly connected routes, one for advertising to peer 3ffe:502:812:1::1, one for advertising to external peers other than 3ffe:502:812:1::1, and one for internal peers.

For static and directly connected routes, configure an `ipv6 prefix-list` filter to permit routes in the 3ffe:501::/32 range and a `route-map` filter from which to call the filter.

For peer 3ffe:502:812:1::1, configure a `route-map` filter to add two AS numbers to direct and static routes.

For external peers other than 3ffe:502:812:1::1, configure an `ip as-path access-list` filter to permit routes whose `AS_PATH` attribute contains one AS, and a `route-map` filter that references the `ip as-path access-list` filter.

For internal peers, configure `route-map` to permit BGP4+ routes and denies all others.

Command examples

1.
```
(config)# ipv6 prefix-list LONGER3ffe0501 seq 10 permit
3ffe:501::/32 ge 32 le 128
(config)# route-map LONGER3ffe0501PERMIT permit 10
(config-route-map)# match ipv6 address prefix-list
LONGER3ffe0501
(config-route-map)# exit
```

Configures `route-map` to permit routes in the 3ffe:501::/32 range. This filter is used to redistribute static and direct routes.

2.
```
(config)# ip as-path access-list 1 permit "^[0-9]+$"
(config)# route-map BGPEXTOUT permit 10
(config-route-map)# match protocol connected static
(config-route-map)# exit
(config)# route-map BGPEXTOUT permit 20
(config-route-map)# match protocol bgp
```

```
(config-route-map)# match as-path 1
(config-route-map)# exit
```

Configures route-map to permit direct routes, static routes, and only those BGP4+ routes whose AS_PATH attribute contains one AS. This route-map filter is used to filter advertisement to external peers.

3.

```
(config)# route-map BGP81211OUT permit 10
(config-route-map)# match protocol connected static
(config-route-map)# set as-path prepend count 2
(config-route-map)# exit
(config)# route-map BGP81211OUT permit 20
(config-route-map)# match protocol bgp
(config-route-map)# match as-path 1
(config-route-map)# set as-path prepend count 2
(config-route-map)# exit
```

Configures route-map to permit direct routes, static routes, and only those BGP4+ routes whose AS_PATH attribute contains one AS, and adds two AS numbers to those routes. This filter is used to filter advertisement to peer 3ffe:502:812:1::1.

4.

```
(config)# route-map BGPINTOUT permit 10
(config-route-map)# match protocol bgp
(config-route-map)# exit
```

Configures route-map to permit only BGP4+ routes. This filter is used to filter advertisement to internal peers.

5.

```
(config)# router bgp 65531
(config-router)# address-family ipv6
(config-router-af)# redistribute connected route-map
LONGER3ffe0501PERMIT
(config-router-af)# redistribute static route-map
LONGER3ffe0501PERMIT
(config-router-af)# exit
```

Configures the redistribute function to advertise only those static and directly connected routes permitted by the route-map filter LONGER3ffe0501PERMIT.

6.

```
(config-router)# neighbor 3ffe:502:811:1::1 remote-as 65532
(config-router)# address-family ipv6
(config-router-af)# neighbor 3ffe:502:811:1::1 route-map
BGPEXTOUT out
(config-router-af)# exit
```

Applies the filter BGPEXTOUT when advertising routes to external peers.

7.
```
(config-router)# neighbor 3ffe:502:812:1::1 remote-as 65533
(config-router)# address-family ipv6
(config-router-af)# neighbor 3ffe:502:812:1::1 route-map
BGP81211OUT out
(config-router-af)# exit
```

Applies the filter `GP81211OUT` when advertising routes to external peer 3ffe:502:812:1::1.

8.
```
(config-router)# neighbor 3ffe:501:811:ff01::1 remote-as 65531
(config-router)# address-family ipv6
(config-router-af)# neighbor 3ffe:501:811:ff01::1 route-map
BGPINTOUT out
```

Applies the filter `BGPINTOUT` when advertising routes to internal peers.

9.
```
(config-router-af)# end
# clear ipv6 bgp * out
```

Applies the changes to the advertised route filtering configuration.

29.2.8 Extranet [OS-L3SA]

To perform communication from a given VRF to a network in a different VRF, configure a route filter so that a specific route in the remote VRF is imported into the local VRF.

(1) Importing a specific VRF route

Configure route filters to allow communication between VRFs. To do this, a VRF 2 route (2001:db8:1:1::/64) is imported into VRF 3, and a VRF 3 route (2001:db8:1:3::/64) is imported to VRF 2.

Points to note

Use the `ipv6 import inter-vrf` command to perform inter-VRF route filtering VRF. Use `route-map` to filter routes by VRF ID. Use `prefix-list` to supply the destination network conditions for the `route-map`.

Configure `route-map` to permit only the VRF 2 route. This `route map` is referenced from `ipv6 import inter-vrf` of VRF 3. Next, configure `route-map` to permit only the VRF 3 route. This `route-map` is referenced from `ipv6 import inter-vrf` of VRF 2.

Command examples

1.
```
(config)# ipv6 prefix-list PERMITVRF2 seq 10 permit
2001:db8:1:1::/64
(config)# route-map VRF2PERMIT permit 10
(config-route-map)# match vrf 2
(config-route-map)# match ipv6 address prefix-list PERMITVRF2
(config-route-map)# exit
```

Configures `route-map` to permit the VRF 2 route.

2.
```
(config)# vrf definition 3
```

```
(config-vrf)# ipv6 import inter-vrf VRF2PERMIT
(config-vrf)# exit
```

Applies the filter settings in step 1 to the VRF 3 extranet so that the VRF 2 route is imported into VRF 3.

3.
```
(config)# ipv6 prefix-list PERMITVRF3 seq 10 permit
2001:db8:1:3::/64
(config)# route-map VRF3PERMIT permit 10
(config-route-map)# match vrf 3
(config-route-map)# match ipv6 address prefix-list PERMITVRF3
(config-route-map)# exit
```

Configures `route-map` to permit the VRF 3 route.

4.
```
(config)# vrf definition 2
(config-vrf)# ipv6 import inter-vrf VRF3PERMIT
(config-vrf)# exit
```

Applies the filter settings in step 3 to the VRF 2 extranet so that the VRF 3 route is imported into VRF 2.

Notes

If the `route-map` referenced by `ipv6 import inter-vrf` has not been configured, all routes in the other VRF or the global network are imported. To prevent unwanted routes from being imported, always configure `route-map` first, and then configure `ipv6 import inter-vrf`.

(2) Advertising routes between VRFs using a protocol

Implement the route (2001:db8:1:3::/64) for VRF3 into the network for VRF2. Use OSPFv3 for VRF2 to advertize the implemented route for VRF3.

Points to note

Use `ipv6 import inter-vrf` to perform inter-VRF route filtering. Use `route-map` to filter routes by VRF. Use `prefix-list` to supply the destination network conditions for the `route-map`. To advertise a route imported by OSPFv3 from another VRF or the global network, configure `redistribute`.

Configure `route-map` to permit only the VRF 3 route. Next, configure `ipv6 import inter-vrf` to reference the `route-map` so that the VRF 3 route is imported into VRF 2. Finally, configure `redistribute` for the OSPFv3 of VRF 2 to advertise the route imported from the other VRF or global network.

Command examples

1.
```
(config)# ipv6 prefix-list PERMITVRF3 seq 10 permit
2001:db8:1:3::/64
(config)# route-map VRF3TO2 permit 10
(config-route-map)# match vrf 3
(config-route-map)# match ipv6 address prefix-list PERMITVRF3
(config-route-map)# exit
```

Configures `route-map` to permit the VRF 3 route.

2.
```
(config)# vrf definition 2
(config-vrf)# ipv6 import inter-vrf VRF3TO2
(config-vrf)# exit
```

Applies the filter settings in step 1 to the VRF 2 extranet so that the VRF 3 route is imported into VRF 2.

3.
```
(config)# ipv6 router ospf 1 vrf 2
(config-rtr)# redistribute extra-vrf
```

Advertises the route imported from the other VRF or the global network in VRF 2 OSPFv3 domain 1.

Notes

If the `route-map` referenced by `ipv6 import inter-vrf` has not been configured, all routes in the other VRF or the global network are imported. To prevent unwanted routes from being imported, always configure a `route-map` first, and then configure `ipv6 import inter-vrf`.

(3) Changing the distance for a specific VRF

VRF 2 and VRF 3 routes are imported into the global network. For the VRF 2 route only, the distance is set to 150.

Points to note

Use `ipv6 import inter-vrf` to perform inter-VRF route filtering. Use `route-map` to filter routes by VRF.

First, configure a `route-map` to permit the VRF 2 route and changes its distance to `150`. Next, in this `route-map`, specify a setting to permit the VRF 3 route with a different sequence number.

Finally, configure a filter that changes the distance of a specific VRF by ensuring that `ipv6 import inter-vrf` references the `route-map`.

Command examples

1.
```
(config)# route-map VRF2AND3PERMIT permit 10
(config-route-map)# match vrf 2
(config-route-map)# set distance 150
(config-route-map)# exit
```

Configures `route-map` to permit the VRF 2 route and change its distance to `150`.

2.
```
(config)# route-map VRF2AND3PERMIT permit 20
(config-route-map)# match vrf 3
(config-route-map)# exit
```

Configures `route-map` to permit the VRF 3 route.

3.
```
(config)# vrf definition global
```

```
(config-vrf)# ipv6 import inter-vrf VRF2AND3PERMIT
```

Applies the filter settings in steps 1 and 2 to the extranet of the global network so that the VRF 2 and VRF 3 routes are imported into the global network, and the distance of the VRF 2 route is changed to `150`.

Notes

If the `route-map` referenced by `ipv6 import inter-vrf` has not been configured, all routes in the other VRF or the global network are imported. To prevent unwanted routes from being imported, always configure `route-map` first, and then configure `ipv6 import inter-vrf`.

29.3 Operation

The following table describes the operation commands for IPv6 route filtering.

Table 29-22: List of operation commands

Command name	Description
show ipv6 route	Shows the IPv6 unicast routes in the routing table.
show ipv6 rip	Shows information about the RIPng protocol.
show ipv6 ospf	Shows information related to the OSPFv3 protocol.
show ipv6 bgp	Shows information about the BGP4+ protocol.
clear ipv6 bgp	Clears BGP4+ sessions or BGP4+-related information, or filters inbound or outbound routes using new BGP filter information.
show ipv6 vrf	Shows the IPv6 information of a VRF.
restart unicast[#]	Restarts the unicast routing program.
dump protocols unicast[#]	Outputs the trace information and control table information collected by the unicast routing program to a file.
erase protocol-dump unicast[#]	Deletes the file of trace information and control table information generated by the unicast routing program.

[#] See *8. Routing Protocol Common to IPv4 and IPv6* in the manual *Operation Command Reference Vol. 2 For Version 11.10.*

29.3.1 Checking routes received by RIPng (prior to learned route filtering)

To check routes received by the RIPng, use the `show ipv6 rip` operation command with the `received-routes` parameter specified.

Figure 29-3: Example of displaying RIPng-received routes

```
> show ipv6 rip received-routes
Date 20XX/07/14 12:00:00 UTC
Status Codes: * valid, > active, r RIB failure

Neighbor Address: fe80::200:87ff:fe28:90d7%VLAN0007
   Destination                                Next Hop
         Interface      Metric   Tag   Timer
*> 3ffe:3b01:6705:1::/64                      fe80::200:87ff:fe28:90d7%VLAN0007
         VLAN0007       1        0      5s
```

Notes

The command output does not include routes that are excluded by learned route filtering or do not have priority under RIPng.

29.3.2 Checking OSPFv3 routes computed by SPF [OS-L3SA]

If a filter invalidates an external AS route computed by the SPF algorithm of the OSPFv3 protocol, the route is still entered into the routing table but as an invalid route. You can check the external AS routes that the OSPFv3 protocol has generated, including invalid routes, by executing the `show ipv6 route` operation command with the `all-routes` and `-T ospf external` parameters specified.

Figure 29-4: Example of displaying OSPFv3 external AS routes

```
> show ipv6 route all-routes -T ospf external
```

```
Date 20XX/07/14 12:00:00 UTC
Status Codes: * valid, > active, r RIB failure
Total: 2 routes
   Destination                                 Next Hop
         Interface     Metric  Protocol     Age
*> 3ffe:3b21:6705:1::/64                      fe80::200:87ff:fe28:90d7%VLAN0007
         VLAN0007      1/1     OSPFv3 ext2  24m 33s , Tag: 100
*  3ffe:8703:2005:1::/64                      fe80::200:87ff:fe28:90d7%VLAN0007
         VLAN0007      1/1     OSPFv3 ext2  26m 52s , Tag: 100
```

29.3.3 Checking routes received by BGP4+ (prior to learned route filtering) [OS-L3SA]

To check routes received by the BGP4+ protocol, use the `show ipv6 bgp` operation command with the `received-routes` parameter specified.

Figure 29-5: Example of displaying BGP4+-received routes

```
> show ipv6 bgp received-routes
Date 20XX/07/14 12:00:00 UTC
BGP4+ Peer: 3ffe:177:7:7::145, Remote AS: 1000
Local AS: 200, Local Router ID: 200.201.202.203
Status Codes: d dampened, * valid, > active, S Stale, r RIB failure
Origin Codes: i - IGP, e - EGP, ? - incomplete
   Network                                     Next Hop
         MED    LocalPref Path
*> 3ffe:3b11:6705:1::/64                      fe80::200:87ff:fe28:90d7%VLAN0007
         -      -         1000 i
*  3ffe:8703:2005:1::/64                      fe80::200:87ff:fe28:90d7%VLAN0007
         -      -         1000 i
```

Notes

The command output does not include routes that are excluded by learned route filtering or do not have priority under BGP4+.

To include detailed information about route attributes in the command output, use the `show ipv6 bgp` operation command with the `received-routes` and `-F` parameters specified. Use this method to check the `ORIGIN`, `AS_PATH`, `MED`, `LOCAL_PREF`, and `COMMUNITIES` attributes of the routes.

Figure 29-6: Example of displaying detailed information about BGP4+-received routes

```
> show ipv6 bgp received-routes  -F
Date 20XX/07/14 12:00:00 UTC
BGP4+ Peer: 3ffe:177:7:7::145, Remote AS: 1000
Local AS: 200, Local Router ID: 200.201.202.203
Status Codes: d dampened, * valid, > active, S Stale, r RIB failure
Route 3ffe:3b11:6705:1::/64
*> Next Hop fe80::200:87ff:fe28:90d7%VLAN0007
     MED: -, LocalPref: -, Type: External route
     Origin: IGP, IGP Metric: 0
     Path: 1000
     Next Hop Attribute: 3ffe:177:7:7::145
                         fe80::200:87ff:fe28:90d7

Route 3ffe:8703:2005:1::/64
*  Next Hop fe80::200:87ff:fe28:90d7%VLAN0007
     MED: -, LocalPref: -, Type: External route
     Origin: IGP, IGP Metric: 0
     Path: 1000
     Next Hop Attribute: 3ffe:177:7:7::145
                         fe80::200:87ff:fe28:90d7
     Communities: 300:300
```

Notes

The command output does not include routes that are excluded by learned route filtering or do not have priority under BGP4+.

29.3.4 Checking routes resulting from route filtering

Routes permitted by learned route filtering are entered into the routing table. You can check the results of learned route filtering by viewing the routes in the routing table.

To display every route in the routing table including invalid routes, execute the `show ipv6 route` operation command with the `all-routes` parameter specified.

Figure 29-7: Example of displaying routes in routing table (including invalid routes)

```
> show ipv6 route all-routes
Date 20XX/07/14 12:00:00 UTC
Status Codes: * valid, > active, r RIB failure
Total: 11 routes
   Destination                              Next Hop
         Interface   Metric  Protocol    Age                              _
*> ::1/128                                  ::1                            |
         localhost   0/0     Connected    4h 44m                           |
*> 3ffe:177:7:7::/64                        3ffe:177:7:7::1                |
         VLAN0007    0/0     Connected   39m 41s                           |
*  3ffe:177:7:7::/64                        3ffe:177:7:7::1                |
         VLAN0007    1/-     OSPFv3 intra 6m 52s                           |
*> 3ffe:177:7:7::1/128                      ::1                            |
         localhost   0/0     Connected   39m 41s                           |
*> 3ffe:3b01:6705:1::/64                 fe80::200:87ff:fe28:90d7%VLAN0007 |
         VLAN0007    2/0     RIPng        2s                               |
*> 3ffe:3b11:6705:1::/64                 fe80::200:87ff:fe28:90d7%VLAN0007 #
         VLAN0007    -/-     BGP4+        4m  5s                           |
*> 3ffe:3b21:6705:1::/64                 fe80::200:87ff:fe28:90d7%VLAN0007 |
         VLAN0007    1/1     OSPFv3 ext2  4m  3s                           |
*> 3ffe:8703:2005:1::/64                    3ffe:177:7:7::145              |
         VLAN0007    0/0     Static       1m 15s                           |
*  3ffe:8703:2005:1::/64                 fe80::200:87ff:fe28:90d7%VLAN0007 |
         VLAN0007    -/-     BGP4+       8m 27s                            |
*  3ffe:8703:2005:1::/64                 fe80::200:87ff:fe28:90d7%VLAN0007 |
         VLAN0007    1/1     OSPFv3 ext2  6m 22s                           |
*  3ffe:8703:2005:1::/64                 fe80::200:87ff:fe28:90d7%VLAN0007 |
         VLAN0007    2/0     RIPng        2s                              _|
```

#

The characters `*` and `>` at the beginning of an entry signify the following:

*: Signifies a valid route. Its absence indicates an invalid route.

>: Signifies a prioritized route. Only prioritized routes are used for packet transfer.

To check only those routes that were learned by a specific protocol, execute the `show ipv6 route` operation command with the `all-routes` parameter and the protocol name specified.

Figure 29-8: Example of displaying routes in the routing table (RIPng only, includes invalid routes)

```
> show ipv6 route all-routes rip
Date 20XX/07/14 12:00:00 UTC
Status Codes: * valid, > active, r RIB failure
Total: 2 routes
   Destination                                 Next Hop
         Interface      Metric   Protocol  Age
*> 3ffe:3b01:6705:1::/64                      fe80::200:87ff:fe28:90d7%VLAN0007
         VLAN0007       2/0      RIPng      3s
```

```
*  3ffe:8703:2005:1::/64                          fe80::200:87ff:fe28:90d7%VLAN0007
         VLAN0007       2/0      RIPng      3s
```

If the various protocols generate more than one route to the same destination network, you will need to check which protocol supplied the prioritized route, and the priority of each route. The priority of each route is determined by its distance.

To display distances, execute the `show ipv6 route` operation command with the `all-routes` and `-P` parameters specified. The distance is the first value in the `Distance` column at the end of each line.

Figure 29-9: Example of displaying distances in the routing table

```
> show ipv6 route all-routes -P
Date 20XX/07/14 12:00:00 UTC
Status Codes: * valid, > active, r RIB failure
Total: 11 routes
   Destination                              Next Hop
         Interface  Metric  Protocol    Age
*> ::1/128                                  ::1
         localhost  0/0     Connected    4h 46m , Distance: 0/0/0
*> 3ffe:177:7:7::/64                        3ffe:177:7:7::1
         VLAN0007   0/0     Connected   42m  0s , Distance: 0/0/0
*  3ffe:177:7:7::/64                        3ffe:177:7:7::1
         VLAN0007   1/-     OSPFv3 intra 9m 11s , Distance: 110/1/0
*> 3ffe:177:7:7::1/128                      ::1
         localhost  0/0     Connected   42m  0s , Distance: 0/0/0
*> 3ffe:3b01:6705:1::/64                    fe80::200:87ff:fe28:90d7%VLAN0007
         VLAN0007   2/0     RIPng       16s     , Distance: 120/0/0
*> 3ffe:3b11:6705:1::/64                    fe80::200:87ff:fe28:90d7%VLAN0007
         VLAN0007   -/-     BGP4+        6m 24s , Distance: 20/0/0
*> 3ffe:3b21:6705:1::/64                    fe80::200:87ff:fe28:90d7%VLAN0007
         VLAN0007   1/1     OSPFv3 ext2  6m 22s , Distance: 110/1/0
*> 3ffe:8703:2005:1::/64                    3ffe:177:7:7::145
         VLAN0007   0/0     Static       3m 34s , Distance: 2/0/0
*  3ffe:8703:2005:1::/64                    fe80::200:87ff:fe28:90d7%VLAN0007
         VLAN0007   -/-     BGP4+       10m 46s , Distance: 20/0/0
*  3ffe:8703:2005:1::/64                    fe80::200:87ff:fe28:90d7%VLAN0007
         VLAN0007   1/1     OSPFv3 ext2  8m 41s , Distance: 110/1/0
*  3ffe:8703:2005:1::/64                    fe80::200:87ff:fe28:90d7%VLAN0007
         VLAN0007   2/0     RIPng       16s     , Distance: 120/0/0
```

To display the distances for routes to a specific destination, execute the `show ipv6 route` operation command with the `all-routes` parameter and the destination network specified. The route's distance is the first value on the `Distance` line in the detailed routing information.

Figure 29-10: Example of displaying routes in the routing table (includes invalid routes, specified destination only)

```
> show ipv6 route all-routes 3ffe:8703:2005:1::/64
Date 20XX/07/14 12:00:00 UTC
Route codes: *  = active,   + = changed to active recently
            ' ' = inactive, - = changed to inactive recently
             r  = RIB failure

Route 3ffe:8703:2005:1::/64
Entries 4 Announced 1 Depth 0 <>

* NextHop 3ffe:177:7:7::145, Interface: VLAN0007
     Protocol <Static>
     Source Gateway ----
     Metric/2    : 0/0
     Distance/2/3: 2/0/0
     Tag : 0, Age :  4m 35s
     AS Path : IGP (Id 1)
     Communities: -
     LocalPref : -
     RT State: <Remote Int Active Gateway>

  NextHop fe80::200:87ff:fe28:90d7%VLAN0007, Interface: VLAN0007
     Protocol <BGP4+>
     Source Gateway fe80::200:87ff:fe28:90d7%VLAN0007
     Metric/2    : -/-
     Distance/2/3: 20/0/0
     Tag : 0, Age : 11m 47s
     AS Path : 1000 IGP (Id 3)
     Communities: -
     LocalPref : 100
     RT State: <Ext Gateway>
```

To check detailed attribute information for routes, use the `show ipv6 route` operation command with the `all-routes` and `-F` parameters specified.

Figure 29-11: Example of displaying routes in the routing table (including invalid routes, and detailed attributes)

```
> show ipv6 route all-routes -F
Date 20XX/07/14 12:00:00 UTC
Status Codes: * valid, > active, r RIB failure
Total: 11 routes
   Destination                               Next Hop
       Interface   Metric  Protocol     Age
*> ::1/128                                   ::1
       localhost   0/0     Connected     4h 55m , Distance: 0/0/0, Tag: 0, AS
-Path: IGP (Id 1), Communities: -, LocalPref: -, <NoAdvise Active Retain>
*> 3ffe:177:7:7::/64                         3ffe:177:7:7::1
       VLAN0007    0/0     Connected    51m  2s , Distance: 0/0/0, Tag: 0, AS
-Path: IGP (Id 1), Communities: -, LocalPref: -, <Active Retain>
*  3ffe:177:7:7::/64                         3ffe:177:7:7::1
       VLAN0007    1/-     OSPFv3 intra 18m 13s , Distance: 110/1/0, Tag: 0,
AS-Path: IGP (Id 1), Communities: -, LocalPref: -, <Int Gateway>
*> 3ffe:177:7:7::1/128                       ::1
       localhost   0/0     Connected    51m  2s , Distance: 0/0/0, Tag: 0, AS
-Path: IGP (Id 1), Communities: -, LocalPref: -, <NoAdvise Int Active Retain>
*> 3ffe:3b01:6705:1::/64                     fe80::200:87ff:fe28:90d7%VLAN0007
       VLAN0007    2/0     RIPng         4s     , Distance: 120/0/0, Tag: 0,
AS-Path: IGP (Id 1), Communities: -, LocalPref: -, <Int Active Gateway>
*> 3ffe:3b11:6705:1::/64                     fe80::200:87ff:fe28:90d7%VLAN0007
       VLAN0007    -/-     BGP4+         3m  6s , Distance: 20/0/0, Tag: 0, A
S-Path: 1000 IGP (Id 3), Communities: -, LocalPref: 100, <Ext Active Gateway>
*> 3ffe:3b21:6705:1::/64                     fe80::200:87ff:fe28:90d7%VLAN0007
       VLAN0007    1/1     OSPFv3 ext2  15m 24s , Distance: 110/1/0, Tag: 0,
 AS-Path: IGP (Id 1), Communities: -, LocalPref: -, <Int Ext Active Gateway>
*> 3ffe:8703:2005:1::/64                     3ffe:177:7:7::145
       VLAN0007    0/0     Static       12m 36s , Distance: 2/0/0, Tag: 0, AS
-Path: IGP (Id 1), Communities: -, LocalPref: -, <Remote Int Active Gateway>
*  3ffe:8703:2005:1::/64                     fe80::200:87ff:fe28:90d7%VLAN0007
       VLAN0007    -/-     BGP4+         3m  6s , Distance: 20/0/0, Tag: 0, A
S-Path: 1000 IGP (Id 5), Communities: 300:300, LocalPref: 100, <Ext Gateway>
*  3ffe:8703:2005:1::/64                     fe80::200:87ff:fe28:90d7%VLAN0007
       VLAN0007    1/1     OSPFv3 ext2  17m 43s , Distance: 110/1/0, Tag: 0,
 AS-Path: IGP (Id 1), Communities: -, LocalPref: -, <Int Ext Gateway>
*  3ffe:8703:2005:1::/64                     fe80::200:87ff:fe28:90d7%VLAN0007
       VLAN0007    2/0     RIPng         4s     , Distance: 120/0/0, Tag: 0,
AS-Path: IGP (Id 1), Communities: -, LocalPref: -, <Int Gateway>
```

29.3.5 Checking routes prior to advertised route filtering

In basic terms, the routes subject to advertisement are the prioritized routes contained in the routing table. You can check which routes are subject to advertised route filtering by displaying the routes in the routing table.

To display the prioritized routes in the routing table, execute the `show ipv6 route` operation command.

Figure 29-12: Example of displaying routes in the routing table

```
> show ipv6 route
Date 20XX/07/14 12:00:00 UTC
Total: 7 routes
Destination                                 Next Hop
      Interface     Metric  Protocol    Age
::1/128                                     ::1
      localhost     0/0     Connected    5h  7m
3ffe:177:7:7::/64                           3ffe:177:7:7::1
      VLAN0007      0/0     Connected    1h  2m
3ffe:177:7:7::1/128                         ::1
      localhost     0/0     Connected    1h  2m
3ffe:3b01:6705:1::/64                       fe80::200:87ff:fe28:90d7%VLAN0007
      VLAN0007      2/0     RIPng       35s
3ffe:3b11:6705:1::/64                       fe80::200:87ff:fe28:90d7%VLAN0007
      VLAN0007      -/-     BGP4+       14m 29s
3ffe:3b21:6705:1::/64                       fe80::200:87ff:fe28:90d7%VLAN0007
      VLAN0007      1/1     OSPFv3 ext2 26m 47s
3ffe:8703:2005:1::/64                       3ffe:177:7:7::145
      VLAN0007      0/0     Static      23m 59s
```

To limit the command output to prioritized routes learned by a specific protocol, execute the `show ipv6 route` operation command with the protocol specified as a parameter.

Figure 29-13: Example of displaying routes in the routing table (BGP4+ only)

```
> show ipv6 route bgp
Date 20XX/07/14 12:00:00 UTC
Total: 1 routes
Destination                                 Next Hop
      Interface      Metric   Protocol  Age
3ffe:3b11:6705:1::/64                       fe80::200:87ff:fe28:90d7%VLAN0007
      VLAN0007       -/-      BGP4+     34m  8s
```

To check detailed attribute information for the prioritized routes in the routing table, execute the `show ipv6 route` operation command with the `-F` parameter specified.

Figure 29-14: Example of displaying routes in the routing table (detailed)

```
> show ipv6 route -F
Date 20XX/07/14 12:00:00 UTC
Total: 7 routes
Destination                               Next Hop
      Interface   Metric  Protocol    Age
::1/128                                   ::1
      localhost   0/0     Connected    5h 27m , Distance: 0/0/0, Tag: 0, AS-Pa
th: IGP (Id 1), Communities: -, LocalPref: -, <NoAdvise Active Retain>
3ffe:177:7:7::/64                         3ffe:177:7:7::1
      VLAN0007    0/0     Connected    1h 22m , Distance: 0/0/0, Tag: 0, AS-Pa
th: IGP (Id 1), Communities: -, LocalPref: -, <Active Retain>
3ffe:177:7:7::1/128                       ::1
      localhost   0/0     Connected    1h 22m , Distance: 0/0/0, Tag: 0, AS-Pa
th: IGP (Id 1), Communities: -, LocalPref: -, <NoAdvise Int Active Retain>
3ffe:3b01:6705:1::/64                     fe80::200:87ff:fe28:90d7%VLAN0007
      VLAN0007    2/0     RIPng       13s     , Distance: 120/0/0, Tag: 0, AS-
Path: IGP (Id 1), Communities: -, LocalPref: -, <Int Active Gateway>
3ffe:3b11:6705:1::/64                     fe80::200:87ff:fe28:90d7%VLAN0007
      VLAN0007    -/-     BGP4+       34m 56s , Distance: 20/0/0, Tag: 0, AS-P
ath: 1000 IGP (Id 3), Communities: -, LocalPref: 100, <Ext Active Gateway>
3ffe:3b21:6705:1::/64                     fe80::200:87ff:fe28:90d7%VLAN0007
      VLAN0007    1/1     OSPFv3 ext2 47m 15s , Distance: 110/1/0, Tag: 0, AS
-Path: IGP (Id 1), Communities: -, LocalPref: -, <Int Ext Active Gateway>
3ffe:8703:2005:1::/64                     3ffe:177:7:7::145
```

```
      VLAN0007    0/0     Static      44m 27s , Distance: 2/0/0, Tag: 0, AS-Pa
th: IGP (Id 1), Communities: -, LocalPref: -, <Remote Int Active Gateway>
```

The BGP4+ protocol sometimes advertises routes that do not have priority in the routing table. To include non-priority BGP+4 routes in the output of the `show ipv6 route` operation command, execute the command with the `all-routes` and `bgp` parameters specified.

Figure 29-15: Example of displaying routes in the routing table (includes invalid routes, BGP4+ only)

```
> show ipv6 route all-routes bgp
Date 20XX/07/14 12:00:00 UTC
Status Codes: * valid, > active, r RIB failure
Total: 2 routes
   Destination                                  Next Hop
      Interface  Metric  Protocol  Age                                    _
*> 3ffe:3b11:6705:1::/64                  fe80::200:87ff:fe28:90d7%VLAN0007 |
      VLAN0007   -/-     BGP4+     35m 57s                                  |
*  3ffe:8703:2005:1::/64                  fe80::200:87ff:fe28:90d7%VLAN0007 |
      VLAN0007   -/-     BGP4+     35m 57s                                 _|
                                                                           #
```

#

The characters `*` and `>` at the beginning of an entry signify the following:

*: Signifies a valid route. Its absence indicates an invalid route.

>: Signifies a prioritized route. Only prioritized routes are used for packet transfer.

29.3.6 Checking RIPng advertised routes

To check the routes that RIPng advertises, execute the `show ipv6 rip` operation command with the `advertised-routes` parameter specified. The command output includes the target interface name, the route advertised to that address, and the attributes of the route.

Figure 29-16: Example of displaying RIPng advertised routes

```
> show ipv6 rip advertised-routes
Date 20XX/07/14 12:00:00 UTC

Target Interface: VLAN0006
Destination                                 Next Hop
      Interface      Metric   Tag   Age
3ffe:3b01:6705:1::/64                       fe80::200:87ff:fe28:90d7%VLAN0007
      VLAN0007       2        0      3s
```

29.3.7 Checking OSPFv3 advertised routes [OS-L3SA]

AS External LSAs contain the OSPFv3 routes selected for advertisement by advertised route filtering.

To check which of the AS External LSAs have been generated by the local device itself, execute the `show ipv6 ospf` operation command with the `database`, `external`, and `self-originate` parameters specified.

Figure 29-17: Example of displaying AS External LSAs (generated by local device)

```
> show ipv6 ospf database external self-originate
Date 20XX/07/14 12:00:00 UTC
Domain: 1
Local Router ID: 177.7.7.4
LS Database: AS-external-LSA
Advertising Router: 177.7.7.4
    LSID: 0000000a, Age: 298, Length: 36
    Sequence: 80000001, Checksum: 6c76
    Prefix: 3ffe:177:7:7::/64                                    ...1
```

```
        Prefix Options: <>
        Type: 2, Metric: 20, Tag: 100
```

1. The `Prefix` (3ffe:177:7:7::/64) is the route's destination network.

29.3.8 Checking BGP4+ advertised routes [OS-L3SA]

To check which routes are advertised by BGP4+, execute the `show ipv6 bgp` operation command with the `advertised-routes` parameter specified.

Figure 29-18: Example of displaying BGP4+ advertised routes

```
> show ipv6 bgp advertised-routes
Date 20XX/07/14 12:00:00 UTC
BGP4+ Peer: 3ffe:192:158:1::145, Remote AS: 1000
Local AS: 200, Local Router ID: 200.201.202.203
Origin Codes: i - IGP, e - EGP, ? - incomplete
Network                                    Next Hop
      MED    LocalPref Path
3ffe:3b11:6705:1::/64                      fe80::200:87ff:fe28:90d7%VLAN0007
      -      -         200 1000 i
3ffe:8703:2005:1::/64                      fe80::200:87ff:fe28:90d7%VLAN0007
      -      -         200 1000 i
```

To include detailed information about route attributes in the command output, use the `show ipv6 bgp` operation command with the `advertised-routes` and `-F` parameters specified. Use this method to check the `ORIGIN`, `AS_PATH`, `MED`, `LOCAL_PREF`, and `COMMUNITIES` attributes of the routes.

Figure 29-19: Example of displaying detailed information about BGP4+ advertised routes

```
> show ipv6 bgp advertised-routes -F
Date 20XX/07/14 12:00:00 UTC
BGP4+ Peer: 3ffe:192:158:1::145, Remote AS: 1000
Local AS: 200, Local Router ID: 200.201.202.203
Status Codes: d dampened, * valid, > active, S Stale, r RIB failure
Route 3ffe:3b11:6705:1::/64
*> Next Hop fe80::200:87ff:fe28:90d7%VLAN0007
     MED: -, LocalPref: -, Type: External route
     Origin: IGP, IGP Metric: 0
     Path: 200 1000
     Next Hop Attribute: 3ffe:192:158:1::1
                         fe80::4048:47ff:fe10:4
     Communities: 200:1200
Route 3ffe:8703:2005:1::/64
*  Next Hop fe80::200:87ff:fe28:90d7%VLAN0007
     MED: -, LocalPref: -, Type: External route
     Origin: IGP, IGP Metric: 0
     Path: 200 1000
     Next Hop Attribute: 3ffe:192:158:1::1
                         fe80::4048:47ff:fe10:4
     Communities: 200:1200
```

29.3.9 Checking extranet [OS-L3SA]

To display only the imported routes, specify `extra-vrf` as the protocol in the `show ipv6 route` operation command.

Figure 29-20: Example of show ipv6 route command output

```
> show ipv6 route vrf 2 extra-vrf
Date 20XX/12/20 12:00:00 UTC
VRF:2  Total: 1 routes
Destination                             Next Hop
     Interface     Metric  Protocol  Age
3ffe:210:6705:1::/64                    3ffe:501:811:ff01::1
     VLAN0010      0/0     Extra-Vrf 365d
:
```

```
>
> show ipv6 route vrf 3 extra-vrf
Date 20XX/12/20 12:00:00 UTC
VRF:3  Total: 1 routes
Destination                              Next Hop
     Interface     Metric  Protocol  Age
3ffe:109:2011:1::/64                     3ffe:500:811:1000::1
     VLAN0011      0/0     Extra-Vrf 365d
>
```

Chapter
30. Description of IPv6 Multicasting

Multicasting is a term used to describe sending the same information to selected groups within a network. This chapter describes multicasting as implemented in IPv6 networks.

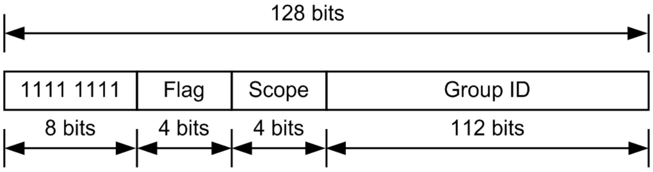
30.1 Overview of IPv6 multicasting

The functionality implemented by IPv6 multicasting is the same as the IPv4 multicasting functionality. For details about the IPv4 multicasting functionality, see *14.1 Overview of IPv4 multicasting*. Note that IPv4 multicasting and IPv6 multicasting can operate independently of each other. Both IPv4 multicasting and IPv6 multicasting can be set separately on the same router.

30.1.1 IPv6 multicast addresses

In IPv6 multicast communication, IPv6 addresses, the six highest-order bits of which are FF (in hexadecimal), are used as destination addresses. An IPv6 multicast address is a logical group address that exists only within a group participating in the sending and receiving of multicast data. The following figure shows the IPv6 multicast address format.

Figure 30-1: Multicast address format



30.1.2 IPv6 multicast routing functionality

The Switch forwards received IPv6 multicast packets based on IPv6 multicast forwarding entries. IPv6 multicast routing functionality consists of the following three types of functionality:

- IPv6 multicast group management functionality

  This functionality sends and receives IPv6 group membership information to learn about the existence of IPv6 multicast groups. For the Switch, the MLD (Multicast Listener Discovery) protocol is used.

- IPv6 path control functionality

  This functionality sends and receives routing information to determine forwarding paths and to create IPv6 multicast routing information and IPv6multicast forwarding entries. PIM-SM (including PIM-SSM) is used to collect routing information.

- IPv6 forwarding functionality

  This functionality forwards IPv6 multicast packets by using hardware and software based on IPv6 multicast forwarding entries.

  The IPv6 forwarding functionality of the Switch can be used with the QoS functionality or Filter functionality to enable the QoS functionality or filter out unnecessary packets in IPv6 multicast communication.

30.2 IPv6 multicast group management functionality

IPv6 multicast group management functionality learns about the existence of IPv6 multicast group members on networks to which the router is directly connected by sending and receiving IPv6 group membership information between routers and hosts. The Switch supports MLD as the management protocol for implementing IPv6 multicast group management functionality.

30.2.1 Overview of MLD

MLD is the IPv6 multicast group-management protocol that is used between routers and hosts, and provides functionality equivalent to the functionality of IGMP, which is used for IPv4 multicasting.

Using the MLD protocol, the router asks hosts whether they have joined IPv6 multicast groups, and the hosts reply to the router. In this way, the router recognizes whether hosts have joined or left IPv6 multicast groups in order to control whether to forward or block IPv6 multicast packets. The MLD protocol functionality is the same as the IGMP functionality, except that the MLD protocol uses IPv6 addresses for communication.

Two versions of MLD (MLDv1 and MLDv2) have been described in RFCs.

MLDv2 is a protocol extended from MLDv1 to implement IPv6 multicast group management functionality by introducing functionality to filter senders so that multicast packets are only received from specified senders. Because senders can be specified when the joining or leaving of an IPv6 multicast group is reported, MLDv2 and PIM-SSM can be used in combination to implement more efficient IPv6 multicast forwarding.

The format and setting values for MLDv1 messages sent by the Switch follows RFC 2710. Likewise, the format and setting values for MLDv2 messages follow RFC 3810.

30.2.2 MLD operation

(1) MLDv1 operation

The following table describes the specifications that the Switch supports for MLDv1 messages.

Table 30-1: MLDv1 messages

Type		Meaning	Supported	
			Sending	Receiving
Multicast Listener Query	General Query	Query to check the hosts that have joined the IPv6 multicast group (query to all groups)	Y	Y
	Group-Specific Query	Query to check the hosts that have joined the IPv6 multicast group (query to a specific group)	Y	Y
Multicast Listener Report		Report on the IPv6 multicast group the host has joined	N	Y
Multicast Listener Done		Report on the IPv6 multicast group the host has left	N	Y

Legend: Y: Supported, N: Not supported

(2) MLDv2 operation

MLDv2 enables source filtering functionality by specifying a filter mode and sender list. There are two filter modes:

- INCLUDE: Only packets from senders included in the specified sender list are forwarded.

- EXCLUDE: Only packets from senders that are not included in the specified sender list are forwarded.

The following table describes the specifications that the Switch supports for MLDv2 messages.

Table 30-2: MLDv2 messages

Type		Meaning	Supported	
			Sending	Receiving
Version 2 Multicast Listener Query	General Query	Query to check the hosts that have joined the IPv6 multicast group (query to all groups)	Y	Y
	Multicast Address Specific Query	Query to check the hosts that have joined the IPv6 multicast group (query to a specific group)	Y	Y
	Multicast Address and Source Specific Query	Query to check the hosts that have joined the IPv6 multicast group (query to a specific sender and group)	Y	Y
Version 2 Multicast Listener Report	Current State Report	Report on the IPv6 multicast group and the filter mode of the host	N	Y
	State Change Report	Report on the changes to the IPv6 multicast group and filter mode of the host	N	Y

Legend: Y: Supported, N: Not supported

The filter mode and sender list can be changed after the host has joined a group by changing the Multicast Address Record specification in a Report message. The following table describes the Multicast Address Record types supported by the Switch.

Table 30-3: Multicast Address Record types

Type		Meaning	Supported
Current State Report	MODE_IS_INCLUDE	Indicates that the mode is INCLUDE	Y
	MODE_IS_EXCLUDE	Indicates that the mode is EXCLUDE	Y (the sender list is disregarded)
State Change Report	CHANGE_TO_ INCLUDE_MODE	Indicates that the filter mode has changed to INCLUDE	Y
	CHANGE_TO_ EXCLUDE_MODE	Indicates that the filter mode has changed to EXCLUDE	Y (the sender list is disregarded)
	ALLOW_NEW_ SOURCES	Indicates that a sender wishing to receive data has been added	Y
	BLOCK_OLD_ SOURCES	Indicates that a sender wishing to receive data has been deleted	Y

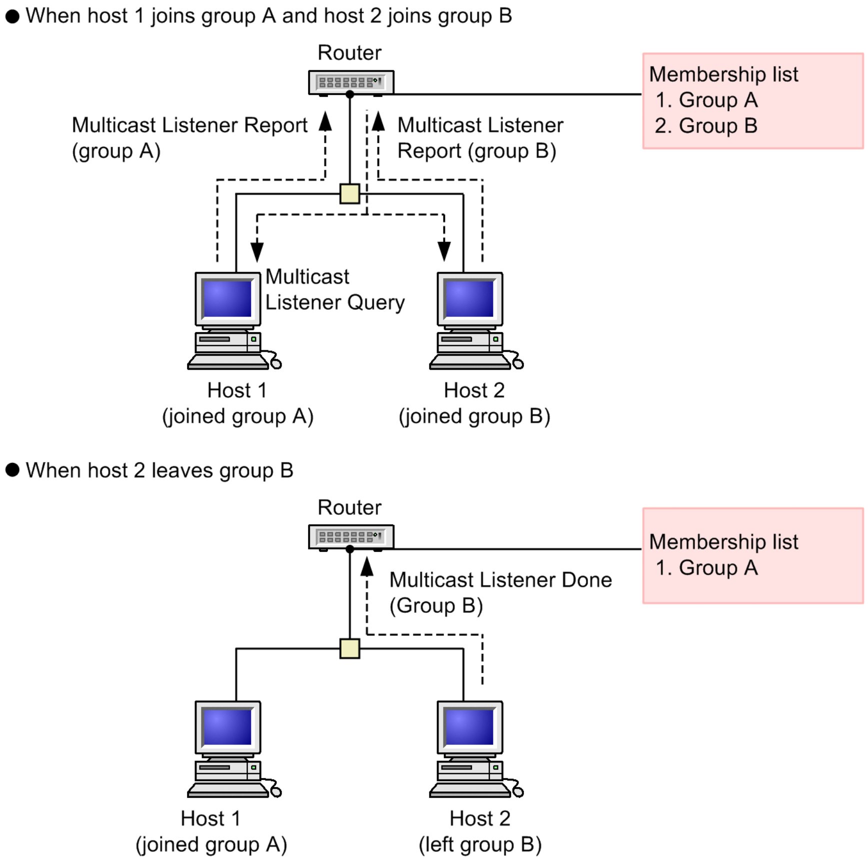
Legend: Y: Supported

The following describes MLDv1 operation that uses MLDv1 messages:

- IPv6 multicast routers regularly send a Multicast Listener Query message to a link-local address ff02::1 for all nodes to obtain information about IPv6 multicast membership on directly connected interfaces.
- When a host receives a Multicast Listener Query, it sends a Multicast Listener Report to the corresponding group to report the join status to the group.
- When a Multicast Listener Report is received from a host, the IPv6 multicast router adds the group to the membership list.
- When a Multicast Listener Done message is received, the group is deleted from the membership list.

The following figure shows how MLDv1 groups are joined and left.

Figure 30-2: Joining and leaving groups (MLDv1)

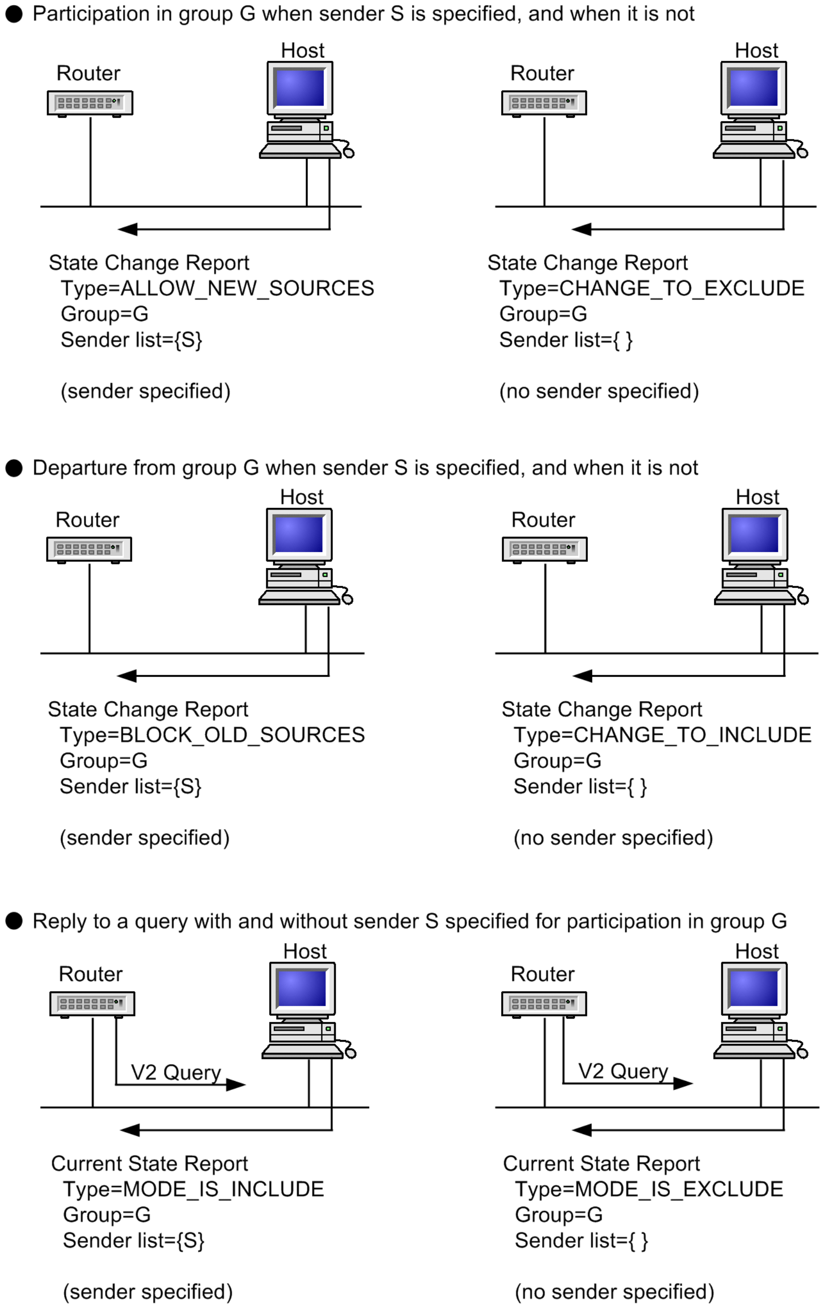


The following describes MLDv2 operation that uses MLDv2 messages:

- IPv6 multicast routers regularly send a Version 2 Multicast Listener Query (General Query) message to the link-local address ff02::1 for all nodes to obtain information about IPv6 multicast membership on directly connected interfaces.
- When a host receives a Version 2 Multicast Listener Query, it sends a Version 2 Multicast listener Report (Current State Report) to ff02::16 to report the join status to the group.
- When an IPv6 multicast router receives a Version 2 Multicast Listener Report (State Change Report) message from a host, the router adds or removes the group from the membership list based on the Multicast Address Record type.

The following figure shows how MLDv2 Report messages are sent from hosts.

Figure 30-3: Joining and leaving groups (MLDv2)



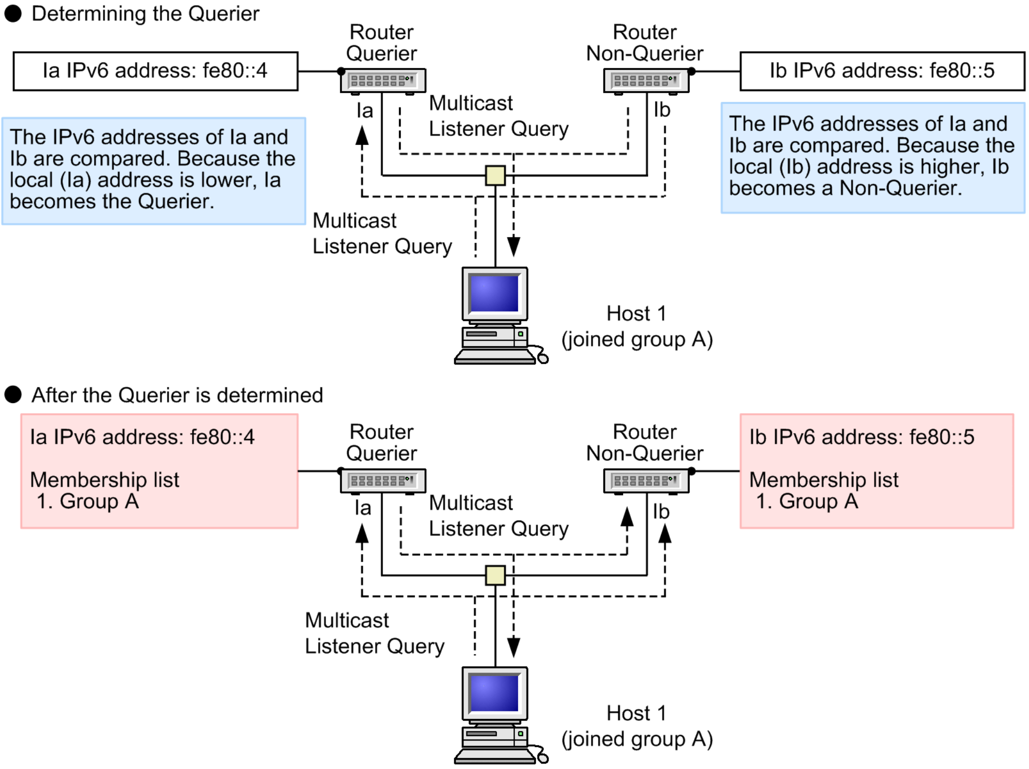
30.2.3 Determining the Querier

MLD routers act as either Queriers or Non-Queriers. If multiple routers exist on the same network, one of the routers becomes the Querier, which regularly sends a Multicast Listener Query message.

Each MLD router on the network compares the IPv6 link-local address of the local interface with the IPv6 link-local addresses of the senders of Multicast Listener Query messages received from other MLD routers. If the address of the local interface is lower than any other addresses, the MLD router runs as the Querier. If the address of the local interface is higher than any of other addresses, the MLD router becomes a Non-Querier, and does not send a Multicast Listener Query message.

This means that only one Querier can exist on the same network. The following figure shows how the Querier and Non-Querier routers are determined.

Figure 30-4: Determining the Querier and Non-Querier routers



When a router becomes the Querier, it runs as the Querier until a Multicast Listener Query with a source IPv6 address lower than the local interface is received, regularly sending Multicast Listener Queries (every 125 seconds by default). A router running as a Non-Querier can become the Querier in the following cases:

- When a Non-Querier monitoring Multicast Listener Query messages sent from the Querier receives a Multicast Listener Query message with the source IPv6 link-local address higher than the local address of the local interface
- When no Multicast Listener Query messages are received within a set time (255 seconds by default)

Determination of the Querier uses only IPv6 link-local addresses that are set on interfaces. Other addresses have no effect on determining the Querier.

The Querier determination method described above is common to both MLDv1 and MLDv2.

30.2.4 Managing IPv6 group members

(1) Managing IPv6 group members when MLDv1 is used

This section explains the registration and removal of IPv6 group members when the MLDv1 protocol is used.

When a router receives a Multicast Listener Report message from a host, the router registers the host as an IPv6 group member. Both the Querier and Non-Querier routers can perform this registration.

If the Querier receives a Multicast Listener Done message that reports that the host is leaving an IPv6 group, the Querier checks whether any other hosts remain in the group. The Querier does this by sending continuous Multicast Listener Query (Group-Specific Query) messages to the relevant group every second. After the Querier sends a Multicast Listener Query (Group-Specific Query) message twice, if a Multicast Listener Report message is not returned within one second, the Querier deletes the relevant group. Non-Queriers ignore the Multicast Listener Done message and receive the Multicast Listener Query (Group-Specific Query) message sent by the Querier twice and deletes the target group if it does not receive Multicast Listener Report within one second.

(2) Managing IPv6 group members when MLDv2 is used

The following explains IPv6 group member registration and deletion when MLDv2 is used.

When a host sends a Report message indicating a request to join a multicast group and a router receives the message, the router registers group information that consists of the group address and the source address. Both the Querier and Non-Querier routers can perform this registration.

When the Querier receives a Report message indicating a leave request from a multicast group, the Querier checks for the existence of other hosts that have joined the group member by sending a message every second indicating the following, based on whether a sender list is specified.

- If no sender list is specified: Multicast Address Specific Query message
- If a sender list is specified: Multicast Address and Source Specific Query message

If the Switch is the Querier, after these messages are sent twice, if no Report is received within one second, the corresponding group information is deleted. If the Switch is a Non-Querier, corresponding group information is deleted after a message is received from the Querier.

30.2.5 MLD timers

The following table describes the MLDv1 timer values used by the Switch.

Table 30-4: MLDv1 timer values

Timer	Description	Default value (in seconds)	Range of values that can be set in configuration mode (in seconds)	Remarks
Query Interval	Interval for sending a Multicast Listener Query message	125	60 to 3600	--
Query Response Interval	Timeout for waiting for a Multicast Listener Report message	10	--	--
Other Querier Present Interval	Querier monitoring interval	255	*Query Interval* x 2 + *Query Response Interval* / 2	Calculated based on the formula to the left.
Startup Query Interval	Interval for sending a General Query message during startup	30	*Query Interval* / 4	Calculated based on the formula to the left.
Last Member Query Interval	Interval for sending a Specific Query message after receiving a Done message	1	--	--
Multicast Listener Interval	Group member retention time	260	*Query interval* x 2 + *Query Response Interval*	Calculated based on the formula to the left.

Legend: --: Not applicable

The following table describes the MLDv2 timer values used by the Switch.

Table 30-5: MLDv2 timer values

Timer	Description	Default value (in seconds)	Range of values that can be set in configuration mode (in seconds)	Remarks
Query Interval	Interval for sending a Multicast Listener Query message	125	60 to 3600	--
Query Response Interval	Timeout for waiting for a Multicast Listener Report message	10	--	--
Other Querier Present Interval	Querier monitoring interval	255	*Query Interval* x 2 + *Query Response Interval* / 2	Calculated based on the formula to the left.
Startup Query Interval	Interval for sending a General Query message during startup	30	*Query Interval* / 4	Calculated based on the formula to the left.
Last Listener Query Interval	Interval for sending a Specific Query message after receiving a leave request message	1	--	--
Multicast Address Listening Interval	Group member retention time	260	*Query Interval* x 2 + *Query Response Interval*	Calculated based on the formula to the left.
Older Version Host Present Interval	Time for switching to MLDv2 multicast address compatibility mode	260	*Query Interval* x 2 + *Query Response Interval*	Calculated based on the formula to the left.

Legend: --: Not applicable

30.2.6 Connecting to MLDv1 and MLDv2 devices

The Switch supports MLDv1 and MLDv2. The `ipv6 mld version` configuration command can be used to set the version of MLD used by each interface. The following table describes the operation status corresponding to each specified version. Version 2 is used by default.

Table 30-6: Operation when the MLD version is specified

Specified version	Operation when the version is specified
version 1	MLDv1 is used for operation. MLDv2 packets are disregarded.
version 2	Both MLDv1 and MLDv2 can be used for operation. MLDv1 and MLDv2 are used for operation for each group address.
version 2 only	MLDv2 is used for operation. MLDv1 packets are disregarded.

(1) Connecting to MLDv1 and MLDv2 routers

If multiple MLD routers exist in the same network such as in redundant configurations, the Querier is determined by mutual Query reception (for details, see *30.2.3 Determining the Querier*). The

Switch does not support connections to MLDv1 routers through interfaces for which the MLD version is set to version 2 or version 2 only. This is because v1 Queries are disregarded, preventing the Querier from being determined. To connect to an MLDv1 router, set the MLD version of the corresponding interface to version 1.

(2) Operation during mixed usage with MLDv1 and MLDv2 hosts

When connecting to a network on which both MLDv1 hosts and MLDv2 hosts are used, make sure that the default MLD version setting is specified for the interface. However, MLDv1 hosts need to be able to receive MLDv2 Query messages as MLDv1 Query messages (as specified by the RFC).

If MLDv1 hosts and MLDv2 hosts are both used and a group join request is received, group member registrations differ depending on the MLD version, as described in the following table.

Table 30-7: Group member registration when MLDv1 hosts and MLDv2 hosts are used together

Group join request	Group member registration
Received by MLDv1	Group members are registered in MLDv1 mode
Received by MLDv2	Group members are registered in MLDv2 mode
Received by MLDv1 and MLDv2	Group members are registered in MLDv1 mode

30.2.7 Static group joins

Use the static group join functionality to forward IPv6 multicast packets on networks without hosts that support MLD.

Interfaces for which static group joining is set run in the same way (without receiving any Multicast Listener Reports) as interfaces that have joined groups.

Because this functionality belongs to MLDv1, it does not work when the MLD version of the corresponding interface is set to version 2 only. Likewise, when version 2 is set, this functionality runs the same as the group join functionality under MLDv1.

30.2.8 Notes that apply when MLD is used

- When static group joining is set due to a configuration change, as many as 125 seconds might be needed to create a (*,G) entry for a PIM-SM group, or a (S,G) entry for a PIM-SSM group.
- When an MLDv2 Report with a sender specified is received for a group for which the SSM address set in the configuration is out of range, multicast packets for all senders are forwarded.

30.3 IPv6 multicast forwarding functionality

Forwarding of multicast packets is performed in hardware and software based on IPv6 multicast forwarding entries. Forwarding information for forwarded IPv6 multicast packets is registered in hardware IPv6 multicast forwarding entries upon forwarding. Registered IPv6 packets are forwarded by hardware, and unregistered IPv6 packets are forwarded based on IPv6 multicast forwarding entries generated by software IPv6 multicast routing information. The IPv6 multicast forwarding functionality is the same as the IPv4 multicast forwarding functionality except that there are restrictions on addresses subject to forwarding in the IPv6 multicast forwarding functionality.

30.3.1 Addresses subject to forwarding

IPv6 multicast forwarding does not support some types of IPv6 multicast addresses. Node-local and link-local IPv6 multicast addresses are not subject to IPv6 multicast forwarding.

For details about IPv6 multicast addresses, see *17.1.5 Multicast addresses*.

30.3.2 IPv6 multicast packet forwarding

Both hardware-based IPv6 multicast packet forwarding and software-based IPv6 multicast packet forwarding are used. The following describes these types of IPv6 multicast packet forwarding:

(1) Hardware-based forwarding

Hardware-based IPv6 multicast packet forwarding consists of the following four steps:

1. Searching of the IPv6 multicast forwarding entries

   When a packet bound for an IPv6 multicast group is received, the hardware multicast forwarding entries are searched for the corresponding entries.

2. Checking the validity of the packet reception interface

   If the search in step 1 returns entries, whether the IPv6 packet was received from a valid interface is checked.

3. Filtering

   The information registered in the IPv6 filtering table is checked to determine whether the packet is to be forwarded.

4. Determining whether to forward the packet based on the hop limit, and decrementing the TTL value

   The hop limit value in the packet is checked to determine whether the packet is to be forwarded. If the packet is to be forwarded, the hop limit value in the packet is decremented.

(2) Software-based forwarding

The processing for software-based IPv6 multicast packet forwarding differs as described for the following conditions:

- If no corresponding entries are found by the search performed in hardware-based forwarding

  When a packet from a given sender bound for a given IPv6 multicast group is first received, it is forwarded by software based on a forwarding entry generated from the IPv6 multicast routing information. At the same time, an IPv6 multicast forwarding entry is registered in hardware.

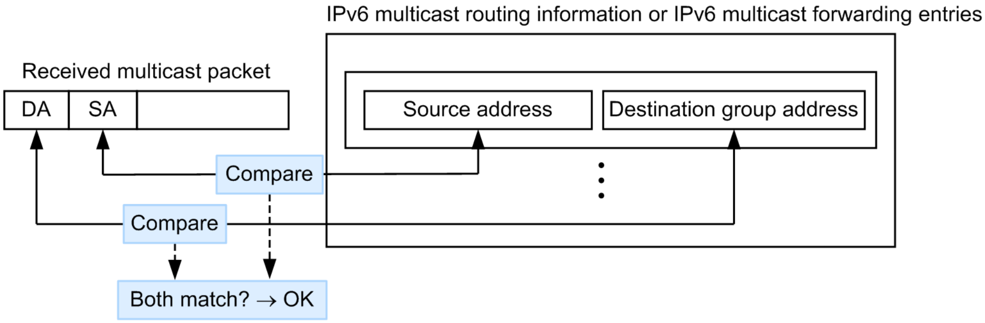
- When IPv6 encapsulation processing is performed

  Forwarding is performed with temporary IPv6 encapsulation up to the rendezvous point, and then decapsulation is performed at the rendezvous point for each forwarding destination.

(3) Searching for IPv6 multicast routing information or IPv6 multicast forwarding entries

When an IPv6 multicast packet is received, the multicast routing information or IPv6 multicast forwarding entries are searched for the entries that match both the DA (destination group address) and SA (source address) of the packet. The following figure shows how IPv6 multicast routing information or IPv6 multicast forwarding entries are searched.

Figure 30-5: How IPv6 multicast routing information or IPv6 multicast forwarding entries are searched


IPv6 multicast routing information or IPv6 multicast forwarding entries
Received multicast packet
DA
SA
Source address
Destination group address
Compare
Compare
Both match? → OK

(4) Negative caching

Negative caching is functionality that uses hardware to discard multicast packets that cannot be forwarded. A negative cache is a forwarding entry for which no forwarding destination interface exists. When a negative cache receives a multicast packet that cannot be forwarded, it registers it in hardware. Then, when a multicast packet is received with the same address as the registered multicast packet, the packet is discarded by using hardware. This prevents undue load when many multicast packets that cannot be forwarded are received.

(5) VRF functionality [OS-L3SA]

When an IPv6 multicast is performed over multiple VRFs, IPv6 multicast forwarding entries can be set for each individual VRF. For different VRFs, IPv6 multicast forwarding entries that have the same IP address can be created. If an IPv6 multicast extranet is used, multicast communication can be performed between different VRFs.

30.3.3 Notes on IPv6 multicast forwarding functionality

Note the following points when using the IPv6 multicast forwarding functionality.

- MTU length of the interface used in IPv6 multicasting

  If the MTU length of the forwarding destination interface is shorter than that of the receiving interface, IPv6 multicast packets whose MTU length is longer than the forwarding destination interface cannot be forwarded properly. Make sure that the same MTU length value is used for all interfaces that are used in IPv6 multicasting.

30.4 IPv6 path control functionality

IPv6 multicast path control functionality creates IPv6 multicast routing information and IPv6 multicast forwarding entries based on neighbor information and group information collected by using the IPv6 multicast routing protocol.

30.4.1 Overview of IPv6 multicast routing protocols

Multicast routing protocols are used for path control. The Switch supports the multicast routing protocols listed below. The format and setting values for IPv6 PIM-SM frames sent by the Switch follows RFC 2362.

- PIM-SM (Protocol Independent Multicast-Sparse Mode)

  This protocol uses the unicast IPv6 path structure to perform multicast path control. After packets have been sent to the rendezvous point, communication is performed on the shortest path.

- PIM-SSM (Protocol Independent Multicast-Source Specific Multicast)

  PIM-SSM is extended PIM-SM functionality. that communicates on the shortest path without using a rendezvous point.

PIM-SM and PIM-SSM can operate simultaneously. but cannot use the same group. Also, if routers running PIM-SM and routers running PIM-DM exist on the same network, multicast packets are not forwarded between routers. To forward multicast packets within the same network, set the same multicast protocol for all routers.

30.4.2 IPv6 PIM-SM

The table below describes supported specifications for IPv6 PIM-SM messages. All messages support both sending and receiving.

Table 30-8: Supported specifications for IPv6 PIM-SM messages

Type	Functionality
PIM-Hello	Detecting nearby PIM routers
PIM-Join/Prune	Joining and pruning multicast delivery trees
PIM-Assert	Determining the forwarder
PIM-Register	Encapsulating multicast packets bound for rendezvous points
PIM-Register-stop	Suppressing Register messages
PIM-Bootstrap	Determining the BSR and distributing rendezvous point information.
PIM-Candidate-RP-Advertisement	Reporting local rendezvous point information from the rendezvous point to the BSR

The following describes the flow of tasks in IPv6 PIM-SM operation:

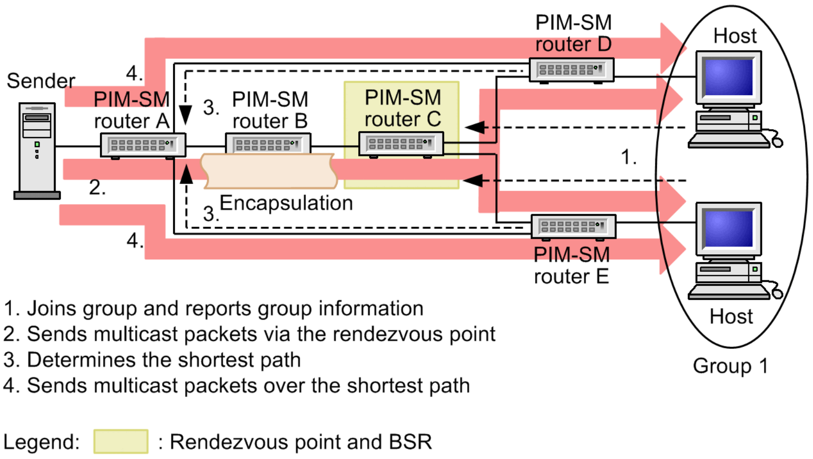
1. Each IPv6 PIM-SM router reports information learned by using MLD to rendezvous points.
2. Rendezvous points acknowledge the existence of each group by receiving group information from each IPv6 PIM-SM router.
3. IPv6 PIM-SM first forms a delivery tree of rendezvous points based on sender in order to deliver multicast packets from sender networks to all group members via the rendezvous points.
4. The existing unicast routing information is used to determine the shortest path from the

senders (form a shortest-path delivery tree from senders), so that multicast packets arrive at the shortest path from the senders to each group.

5. Multicast packet forwarding from senders to each group member over the shortest path is performed.

The following figure provides an overview of PIM-SM operation.

Figure 30-6: Overview of PIM-SM operation

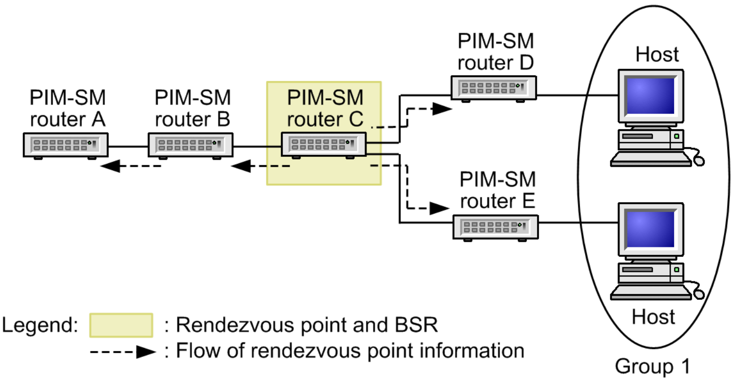

Sender
4.
PIM-SM router A
3.
PIM-SM router B
PIM-SM router C
PIM-SM router D
Host
1.
2.
3.
Encapsulation
4.
PIM-SM router E
Host
Group 1
1. Joins group and reports group information
2. Sends multicast packets via the rendezvous point
3. Determines the shortest path
4. Sends multicast packets over the shortest path
Legend: : Rendezvous point and BSR

(1) Rendezvous points and bootstrap routers (BSRs)

Rendezvous points and bootstrap routers (BSRs) can be set up through configuration. The Switch can use a maximum of 16 BSRs in each network (VPN). Note, however, that separate rendezvous points and bootstrap routers can be used for IPv4 PIM-SM and IPv6 PIM-SM.

The BSR reports the IPv6 address and other rendezvous point information to all multicast interfaces. The information is sent to the link-local multicast address (ff02::d) and reported to all PIM routers on a hop-by-hop basis. The following figure shows the roles of the rendezvous point and the BSR.

Figure 30-7: Roles of the rendezvous point and the bootstrap router (BSR)

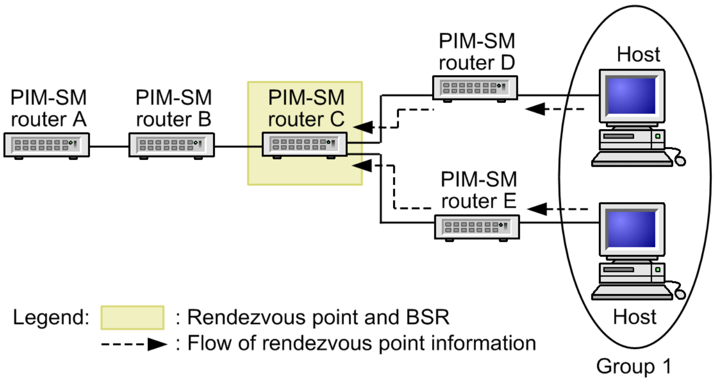


The BSR (PIM-SM router C) reports rendezvous point information to all IPv6 multicast interfaces. Routers receiving rendezvous point information learn the IPv6 address of the rendezvous point, and report rendezvous point information to all other interfaces for which IPv6 PIM routers exist.

(2) Reporting group-joining information to the rendezvous point

Each router reports the group-joining information learned by using MLD to the rendezvous point. Note that the sender and destination IPv6 addresses used for this notification are the device addresses of the corresponding routers. The rendezvous point understands the existence of groups for each interface by receiving IPv6 group information. The following figure shows how group-joining information is reported to the rendezvous point.

Figure 30-8: How group-joining information is reported to the rendezvous point



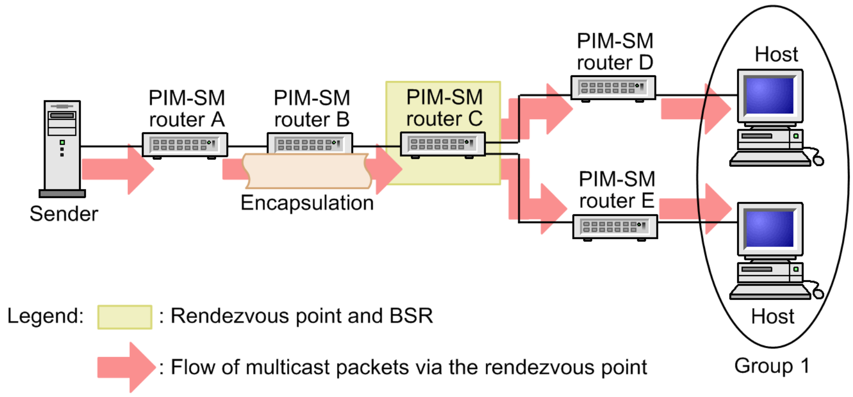
In this diagram, each host joins group 1 by using MLD. PIM-SM router D and PIM-SM router E learn information about group 1, and report this information to the rendezvous point (PIM-SM router C). The rendezvous point (PIM-SM router C) receives this information, and learns that group 1 exists on the interface from which reception occurred.

(3) IPv6 multicast packet communication (encapsulation)

When the sender sends an IPv6 multicast packet bound for group 1, PIM-SM router A IPv6-encapsulates (creates a Register packet) and sends the packet to the rendezvous point (PIM-SM router C). For the Switch, the sender and destination IPv6 addresses used for sending are the device addresses of the corresponding routers. Here, the IP address of the rendezvous point has already been learned. For details, see *(1) Rendezvous points and bootstrap routers (BSRs)*.

When the rendezvous point (PIM-SM router C) receives the IPv6-encapsulated packet, it decapsulates the packet into the original multicast packet bound for group 1, and forwards it to the interface on which group 1 exists. Here, the existence of group 1 has already been learned in *(2) Reporting group-joining information to the rendezvous point*. When PIM-SM routers D and E receive the multicast packet bound for group 1, the packet is forwarded to the interface on which group 1 exists. Here, the existence of group 1 has already been learned by using MLD in *(2) Reporting group-joining information to the rendezvous point*. The following figure shows multicast packet communication (encapsulation) via a rendezvous point.

Figure 30-9: IPv6 multicast packet communication (capsulation)



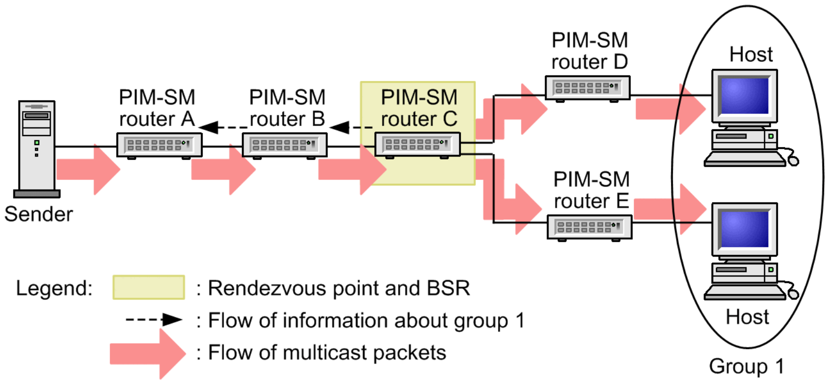
(4) IPv6 multicast packet communication (decapsulation)

When the rendezvous point (PIM-SM router C) receives an IPv6-encapsulated packet, it decapsulates the packet into the original IPv6 multicast packet bound for group 1, and forwards it to the interface on which group 1 exists. (See also *(3) IPv6 multicast packet communication (encapsulation)*).

After this processing has been completed, the rendezvous point reports, over the shortest path, information about group 1 to the sending server determined based on the existing IPv6 unicast routing information. The destination address used for this notification is the link-local multicast address for all PIM routers (ff02::d).

When PIM-SM router B and PIM-SM router A receive information about group 1, they learn that group 1 exists on the interface from which the information was received. Upon receiving an IPv6 multicast packet bound for group 1 from the sending server, PIM-SM router A forwards the packet to the relevant interface without IPv6-encapsulating the packet. When PIM-SM routers B, C, D, and E receive the IPv6 multicast packet, they forward it to the interface on which group 1 exists. The following figure shows IPv6 multicast packet communication (decapsulation).

Figure 30-10: IPv6 multicast packet communication (decapsulation)

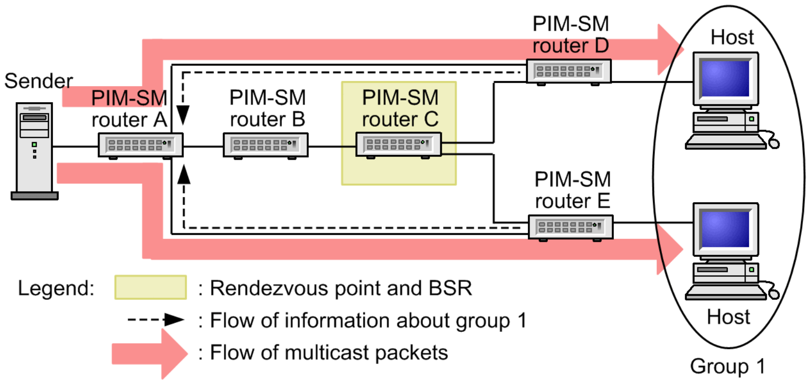


(5) Multicast packet communication over the shortest path

When PIM-SM routers D and E receive an IPv6 multicast packet bound for group 1 from the sending server (as explained in *(4) IPv6 multicast packet communication (decapsulation)*), PIM-SM routers D and E report information about group 1 to the sender over the shortest path (based on the existing IPv6 unicast routing information). The destination address used for this notification is the link-local multicast address for all PIM routers (ff02::d).

When PIM-SM router A receives information about group 1 from PIM-SM routers D and E, it learns that group 1 exists on the interface from which the information was received. When PIM-SM router A receives a multicast packet bound for group 1 from the sending server, it forwards the packet to the relevant interface. The following figure shows multicast packet communication over the shortest path.

Figure 30-11: IPv6 multicast packet communication over the shortest path

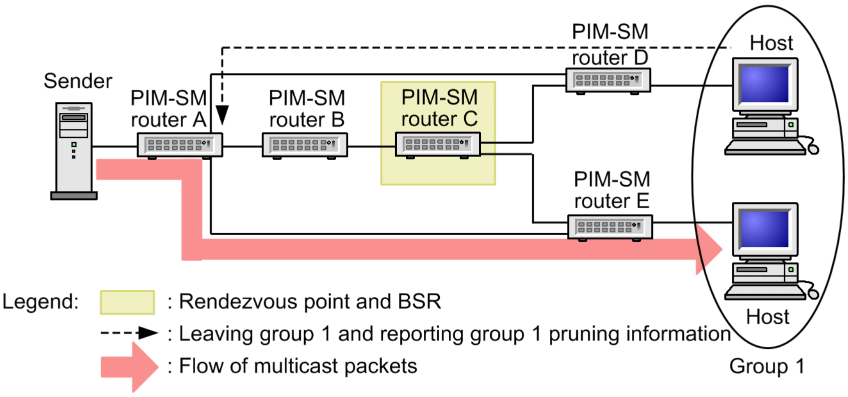


(6) Pruning IPv6 multicast delivery trees

When a host leaves group 1 over MLD, PIM-SM router D reports pruning information for group 1 to the interfaces to which information about group 1 was being reported. The destination address used for this notification is the link-local multicast address for all PIM routers (ff02::d).

When PIM-SM router A receives the pruning information for group 1, it stops forwarding IPv6 multicast packets bound for group 1 to the interface from which the information was received. The following figure shows IPv6 multicast delivery tree pruning.

Figure 30-12: Pruning IPv6 multicast delivery trees


Sender
PIM-SM router A
PIM-SM router B
PIM-SM router C
PIM-SM router D
PIM-SM router E
Host
Host
Group 1
Legend:
: Rendezvous point and BSR
: Leaving group 1 and reporting group 1 pruning information
: Flow of multicast packets

30.4.3 Detecting neighbors

Each IPv6 PIM router regularly distributes an IPv6 PIM-Hello message to all interfaces for which IPv6 PIM is enabled. Distribution of the information is performed by sending the message to the link-local multicast address for all PIM routers (ff02::d). Each IPv6 PIM router dynamically detects neighboring IPv6 PIM routers by receiving IPv6 PIM-Hello messages from them. The Switch supports the Generation ID option for PIM-Hello messages (in accordance with RFC 4601 and draft-ietf-pim-sm-bsr-07).

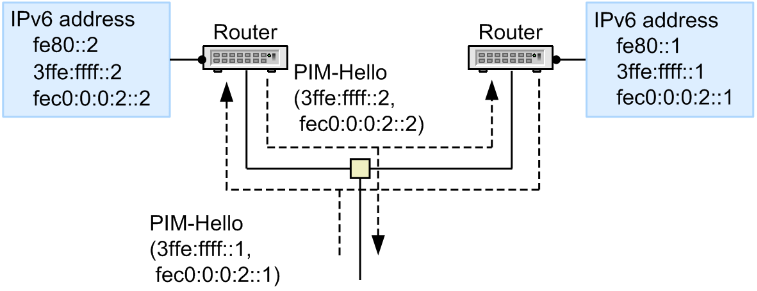
A Generation ID is a 32-bit random number for each multicast interface that is appended when PIM-Hello messages are sent. Generation IDs are regenerated when the status of the multicast interface changes to Up. When the Generation ID option is added to a received PIM-Hello message, the Generation ID is recorded to enable neighboring switches to detect interface failure when the Generation ID changes. When a Generation ID change is detected, updates to neighbor switch information, PIM-Hello messages, PIM Bootstrap messages, and PIM Join/Prune messages are sent without waiting for regular advertisement. This process allows multicast routing information to be relearned quickly.

In addition to the link-local address set for the sender interface, PIM-Hello messages sent by the Switch includes an address list, as optional data (type 24 and 65001) of PIM-Hello messages. By receiving the optional data, the Switch can recognize addresses other than the link-local addresses of neighboring IPv6 PIM routers.

If the next hop of the Switch for a message due to arrive at the IPv6 multicast sender does not have a link-local address, the Switch can detect the IPv6 PIM routers to the sender by using the address list.

The following figure shows how neighboring PIM router addresses are received.

Figure 30-13: PIM-Hello messages used to receive neighboring router addresses



30.4.4 Determining the forwarder

If multiple PIM-SM routers are connected on the same LAN, multicast packets might be duplicated and forwarded on the network.

If multiple PIM-SM routers exist on the same LAN, and two or more of these routers forward multicast packets to the LAN, the PIM-SM routers use PIM-Assert messages to compare the multicast path preferences and metrics, and choose the most appropriate router on the sender network as the forwarder.

Only the router that becomes the forwarder forwards multicast packets on the LAN, in order to prevent duplicate multicast packet forwarding.

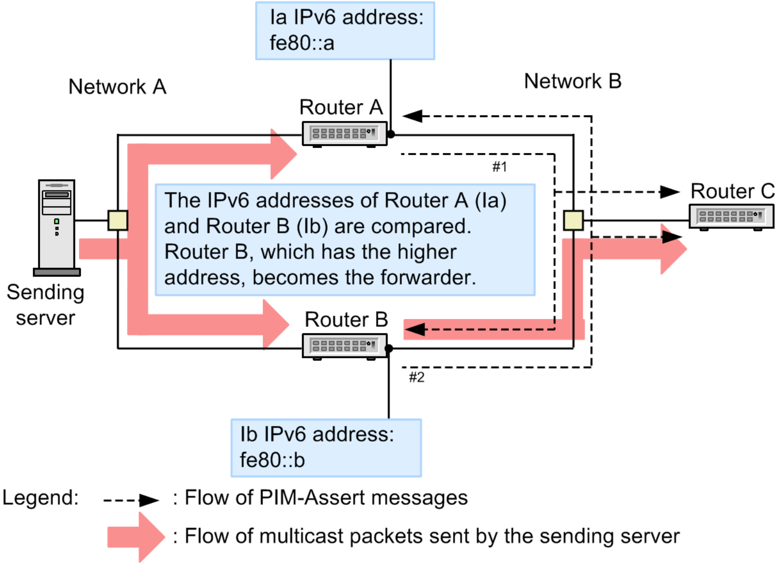
The flow for determining the forwarder by PIM-Assert messages is as follows:

1. Preferences are compared, and the router with the lowest value becomes the forwarder.
2. If the preferences are the same, the metrics are compared, and the router with the lowest value becomes the forwarder.
3. If the metrics are the same, the IP addresses of the routers are compared, and the router with the highest IP address becomes the forwarder.

The Switch sends PIM-Assert messages with the multicast path preference fixed at 101, and the metric fixed at 1024. However, for direct connections with a sender, PIM-Assert messages are sent with the preference and metric both fixed at 0. You can also obtain the distance and metric of the path from the unicast information in configuration mode, and send them as the preference and metric of PIM-Assert messages.

The following figure shows how the forwarder is determined.

Figure 30-14: Determining the forwarder

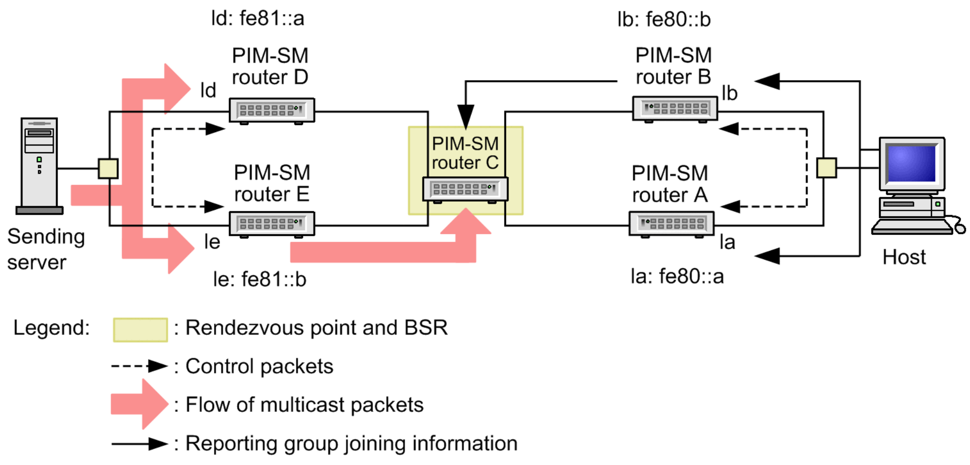

Ia IPv6 address:
fe80::a
Network A
Network B
Router A
#1
Router C
The IPv6 addresses of Router A (Ia) and Router B (Ib) are compared. Router B, which has the higher address, becomes the forwarder.
Sending server
Router B
#2
Ib IPv6 address:
fe80::b
Legend:
: Flow of PIM-Assert messages
: Flow of multicast packets sent by the sending server

#1: PIM-Assert
Network A: preference = 101, metric = 1024, IPv6 address = fe80::a
#2: PIM-Assert
Network A: preference = 101, metric = 1024, IPv6 address = fe80::b

30.4.5 Determining and running the DR

If multiple IPv6 PIM-SM routers exist on the same LAN, a forwarding delegate router (DR) for the LAN is determined. The router with the highest IPv6 link-local address becomes the DR. The DR reports group-joining information from the receiving host and reports it to the rendezvous point. The DR IPv6-encapsulates multicast packets sent by the sending server, and then sends them to the rendezvous point. The following figure shows DR operation.

Figure 30-15: DR operation



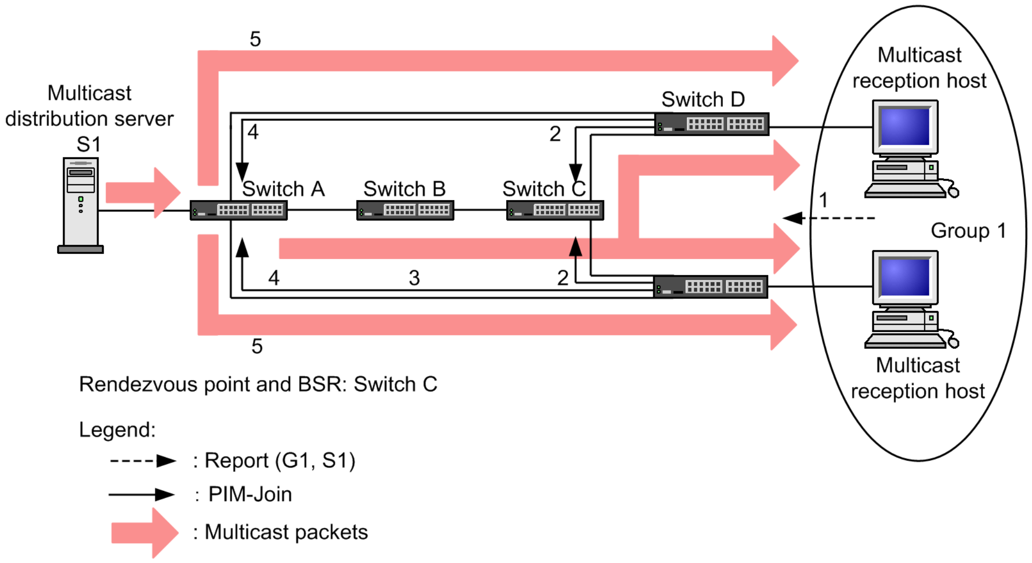
The IPv6 addresses of PIM-SM router A and PIM-SM router B are compared, and if the address of PIM-SM router B is higher, PIM-SM router B becomes the DR and reports group-joining information to the rendezvous point. Similarly, the IPv6 addresses of PIM-SM router D and PIM-SM router E are compared, and if the address of PIM-SM router E is higher, PIM-SM router E becomes the DR and forwards IPv6-encapsulated packets to the rendezvous point.

30.4.6 IPv6 PIM-SM operation when MLDv2 is used

IPv6 PIM-SM operation is as follows when the multicast distribution server (source address S1) sends multicast packets to multicast group G1, and hosts join the group via MLDv2:

1. Hosts send an MLDv2 Report (G1, S1) message as a request for joining the multicast group.
2. Switches receiving the MLDv2 Report (G1, S1) send a PIM-Join message that includes the group address (G1) to the rendezvous point over the shortest path.
3. The rendezvous point receiving the PIM-Join message learns about the existence of the group. A delivery tree from the sender via the rendezvous point is formed so that multicast packets can be delivered from the sender network to each group member via the rendezvous point.
4. The shortest path from the sender to each group member is determined by using the existing unicast routing information (PIM-Join messages are sent to the sender over the shortest path to the sender to form a shortest-path delivery tree).
5. Switches receiving multicast packets sent by multicast distribution server S1 bound for group G1 forward the multicast packets based on the shortest path delivery tree.

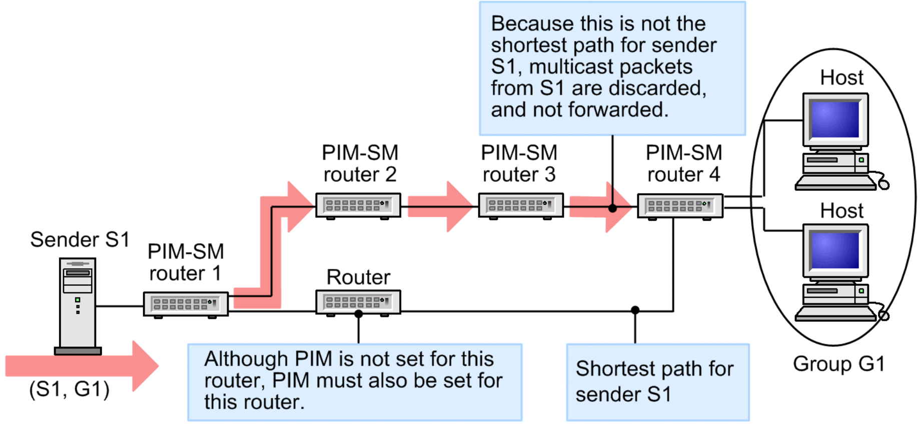
Figure 30-16: Overview of IPv6 PIM-SM operation when MLDv2 is used



30.4.7 Notes on redundant routes

Keep in mind that IPv6 multicast packets are not forwarded in redundant configurations such as that shown in the following figure. IPv6 PIM-SIM settings are required for all routers on any redundant routes.

Figure 30-17: Notes on redundant paths


Because this is not the shortest path for sender S1, multicast packets from S1 are discarded, and not forwarded.
Host
PIM-SM router 2
PIM-SM router 3
PIM-SM router 4
Host
Sender S1
PIM-SM router 1
Router
(S1, G1)
Although PIM is not set for this router, PIM must also be set for this router.
Shortest path for sender S1
Group G1

30.4.8 IPv6 PIM-SM timer specification

The following table describes the timer values used by IPv6 PIM-SM.

Table 30-9: IPv6 PIM-SM timers

Timer name	Description	Defaul tvalue (in secon ds)	Range of values that can be set in configuration mode (in seconds)	Remarks
Hello-Period	Hello sending interval	30	5 to 3600	--
Hello-Holdtime	Adjacency retention period	105	3.5 x *Hello-Period*	Calculated based on the formula to the left.
Assert-Timeout	Assert-based forwarding suppression period	180	--	--
Join/Prune-Period	Join/Prune sending interval	60	30 to 3600	A variance as high as +50% might occur.
Join/ Prune-Holdtime	Retention period for routing information and forwarding destination interfaces	210	3.5 x *Join/ Prune-Period*	Calculated based on the formula to the left.
Deletion-Delay-Ti me	Retention period for multicast forwarding destination interfaces after a Prune message is received[#1]	1/3 x *Join/ Prune- Holdti me*	0 to 300	[#2]
Data-Timeout	Forwarding entry retention period	210	0 (infinite) or 60 to 43200	A gap of as much as +90 seconds might occur.
Register-Suppressi on-Timer	Encapsulation sending suppression period	60	--	A variance as high as 30 seconds might occur.
Probe-Time	Time for sending restart checks for encapsulation sending	5	5 to 60	With the default of 5 seconds, a restart check (Null-Register) for encapsulation sending is sent once, 5 seconds before Register-Suppression-Tim er is up.[#3]
C-RP-Adv-Period	Notification interval for rendezvous point candidates	60	--	--
RP-Holdtime	Rendezvous point retention period	150	2.5 x *C-RP-Adv-Peri od*	Calculated based on the formula to the left.
Bootstrap-Period	BSR message send interval	60	--	--
Bootstrap-Timeout	BSR message retention period	130	2 x *Bootstrap-Peri od* + 10	Calculated based on the formula to the left.
Negative-Cache-H oldtime(PIM-SM)	Negative cache retention period	210	10 to 3600	For PIM-SSM, this is fixed at 10 to 3600

Legend: --: Not applicable

#1

A configured timer value is used if it set in the configuration, however, to the forwarding destination interface, set a value that does not exceed the Join/Prune-Holdtime value included in the PIM-Join/Prune message received with the last Join as the retention period in the interface.

#2

Because the value set in the configuration is given priority for this timer value, operation differs from the standard in RFC 2362. However, if no value is specified in the configuration, operation complies with RFC 2362.

#3

If this timer value is set to 10 or more, the restart check for encapsulation sending is sent multiple times every five seconds. If no value is specified in the configuration, the check is sent only once.

30.4.9 Notes on using IPv6 PIM-SM

Keep the following limitations in mind when configuring a network that uses IPv6 PIM-SM.

The Switch conforms to RFC 2362 (the PIM-SM specification), but differs from some parts of the RFC due to software functionality limitations. The following table describes the differences with the RFC.

Table 30-10: Differences with the RFC

RFC		Switch
Packet format	The RFC contains a field for setting a mask length for the encoding group address and encoding source address.	The mask length of encoding addresses is fixed at 128.
	The RFC contains fields for setting the address family and encoding type for encoding group addresses and encoding source addresses.	The address family for encoding addresses is fixed at 2 (IPv6), and the encoding type is fixed at 0 for the Switch. Connections cannot be established outside of IPv6 to PIM-SM.
	The RFC contains a field for setting the PIM version of a PIM message header.	The PIM version is fixed at 2. Connections to PIM version 1 cannot be established.
Join/Prune fragments	Join/Prune messages can be fragmented even when they exceed the network MTU.	If the size of a Join/Prune message to be sent is large, the Switch splits it into 8-KB parts before sending. Also, Join/Prune messages split before sending are sent by using IP fragments based on the network MTU length.
Connections to PMBR	The RFC contains a specification for connections to PMBRs (PIM Border Routers) and for (*, *, RP) entries.	The Switch does not support connections to PMBR. Also, (*, *, RP) entries are not supported.
Switching to shortest paths	The RFC contains a switching method based on the data rate as an example of the timing for switching to the shortest path.	When the Switch first receives data for the last-hop-router, it switches to the shortest path without checking the data rate.

RFC		Switch
Receiving C-RP-Adv and sending Bootstrap	A Bootstrap message is permitted to be fragmented if the length of a generated message exceeds the maximum packet length. However, to suppress occurrences of fragmentation, the maximum number of rendezvous point candidates should be set.	Only one BSR is allowed per system. In addition, the maximum number of group prefixes that can be set at the rendezvous point is 128. For the Switch, if the size of a Bootstrap message to be sent is large, it is IP-fragmented by MTU length and then sent.
Hello message option	The HoldTime option (type 1) is defined in the RFC.	In addition to the HoldTime option, the neighboring-router address list option (types 24 and 65001) is available. (See *30.4.3 Detecting neighbors*.)

30.4.10 IPv6 PIM-SSM

PIM-SSM is extended PIM-SM functionality. PIM-SM and PIM-SSM can run at the same time. The multicast addresses used by PIM-SSM are assigned by the IANA. The Switches allow the address range of multicast addresses (group addresses) for which PIM-SSM runs to be specified by configuration. PIM-SM runs on addresses other than those specified.

Whereas PIM-SM requires multicast forwarding packets when creating multicast entries, PIM-SSM creates IPv6 multicast forwarding entries by exchanging multicast routing information (PIM-Join), and forwards multicast packets based on the corresponding entries. Note that PIM-SSM does not require a rendezvous point or bootstrap router. This means that packet encapsulation and decapsulation are not needed when multicast packets are forwarded, enabling more efficient multicast forwarding. The Switch provides measures that enable PIM-SSM to run by using MLD.

(1) Supported specifications for IPv6 PIM-SSM messages

The supported specifications for IPv6 PIM-SM messages are the same as those for PIM-SM messages.

(2) Prerequisites for running IPv6 PIM-SSM

The following settings must be specified in configuration mode for the Switch:

- Settings for each switch

  Set the range of group addresses for which PIM-SSM runs.

- Switches directly connected to hosts running MLD

  Set the addresses of the groups and the sender for which PIM-SSM runs for MLD reception.

(3) IPv6 PIM-SSM operation (for MLDv1 or MLDv2 (EXCLUDE mode) hosts)

Sender information is needed to use PIM-SSM. The Switch allows PIM-SSM to be used by setting the sender in configuration mode if MLDv1 is used.

IPv6 PIM-SSM operation is as follows when the multicast distribution server (source address S1) sends multicast packets to multicast group G1:

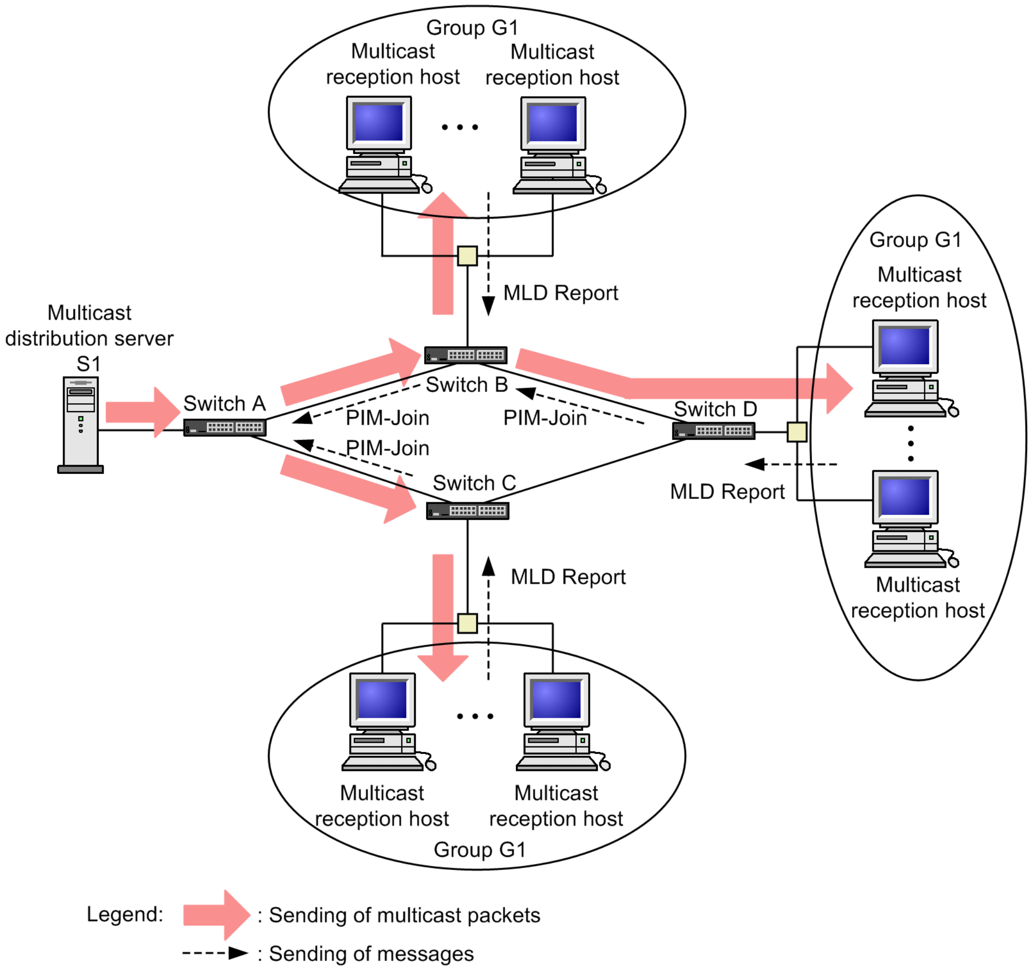
1. Hosts send an MLD Report (G1) message as a request for joining the multicast group.
2. The switch receiving the MLD Report message compares the group address (G1) reported in the message and the group address set in configuration mode. If these group addresses match, a PIM-Join message is sent to the source address (S1) set in configuration over the shortest path (determined by the unicast routing information). In this case, the PIM-Join message contains information about the source address (S1) and group address (G1). Each switch receiving the PIM-Join message sends it hop-by-hop to the source address (S1) over the shortest path. Switches receiving the PIM-Join message learn IPv6 multicast routing

information for the source address (S1) and group address (G1).

3. The multicast packet distribution server (S1) sends multicast packets bound for group 1 (G1). Switches receiving multicast packets forward the packets based on the multicast forwarding entries generated by learned IPv6 multicast routing information.

The following figure provides an overview of IPv6 PIM-SSM operation.

Figure 30-18: Overview of IPv6 PIM-SSM operation (for MLDv1 or MLDv2 (EXCLUDE mode) hosts)



(4) IPv6 PIM-SSM operation (for MLDv2 (INCLUDE mode) hosts)

Sender information is needed to use PIM-SSM. For MLDv2, PIM-SSM can be used by specifying the sender in a Report message.

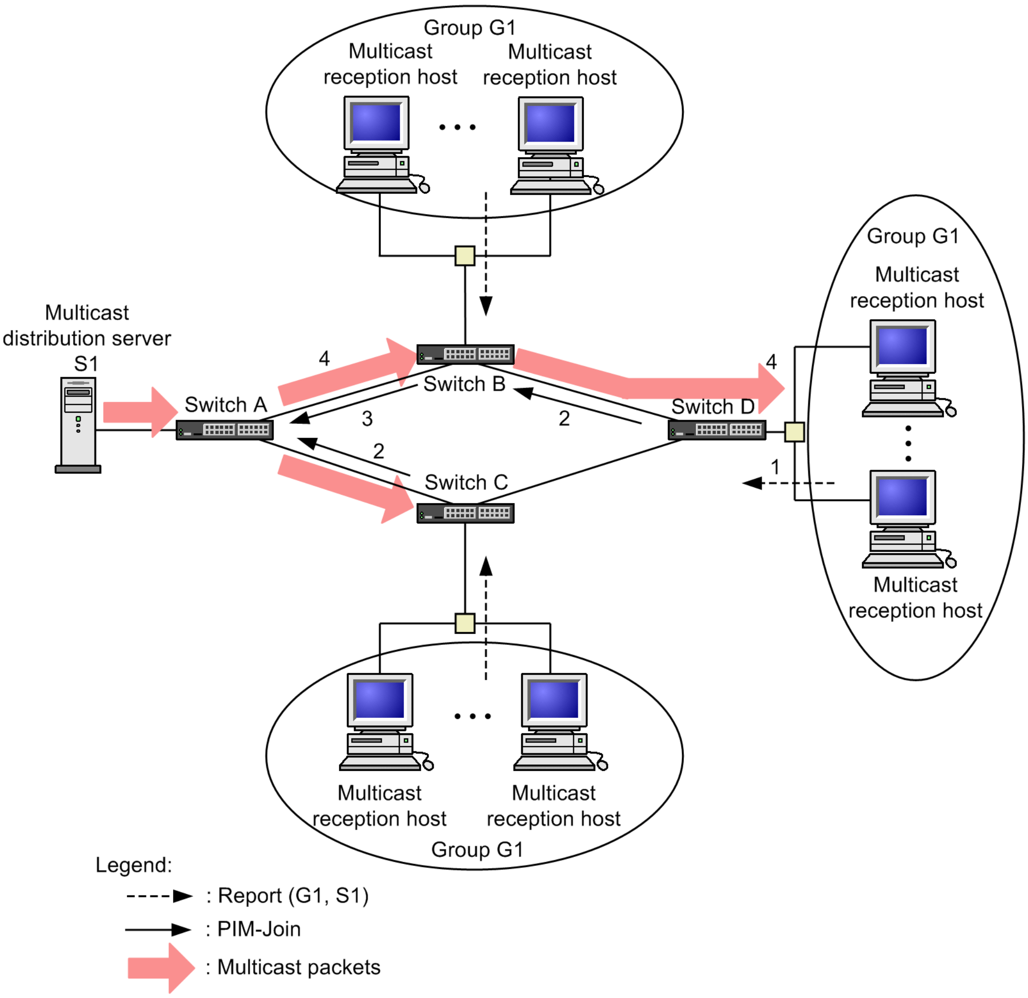
IPv6 PIM-SSM operation is as follows when the multicast distribution server (source address S1) sends multicast packets to multicast group G1:

1. Hosts send an MLDv2 Report (G1, S1) message as a request for joining the multicast group.
2. The switches receiving the MLDv2 Report (G1, S1) message send a PIM-Join message that contains the group address G1 and source address S1 that were contained in the Report message.
3. The switches receiving the PIM-Join message forward the message to source address S1 hop-by-hop over the shortest path. These switches form an S1-to-G1 delivery tree so that multicast packets from source address S1 are forwarded only to the interfaces from which the PIM-Join message was received.

4. Switches receiving multicast packets sent by multicast distribution server S1 bound for group G1 forward the multicast packets based on multicast forwarding information.

The following figure provides an overview of IPv6 PIM-SSM operation.

Figure 30-19: Overview of IPv6 PIM-SSM operation (for MLDv2 (INCLUDE mode) hosts)



(5) IPv6 path control when MLDv1 hosts and MLDv2 hosts are mixed

The following explains IPv6 path control operation when PIM-SSM is set to be used on MLDv1, and both MLDv1 hosts and MLDv2 hosts are used.

If a request to join a group whose address is within the PIM-SSM target address range specified in configuration mode is received, PIM-SSM is used as shown in the table below. If an MLDv1 Report message is received as a group-join request, the source addresses set in configuration mode are used as the sender list. Note, however, that both an MLDv1 Report message and an MLDv2 Report (EXCLUDE mode) message might be received as join requests for the same group address. In that case, a combination of the source addresses set in configuration mode and the sender list contained in the MLDv2 Report (INCLUDE mode) message is used as the sender list.

Table 30-11: IPv6 path control when MLDv1 hosts and MLDv2 hosts are mixed

Subscription group addresses	MLDv1 Report MLDv2 Report (EXCLUDE mode)	MLDv2 Report (INCLUDE mode)
Within the SSM address range	PIM-SSM	PIM-SSM

Subscription group addresses	MLDv1 Report MLDv2 Report (EXCLUDE mode)	MLDv2 Report (INCLUDE mode)
Outside the SSM address range	PIM-SM	PIM-SM

(6) Detecting neighbors

Operation is the same as for PIM-SM (see *30.4.3 Detecting neighbors*).

(7) Determining the forwarder

Operation is the same as for PIM-SM (see *30.4.4 Determining the forwarder*).

(8) Determining and running the DR

Operation is the same as for PIM-SM (see *30.4.5 Determining and running the DR*).

(9) Notes on redundant routes

Operation is the same as for PIM-SM (see *30.4.7 Notes on redundant routes*).

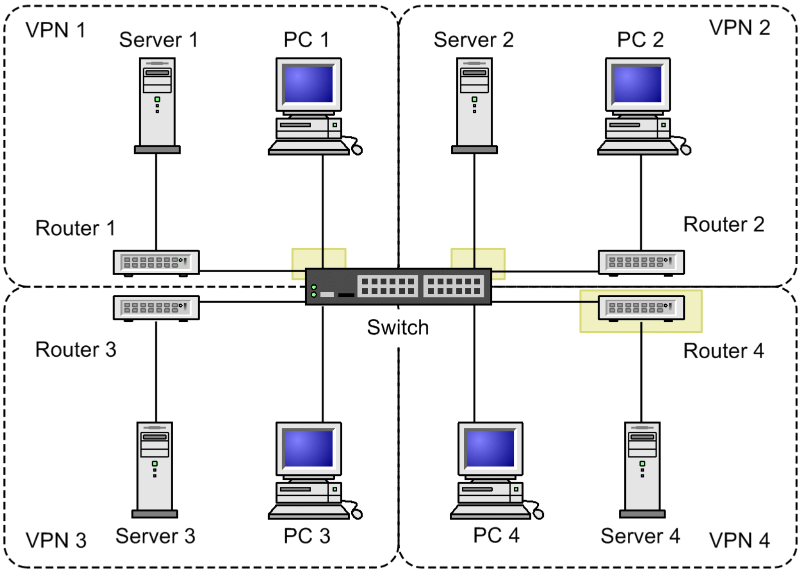
30.4.11 IPv6 multicasting for a VRF [OS-L3SA]

(1) IPv6 multicasting for a VRF

You can connect the Switch to multiple VPNs and use IPv6 multicasting in each VPN. Configure VRF for each VPN and run IPv6 multicasting in each VRF. You can set different rendezvous points, BSRs, timers, and SSM address ranges for VRF-based IPv6 multicasts.

The figure below shows an example configuration in which the Switch is connected to four VPNs. The settings on the Switch in this example are given in the table.

Figure 30-20: IPv6 multicasting for a VRF



Legend: ▨ : Rendezvous point and BSR

Table 30-12: Settings on the Switch

VPN	Operation protocol	Loopback address	Rendezvous point (The values in parentheses are rendezvous point addresses)	SSM address
1	PIM-SM	2001:db8::1	Switch (2001:db8::1)	Not used

VPN	Operation protocol	Loopback address	Rendezvous point (The values in parentheses are rendezvous point addresses)	SSM address
2	PIM-SM/ PIM-SSM	2001:db8::2	Switch (2001:db8::2)	ff30::/12
3	PIM-SSM	2001:db8::2	None	ff35:100::/32
4	PIM-SM	2001:db8::3	Router 4 (2001:db8::1)	Not used

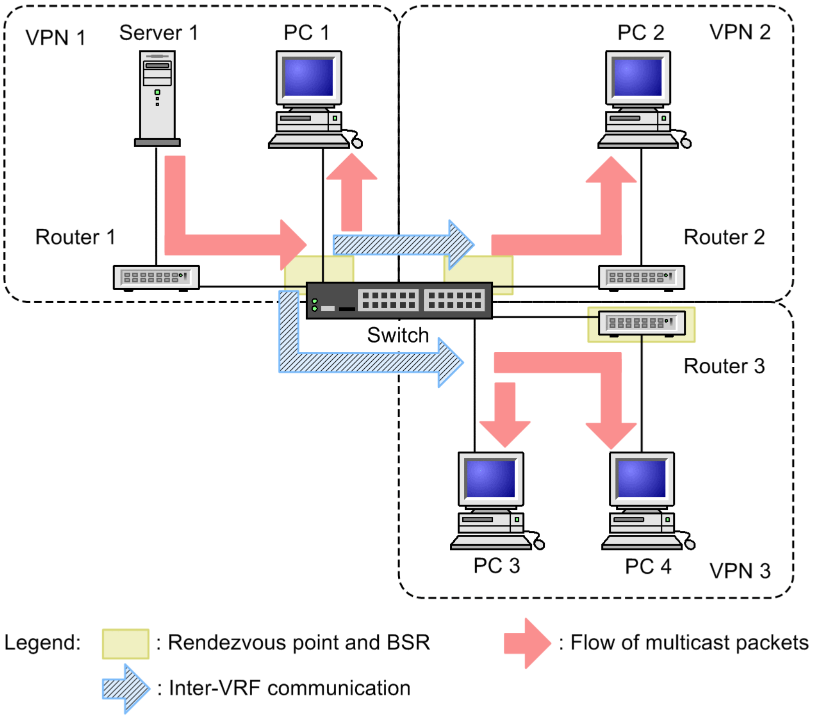
(2) IPv6 multicast extranets

If an IPv6 multicast extranet is used, IPv6 multicast forwarding can be performed between VRFs. In addition, if IPv6 multicast route filtering is used, you can limit the range of group addresses used in the extranet and the VRFs for which forwarding requests from downstream are permitted.

Note that a unicast route to the sender must exist in the unicast extranet to establish the shortest path from the last-hop-router.

The following figure shows the overview of IPv6 multicast extranet operation.

Figure 30-21: Overview of IPv6 multicast extranet operation


VPN 1
Server 1
PC 1
PC 2
VPN 2
Router 1
Router 2
Switch
Router 3
PC 3
PC 4
VPN 3
Legend:
: Rendezvous point and BSR
: Flow of multicast packets
: Inter-VRF communication

(3) PIM-SM VRF gateways

To perform multicast communication by using the PIM-SM protocol, IPv6 multicast packets must be forwarded to the last-hop-router. If PIM-SM is used in an extranet environment, IPv6 multicast packets must also be forwarded to all the last-hop-routers in the VRFs.

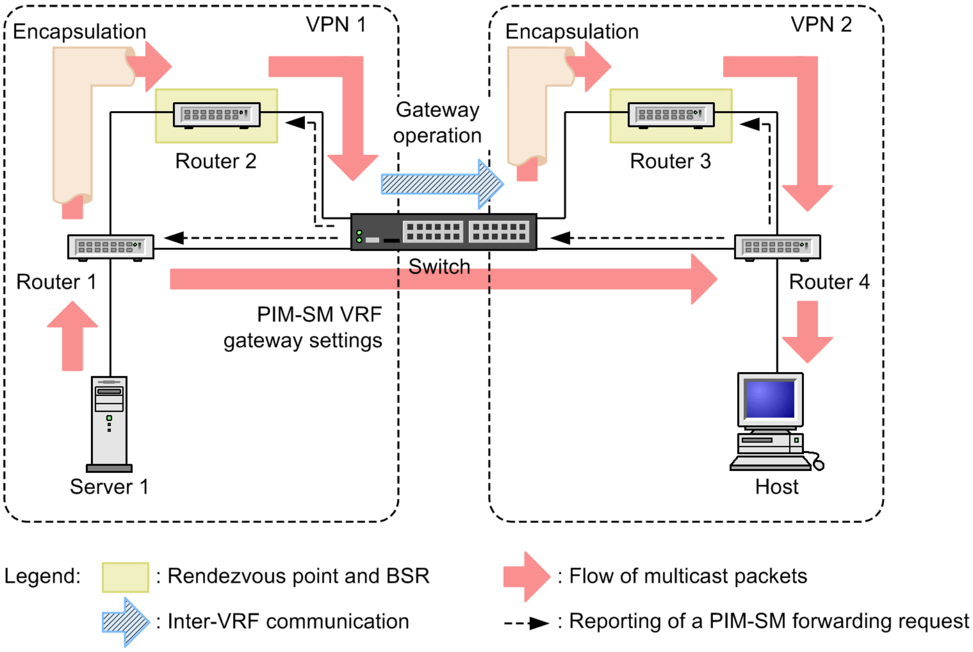
Use a PIM-SM VRF gateway to enable the Switch to forward IPv6 multicast packets to the rendezvous point of each VRF.

The PIM-SM VRF gateway set in the VRF containing the sender (multicast server) operates as the last-hop-router for the specified group in the target VPN, and requests the rendezvous point to forward packets. Upon receiving IPv6 multicast packets from the rendezvous point, the gateway forwards the packets to the destination VRFs.

The PIM-SM VRF gateway in a destination VPN operates as the first-hop-router. The gateway encapsulates IPv6 multicast packets, and sends the encapsulated packets (called register packets) to the rendezvous point of each VRF. When a rendezvous point receives register packets, the rendezvous point decapsulates the packets and transfers the decapsulated IPv6 multicast packets to the last-hop-router, the same way as in normal PIM-SM operation. Next, the shortest-path delivery tree from the last-hop-router to the sender is generated. At the same time, IPv6 multicast packets in the extranet are forwarded by hardware.

As described above, use of PIM-SM VRF gateways allows a PIM-SM-based extranet to be created without changing the settings of any components other than the Switch. The following figure shows an overview of PIM-SM VRF gateway operation.

Figure 30-22: Overview of PIM-SM VRF gateway operation


Encapsulation
VPN 1
Encapsulation
VPN 2
Gateway operation
Router 2
Router 3
Switch
Router 1
Router 4
PIM-SM VRF gateway settings
Server 1
Host
Legend:
: Rendezvous point and BSR
: Inter-VRF communication
: Flow of multicast packets
: Reporting of a PIM-SM forwarding request

(4) Notes on using IPv6 multicast extranets

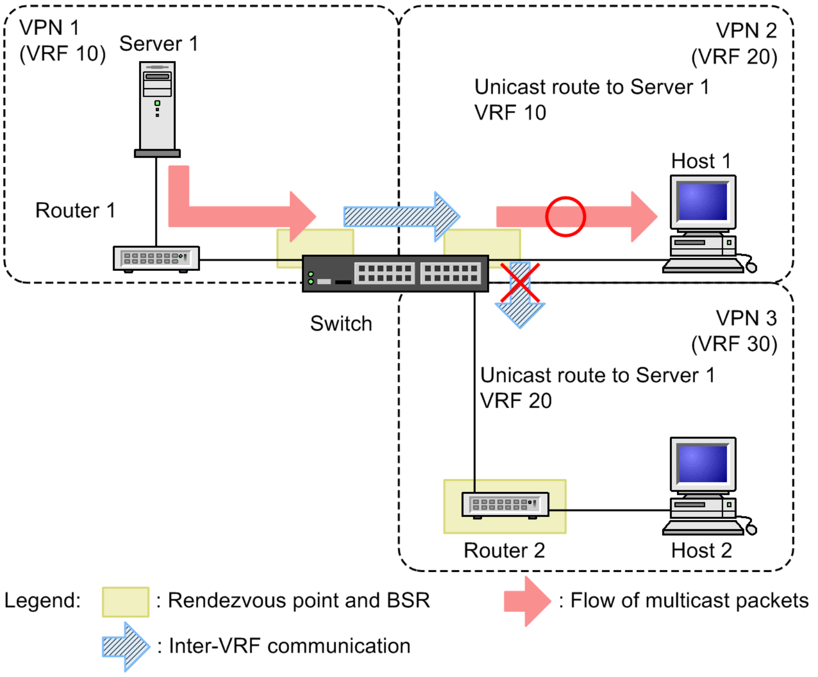
(a) Multi-level VRF forwarding with one device

IPv6 multicast extranets do not permit multi-level VRF forwarding with one device.

For an IPv6 multicast route of a VRF, the same VRF must be used for upstream and downstream interfaces. Any forwarding requests from upstream multicast routing information with a different VRF are ignored. When downstream interfaces of a multicast route use one VRF, and the VRF of an upstream interface is switched to another VRF, the downstream interfaces are detached from the target multicast route.

As indicated in the following figure, when there is a unicast route from VPN 3 to VPN 1 via VPN 2, host 1 in VPN 2 can receive IPv6 multicast packets from server 1 in VPN 1, but host 2 in VPN 3 cannot.

Figure 30-23: Example of an IPv6 multicast extranet in which VRF forwarding is not permitted with one device


VPN 1
(VRF 10)
Server 1
Router 1
Switch
VPN 2
(VRF 20)
Unicast route to Server 1
VRF 10
Host 1
VPN 3
(VRF 30)
Unicast route to Server 1
VRF 20
Router 2
Host 2
Legend:
: Rendezvous point and BSR
: Flow of multicast packets
: Inter-VRF communication

(b) Interconnection of PIM-SM and PIM-SSM

If you want to use an IPv6 multicast extranet to perform multicast forwarding between VRFs, make sure that all group addresses used in the multicast forwarding use the same protocol (for IPv6 multicast VRFs, the group addresses to be used with PIM-SSM can be specified for each VRF).

In an IPv6 multicast extranet, multicast forwarding between different protocols is impossible. The figure below shows an example in which inter-VRF forwarding cannot be performed because the protocols do not match. The settings on the Switch in this example are given in the table that follows.

Figure 30-24: Example of an IPv6 multicast extranet in which inter-VRF forwarding cannot be performed because the protocols do not match

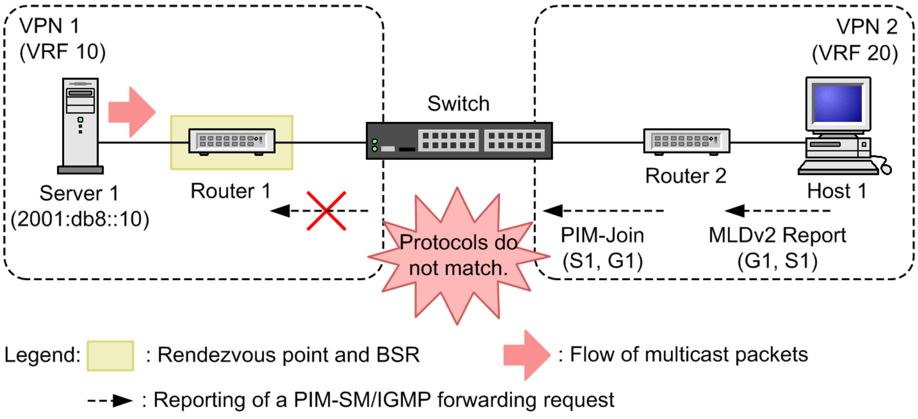


Table 30-13: Settings on the Switch

VPN	Operation protocol	Rendezvous point (The values in parentheses are rendezvous point addresses)	SSM address
1	PIM-SM	Switch (2001:db8::100)	Not used
2	PIM-SSM	None	ff30::/12

G1: ff35::1
S1: 2001:db8::10

30.5 Network design considerations

30.5.1 IPv6 multicast forwarding

Keep the following in mind when using the Switch to forward IPv6 multicast packets.

(1) Notes applying to both IPv6 PIM-SM and IPv6 PIM-SSM

(a) Stopped forwarding due to a routing program restart

For the Switch, when the `restart ipv6-multicast` command is used to restart the IPv6 multicast routing program, IPv6 multicast communication is stopped until multicast routing information is relearned.

(b) Point-to-point lines

To perform IPv6 multicast communication over a point-to-point line for which a unicast static path is set, make sure that you explicitly specify the connection destination address (gateway).

(c) Chronological packet overtaking

When the Switch receives multicast data from a sender and a PIM-Join message from the receiver at the same time, some packets might overtake others due to timing, causing the order of the packets to be switched.

(2) IPv6 PIM-SM

Keep the following in mind when using IPv6 PIM-SM.

(a) Packet loss during software forwarding

When the Switch receives the first IPv6 multicast packet, it sets an IPv6 multicast forwarding entry in hardware so that the forwarding of IPv6 multicast packets is performed by hardware. Note, however, that because software performs the forwarding until the entry has been created in hardware, some packets might be temporarily lost.

(b) Packet overtaking during a switchover from software forwarding to hardware forwarding

The Switch stops forwarding IPv6 multicast packets by software, and starts forwarding IPv6 multicast packets by hardware, when an IPv6 multicast forwarding entry has been set in hardware. Some packets might be overtaken at this point, causing the order of packets to change.

(c) Duplicate forwarding or packet loss during path switching

In the Switch, duplicate forwarding or packet loss might occur temporarily during forwarding of IPv6 multicast packets via the rendezvous point or during a switchover from forwarding via the rendezvous point to forwarding over the shortest path.

For details about IPv6 multicast packet forwarding via the rendezvous point and the switchover from forwarding via a rendezvous point to forwarding over the shortest path, see *30.4.2 IPv6 PIM-SM*.

(d) When the device address setting is required

If the Switch is used as the first-hop router, the IPv6 address set as the local address in the device management information is used for communication up to the rendezvous point. Therefore, when IPv6 PIM-SM is used, unlike when PIM-SM is used, the device address of the switch must also be set even if the switch is not a rendezvous point or BSR.

(e) Device address reachability

If the Switch is used as a rendezvous point and bootstrap router, the IPv6 address set as the local address in the device management information is used as the address for the rendezvous point and bootstrap router. The local address in this device management information must be reachable for route recognition and communication via unicast for all switches that use IPv6 multicast communication.

(f) Static rendezvous point

Static rendezvous point functionality allows the rendezvous point to be specified without using a BSR. The static rendezvous point is set in configuration mode.

The static rendezvous point can also exist with rendezvous point candidates advertised by Bootstrap messages from a BSR. In that case, the static rendezvous point is given priority over the rendezvous point candidates advertised by Bootstrap messages from a BSR.

If a rendezvous point candidate router recognizes that its local address is the rendezvous point router address, the router functions as the rendezvous point. Therefore, if a network using the static rendezvous point is designed without using a BSR, the static rendezvous point settings must also be specified on all rendezvous point candidate routers.

Also, if the static rendezvous point is used, the same settings need to be specified for all routers on the same network.

30.5.2 Redundant paths (path switching due to failure and other reasons)

The following explains what to keep in mind for the Switch when IPv6 multicast paths are redundant.

(1) Using IPv6 PIM-SM

Keep in mind that for IPv6 PIM-SM, it might take some time for IPv6 multicast communication to restart for the following route switching. In the following time indications, the switching time for sender network information (unicast routing information) is represented as *U*.

The time shown here indicates how long it takes the Switch to perform switching. Because of this, join notification time (from the connection request from the Switch to the upstream router to the arrival of the multicast data from the upstream) is required to actually restart multicast forwarding.

- When a prioritized route is switched, the following time might be required until communication restarts:

  U + 20 seconds

- When a prioritized route is switched to a redundant route due to line failure, the following time is required until communication restarts:

  For *U* < 5: 5 to 10 seconds
  For *U* >= 5: *U* + 0 to 135 seconds

- When a redundant route is switched to a prioritized route due to line restoration, the following time might be required until the prioritized route is used in communication:

  0 seconds

  The following time is necessary to perform switching back.
  U + (*PIM-Hello-message-sending-interval-for-sender-direction* + 20) seconds
  (Default: *U* + 30 + 20 = *U* + 50 seconds)

- When a rendezvous point and BSR are switched to the Switch (the Switch becomes the rendezvous point and BSR, such as due to failure or configuration), the following time is required until communication restarts.

  The time until communication restarts differs depending on the rendezvous point or BSR. The default values are shown in parentheses.

- For rendezvous point switching: 285 seconds

  RP-Holdtime `(150 seconds) +` *Query-interval* `(125 seconds) +` *Query Response Interval* `(10 seconds)`

- For BSR switching: At most 385 seconds

  Bootstrap-Timeout `(130 seconds) +` *BS_Rand_Override* `(0 to 60 seconds) +`
  Bootstrap-Period `(60 seconds)`
  `+` *Query-interval* `(125 seconds) +` *Query Response Interval* `(10 seconds)`

- When a DR is switched to the Switch, the following time is required until communication restarts. The default values are shown in parentheses.
  - For DR switching: 240 seconds

    Hello-Holdtime `(105 seconds) +` *Query-interval* `(125 seconds) +` *Query Response Interval* `(10 seconds)`

IPv6 multicast communication might stop at these times not only for redundant route switching due to failure, but also when path switching is performed explicitly due to a configuration change. Plan ahead when changing the system configuration.

Especially when switching the device that is running the rendezvous point and BSR, make sure that you set a higher priority value on the new device in configuration mode than the priority value of the current device.

(2) Using IPv6 PIM-SSM

Keep in mind that for IPv6 PIM-SSM, it might take some time for IPv6 multicast communication to restart for the following path switching. In the following time indications, the switching time for sender network information (unicast routing information) is represented as U.

The time shown here indicates how long it takes the Switch to perform switching. Because of this, join notification time (from the connection request from the Switch to the upstream router to the arrival of the multicast data from the upstream) is required to actually restart multicast forwarding.

- When a prioritized route is switched, the following time might be required until communication restarts:

  U `+ 20 seconds`

- When a prioritized route is switched to a redundant route due to line failure, the following time is required until communication restarts:

  `For` *U* `< 5: 5 to 10 seconds`
  `For` *U* `>= 5:` *U* `+ 0 to 135 seconds`

- When a redundant route is switched to a prioritized route due to line restoration, the following time might be required until the prioritized route is used in communication:

  `0 seconds`

  The following time is necessary to perform switching back.
  U `+ (`*PIM-Hello-message-sending-interval-for-sender-direction* `+ 20) seconds`
  `(Default:` *U* `+ 30 + 20 =` *U* `+ 50 seconds)`

- When a DR is switched to the Switch, the following time is required until communication restarts. The default values are shown in parentheses.
  - For DR switching: 240 seconds

Hello-Holdtime `(105 seconds) +` *Query-interval* `(125 seconds) +` *Query Response Interval* `(10 seconds)`

30.5.3 Examples of appropriate network configurations

(1) Configuration using IPv6 PIM-SM

This configuration is appropriate in the following cases:

- The number of users sending multicast packets is not limited.
- The number of users sending multicast packets is large.

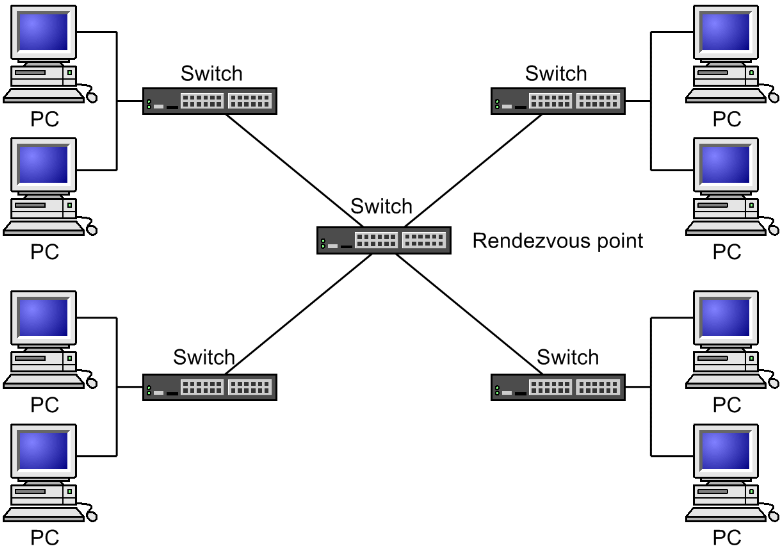
Network environment

1. The IPv6 unicast routing protocol must run on all routers.
2. IPv6 PIM-SM is used as the IPv6 multicast routing protocol between the Switches.
3. Either MLDv1 or MLDv2 is used for communication between a group and the Switch.
4. One switch is used as the rendezvous point and BSR.

Configuration diagram

The following diagram shows a configuration.

Figure 30-25: Configuration using IPv6 PIM-SM



(2) Configuration using IPv6 PIM-SSM

This configuration is appropriate in the following cases:

- The number of users sending multicast packets is limited (such as for distribution servers).
- Users receiving multicast can specify the address of the MLDv2-compatible sending server.
- Broadband multicast communication is used.
- Multi-channel multicast communication is used.

Network environment

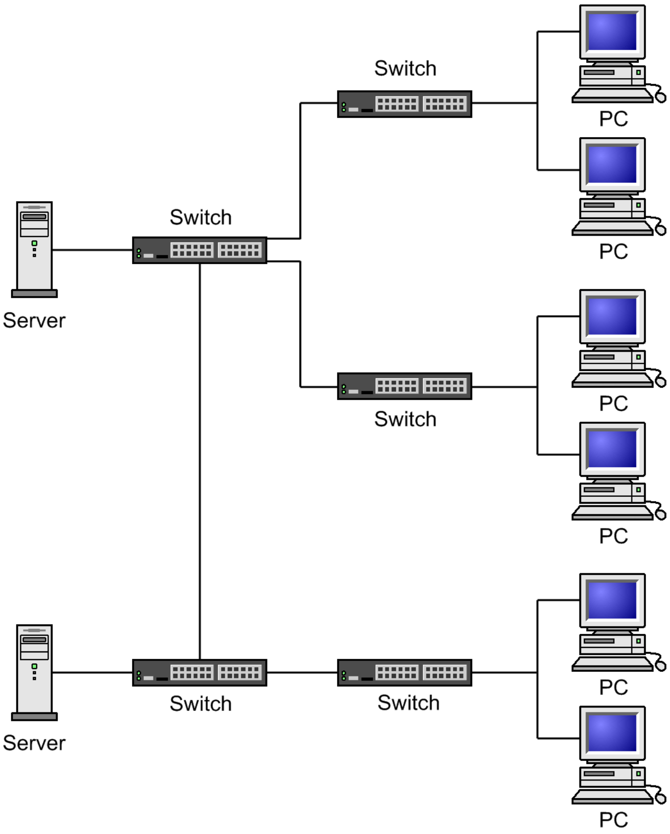
1. The IPv6 unicast routing protocol must run on all routers.
2. IPv6 PIM-SSM is used as the IPv6 multicast routing protocol between the Switches. IPv6 PIM-SSM is extended PIM-SM functionality.
3. MLDv1 or MLDv2 is used for group management control between the Switch and a

group (the settings for linking with SSM by using MLDv1 must be specified).

Configuration diagram

The following diagram shows a configuration.

Figure 30-26: Configuration using IPv6 PIM-SSM



30.5.4 Notes about network configurations

IPv6 multicasting is appropriate for 1: N unidirectional communication in which data is distributed from one server (sender) to multiple groups (receivers). The following gives notes on network configurations appropriate for IPv6 multicasting.

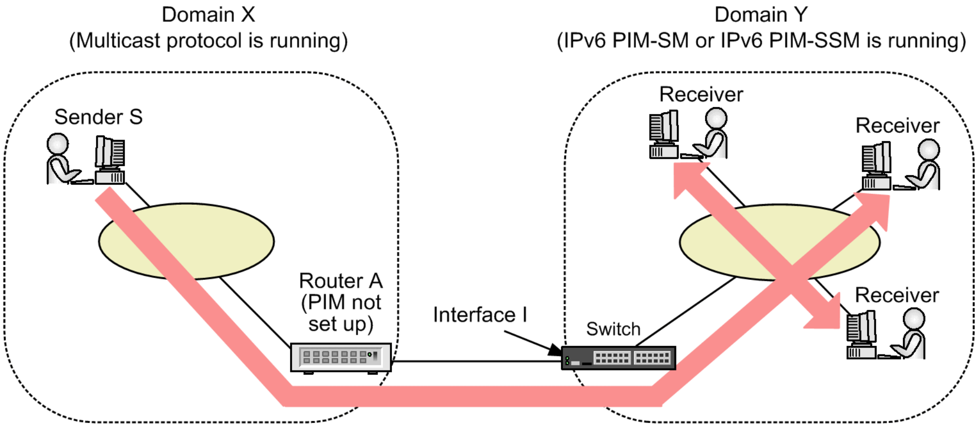
(1) Notes applying to both IPv6 PIM-SM and IPv6 PIM-SSM

(a) Applicable configuration

If IPv6 PIM-SM or IPv6 PIM-SSM (abbreviated hereafter to PIM) is used, PIM must be set up on all routers existing on the paths from the sender to receivers. If there is a router on which PIM is not set up on a path, multicast packets can be forwarded no further. If PIM is not set up on neighboring routers, packet forwarding becomes possible by executing the `ipv6 pim direct` configuration command.

Figure 30-27: Example of an applicable configuration in which the ipv6 pim direct configuration command is used shows an example configuration in which the `ipv6 pim direct` configuration command is applied. Router A and the Switch do not communicate via PIM because they belong to different multicast domains. In this example, sender S in domain X wants to send multicast data to the receivers in domain Y. However, Router A and the Switch do not communicate via PIM, so the multicast data sent by Sender S is discarded at the Switch. In such situations, if the `ipv6 pim direct` configuration command is used to set sender S for interface I of the Switch, multicast packets can be forwarded inside of domain Y.

Figure 30-27: Example of an applicable configuration in which the ipv6 pim direct configuration command is used



The `ipv6 pim direct` configuration command is applicable to configurations such as the one shown in the above figure. If the command is used in other configurations, forwarding multicast packets might no longer be possible.

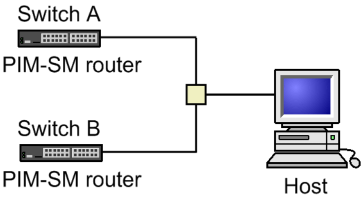
(b) Configurations requiring special attention

Use caution if using IPv6 PIM-SM or IPv6 PIM-SSM in the following configurations:

- In a configuration such as the one in the figure below, multiple routers directly connect on the host on the same network. For such configurations, make sure that PIM-SM always runs at the interface.

  If only MLD runs and PIM-SM does not run at an interface on which multiple routers exist on the same network, multicast data might be forwarded twice.

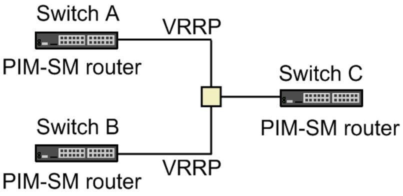
  Figure 30-28: Configuration requiring special attention (when connecting multiple routers to a host)



- In a configuration, such as the one in the figure below, in which the PIM protocol cannot detect the upstream router, multicast communication cannot be performed. This is because the environment has a static route for which the virtual interface set as the gateway has VRRP set by Switch C for Switch A and for Switch B. (This also applies when PIM-SSM is used.)

  To perform multicast communication in this configuration, a static route needs to be set for Switch C, with the real address of Switch A or Switch B used as the gateway address to the rendezvous point address, BSR address, and multicast data source address.

  Figure 30-29: Configuration requiring special attention (when configuring VRRP setting)

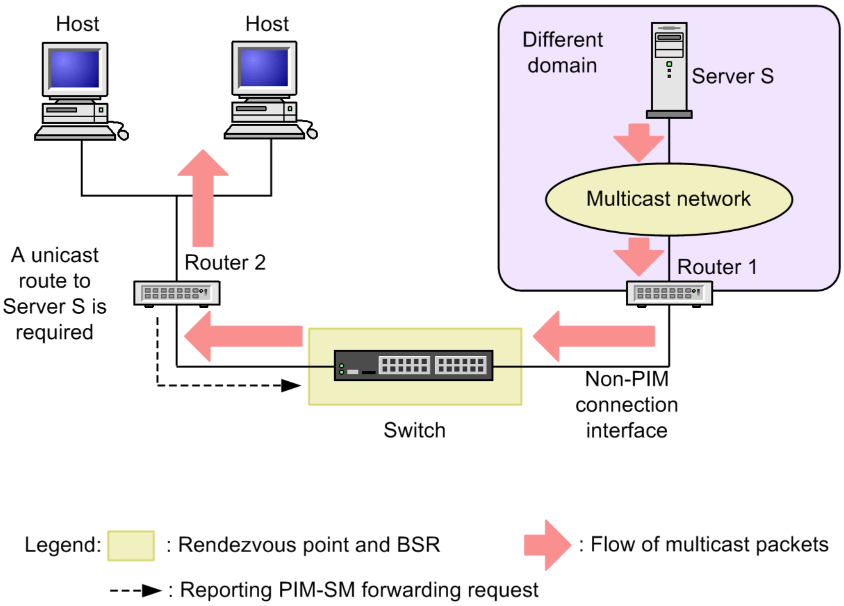


- When multicast forwarding is performed between routers in different domains without using the PIM-SM or PIM-SSM protocol, the interface between these routers is called a non-PIM connection interface.

  As shown in the figure below, when multicast packets are forwarded by multicasting from Server S in a different domain over a non-PIM connection interface, the following settings must be specified:

  - On Router 2 connected downstream to the Switch, set a unicast route to Server S.
  - For the non-PIM connection interface of the Switch, set the `ipv6 pim direct` configuration command.

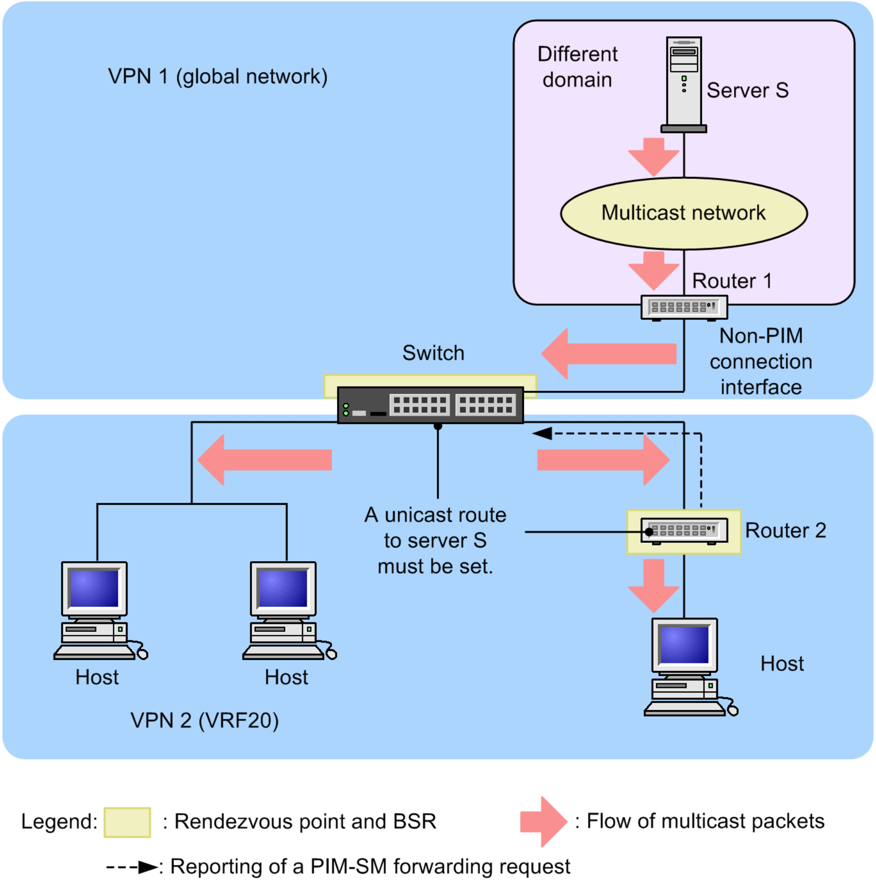
  Figure 30-30: Configurations requiring special attention (when packets are forwarded over a non-PIM connection interface)


Host
Host
Different domain
Server S
Multicast network
A unicast route to Server S is required
Router 2
Router 1
Switch
Non-PIM connection interface
Legend:
: Rendezvous point and BSR
: Flow of multicast packets
: Reporting PIM-SM forwarding request

- As shown in the figure below, when multicast packets are forwarded by multicasting to VPN2 (VRF20) from Server S in a different domain in VPN1 (global network) upstream of a non-PIM connection interface, the following settings must be specified: **[OS-L3SA]**

  - For the non-PIM connection interface of the Switch in the upstream VRF, set the `ipv6 pim direct` configuration command. In addition, configure an IPv6 multicast extranet, and then set a unicast route to Server S in the forwarding destination VRF of the Switch.
  - On the downstream router (Router 2) in the forwarding destination VPN, set a unicast route to Server S.

Figure 30-31: Configuration requiring special attention (when packets are forwarded via an extranet over a non-PIM connection interface)


VPN 1 (global network)
Different domain
Server S
Multicast network
Router 1
Non-PIM connection interface
Switch
A unicast route to server S must be set.
Router 2
Host
Host
Host
VPN 2 (VRF20)
Legend:
: Rendezvous point and BSR
: Flow of multicast packets
: Reporting of a PIM-SM forwarding request

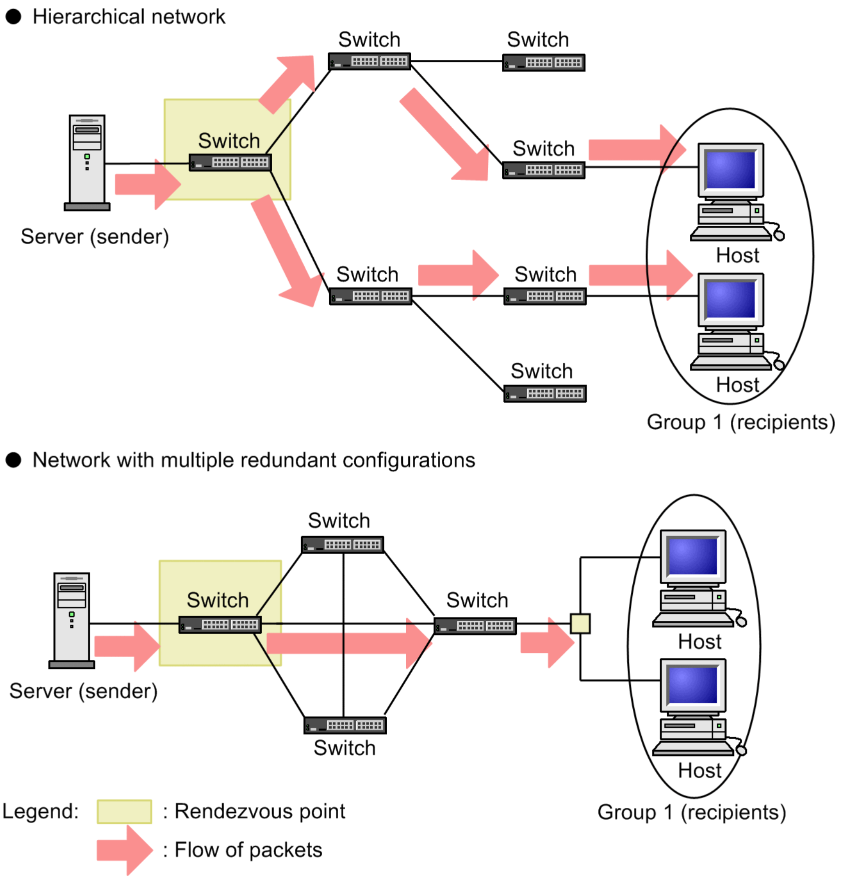
(2) IPv6 PIM-SM

(a) Recommended configurations

We recommend network configurations based on IPv6 PIM-SM for hierarchical network configurations and configurations with redundant routes. However, special care is needed for the placement of the rendezvous point. We recommend that the rendezvous point be as close as possible to the sender to reduce the processing load needed to change the IPv6 multicast sending path when the IPv6 PIM-SM mode is changed.

The following figure shows a recommended network configuration for IPv6 PIM-SM.

Figure 30-32: Recommended network configuration for IPv6 PIM-SM



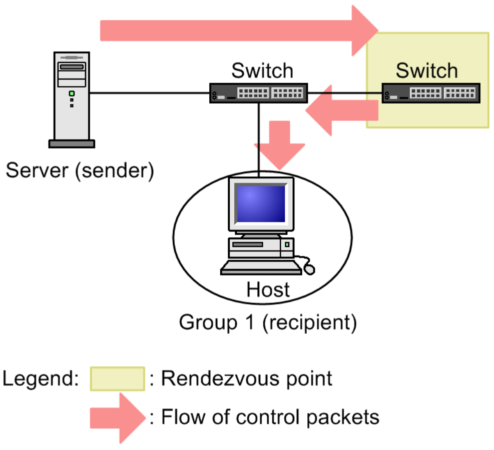
(b) Inappropriate configurations

Do not use IPv6 PIM-SM for configurations such as the following:

- Configuration in which a recipient exists between the sender and rendezvous point

  When IPv6 multicast communication with group 1 is performed from the server in the following configuration, forwarding via the rendezvous point cannot be done efficiently.

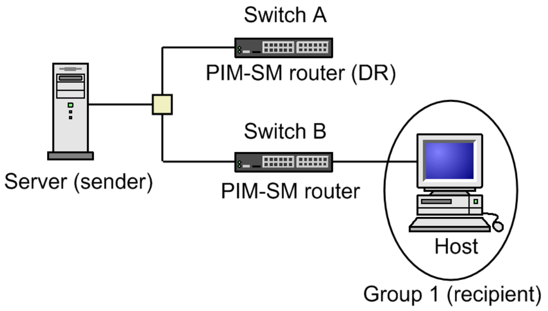
  Figure 30-33: Inappropriate configuration (when a recipient exists between a sender and a rendezvous point)



- Configuration in which multiple IPv6 PIM-SM routers are running on the same line as the sender

When the server sends IPv6 multicast data in the following configuration, undue load is placed on the non-DR IPv6 PIM-SM router, which might have a significant impact on other functionality on the Switch. In this configuration, use separate lines for Switches A and B.

Figure 30-34: Inappropriate configuration (when connecting a sender to multiple routers)

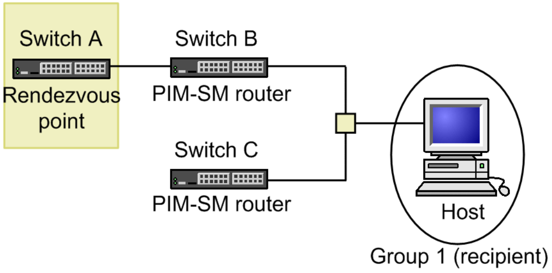


- Configuration in which multiple IPv6 PIM-SM routers are running on the same line as the IPv6 multicast group (recipient), and one of the IPv6 PIM-SM routers does not connect to the rendezvous point

  When IPv6 multicast communication is bound for group 1 in the following configuration, the shortest path between the sender and group 1 might not be established.

  Make sure that Switch A connects to the rendezvous point without going through Switch B, and that Switch B connects to the rendezvous point without going through Switch A.

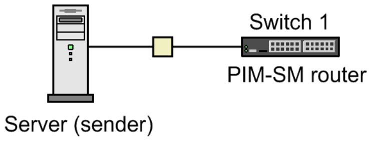
Figure 30-35: Inappropriate configuration (when a router is not connected to a rendezvous point)



- Configuration in which there are no recipients

  If the server sends a large volume of IPv6 multicast data in the following configuration, Switch 1 bears the load of discarding the data. This load might have a significant impact on other functionality of the Switch. When using IPv6 multicasting, make sure that there is at least one recipient.

Figure 30-36: Inappropriate configuration (configuration in which there are no recipients)



(3) IPv6 PIM-SSM

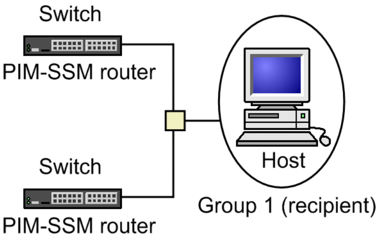
(a) Configurations requiring special attention

Use caution when using IPv6 PIM-SSM in the following configurations:

- Configuration in which multiple IPv6 PIM-SSM routers are running on the same line as the IPv6 multicast group (receiver)

If MLDv1 is used to run PIM-SSM in the following configuration, make sure that the `ipv6 pim ssm` and `ipv6 mld ssm-map static` configuration commands are used to set up all routers on the same line.

Figure 30-37: Configuration requiring special attention (when connecting multiple routers to a host)



(b) Notes applying when multiple addresses are set on the terminal side

Care is needed if multiple IPv6 addresses are assigned to a terminal that sends data by SSM communication. Accordingly, make sure that the source address of the send data matches the source address information set on the Switch by using the `ipv6 mld ssm-map static` configuration command. These addresses might not match when, for example, the RA automatic address setting function is used, in which case the terminal performs communication by using the automatically set address.

(4) PIM-SM VRF gateways [OS-L3SA]

(a) Configurations requiring special attention

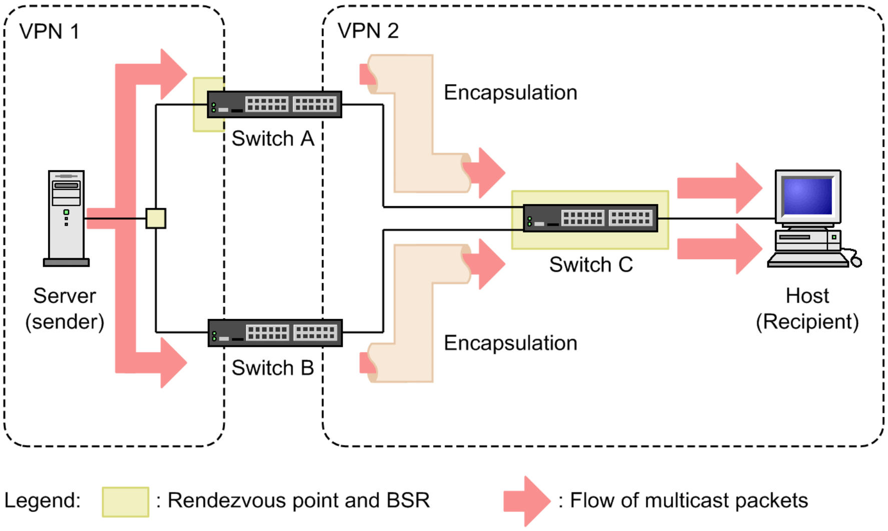
The following configuration needs special attention:

- In an IPv6 multicast extranet that uses PIM-SM VRF gateways, if two or more VRF border routers are used to create a redundant configuration, make sure that one of the VRF border routers is set as the rendezvous point.

  If the rendezvous point is not a VRF border router, each VRF border router forwards the same IPv6 multicast packets until the shortest-path delivery tree is generated.

  The following figure shows an example configuration which uses PIM-SM VRF gateways and forwards the same encapsulated packets twice.

Figure 30-38: Configuration requiring special attention (when deploying more than one VRF boundary router for a redundant configuration)



Chapter

31. Settings and Operation for IPv6 Multicasting

This chapter describes how to set up and check the status of an IPv6 multicast configuration.

31.1 Configuration

31.1.1 List of configuration commands

The following table describes the configuration commands for IPv6 multicasting.

Table 31-1: List of configuration commands

Command name	Description
ipv6 mld fast-leave	Enables the function that allows an MLD listener to immediately leave the group when the MLD listener is the only MLD listener on the same link.
ipv6 mld group-limit	Specifies the maximum number of groups that can run on an interface.
ipv6 mld query-interval	Changes the sending interval for query messages.
ipv6 mld router	Enables MLD.
ipv6 mld source-limit	Specifies the maximum number of sources during group participation.
ipv6 mld ssm-map enable	Enables IPv6 PIM-SSM mapping operation to be used with MLDv1 or MLDv2 (EXCLUDE mode).
ipv6 mld ssm-map static	Sets the group address and source address for which PIM-SSM runs.
ipv6 mld static-group	Enables static additions to MLD groups.
ipv6 mld version	Changes the MLD version.
ipv6 multicast-routing	Enables use of the IPv6 multicast functionality.
ipv6 pim	Sets IPv6 PIM-SM.
ipv6 pim assert-metric	Changes the metric used in assert messages.
ipv6 pim assert-preference	Changes the preference value used in assert messages.
ipv6 pim bsr candidate bsr	Sets the BSR.
ipv6 pim bsr candidate rp	Sets the rendezvous point.
ipv6 pim deletion-delay-time	Changes the deletion delay time.
ipv6 pim direct	Enables the function that treats a remote multicast server address as a directly connected server.
ipv6 pim hello-interval	Changes the sending interval for hello messages.
ipv6 pim join-prune-interval	Changes the sending interval for join or prune messages.
ipv6 pim keep-alive-time	Changes the keep-alive time.
ipv6 pim max-interface	Changes the maximum number of interfaces that can run IPv6 PIM.
ipv6 pim mcache-limit	Specifies the maximum number of multicast forwarding entries.
ipv6 pim mroute-limit	Specifies the maximum number of multicast routing information entries.
ipv6 pim negative-cache-time	Changes the negative cache time.
ipv6 pim register-probe-time	Specifies the register probe time.
ipv6 pim rp-address	Sets the static rendezvous point.
ipv6 pim rp-mapping-algorithm	Specifies the rendezvous point selection algorithm.

Command name	Description
ipv6 pim ssm	Sets the IPv6 PIM-SSM address.
ipv6 pim vrf-gateway	Sets the PIM-SM VRF gateway.

31.1.2 Overview of configuration

Refer to the setting examples below according to the configuration used.

Note that before IPv6 can be used, the `swrt_table_resource` command must be used to set the mode that allows IPv6 resources to be used. For details about the `swrt_table_resource` command, see the manual *Configuration Command Reference Vol. 1 For Version 11.10*.

- ■ When PIM-SM is used
  - Configuring IPv6 multicast routing
  - Configuring IPv6 PIM-SM
  - Configuring an IPv6 PIM-SM rendezvous point candidate (when using the local switch as a rendezvous point)
  - Configuring an IPv6 PIM-SM BSR candidate (when using the local switch as the BSR)
  - Configuring MLD settings
- ■ When PIM-SM (static rendezvous point) is used
  - Configuring IPv6 multicast routing
  - Configuring IPv6 PIM-SM
  - Configuring an IPv6 PIM-SM rendezvous point candidate (when using the local switch as a rendezvous point)
  - Configuring an IPv6 PIM-SM static rendezvous point
  - Configuring MLD settings
- ■ When PIM-SSM is used
  - Configuring IPv6 multicast routing
  - Configuring IPv6 PIM-SM
  - Configuring IPv6 PIM-SSM
  - Configuring MLD settings
- ■ When PIM-SM is used in a VRF
  - Configuring IPv6 multicast routing for a VRF
  - Configuring IPv6 PIM-SM for a VRF
  - Configuring an IPv6 PIM-SM rendezvous point candidate for a VRF (when using the local switch as a rendezvous point in the target VPN)
  - Configuring an IPv6 PIM-SM BSR candidate for a VRF (when using the local switch as the BSR in the target VPN)
  - Configuring MLD for a VRF
- ■ When PIM-SM (static rendezvous point) is used in a VRF
  - Configuring IPv6 multicast routing for a VRF
  - Configuring IPv6 PIM-SM for a VRF
  - Configuring an IPv6 PIM-SM rendezvous point candidate for a VRF (when using the

local switch as a rendezvous point in the target VPN)

- Configuring an IPv6 PIM-SM static rendezvous point for a VRF
- Configuring MLD for a VRF

■ When PIM-SSM is used in a VRF

- Configuring IPv6 multicast routing for a VRF
- Configuring IPv6 PIM-SM for a VRF
- Configuring IPv6 PIM-SSM for a VRF
- Configuring MLD for a VRF

■ When PIM-SM is used for a VRF (extranet) (PIM-SM VRF gateway)

- Configuring IPv6 multicast routing for a VRF
- Configuring IPv6 PIM-SM for a VRF
- Configuring an IPv6 PIM-SM rendezvous point candidate for a VRF (when using the local switch as a rendezvous point)
- Configuring an IPv6 PIM-SM BSR candidate for a VRF (when using the local switch as the BSR)
- Configuring a PIM-SM VRF gateway

■ When PIM-SSM is used for a VRF (extranet)

- Configuring IPv6 multicast routing for a VRF
- Configuring IPv6 PIM-SM for a VRF
- Configuring IPv6 PIM-SSM for a VRF
- Configuring an IPv6 multicast extranet

31.1.3 Configuring IPv6 multicast routing

Points to note

To enable IPv6 multicast routing with the Switch, you must use global configuration mode for configurations, and you must set the loopback address of the Switch for the `loopback 0` interface. In the following procedure, 2001:db8::b is used as the loopback address of the Switch.

Note that, in addition to the settings described here, you must configure IPv6 PIM on at least one interface by using the `ipv6 pim` command.

Command examples

1.
```
(config)# interface loopback 0
(config-if)# ipv6 address 2001:db8::b
(config-if)# exit
```
Sets the loopback address.

2.
```
(config)# ipv6 multicast-routing
```
Enables the IPv6 multicast functionality.

31.1.4 Configuring IPv6 PIM-SM

Points to note

IPv6 PIM-SM (sparse mode) needs to be set on interfaces running IPv6 multicast routing. For the configuration, use interface configuration mode. The following shows a PIM-SM configuration example in which 2001:db8::a/16 is used as the interface IPv6 address.

Command examples

1.
```
(config)# interface vlan 10
(config-if)# ipv6 address 2001:db8::a/16
(config-if)# ipv6 enable
```
Sets the IPv6 address.

2.
```
(config-if)# ipv6 pim
```
Specifies operation as an IPv6 PIM-SM (sparse mode).

31.1.5 Configuring an IPv6 PIM-SM rendezvous point candidate

Points to note

The example below shows how to use the Switch as a rendezvous point candidate by using global configuration mode for configurations. Use the address set for the `loopback 0` interface as the rendezvous point address. In the following procedure, 2001:db8::b is used as the loopback address of the Switch, and ff15::/16 is used as the multicast group address to be managed.

Command examples

1.
```
(config)# ipv6 access-list GROUP1
(config-ipv6-acl)# permit ipv6 any ff15::/16
(config-ipv6-acl)# exit
```
Creates an access list for the multicast group addresses to be managed.

2.
```
(config)# ipv6 pim bsr candidate rp 2001:db8::b group-list
GROUP1
```
Sets the Switch as a rendezvous point candidate. (The access list created in step 1 is used to specify the multicast group address to be managed.)

31.1.6 Configuring an IPv6 PIM-SM BSR candidate

Points to note

The example below shows how to use a Switch as a BSR candidate by using global configuration mode for the configuration. Use the address set for the `loopback 0` interface as the BSR address. In the following procedure, 2001:db8::b is used as the loopback address of the Switch.

Command examples

1.
```
(config)# ipv6 pim bsr candidate bsr 2001:db8::b
```
Sets the Switch as a BSR candidate.

31.1.7 Configuring an IPv6 PIM-SM static rendezvous point

Points to note

The example below shows how to specify the static rendezvous point by using global configuration mode for the configuration. In this example, 2001:db8::b is used as the switch address of the static rendezvous point.

Command examples

1.
```
(config)# ipv6 pim rp-address 2001:db8::b
```

Sets 2001:db8::b as the rendezvous point.

31.1.8 Configuring IPv6 PIM-SSM

(1) Configuring IPv6 PIM-SSM addresses

Points to note

The following configuration is performed in global configuration mode to enable use of IPv6 PIM-SSM for the Switches. The configuration causes IPv6 PIM-SSM to run within the specified SSM address range on interfaces for which IPv6 PIM-SM is set. Only one SSM address setting can be used on the Switch. In this example, ff35::/16 is specified as the SSM address range for which PIM-SSM runs.

Command examples

1.
```
(config)# ipv6 access-list GROUP2
(config-ipv6-acl)# permit ipv6 any ff35::/16
(config-ipv6-acl)# exit
```

Creates an access list for an SSM address range.

2.
```
(config)# ipv6 pim ssm range GROUP2
```

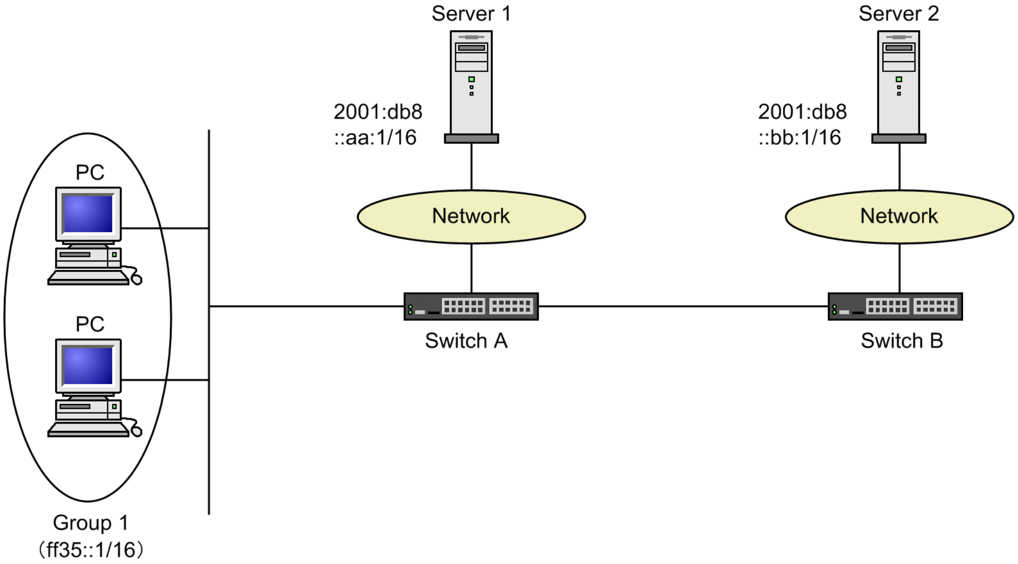
Enables the use of IPv6 PIM-SSM. (The access list created in step 1 is used to specify the SSM address range.)

(2) Enabling linked operation of IPv6 PIM-SSM when MLDv1 or MLDv2 (EXCLUDE mode) is used

Points to note

Because source addresses cannot be differentiated with MLDv1 or MLDv2 (EXCLUDE mode), linkage to the PIM-SSM cannot be performed. With the Switch, the group address and source address for which PIM-SSM is run can be set to link with PIM-SSM. In the following PIM-SSM configuration example, ff35::1 is used as the group address when two servers are used with 2001:db8::aa:1 as the source address of server 1, and 2001:db8::bb:1 as the source address of server 2.

Figure 31-1: PIM-SSM configuration example



Command examples

1.
```
(config)# ipv6 access-list GROUP3
(config-ipv6-acl)# permit ipv6 any host ff35::1
(config-ipv6-acl)# exit
```
Creates the access list for which the group address is specified.

2.
```
(config)# ipv6 mld ssm-map static GROUP3 2001:db8::aa:1
(config)# ipv6 mld ssm-map static GROUP3 2001:db8::bb:1
```
Sets the group address for which PIM-SSM is run, and the source addresses for server 1 and server 2. (The access list created in step 1 is used to specify the group address.)

3.
```
(config)# ipv6 mld ssm-map enable
```
Enables the use of IPv6 PIM-SSM.

31.1.9 Configuring MLD settings

Points to note

MLD must be configured for interfaces on which MLD will run.

Command examples

1.
```
(config-if)# ipv6 mld router
```
Specifies that MLD runs on the interface in mixed mode for versions 1 and 2 (default).

31.1.10 Configuring IPv6 multicast routing for a VRF [OS-L3SA]

Points to note

To use IPv6 multicast routing in VRFs, you must set an address on a loopback interface for each VRF, and then specify the settings below in global configuration mode. In this example IPv6 multicast routing configuration, the loopback address of VRF 10 is 2001:db8::10.

Note that, in addition to the settings shown here, you must configure IPv6 PIM on at least one interface for each VRF by using the `ipv6 pim` command.

Command examples

1.
```
(config)# vrf definition 10
(config-vrf)# exit
```
Configures VRF 10.

2.
```
(config)# interface loopback 30
(config-if)# vrf forwarding 10
(config-if)# ipv6 address 2001:db8::10
(config-if)# exit
```
Sets a loopback address for the `loopback 30` loopback interface of VRF 10.

3.
```
(config)# ipv6 multicast-routing vrf 10
```
Enables IPv6 multicasting in VRF 10.

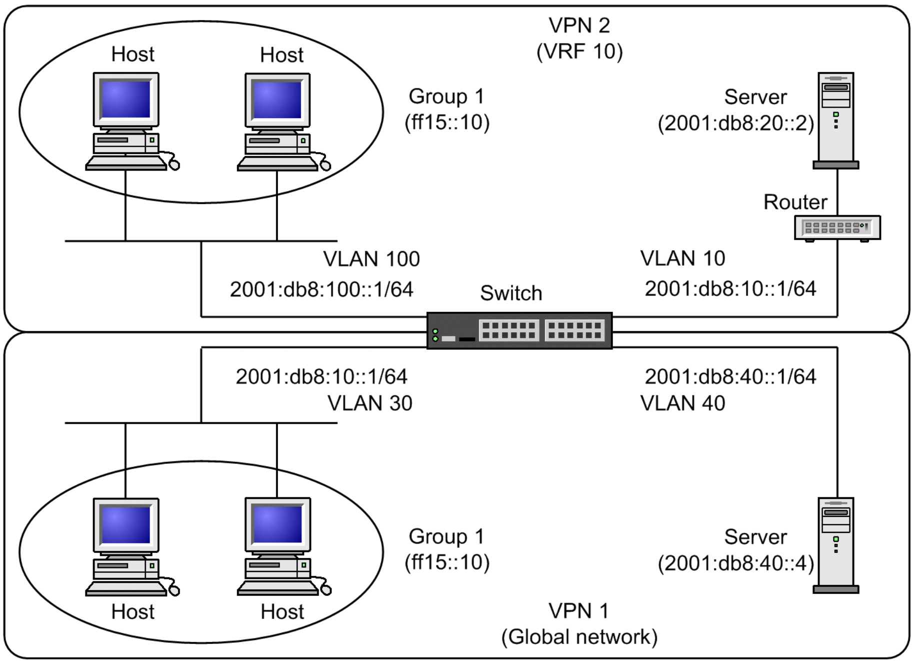
31.1.11 Configuring IPv6 PIM-SM for a VRF [OS-L3SA]

Points to note

To use IPv6 PIM-SM in a VRF, configure the IPv6 multicast routing functionality for the VRF, and then configure IPv6 PIM-SM (`sparse` mode) on one or more interfaces in the VRF. You must also configure these even in a VRF that has no neighboring routers (for example, the sender and recipient of IPv6 multicasting are directly connected to the Switch).

The settings for IPv6 PIM-SM (sparse mode) are specified in interface configuration mode. The following figure shows an example IPv6 PIM-SM configuration in which VPN 2 is associated with VRF 10 and 2001:db8:10::1/64 is used as the IPv6 address of the VLAN 10 interface in VRF 10.

Figure 31-2: IPv6 PIM-SM configuration example for a VRF



Command examples

1.
```
(config)# interface vlan 10
```
Configures VLAN 10.

2.
```
(config-if)# vrf forwarding 10
```
Configures VLAN 10 in VRF 10.

3.
```
(config-if)# ipv6 address 2001:db8:10::1/64
(config-if)# ipv6 enable
```
Sets an IPv6 address for VLAN 10.

4.
```
(config-if)# ipv6 pim
(config-if)# exit
```
Sets IPv6 PIM-SM for VLAN 10.

31.1.12 Configuring an IPv6 PIM-SM rendezvous point candidate for a VRF [OS-L3SA]

Points to note

The configuration below is performed in global configuration mode to use the Switch as a rendezvous point candidate in a VRF. In the procedure below, 2001:db8::10 is used as the loopback address of VRF 10, and ff15::/16 is used as the multicast group address to be managed.

Command examples

1.
```
(config)# ipv6 access-list GROUP1
(config-ipv6-acl)# permit ipv6 any ff15::/16
(config-ipv6-acl)# exit
```
   Creates an access list for the multicast group addresses to be managed in VRF 10.

2.
```
(config)# ipv6 pim vrf 10 bsr candidate rp 2001:db8::10
group-list GROUP1
```
   Sets the Switch as the rendezvous point candidate for VRF 10. (The access list created in step 1 is used to specify the multicast group address to be managed.)

31.1.13 Configuring an IPv6 PIM-SM BSR candidate for a VRF [OS-L3SA]

Points to note

The configuration below is performed in global configuration mode to use the Switch as a BSR candidate in a VRF. In the following procedure, 2001:db8::10 is used as the loopback address of the Switch.

Command examples

1.
```
(config)# ipv6 pim vrf 10 bsr candidate bsr 2001:db8::10
```
   Sets the Switch as a BSR candidate of VRF 10.

31.1.14 Configuring an IPv6 PIM-SM static rendezvous point for a VRF [OS-L3SA]

Points to note

The configuration below is performed in global configuration mode to specify a static rendezvous point for a VRF. In the procedure below, 2001:db8::b is used as the IPv6 address of the VRF 10 static rendezvous point.

Command examples

1.
```
(config)# ipv6 pim vrf 10 rp-address 2001:db8::10
```
   Specifies 2001:db8::10 as the rendezvous point for VRF 10.

31.1.15 Configuring IPv6 PIM-SSM for a VRF [OS-L3SA]

(1) Configuring IPv6 PIM-SSM addresses

Points to note

The configuration below is performed in global configuration mode to use IPv6 PIM-SSM for the Switch in a VRF. These settings cause IPv6 PIM-SSM to run within the specified SSM address range on the VRF interfaces for which IPv6 PIM-SM is set. The Switch can set only one SSM address for each VRF. In this example, the default (ff30::/12) is used for the SSM address range for which IPv6 PIM-SSM runs in VRF 10.

Command examples

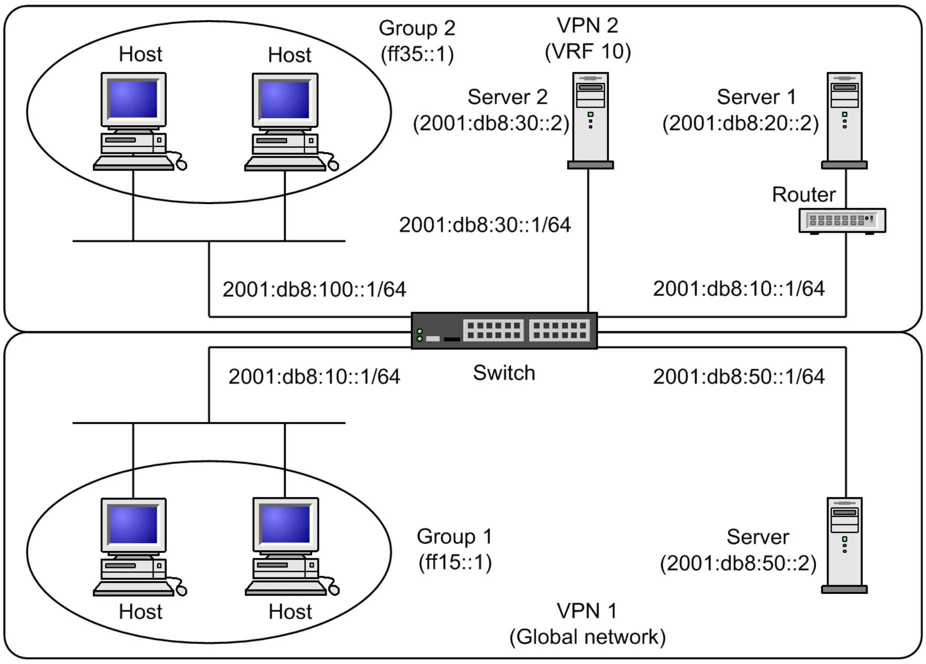
1.
```
(config)# ipv6 pim vrf 10 ssm default
```
   Enables the use of IPv6 PIM-SSM for VRF 10 (with an SSM address range of ff30::/12).

(2) Enabling linked operation of IPv6 PIM-SSM when MLDv1 or MLDv2 (EXCLUDE mode) is used

Points to note

Because source addresses cannot be differentiated with MLDv1 or MLDv2 (EXCLUDE mode), linkage to the IPv6 PIM-SSM cannot be performed. With the Switch, the group address and source address for which IPv6 PIM-SSM is run can be set to link with IPv6 PIM-SSM. This functionality must be set for each VRF. The group address for which IPv6 PIM-SSM is run needs to be within the SSM address range specified for the target VRF in the IPv6 PIM-SSM address setting. The figure below shows an IPv6 PIM-SSM configuration example in which VPN 2 is associated with VRF 10 and ff35::1 is used as the group address for VPN 2. When two servers are used in the same VPN, 2001:db8:20::2 is used as the source address of Server 1 and 2001:db8:30::2 is used as the source address of Server 2.

Figure 31-3: IPv6 PIM-SSM configuration example for a VRF



Command examples

1.
```
(config)# ipv6 access-list GROUP2
(config-ipv6-acl)# permit ipv6 any host ff35::1
(config-ipv6-acl)# exit
```
   Creates an access list for the multicast group addresses to be managed in VRF 10.

2.
```
(config)# ipv6 mld ssm-map vrf 10 static GROUP2 2001:db8:20::2
(config)# ipv6 mld ssm-map vrf 10 static GROUP2 2001:db8:30::2
```
   For VRF 10, sets the group address for which IPv6 PIM-SSM is run in VPN 2, and the source addresses of server 1 and server 2 (the access list created in step 1 is used to specify the group address).

3.
```
(config)# ipv6 mld vrf 10 ssm-map enable
```
   Enables IPv6 PIM-SSM to be used for MLDv1 and MLDv2 (EXCLUDE mode) on VRF 10.

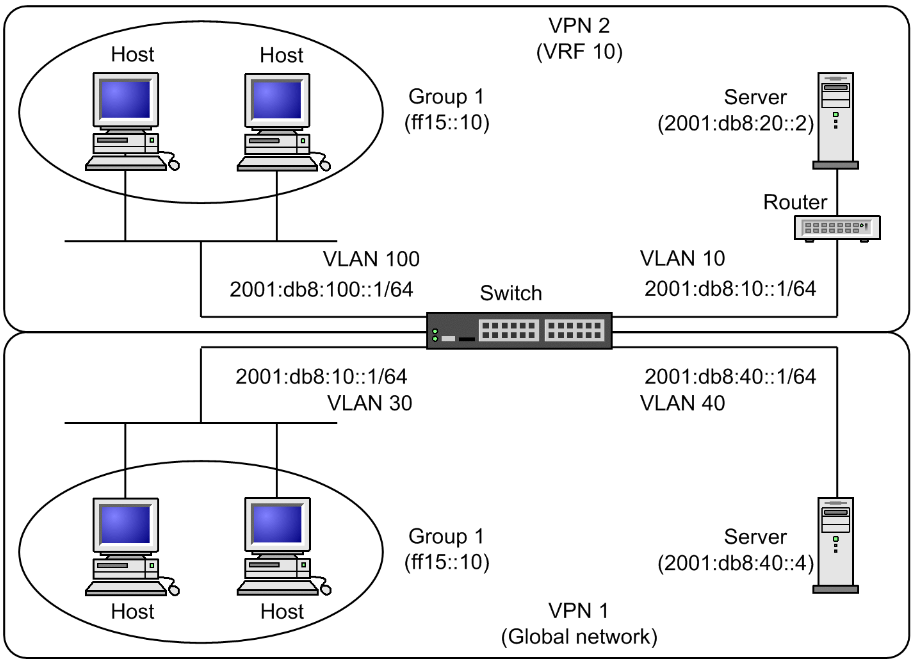
31.1.16 Configuring MLD for a VRF [OS-L3SA]

Points to note

The example below shows how to configure MLD for a VRF to run MLD on the VRF.

By default, MLD uses a mixed mode of versions 1 and 2. Use the `ipv6 mlds` configuration command to change the MLD version. The figure below shows an MLD configuration example in which VPN 2 is associated with VRF 10 and 2001:db8:100::1/64 is used as the IPv6 address of the VLAN 100 interface in VRF 10.

Figure 31-4: MLD configuration example for a VRF



Command examples

1. `(config)# interface vlan 100`

   Configures VLAN 100.

2. `(config-if)# vrf forwarding 10`

   Configures VLAN 10 in VRF 100.

3.
   ```
   (config-if)# ipv6 address 2001:db8:100::1/64
   (config-if)# ipv6 enable
   ```

   Sets an IPv6 address for VLAN 100.

4.
   ```
   (config-if)# ipv6 mld router
   (config-if)# exit
   ```

   Sets MLD for VLAN 100.

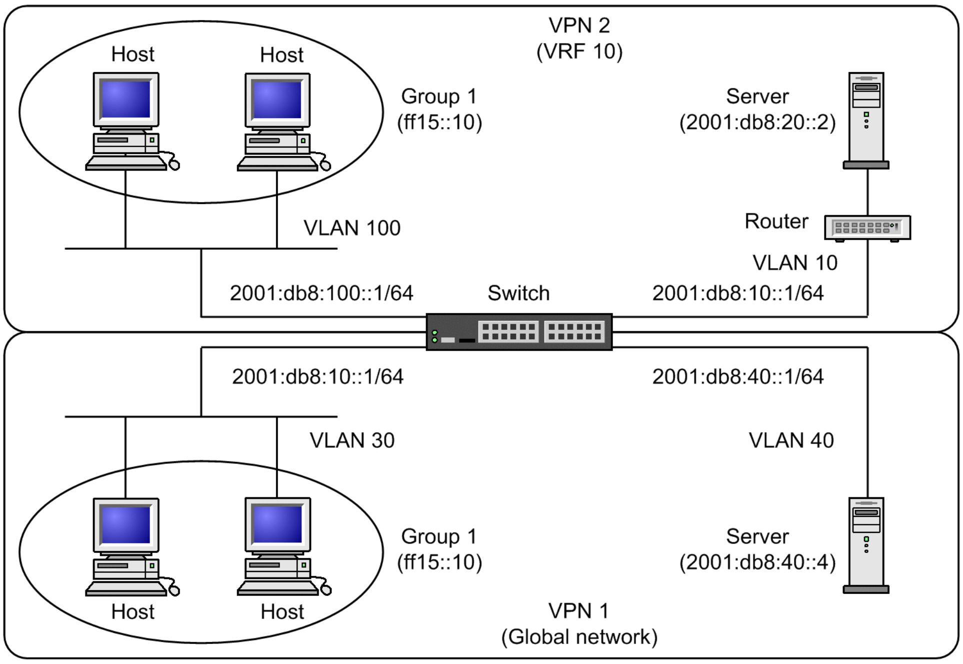
31.1.17 Configuring an IPv6 multicast extranet [OS-L3SA]

Points to note

In an IPv6 multicast extranet, a unicast extranet to the sender must be configured for the forwarding destination VRF and a unicast route must exist.

Configure multicast route filtering for a VRF in which the sender exists. If no conditions are specified for the route filtering, all multicast addresses can be forwarded to all VRFs on which multicasts run. Specify all the multicast route filtering settings in global configuration mode. The following figure shows a PIM-SSM configuration example in which VPN 2 is associated with VRF 10, and 2001:db8:100::1/64 and 2001:db8:10::1/64 are used as the interface IP addresses for VRF 10. In this case, a unicast route to the server (2001:db8:20::2) in VPN 2 (VRF 10) must exist in VPN 1 (global network).

Figure 31-5: PIM-SSM configuration example for a VRF (IPv6 multicast extranet)



Command examples

1.
```
(config)# route-map MLT6EXNET permit 10
(config-route-map)# exit
```
Creates a `route-map` that permits all multicast forwarding requests.

2.
```
(config)# vrf definition 10
(config-vrf)# ipv6 import multicast inter-vrf MLT6EXNET
(config-vrf)# exit
```
Applies the setting that permits IPv6 multicast forwarding requests from all VRFs to VRF 10.

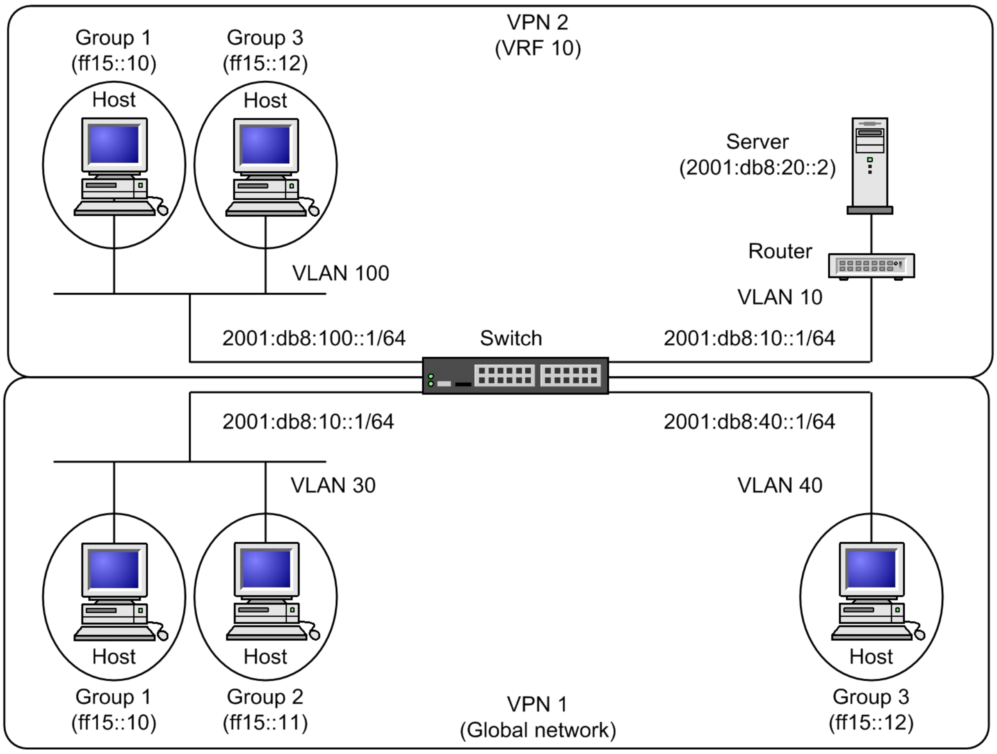
31.1.18 Configuring a PIM-SM VRF gateway [OS-L3SA]

Points to note

In an IPv6 multicast extranet, a unicast extranet to the sender must be configured for the forwarding destination VRF and a unicast route must exist.

To use multicast extranets with PIM-SM to perform communication between multicast VRFs, a PIM-SM VRF gateway must be configured. The PIM-SM VRF gateway must be set for the VRF that has the multicast sender. Specify the settings in global configuration mode. The PIM-SM VRF gateway is configured by specifying all group addresses used for the extranet in multicast route filtering as host addresses. At this time, a range of group addresses specified with a prefix are not subject to control by the PIM-SM VRF gateway. The figure below shows a configuration example in which packets of group addresses ff15::10, ff15::11, and ff15::12 are forwarded from VRF 10 to the global network. In this case, a unicast route to the server (2001:db8:20::2) in VPN 2 (VRF 10) must exist in VPN 1 (global network).

Figure 31-6: PIM-SM configuration example for a VRF (PIM-SM VRF gateway)



Command examples

1.
```
(config)# ipv6 access-list MLT6GROUP
(config-ipv6-acl)# permit ipv6 host ff15::10 any
(config-ipv6-acl)# permit ipv6 host ff15::11 any
(config-ipv6-acl)# permit ipv6 host ff15::12 any
(config-ipv6-acl)# exit
(config)# route-map MLT6EXNET permit 10
(config-route-map)# match ipv6 address MLT6GROUP
(config-route-map)# exit
```

Specifies ff15::10, ff15::11, and ff15::12 as the group addresses used for the PIM-SM VRF gateway.

2.
```
(config)# vrf definition 10
(config-vrf)# ipv6 import multicast inter-vrf MLT6EXNET
```

```
(config-vrf)# exit
```

Specifies the groups that will be forwarded from VRF 10 to other VRFs.

3. `(config)# ipv6 pim vrf 10 vrf-gateway`

   Configures the PIM-SM VRF gateway for VRF 10.

31.2 Operation

31.2.1 List of operation commands

The following table describes the operation commands for IPv6 multicast.

Table 31-2: List of operation commands

Command name	Description
show ipv6 mcache	Shows a list of all multicast paths.
show ipv6 mroute	Shows PIM-SM multicast routing information.
show ipv6 pim interface	Shows the status of the IPv6 PIM-SM/SSM interface.
show ipv6 pim neighbor	Shows neighbor information for the IPv6 PIM-SM/SSM interface.
show ipv6 pim mcache	Shows multicast forwarding entries for IPv6 PIM-SM/SSM.
show ipv6 pim bsr	Shows the IPv6 PIM-SM BSR information.
show ipv6 pim rp-mapping	Shows the IPv6 PIM-SM rendezvous point information.
show ipv6 pim rp-hash	Shows the rendezvous point information for each IPv6 PIM-SM group.
show ipv6 mld interface	Shows the status of the MLD interface.
show ipv6 mld group	Shows MLD information.
show ipv6 rpf	Shows PIM RPF information.
show ipv6 multicast statistics	Shows IPv6 multicast statistics.
clear ipv6 multicast statistics	Clears IPv6 multicast statistics.
restart ipv6-multicast	Restarts the IPv6 multicast routing program.
debug protocols ipv6-multicast	Enables the IPv6 multicast routing program to log event information to syslog.
no debug protocols ipv6-multicast	Stops the logging of event information to syslog by the IPv6 multicast routing program.
dump protocols ipv6-multicast	Obtains a dump of control table information and event trace information being collected by the IPv6 multicast routing program.
erase protocol-dump ipv6-multicast	Deletes the dump of the event trace information file, control table information file, and core file created by the IPv6 multicast routing program.

31.2.2 Checking routes to IPv6 multicast group addresses

When IPv6 multicast routing information has been set on the Switch, use the `show ipv6 mcache` and `show netstat multicast` commands to check whether routes to the destination addresses exist. If there are no routes or if `outgoing` entries are incorrect, conduct the necessary checks. For details, see *31.2.3 Checking IPv6 PIM-SM information* and *31.2.4 Checking MLD information*.

The `show ipv6 mcache` command displays the IPv6 multicast forwarding entries retained in the IPv6 multicast routing program. The `show netstat multicast` command displays the IPv6 multicast forwarding entries registered in hardware.

Note that the `show netstat multicast` command also displays the negative cache (packet discarding entry for which no output interface exists).

Figure 31-7: Results of executing the show ipv6 mcache command

```
> show ipv6 mcache
Date 20XX/04/20 15:20:00 UTC
Total: 1 route
- Forwarding entry  ----------------------------------------------------------
Group Address                            Source Address
ff15::2                                  2001:db8::100
    uptime: 00:20    expires: 02:40    flags:
    incoming:
        VLAN0002
    outgoing:
        VLAN0001
        VLAN0003
>
```

Figure 31-8: Results of executing the show netstat multicast command

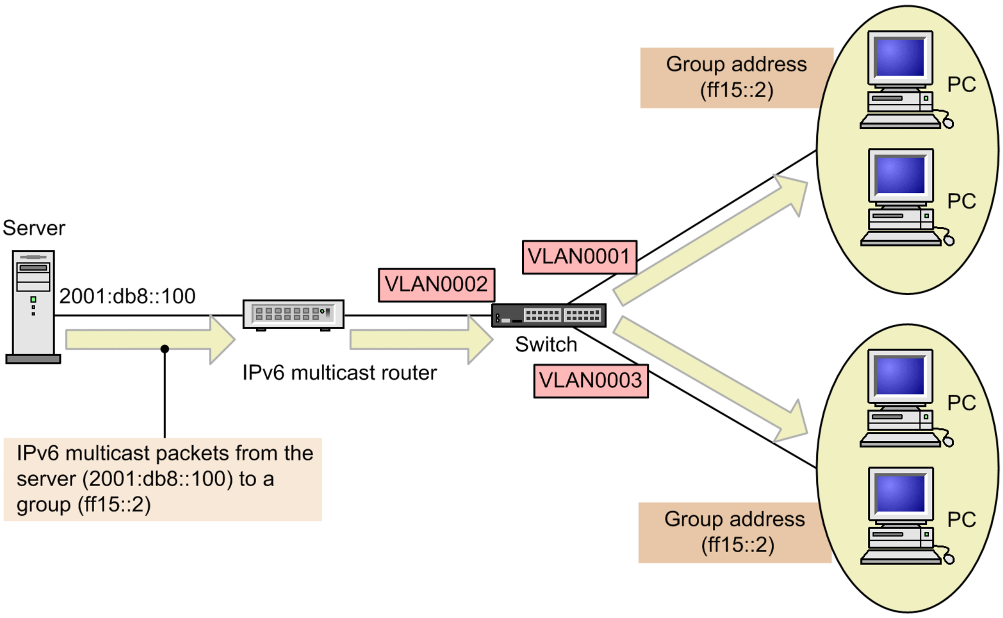
```
> show netstat multicast
Date 20XX/04/10 15:20:00 UTC
Virtual Interface Table is empty

Multicast Forwarding Cache is empty

IPv6 Virtual Interface Table
 Mif   Rate      PhyIF         Pkts-In   Pkts-Out
   0      0   VLAN0004               0          0
   1      0   VLAN0002               0          0
   2      0   VLAN0001               0          0
   3      0   VLAN0003               0          0

IPv6 Multicast Forwarding Cache
 Origin                 Group           Packets Waits In-Mif Out-Mifs
 2001:db8::100          ff15::2               0     0    1    2  3

Total no. of entries in cache: 1
>
```



31.2.3 Checking IPv6 PIM-SM information

The following are the types of information you can check in the IPv6 multicast routing information for the Switch when the PIM-SM functionality is set.

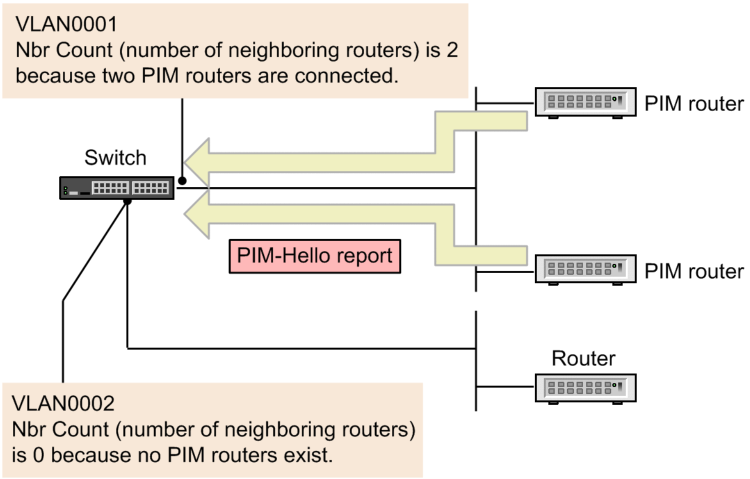
(1) Interface information

The following describes the checking that is performed by executing the `show ipv6 pim interface` command.

Figure 31-9: Results of executing the show ipv6 pim interface command

```
> show ipv6 pim interface
Date 20XX/08/01 15:20:00 UTC
Interface      Component  Vif Nbr    Hello DR                        This
                              Count  Intvl Address                   System
VLAN0001       PIM-SM       1     2     30 fe80::200:87ff:fe10:a95a Y
(data omitted)
```

- Check whether the execution results include the expected interface name. If the expected interface name is not included, IPv6 PIM-SM is not running on the interface. In configuration mode, make sure that IPv6 PIM is enabled on that interface. Also, make sure that a failure has not occurred for the interface.
- Check the `Nbr Count` value (number of neighboring PIM routers) for the target interface. If this value is 0, neighboring routers either do not exist, or are not reporting PIM hello messages. Check the neighboring routers.

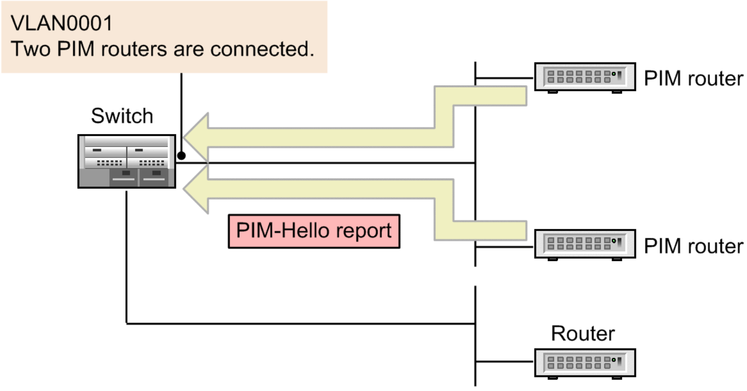

VLAN0001
Nbr Count (number of neighboring routers) is 2 because two PIM routers are connected.
Switch
PIM router
PIM-Hello report
PIM router
Router
VLAN0002
Nbr Count (number of neighboring routers) is 0 because no PIM routers exist.

(2) Neighbor information

Execute the `show ipv6 pim neighbor` command to check for the presence of neighbors for the target interface. If a particular neighbor does not exist, the neighboring routers might not be reporting PIM Hello messages. Check the neighboring routers.

Figure 31-10: Results of executing the show ipv6 pim neighbor command

```
> show ipv6 pim neighbor
Date 20XX/08/01 15:20:00 UTC
Neighbor Address         Interface Uptime  Expires
fe80::200:87ff:fea0:abcd VLAN0001  00:05   01:40
fe80::200:87ff:feb0:1234 VLAN0001  00:05   01:40
(data omitted)
```

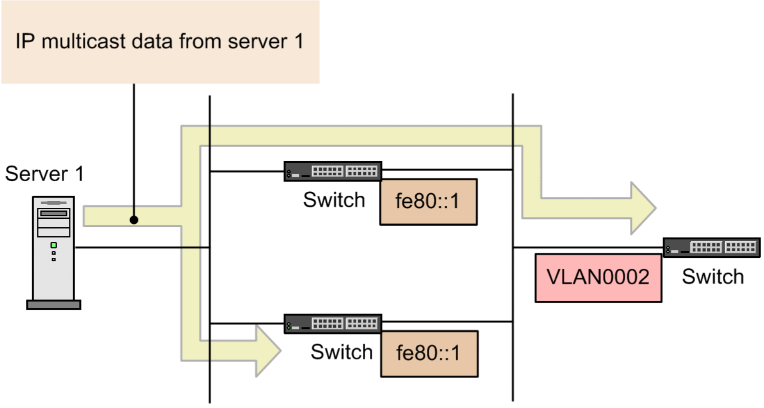


(3) Sender routing information

Execute the `show ipv6 rpf` command to check sender routing information.

Figure 31-11: Results of executing the show ipv6 rpf command

```
> show ipv6 rpf 2001:db8::100
Date 20XX/08/01 15:20:00 UTC
Incoming: VLAN0002 Upstream: fe80::1
(data omitted)
```



(4) PIM-SM BSR information

Execute the `show ipv6 pim bsr` command and check whether the BSR address is displayed. If `----` is displayed, the BSR might not be sending bootstrap messages, or might not exist. Check the BSR. Note that BSRs cannot be used for PIM-SSM.

Figure 31-12: Results of executing the show ipv6 pim bsr command

```
> show ipv6 pim bsr
Date 20XX/08/01 15:20:00 UTC
Status : Not Candidate Bootstrap Router
BSR Address : 2001:db8::1
    Priority: 100    Hash mask length: 30
    Uptime   : 03:00
    Bootstrap Timeout : 130 seconds
>
```

(5) PIM-SM rendezvous point information

Execute the `show ipv6 pim rp-mapping` command to make sure that the C-RP address is displayed for the corresponding IPv6 multicast group address. If it is not displayed, the BSR might not be sending bootstrap messages, or the rendezvous point or BSR might not exist. Check the rendezvous point and BSR. Note that rendezvous points cannot be used for PIM-SSM.

Figure 31-13: Results of executing the show ipv6 pim rp-mapping command

```
> show ipv6 pim rp-mapping brief
Date 20XX/08/01 15:20:00 UTC
Status : Not Candidate Rendezvous Point
Total: 2 routes, 2 groups, 1 RP
Group/Masklen                  C-RP Address
ff15:100::/32                  2001:db8::1
ff15:200::/64                  2001:db8::1
>
```

(6) PIM-SM routing information

Execute the `show ipv6 mroute` command to check whether a path to the corresponding destination address exists. If no `(S,G)` entry exists, check whether a `(*,G)` entry exists. If no `(*,G)` entry exists, and both incoming and outgoing are invalid, check the neighboring router. Note that `(*,G)` cannot be used for PIM-SSM, since it does not exist.

Figure 31-14: Displaying PIM-SM multicast routing information

```
> show ipv6 mroute
Date 20XX/04/20 15:20:00 UTC
Total: 4 routes, 2 groups, 2 sources

(S,G) 2 routes  --------------------------------------------------
Group Address                       Source Address
ff15:100::50                        2001:db8::100
    uptime 02:00    expires 02:30    assert 00:00    flags F  protocol SM
    incoming: VLAN0002       upstream: Direct  reg-sup: 30s
    outgoing: VLAN0003  uptime 02:30    expires --:--

ff15:200::1                         2001:db8::200
    uptime 02:00    expires 02:30    assert 00:00    flags F  protocol SM
    incoming: VLAN0001       upstream: Direct  reg-sup: 30s
    outgoing: VLAN0003  uptime 02:30    expires --:--

(*,G) 2 routes  --------------------------------------------------
Group Address                       RP Address
ff15:100::50                        2001:db8::1
    uptime 02:00    expires --:--    assert 00:00    flags R  protocol SM
    incoming: VLAN0001       upstream: This System
    outgoing: VLAN0003  uptime 02:30    expires --:--

ff15:200::1                         2001:db8::2
    uptime 02:00    expires --:--    assert 00:00    flags R  protocol SM
    incoming: VLAN0001       upstream: fe80::1200:87ff:fe10:1234
    outgoing: VLAN0003  uptime 02:30    expires --:--
              VLAN0004  uptime 02:30    expires --:--
>
```

31.2.4 Checking MLD information

The following are the types of information you can check in the IPv6 multicast routing information for the Switch when the MLD functionality is set.

(1) Interface information

The following describes the checking that is performed by executing the `show ipv6 mld interface` command:

- Check the interfaces displayed in the `Interface` field. MLD is running on the displayed interfaces. If the expected interface is not displayed, check the `mld` configuration setting. In addition, check whether a failure has occurred on the interface.
- Check the `Group Count` value (number of groups hosts have joined) for the target interface. If the value is 0, either groups hosts have joined might not exist, or the group member hosts might not advertise MLD-Report. Check the hosts.

- Check whether the version displayed in the `Version` field allows connection with the host used by the interface.
- If a code is displayed in the `Notice` field, MLD packets have been discarded. Use the code to determine why the packets were discarded.

Figure 31-15: Results of executing the show ipv6 mld interface command

```
> show ipv6 mld interface
Date 20XX/04/10 15:10:00 UTC
Total: 10 Interfaces
Interface   Version   Flags  Querier             Expires  Group Count  Notice
VLAN0001       1      S      fe80::10:87ff:2959   02:30             4  L
VLAN0003       2             fe80::10:87ff:2959   01:30             2
VLAN0004      (2)            fe80::10:87ff:2959     -               5   QR
VLAN0005       1             fe80::1234           01:00             3   Q
VLAN0006       1             fe80::2592           02:30             6
(data omitted)
```

(2) Group information

Execute the `show ipv6 mld group` command and check the groups displayed in the `Group Address` field. If there are none, check for the following:

- The group members (hosts) might not advertise MLD-Report. Check the hosts.
- Check the version of the MLD interfaces for the Switch and for each host, and make sure that the Switch can connect to the hosts.
- If hosts ignore MLDv2 Query, MLDv2 cannot be used. Set the MLD version of the interface to 1.

Figure 31-16: Results of executing the show ipv6 mld group command

```
> show ipv6 mld group brief
Date 20XX/08/01 15:20:00 UTC
Total: 20 groups
Group Address   Interface       Version  Mode      Source Count
ff15::100::50   VLAN0001           1     EXCLUDE              9
ff15::100::60   VLAN0003           2     INCLUDE              2
ff15::200::1    VLAN0003           1     EXCLUDE              0
ff15::200::2    VLAN0004           2     EXCLUDE              1
(data omitted)
```

Chapter

32. IPv6 Multicast Route Filtering [OS-L3SA]

This chapter provides an overview of IPv6 multicast route filtering and explains how to use it.

32.1 Description of IPv6 multicast route filtering

32.1.1 Overview of IPv6 multicast route filtering

IPv6 multicast route filtering controls IPv6 multicast routes by filtering them. This functionality is used only for IPv6 multicast extranets.

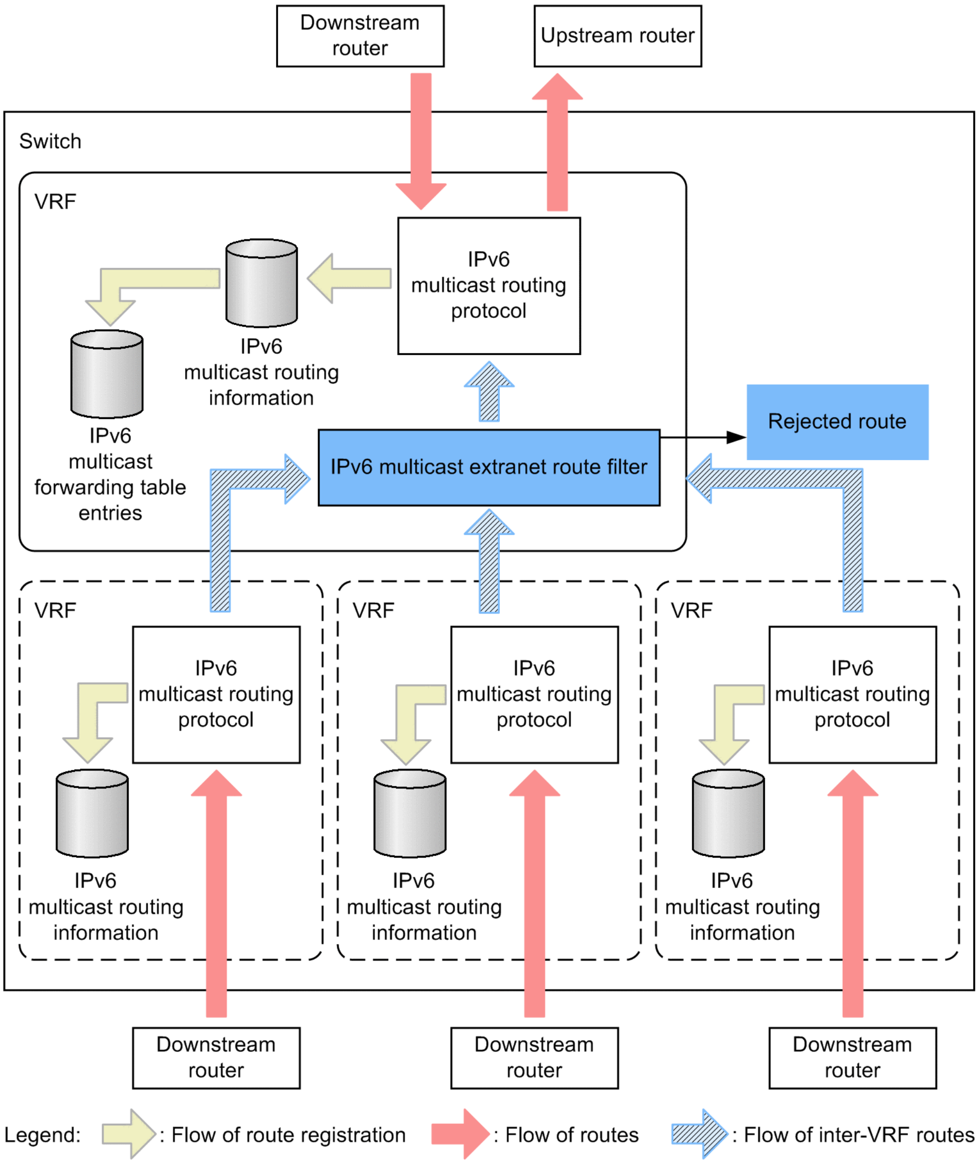
(1) Filtering routes in IPv6 multicast extranets

To enable an IPv6 multicast extranet, forwarding requests need to be exchanged between different VRFs. The Switch uses a method in which routing protocols running in VRFs exchange forwarding requests. When routes are filtered in an IPv6 multicast extranet, the routing protocols running in VRFs filter the forwarding requests. This functionality enables each VRF to determine whether to receive forwarding requests for each destination IP address in a multicast packet. Note that IPv6 multicast routing protocols follow the practice of the route filters used in a unicast extranet because they reference the routing information in the unicast extranet regarding the source IP addresses.

When route filtering is not configured in an IPv6 multicast extranet, the Switch discards all the forwarding requests between the VRFs.

The following figure shows the concept of the IPv6 multicast extranet route filtering.

Figure 32-1: Concept of IPv6 multicast extranet route filtering


Downstream router
Upstream router
Switch
VRF
IPv6 multicast routing protocol
IPv6 multicast routing information
IPv6 multicast forwarding table entries
IPv6 multicast extranet route filter
Rejected route
VRF
IPv6 multicast routing protocol
IPv6 multicast routing information
VRF
IPv6 multicast routing protocol
IPv6 multicast routing information
VRF
IPv6 multicast routing protocol
IPv6 multicast routing information
Downstream router
Downstream router
Downstream router
Legend:
: Flow of route registration
: Flow of routes
: Flow of inter-VRF routes

32.1.2 IPv6 multicast route filtering

For details about IPv6 multicast route filtering, see *29.1.2 Filtering methods*.

The following table describes how IPv6 multicast route filtering behaves when a configuration command is executed.

Table 32-1: Behavior of IPv6 multicast route filtering when a configuration command is executed

Configuration commands	Description
ipv6 prefix-list	This command is not supported, and is ignored if specified.
ipv6 access-list	Only `permit` is used. The IP addresses specified with `deny` are ignored.
route-map	Only `permit` is used. `route-map` specified with `deny` is ignored.
ip as-path access-list	This command is not supported, and is ignored if specified.

Configuration commands	Description
ip community-list standard	This command is not supported, and is ignored if specified.
ip community-list	This command is not supported, and is ignored if specified.

32.1.3 IPv6 multicast extranets

(1) Inter-VRF route filtering

Routes between VRFs can be filtered. If a Switch decides not to include a route in the routing table as a result of filtering, no IPv6 multicast routing information is generated for the route.

(a) Method of applying filters

The filter is set at the upstream VFR, and routes notified from the forwarding destination VRF are filtered using the `ipv6 import multicast inter-vrf` configuration command to select permitted group addresses. The routes permitted by this filtering are added to the IPv6 multicast routing information. The routes to which no filters are applicable are not included.

The table below describes the configuration command used in the route filtering between IPv6 multicast VRFs.

Table 32-2: Configuration command used in the route filtering between IPv6 multicast VRFs

Command name	Filtered routes
ipv6 import multicast inter-vrf	Forwarding requests from a VRF specified as `route-map` are filtered.

The following table describes filter conditions for `route-map` in an IPv6 multicast extranet. Other conditions are ignored.

Table 32-3: Filter conditions for route-map in an IPv6 multicast extranet

Route attribute used as conditions	Description	Configuration commands
Destination IPv6 multicast group address	Specifies the identifier of an access list as a condition and uses the specified filter to filter the IPv6 multicast group address of the destination. A match is assumed if the filter action is permit. If this condition is not specified, all IPv6 multicast group addresses can be permitted.	match ipv6 address ipv6 access-list
VRF ID	Specifies a VRF ID as a condition and compares it with the VRF ID in a route. The forwarding requests from the specified VRF is permitted when the IDs match. If the same VRF ID is specified as a VRF already specified with this command, that ID is ignored. This enables grouping of multiple VRFs and use of the same `route-map` for the grouped VRFs. If this condition is not specified, the forwarding requests from all VRFs are permitted.	match vrf

(b) Setting inter-VRF routes

Specify an inter-VRF route filter. The routes which received the forwarding request from another VRF or the global network are included in the local VRF IPv6 multicast routing information according to the filter conditions. The included route is added as a forwarding-destination interface in the IPv6 multicast routing information. If you execute the `match vrf` configuration command for an IPv6 multicast inter-VRF route filter, the filter compares the VRF ID of the route source VRF against the VRF ID specified by the command. If you do not specify the `match vrf` command, the same filter conditions are applied to all other VRFs and global networks.

(c) Advertising inter-VRF routes by using protocols

If you apply a route filter to a VRF, the filter determines whether to permit forwarding requests from other VRFs or the global network. When a VRF receives a forwarding request from another VRF or the global network and finds a match after it runs the route through a filter, the VRF creates IPv6 multicast routing information and sends the route to the upstream router (if there is one).

For a VRF or the global network to be able to send a forwarding request to the local VRF, in the unicast extranet, configure the local VRF so that it contains the source IP address in the packets sent to the VRFs and global networks from which routes are requested.

32.2 Configuration

32.2.1 List of configuration commands

The following table describes the configuration commands for IPv6 multicast route filtering.

Table 32-4: List of configuration commands

Command name	Description
ipv6 access-list#1	Configures an access list to serve as an IPv6 address filter.
match ipv6 address#2	Configures `route-map` to use the IPv6 address as filter conditions.
match vrf#2	Configures `route-map` to use the VRF value as a filter condition.
ipv6 import multicast inter-vrf#3	Uses a filter to control IPv6 multicast forwarding requests from other VRFs or global networks.

#1

See *19. Access Lists* in the manual *Configuration Command Reference Vol. 1 For Version 11.10*.

#2

See *14. Route Filters (IPv4 and IPv6)* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

#3

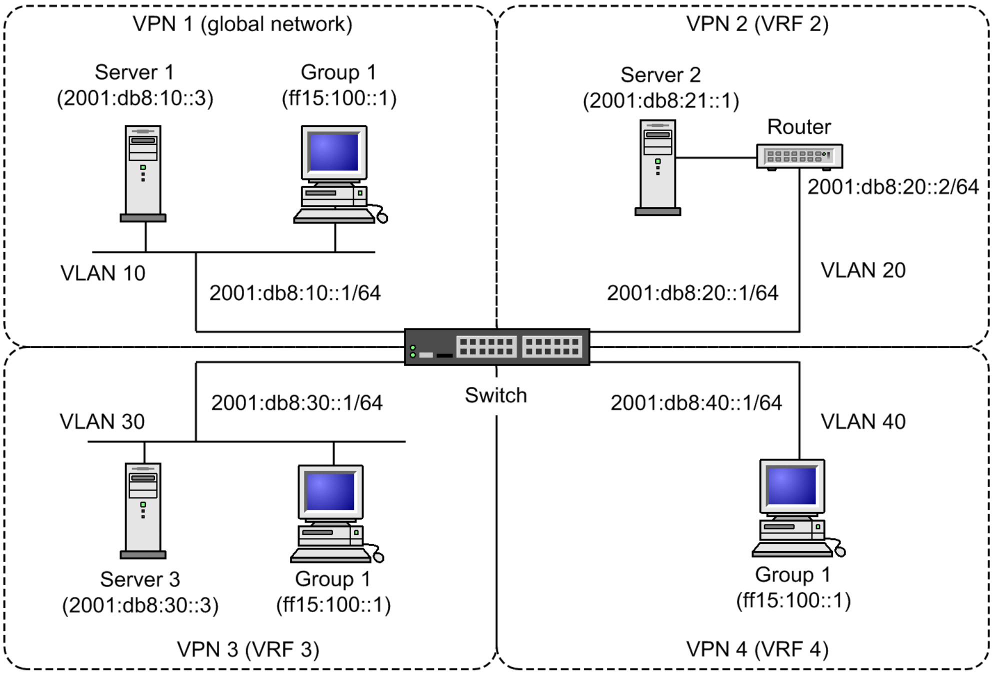
See *30. VRF [OS-L3SA]* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

32.2.2 IPv6 multicast extranets

Configure an IPv6 multicast extranet as described in the figure below.

You can use IPv6 multicast extranet route filtering to apply restrictions.

Figure 32-2: IPv6 multicast extranet configuration example



(1) Permitting requests from all VRFs

Configure VRF 2 so that VRF 2 permits the IPv6 multicast forwarding requests from all VRFs and the global network.

Configure a unicast extranet in advance, and then configure VRF 2 so that it contains the source IP address in the IPv6 multicast packets sent to forwarding destination VRFs and the global network.

Points to note

If filter conditions are not specified in `route-map`, all conditions are permitted.

Command examples

1.
```
(config)# route-map MLT6EXNET permit 10
(config-route-map)# exit
```
Permits all filter conditions.

2.
```
(config)# vrf definition 2
(config-vrf)# ipv6 import multicast inter-vrf MLT6EXNET
(config-vrf)# exit
```
Applies the filter created in step 1 to VRF 2 in the IPv6 multicast extranet, and permits the IPv6 multicast forwarding requests from all VRFs and the global network.

(2) Permitting forwarding requests from specific VRFs

Configure VRF 2 so that VRF 2 permits IPv6 multicast forwarding requests from VRF 3 and VRF 4.

Configure a unicast extranet in advance, and configure VRF 2 so that it contains the source IP address in the IPv6 multicast packets sent to VRF 3 and VRF 4.

Points to note

If this setting is not specified, VRF 2 receives the IPv6 multicast forwarding requests from all VRFs.

Command examples

1.
```
(config)# route-map MLT6EXNET permit 10
(config-route-map)# match vrf 3 4
(config-route-map)# exit
```
Permits the IPv6 multicast forwarding requests from only VRF 3 and VRF 4.

2.
```
(config)# vrf definition 2
(config-vrf)# ipv6 import multicast inter-vrf MLT6EXNET
(config-vrf)# exit
```
Applies the filter created in step 1 to VRF 2 in the IPv6 multicast extranet, and permits the IPv6 multicast forwarding requests from VRF 3 and VRF 4.

(3) Permitting forwarding requests from specific group addresses

Set a configuration to permit IPv6 multicast forwarding requests from only the group addresses in the ff15:100::/32 range.

Configure a unicast extranet in advance, and then configure VRF 2 so that it contains the source IP address in the IPv6 multicast packets sent to forwarding VRFs and the global network.

Points to note

When you set a range of group addresses for use in an extranet, other group addresses are assigned for communication only within VRFs. The locally used group addresses can be used for different purposes in each VRF.

If you do not set this configuration, all group addresses (ff00::/8) are used in an extranet.

Command examples

1.
```
(config)# ipv6 access-list MLT6GROUP
(config-ipv6-acl)# permit ipv6 ff15:100::/32 any
(config-ipv6-acl)# exit
(config)# route-map MLT6EXNET permit 10
(config-route-map)# match ipv6 address MLT6GROUP
(config-route-map)# exit
```
Sets ff15:100::/32 as the range of group addresses to be used in the extranet.

2.
```
(config)# vrf definition 2
(config-vrf)# ipv6 import multicast inter-vrf MLT6EXNET
(config-vrf)# exit
```
Applies the filter created step 1 to VRF 2 in the IPv6 multicast extranet, and allows VRF 2 to accept forwarding requests only from the group addresses in the ff15:100::/32 range.

(4) Configuring a bidirectional IPv6 multicast extranet

This configuration allows mutual communication among the global network, VRF 2, VRF 3, and VRF 4 in the IPv6 multicast extranet.

Configure a unicast extranet in advance, and then configure the VRF or global network you want to connect so that it contains the source IP address in IPv6 multicast packets sent to the global network, VRF 2, VRF 3, and VRF 4.

Points to note

The VRF that is specified with the `match vrf` command of `route-map` does not register routes from another VRF that have the same VRF ID when it imports routes. By writing all the VRFs that perform bidirectional communication in one route map, the VRFs can share the same route map when they import routes.

Command examples

1.
```
(config)# ipv6 access-list MLT6GROUP
(config-ipv6-acl)# permit ipv6 ff15:100::/32 any
(config-ipv6-acl)# exit
(config)# route-map MLT6EXNET permit 10
(config-route-map)# match vrf global 2 3 4
(config-route-map)# match ipv6 address MLT6GROUP
(config-route-map)# exit
```

Permits IPv6 multicast forwarding requests from global networks, VRF 2, VRF 3, and VRF 4 to the group address range ff15:100::/32.

2.
```
(config)# vrf definition global
(config-vrf)# ipv6 import multicast inter-vrf MLT6EXNET
(config-vrf)# exit
(config)# vrf definition 2
(config-vrf)# ipv6 import multicast inter-vrf MLT6EXNET
(config-vrf)# exit
(config)# vrf definition 3
(config-vrf)# ipv6 import multicast inter-vrf MLT6EXNET
(config-vrf)# exit
(config)# vrf definition 4
(config-vrf)# ipv6 import multicast inter-vrf MLT6EXNET
(config-vrf)# exit
```

Applies the filter created in step 1 to the global network, VRF 2, VRF 3 and VRF 4 in the IPv6 multicast extranet, and permits IPv6 multicast forwarding requests among them.

32.3 Operation

32.3.1 List of operation commands

The following table describes the operation commands for IPv6 multicast route filtering.

Table 32-5: List of operation commands

Command name	Description
show ipv6 mcache#	Lists the IPv6 multicast forwarding entries.
show ipv6 mroute#	Lists the IPv6 multicast routing information.
show ipv6 multicast resources#	Displays the number of entries used in IPv6 multicast routing.

#

See *14. IPv6 Multicast Routing Protocols* in the manual *Operation Command Reference Vol. 2 For Version 11.10*.

32.3.2 Checking an IPv6 multicast extranet

Use the `show ipv6 mroute` and `show ipv6 mcache` commands to reference the inter-VRF forwarding entries used in an IPv6 multicast extranet. For the entries that issue forwarding requests to different VRFs, the VRF IDs of the forwarding destinations are displayed in the `incoming` section. For the entries that permit forwarding requests from different VRFs, the VRF IDs are displayed in the `outgoing` section.

Figure 32-3: Results of executing the show ipv6 mroute command

```
> show ipv6 mroute vrf all group ff15:100::1 source 2001:db8:21::1
Date 20XX/12/10 12:40:30 UTC
Total: 4 routes
VRF: global  Total: 1 route, 1 group, 1 source

(S,G) 1 route  -------------------------------------------------
Group Address                      Source Address
ff15:100::1                        2001:db8:21::1
    uptime: 02:00    expires: 02:30    assert: 00:00    flags: FL  protocol SM
    incoming: VRF 2          upstream: Extra  reg-sup: 0s
    outgoing: VLAN0010  uptime 02:30    expires --:--

VRF: 2  Total: 1 routes, 1 groups, 1 sources

(S,G) 1 route  -------------------------------------------------
Group Address                      Source Address
ff15:100::1                        2001:db8:21::1
    uptime: 02:00    expires: 02:30    assert: 00:00    flags: LV  protocol SM
    incoming: VLAN0020       upstream: 2001:db8:20::2  reg-sup: 0s
    outgoing: global    uptime 02:30
    outgoing: VRF 3     uptime 02:30
    outgoing: VRF 4     uptime 02:30

VRF: 3  Total: 1 route, 1 group, 1 source

(S,G) 1 route  -------------------------------------------------
Group Address                      Source Address
ff15:100::1                        2001:db8:21::1
    uptime: 02:00    expires: 02:30    assert: 00:00    flags: FL  protocol SM
    incoming: VRF 2          upstream: Extra  reg-sup: 0s
    outgoing: VLAN0030  uptime 02:30    expires --:--
```

```
VRF: 4  Total: 1 route, 1 group, 1 source

(S,G) 1 route  --------------------------------------------------
Group Address                       Source Address
ff15:100::1                         2001:db8:21::1
    uptime: 02:00    expires: 02:30    assert: 00:00    flags: FL  protocol SM
    incoming: VRF 2          upstream: Extra  reg-sup: 0s
    outgoing: VLAN0040  uptime 02:30    expires --:--

>
```

Figure 32-4: Results of executing the show ipv6 mcache command

```
> show ipv6 mcache vrf all group ff15:100::1 source 2001:db8:21::1
Date 20XX/12/10 12:42:30 UTC
Total: 4 routes
VRF: global  Total: 1 route
- Forwarding entry  ----------------------------------------------------------
Group Address                            Source Address
ff15:100::1                              2001:db8:21::1
    uptime: 00:20    expires: 02:40    flags: U
    incoming:
        VRF 2
    outgoing:
        VLAN0010

VRF: 2  Total: 1 route
- Forwarding entry  ----------------------------------------------------------
Group Address                            Source Address
ff15:100::1                              2001:db8:21::1
    uptime: 00:20    expires: 02:40    flags: U
    incoming:
        VLAN0020
    outgoing:
        VLAN0010      global
        VLAN0030      VRF 3
        VLAN0040      VRF 4

VRF: 3  Total: 1 route
- Forwarding entry  ----------------------------------------------------------
Group Address                            Source Address
ff15:100::1                              2001:db8:21::1
    uptime: 00:20    expires: 02:40    flags:D
    incoming:
        VRF 2
    outgoing:
        VLAN0030

VRF: 4  Total: 1 route
- Forwarding entry  ----------------------------------------------------------
Group Address                            Source Address
ff15:100::1                              2001:db8:21::1
    uptime: 00:20    expires: 02:40    flags:D
    incoming:
        VRF 2
    outgoing:
        VLAN0040

>
```

Chapter

33. Network Partitioning [OS-L3SA]

This chapter describes network partitioning, network configuration examples, and how to use VRF functionality.

33.1 Description

33.1.1 Overview of network partitioning

Network partitioning enables consolidation of networks that are created for different departments or jobs into one network. The following table and figures show how network consolidation is enabled through network partitioning.

Table 33-1: Network consolidation category

Type	Description
Physical consolidation	The devices and lines in different networks are physically integrated into one network while the networks are kept logically separate.
Operational and management consolidation	Layer 3 functionality distributed among different networks is controlled at one location to integrate operation and management.

Figure 33-1: Physical consolidation

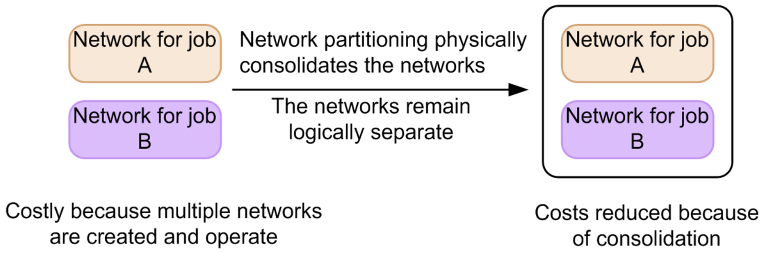
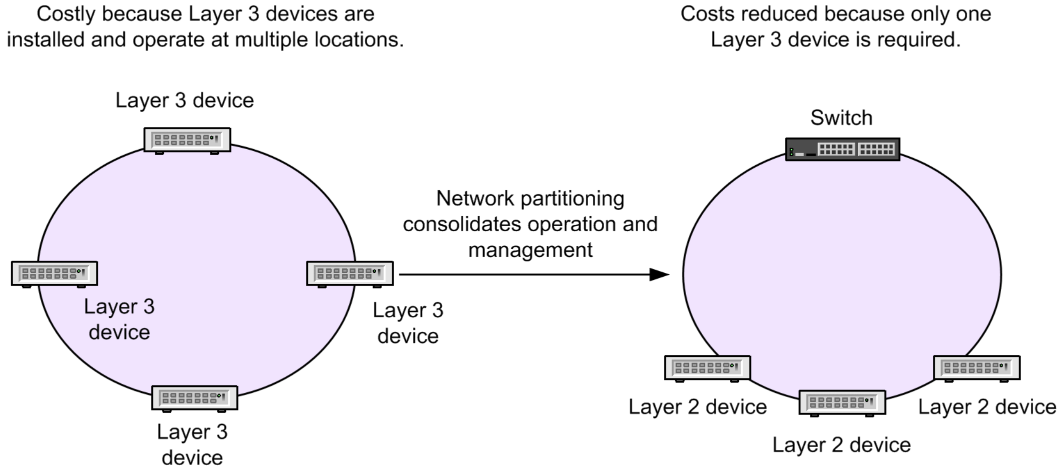

Network for job A
Network for job B
Network partitioning physically consolidates the networks
The networks remain logically separate
Network for job A
Network for job B
Costly because multiple networks are created and operate
Costs reduced because of consolidation

Figure 33-2: Operational and management consolidation


Costly because Layer 3 devices are installed and operate at multiple locations.
Costs reduced because only one Layer 3 device is required.
Layer 3 device
Layer 3 device
Layer 3 device
Layer 3 device
Network partitioning consolidates operation and management
Switch
Layer 2 device
Layer 2 device
Layer 2 device

Network partitioning uses VRF and VLANs to logically partition a network to create VPNs. The following table describes the advantages of network partitioning.

Table 33-2: Advantages of network partitioning

Advantage	Description
Cost reduction	• Fewer devices and lines are required to create a network thanks to physical consolidation, reducing the initial cost of building a network. • Operating costs can be reduced because only one Layer 3 device is required for central management.
Robust security	• Networks can be kept logically separate.

Advantage	Description
Ease of network consolidation	• Multiple networks can be consolidated into one physical network. • Because the networks are logically separate, the same IP addresses can be used in different networks without any need to change them.
Low power consumption	• The network is environmentally friendly because fewer devices and lines are required.

33.1.2 VRF

One of the functions that enable network partitioning is a VRF. A VRF is functionality that retains logically separated routing tables inside the Switch and transfers packets according to each routing table. These distinct routing spaces are called VRF instances. Different VRFs can share the same IP addresses. Routing protocols run independently for each VRF instance.

A VLAN belongs to either a VRF instance or a global network. A global network is a network that is not configured with VRF. Global networks are instances without VRF and they are separate from VRFs.

The following table describes which types of functionality support VRF.

Table 33-3: VRF support status

Item		Supported	Remarks
VLAN	Port VLANs	Y	None
	Protocol VLANs	Y	
	MAC VLANs	Y	
	Tag translation	Y	
	VLAN tunneling	Y	
Layer 2 protocol	Spanning Tree Protocols	Y	None
	Ring Protocol	Y	None
IGMP snooping and MLD snooping		Y	None
Filters		Y	None
QoS		Y	None
Layer 2 authentication	IEEE 802.1X	--	Mutually exclusive with VRF on each device.
	Web authentication	--	Mutually exclusive with VRF on each device.
	MAC-based authentication	--	Mutually exclusive with VRF on each device.
	Authentication VLANs	--	Mutually exclusive with VRF on each device.
DHCP snooping		Y	None
High-reliability functionality	GSRP	Y	None
	VRRP	Y	
	Uplink redundancy	Y	
Failure detection functionality	IEEE 802.3ah/UDLD	Y	None

Item		Supported	Remarks
	Storm control	Y	
	L2 loop detection	Y	
	CFM	Y	
Remote network management	SNMP	Y	For details about SNMP configuration in VRF, see the following: • *22.2 Configuration* in the manual *Configuration Guide Vol. 2 For Version 11.10*
	MIB	P	Some MIBs are used only for global networks. For details, see the following: • *2. Standard MIBs (RFC-Compliant and IETF Draft MIBs)* in the manual *MIB Reference For Version 11.10* • *3. Private MIBs* in the manual *MIB Reference For Version 11.10*
	Traps	P	Some traps are used only for global networks. For details, see the following: • *4. Supported MIB Traps* in the manual *MIB Reference For Version 11.10*
	syslog output	Y	For details about the configuration of syslog output to VRF, see the following: • *23.2.3 Configuring the output of log information to the syslog in VRF [OS-L3SA]* in the manual *Configuration Guide Vol. 2 For Version 11.10*
	Email output	--	Only for global networks
	sFlow statistics	P	Statistics regarding VRF are also collected. However, the extended data formats for routers and gateways are invalid for information collected on the interfaces configured with VRF. Install collectors in global networks.
Neighboring devices information management	LLDP	P	Organizationally-defined TLV extensions apply only to global networks. For details, see the following: • *25.1.3 Notes on using LLDP* in the manual *Configuration Guide Vol. 2 For Version 11.10*
	OADP	P	For VLANs with VRF, OADP does not collect address information. For details, see the following: • *26.1.3 Notes on using OADP* in the manual *Configuration Guide Vol. 2 For Version 11.10*

Item		Supported	Remarks
Port mirroring		Y	None
Layer 3 forwarding	IPv4 unicast forwarding	Y	See this chapter, as well as the following: • *7.11 Description of a VRF [OS-L3SA]* • *7.12 VRF configuration [OS-L3SA]* • *7.13 VRF operation [OS-L3SA]*
	IPv4 unicast inter-VRF forwarding	Y	For details, see the following: • *7.11.3 Extranet* • *8.2.7 Configuring static routes across VRFs [OS-L3SA]* • *13.1.6 Extranet [OS-L3SA]* • *13.2.8 Extranet [OS-L3SA]* • *13.3.10 Checking extranet [OS-L3SA]*
	IPv4 multicast forwarding	Y	For details, see the following: • *14.4.5 IPv4 multicasting for a VRF [OS-L3SA]* • *15.1 Configuration* • *16. IPv4 Multicast Route Filtering [OS-L3SA]*
	IPv4 multicast inter-VRF forwarding	Y	
	IPv6 unicast forwarding	Y	See this chapter, as well as the following: • *23.11 Description of VRF [OS-L3SA]* • *23.12 VRF configuration [OS-L3SA]* • *23.13 VRF operation [OS-L3SA]*
	IPv6 unicast inter-VRF forwarding	Y	For details, see the following: • *23.11.3 Extranet* • *24.2.7 Configuring a static route between VRFs [OS-L3SA]* • *29.1.6 Extranet [OS-L3SA]* • *29.2.8 Extranet [OS-L3SA]* • *29.3.9 Checking extranet [OS-L3SA]*
	IPv6 multicast forwarding	Y	For details about IPv6 multicast forwarding with VRF, see the following: • *30.4.11 IPv6 multicasting for a VRF [OS-L3SA]* • *31.1 Configuration* • *32. IPv6 Multicast Route Filtering [OS-L3SA]*
	IPv6 multicast inter-VRF forwarding	Y	
Null interface		Y	Global networks and VRFs share one null interface.

Item		Supported	Remarks
Policy-based routing		Y	Inter-VRF routing is available. For details about configuration of inter-VRF policy-based routing, see the following: • *4.2.3 Configuring an extranet with policy-based routing*
RA		Y	None
DHCP and BOOTP relay agents		Y	For details about configuration of relay agents in VRF, see the following: • *5.2.4 Configuring a VRF configuration [OS-L3SA]* • *5.2.5 Configuring an extranet configuration [OS-L3SA]*
DHCP server functionality		--	Only for global networks
IPv6 DHCP relay		--	Only for global networks
IPv6 DHCP server functionality		--	Only for global networks
IPv4 static routing		Y	Inter-VRF routing is available. For details about configuration, see the following: • *8.2.6 Configuring a static route in a VRF [OS-L3SA]* • *8.2.7 Configuring static routes across VRFs [OS-L3SA]*
IPv6 static routing		Y	Inter-VRF routing is available. For details about configuration, see the following: • *24.2.6 Configuring a static route for a VRF [OS-L3SA]* • *24.2.7 Configuring a static route between VRFs [OS-L3SA]* • *24.2.8 Configuring a static route across VRFs by using an IPv6 link-local address as the next hop [OS-L3SA]*
IPv4 unicast routing protocols	RIP	Y	For details about configuration, see the following: • *9.2.8 Applying RIP for a VRF [OS-L3SA]* • *10.2.7 Applying OSPF for a VRF*
	OSPF	Y	
	BGP4	Y	For details, see the following: • *12.1.4 BGP4 functionality for VRFs* • *12.2.11 Configuring BGP4 for a VRF*
	Route filtering	Y	None

Item		Supported	Remarks
IPv6 unicast routing protocols	RIPng	Y	For details about configuration, see the following: • *25.2.5 Applying RIPng for a VRF [OS-L3SA]* • *26.2.7 Applying OSPFv3 for a VRF*
	OSPFv3	Y	
	BGP4+	Y	For details, see the following: • *28.1.4 BGP4+ functionality for VRFs* • *28.2.11 Configuring BGP4+ for a VRF*
	Route filtering	Y	None
IPv4 multicast routing protocols	IGMP	Y	For details, see the following: • *14.4.5 IPv4 multicasting for a VRF [OS-L3SA]* • *15.1 Configuration* • *16. IPv4 Multicast Route Filtering [OS-L3SA]*
	PIM-SM	Y	
	PIM-SSM	Y	
IPv6 multicast routing protocols	MLD	Y	For details, see the following: • *30.4.11 IPv6 multicasting for a VRF [OS-L3SA]* • *31.1 Configuration* • *32. IPv6 Multicast Route Filtering [OS-L3SA]*
	PIM-SM	Y	
	PIM-SSM	Y	
Operation and maintenance	ping	Y	None
	traceroute	Y	
	telnet	Y	
	ftp	Y	
	tftp	Y	
	Telnet login	Y	For details, see the following: • *10.1.9 Permitting login from a remote operation terminal when using VRF [OS-L3SA]* in the manual *Configuration Guide Vol. 1 For Version 11.10* • *10.1.10 Setting the IP address that permits login from a remote operation terminal when using VRF [OS-L3SA]* in the manual *Configuration Guide Vol. 1 For Version 11.10*
	FTP login	Y	

Item		Supported	Remarks
	DNS resolver	--	Only for global networks
	NTP	Y	For details about configuration, see the following: • *11.1.6 Synchronizing time on VRF by using NTP [OS-L3SA]* in the manual *Configuration Guide Vol. 1 For Version 11.10*

Legend: Y: Supported, P: Partially supported, --: Not supported

33.1.3 Network configuration examples

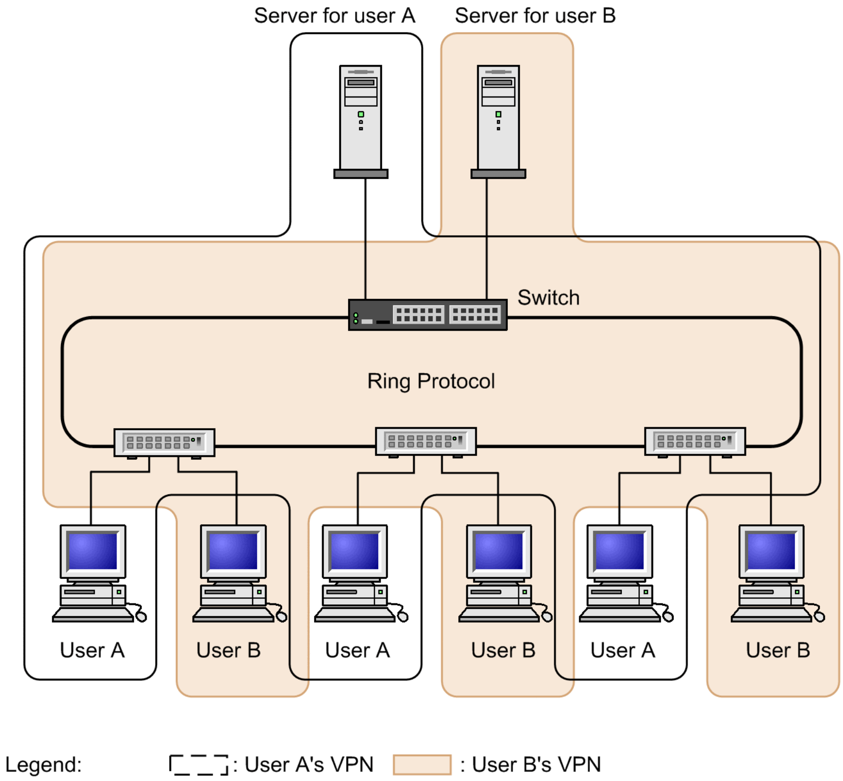
You can use network partitioning to create a variety of networks. This subsection describes how to create networks with typical examples of applying network partitioning.

(1) Network partitioning with the Ring Protocol

One of the applications for which network partitioning is most suitable is a network that uses the Ring Protocol. By using the Ring Protocol, you can create a highly reliable network because the Ring Protocol can quickly switch routes if a failure occurs. Network partitioning can centralize Layer 3 functionality in one location, which simplifies network operation.

The following figure shows an example of how to use network partitioning to create a network that uses the Ring Protocol. In the figure, user A and user B belong to different VPNs and are unable to communicate with each other.

Figure 33-3: Network partitioning with the Ring Protocol

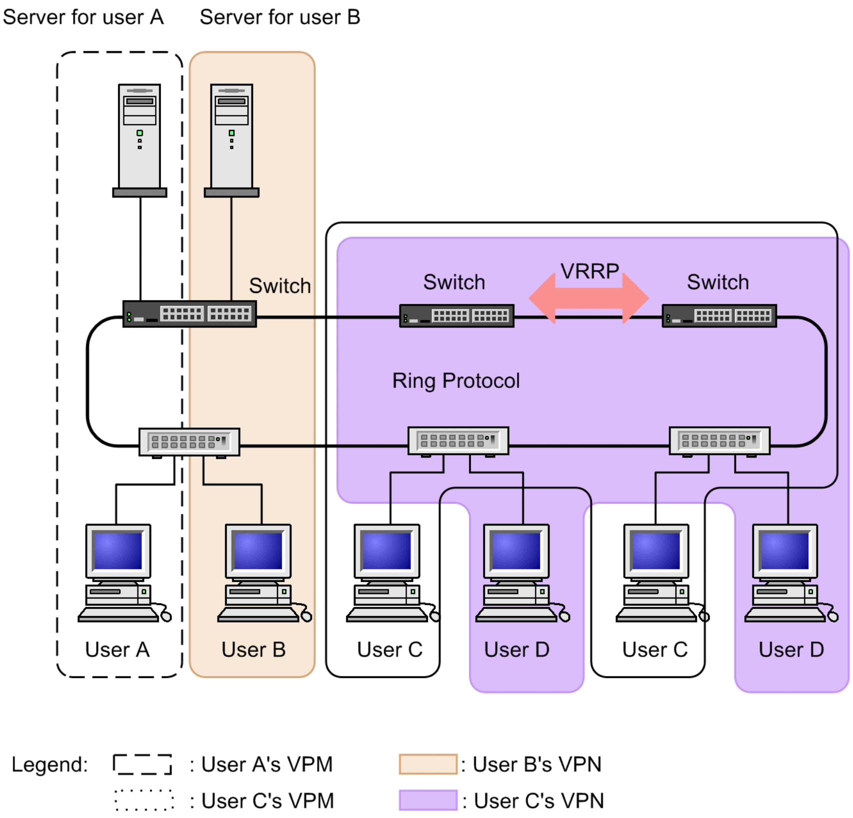


(2) Adding Layer 3 concentrators

When the number of VRFs or sites in a network is too large, you can add Layer 3 devices to handle them separately. Run VRRP to further improve the reliability of devices when high reliability is required.

The following figure shows an example of adding Layer 3 concentrators to a network.

Figure 33-4: Adding Layer 3 concentrators

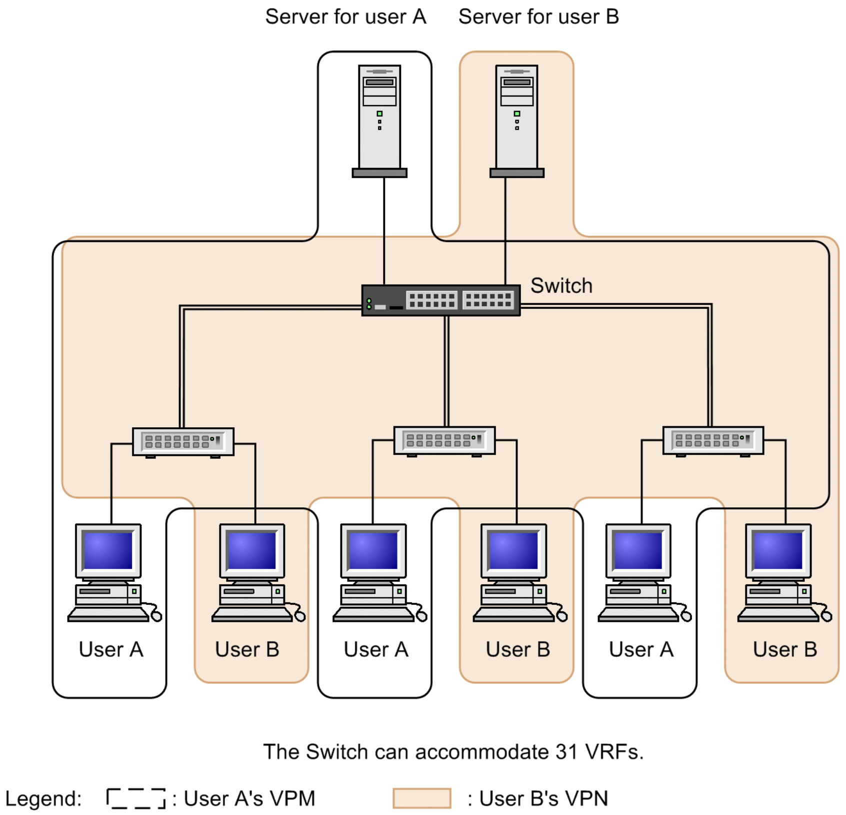


(3) Creating a network without Layer 2 protocols

The VRF functionality can be used for networks that do not use Layer 2 protocols.

The following figure shows an example of creating a network without Layer 2 protocols.

Figure 33-5: Creating a network without Layer 2 protocols



(4) Creating an extranet

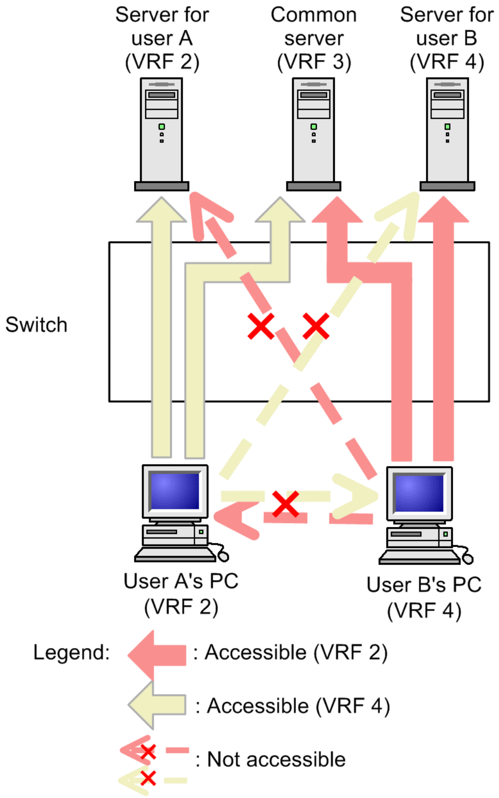
In an extranet, communication is usually disabled between VRFs, but is allowed between special VRFs. This method allows you to create a network that permits users to access a common server while maintaining security between users.

Use one of the following inter-VRF forwarding techniques to create an extranet:

- Route exchange between VRFs
- Static routing across VRFs
- Policy-based routing

The figure below shows an example of an extranet configuration.

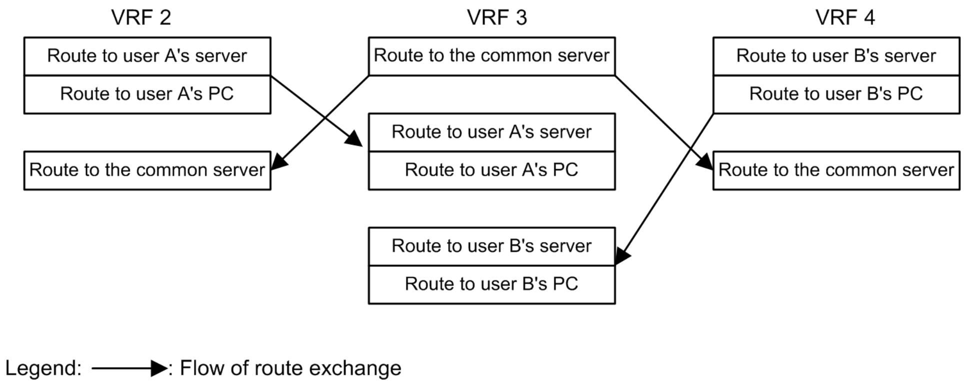
Figure 33-6: Extranet configuration via route exchange



- User A (VRF 2) and user B (VRF 4) cannot communicate with each other because they do not share routing information.
- User A (VRF 2) and the common server (VRF 3) can communicate with each other, and user B (VRF 4) and the common server (VRF 3) can communicate with each other, because routes are exchanged between them.

The following figure shows the flow of routing information stored on the Switch and the flow of route exchange.

Figure 33-7: Routing information on the Switch

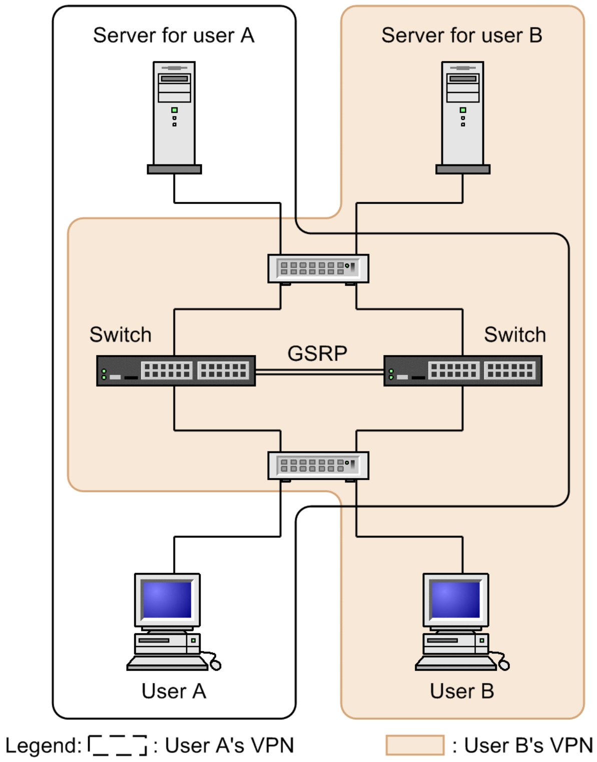


(5) Network partitioning with the GSRP

VRF can use GSRP to provide redundancy. GSRP enables high-speed device switching in the event of a failure, allowing you to create a highly reliable network. Another advantage is that GSRP can by itself provide redundancy for Layer 2 and Layer 3.

The following figure shows an example of network partitioning configuration that uses GSRP. In the figure, user A and user B belong to different VPNs and are unable to communicate with each other.

Figure 33-8: Network partitioning with the GSRP



33.2 Configuration

33.2.1 List of configuration commands

The following table describes the configuration commands for network partitioning.

Table 33-4: List of configuration commands

Command name	Description
ipv6 maximum routes	Specifies the maximum number of IPv6 routes in a VRF and the threshold for outputting an operation warning message.
maximum routes	Specifies the maximum number of IPv4 routes in a VRF and the threshold for outputting an operation warning message.
vrf definition	Configures a VRF.
arp-limit#1	Configures the maximum number of ARP entries for VRF.
vrf forwarding#1	Specifies VRF on an interface.
nd-limit#2	Configures the maximum number of NDP entries for VRF.

#1

See *2. IPv4, ARP, and ICMP* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

#2

See *16. IPv6, NDP, and ICMPv6* in the manual *Configuration Command Reference Vol. 2 For Version 11.10*.

33.2.2 Configuring VRF

To run VRF on a Switch, you need to configure VRF, assign VRF to interfaces, and configure VRF for different types of functionality. For details about how to configure VRF for each functionality, see the configuration section of the relevant functionality.

Points to note

Configure VRF, and then assign VRF to interfaces.

Command examples

1.
```
(config)# vrf definition 2
(config-vrf)# maximum routes 500 70
(config-vrf)# arp-limit 50
(config-vrf)# ipv6 maximum routes 400 70
(config-vrf)# nd-limit 50
(config-vrf)# exit
```

Configures VRF 2. The maximum number of IPv4 routes is set to 500, and the threshold for outputting an operation warning message is set to 70%. The maximum number of ARP entries is set to 50. The maximum number of IPv6 routes is set to 400, and the threshold for outputting an operation warning message is set to 70%. The maximum number of NDP entries is set to 50.

2.
```
(config)# interface loopback 2
(config-if)# vrf forwarding 2
(config-if)# ip address 192.168.0.2
(config-if)# ipv6 address 2001:db8::2
(config-if)# exit
```

   Specifies VRF for the loopback interface. Specifies VRF 2 for loopback interface 2. Sets 192.168.0.2 as the IPv4 address and 2001:db8::2 as the IPv6 address.

3.
```
(config)# interface vlan 10
(config-if)# vrf forwarding 2
(config-if)# ip address 192.168.10.1 255.255.255.0
(config-if)# ipv6 enable
(config-if)# ipv6 address 2001:db8:10::1/64
```

   Specifies VRF for a VLAN interface. Sets VRF 2, IPv4 address 192.168.10.1, and subnet mask 255.255.255.0 for VLAN ID 10. Enables IPv6, and sets 2001:db8:10::1 as the IPv6 address and 64 as the prefix length.

33.3 Operation

33.3.1 List of operation commands

The following table describes the operation commands for network partition.

Table 33-5: List of operation commands

Command name	Description
show vlan[#1]	Shows VLAN information.
show ip vrf[#2]	Shows IPv4 information for VRF.
show ipv6 vrf[#3]	Shows the IPv6 information for VRF.

#1

See *19. VLAN* in the manual *Operation Command Reference Vol.1 For Version 11.10*.

#2

See *6. IPv4 Routing Protocols* in the manual *Operation Command Reference Vol. 2 For Version 11.10*.

#3

See *13. IPv6 Routing Protocols* in the manual *Operation Command Reference Vol. 2 For Version 11.10*.

33.3.2 Checking VRF information

Use the `show ip vrf` command to check the IPv4 routing information in VRFs and the status of IPv4 interfaces. Use the `show ipv6 vrf` command to check the IPv6 routing information in VRFs and the status of IPv6 interfaces.

Figure 33-9: Results of executing the show ip vrf command

```
> show ip vrf 2
Date 20XX/12/10 12:00:00 UTC
VRF              Routes        ARP
2                270/500       7/50
>
```

Figure 33-10: Results of executing the show ip vrf detail command

```
> show ip vrf 2 detail
Date 20XX/12/10 12:00:00 UTC
VRF 2
  Maximum routes: 500, Warn threshold: 70%, Current routes: 270
  Maximum ARP entries: 50, Current ARP entries: 7
  Import inter-vrf: -
Interface
Name           Local              Remote          Status
VLAN0010       192.168.10.1/24    192.168.10.255  Up
loopback2      127.0.0.1/8        127.0.0.1       Up
loopback2      192.168.0.2/32     192.168.0.2     Up
>
```

Figure 33-11: Results of executing the show ipv6 vrf command

```
> show ipv6 vrf 2
Date 20XX/12/10 12:00:00 UTC
VRF              Routes          Neighbor
2                200/400         7/50
>
```

Figure 33-12: Results of executing the show ipv6 vrf detail command

```
> show ipv6 vrf 2 detail
Date 20XX/12/10 12:00:00 UTC
VRF 2
  Maximum routes: 400, Warn threshold: 70%, Current routes: 200
  Maximum Neighbor entries: 50, Current Neighbor entries: 7
  Import inter-vrf: -
Interface
Name           Address                                     Status
VLAN0010       2001:db8:10::1/64                           Up
VLAN0010       fe80::212:e2ff:fe20:b000%VLAN0010/64        Up
loopback2      ::1/128                                     Up
loopback2      2001:db8::2/128                             Up
loopback2      fe80::1%loopback2/64                        Up
>
```

Use the `show vlan` command to check the VRF to which a VLAN belongs.

Figure 33-13: Results of executing the show vlan command

```
> show vlan 10
Date 20XX/12/10 12:00:00 UTC
VLAN counts:1
VLAN ID:10    Type:Port based      Status:Up
  Learning:On            Tag-Translation:
  BPDU Forwarding:       EAPOL Forwarding:
  Router Interface Name:VLAN0010
  VRF:2
  IP Address:192.168.10.1/24
             2001:db8:10::1/64
             fe80::212:e2ff:fe22:9db3/64
  Source MAC address: 0012.e222.9db3(System)
  Description:VLAN0010
  Spanning Tree:PVST+(802.1D)
  AXRP RING ID:      AXRP VLAN group:
  GSRP ID:      GSRP VLAN group:    L3:
  IGMP snooping:     MLD snooping:
  Untagged(8)    :1/0/5-12
  Tagged(2)      :1/0/25-26
>
```

Appendix

A. Relevant standards

A.1 IP, ARP, and ICMP

Table A-1: Relevant standards and recommendations for IP version 4

Name (month and year issued)	Title
RFC 791 (September 1981)	Internet Protocol
RFC 792 (September 1981)	Internet Control Message Protocol
RFC 826 (November 1982)	An Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware
RFC 922 (October 1984)	Broadcasting Internet datagrams in the presence of subnets
RFC 950 (August 1985)	Internet Standard Subnetting Procedure
RFC 1027 (October 1987)	Using ARP to implement transparent subnet gateways
RFC 1122 (October 1989)	Requirements for Internet hosts-communication layers
RFC 1519 (September 1993)	Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy
RFC 1812 (June 1995)	Requirements for IP Version 4 Routers

A.2 DHCP and BOOTP relay agents

Table A-2: Relevant standards and recommendations for DHCP and BOOTP relay agents

Name (month and year issued)	Title
RFC 1542 (October 1993)	Clarifications and Extensions for the Bootstrap Protocol
RFC 1812 (June 1995)	Requirements for IP Version 4 Routers
RFC 2131 (March 1997)	Dynamic Host Configuration Protocol

A.3 DHCP server functionality

Table A-3: Relevant standards for the DHCP server functionality

Name (month and year issued)	Title
RFC 2131 (March 1997)	Dynamic Host Configuration Protocol
RFC 2132 (March 1997)	DHCP Options and BOOTP Vendor Extensions
RFC 2136 (April 1997)	Dynamic Updates in the Domain Name System (DNS UPDATE)
RFC 3679 (January 2004)	Unused Dynamic Host Configuration Protocol (DHCP) Option Codes

A.4 RIP

Table A-4: Relevant standards and recommendations for RIP

Name (month and year issued)	Title
RFC 1058 (June 1988)	Routing Information Protocol

Name (month and year issued)	Title
RFC 1519 (September 1993)	Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy
RFC 2453 (November 1998)	RIP Version 2
RFC 4822 (February 2007)	RIPv2 Cryptographic Authentication

A.5 OSPF [OS-L3SA]

Table A-5: Relevant standards and recommendations for OSPF

Name (month and year issued)	Title
RFC 1519 (September 1993)	Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy
RFC 2328 (April 1998)	OSPF Version 2
RFC 2370 (July 1998)	The OSPF Opaque LSA Option
RFC 3101 (January 2003)	The OSPF Not-So-Stubby Area (NSSA) Option
RFC 3137 (June 2001)	OSPF Stub Router Advertisement
RFC 3623 (November 2003)	Graceful OSPF Restart
RFC 5309 (October 2008)	Point-to-Point Operation over LAN in Link State Routing Protocols

A.6 BGP4 [OS-L3SA]

Table A-6: Relevant standards and recommendations for BGP4

Name (month and year issued)	Title
RFC 1519 (September 1993)	Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy
RFC 1997 (August 1996)	BGP Communities Attribute
RFC 2385 (August 1998)	Protection of BGP Sessions via the TCP MD5 Signature Option
RFC 2918 (September 2000)	Route Refresh Capability for BGP-4
RFC 4271 (January 2006)	A Border Gateway Protocol 4 (BGP-4)
RFC 4456 (August 2006)	BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)
RFC 5065 (August 2007)	Autonomous System Confederations for BGP
RFC 5492 (February 2009)	Capabilities Advertisement with BGP-4
draft-ietf-idr-avoid-transition-04 (December 2005)	Avoid BGP Best Path Transitions from One External to Another
draft-ietf-idr-restart-13 (July 2006)	Graceful Restart Mechanism for BGP#

#: Only the receiving speaker functionality is supported.

A.7 IPv4 multicasting

Table A-7: Relevant standards and recommendations for IP multicasting

Name (month and year issued)	Title
RFC 2236 (November 1997)	Internet Group Management Protocol, Version2
RFC 2362 (June 1998)	Protocol Independent Multicast-Sparse Mode (PIM-SM): Specification
RFC 2934 (October 2000)	Protocol Independent Multicast MIB for IPv4
RFC 3376 (October 2002)	Internet Group Management Protocol, Version 3
RFC 4601 (August 2006)#2	Protocol Independent Multicast-Sparse Mode (PIM-SM): Specification (revised)
draft-ietf-pim-sm-v2-new-05 (March 2002)#1	Protocol Independent Multicast-Sparse Mode (PIM-SM): Specification (revised)
draft-ietf-pim-sm-bsr-07 (March 2006)#2	Bootstrap Router (BSR) Mechanism for PIM

#1: Only the sections related to PIM-SSM conform to this standard.

#2: Only the sections related to generation IDs of the PIM hello option and fragmentation of bootstrap messages conform to this standard.

A.8 IPv6, NDP, and ICMPv6

Table A-8: Relevant standards and recommendations for an IPv6 network

Name (month and year issued)	Title
RFC 2373 (July 1998)	IP Version 6 Addressing Architecture
RFC 2460 (December 1998)	Internet Protocol, Version 6 (IPv6) Specification
RFC 2461 (December 1998)	Neighbor Discovery for IP Version 6 (IPv6)
RFC 2462 (December 1998)	IPv6 Stateless Address Autoconfiguration
RFC 2463 (December 1998)	Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification
RFC 2710 (October 1999)	Multicast Listener Discovery for IPv6
draft-ietf-ipv6-deprecate-rh0-01 (June 2007)	Deprecation of Type 0 Routing Headers in IPv6

A.9 IPv6 DHCP relays [OP-DH6R]

Table A-9: Relevant standards and recommendations for IPv6 DHCP relays

Name (month and year issued)	Title
RFC 3315 (July 2003)	Dynamic Host Configuration Protocol for IPv6 (DHCPv6)

A.10 IPv6 DHCP servers

Table A-10: Relevant standards and recommendations for IPv6 DHCP servers

Name (month and year issued)	Title
RFC 3315 (July 2003)	Dynamic Host Configuration Protocol for IPv6 (DHCPv6)

Name (month and year issued)	Title
RFC 3319 (July 2003)	Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiation Protocol (SIP) Servers
RFC 3633 (December 2003)	IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6
RFC 3646 (December 2003)	DNS Configuration Options for DHCPv6
RFC 3736 (April 2004)	Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6
RFC 4075 (March 2005)	Simple Network Time Protocol (SNTP) Configuration Option for DHCPv6

A.11 RIPng

Table A-11: Relevant standards and recommendations for RIPng

Name (month and year issued)	Title
RFC 2080 (January 1997)	RIPng for IPv6

A.12 OSPFv3 [OS-L3SA]

Table A-12: Relevant standards and recommendations for OSPFv3

Name (month and year issued)	Title
RFC 2740 (December 1999)	OSPF for IPv6
RFC 3137 (June 2001)	OSPF Stub Router Advertisement
RFC 5309 (October 2008)	Point-to-Point Operation over LAN in Link State Routing Protocols
draft-kompella-ospf-opaquev2-00 (October 2002)	OSPFv2 Opaque LSAs in OSPFv3
draft-ietf-ospf-ospfv3-graceful-restart-04 (May 2006)	OSPFv3 Graceful Restart

A.13 BGP4+ [OS-L3SA]

Table A-13: Relevant standards and recommendations for BGP4+

Name (month and year issued)	Title
RFC 1997 (August 1996)	BGP Communities Attribute
RFC 2385 (August 1998)	Protection of BGP Sessions via the TCP MD5 Signature Option
RFC 2545 (March 1999)	Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing
RFC 2918 (September 2000)	Route Refresh Capability for BGP-4
RFC 4271 (January 2006)	A Border Gateway Protocol 4 (BGP-4)
RFC 4456 (August 2006)	BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)
RFC 4760 (January 2007)	Multiprotocol Extensions for BGP-4
RFC 5065 (August 2007)	Autonomous System Confederations for BGP

Name (month and year issued)	Title
RFC 5492 (February 2009)	Capabilities Advertisement with BGP-4
draft-ietf-idr-avoid-transition-04 (December 2005)	Avoid BGP Best Path Transitions from One External to Another
draft-ietf-idr-restart-13 (July 2006)	Graceful Restart Mechanism for BGP#

#: Only the receiving speaker functionality is supported.

A.14 IPv6 multicasting

Table A-14: Relevant standards and recommendations for IPv6 multicasting

Name (month and year issued)	Title
RFC 2362 (June 1998)	Protocol Independent Multicast-Sparse Mode (PIM-SM): Specification
RFC 2710 (October 1999)	Multicast Listener Discovery (MLD) for IPv6
RFC 3810 (June 2004)	Multicast Listener Discovery Version 2 (MLDv2) for IPv6
RFC 4601 (August 2006)#3	Protocol Independent Multicast-Sparse Mode (PIM-SM): Specification (revised)
draft-ietf-pim-sm-v2-new-03 (July 2001)#1	Protocol Independent Multicast-Sparse Mode (PIM-SM): Specification (revised)
draft-ietf-pim-sm-v2-new-05 (March 2002)#2	Protocol Independent Multicast-Sparse Mode (PIM-SM): Specification (revised)
draft-ietf-pim-sm-bsr-07 (March 2006)#3	Bootstrap Router (BSR) Mechanism for PIM

#1: Only the IPv6-related sections conform to this standard.

#2: Only the sections related to PIM-SSM conform to this standard.

#3: Only the sections related to generation IDs of the PIM hello option and fragmentation of bootstrap messages conform to this standard.

Index