



DANOS-Vyatta edition

Disaggregated Network Operating System Version 2009a

Multicast Routing Configuration Guide
October 2020

Contents

Chapter 1. Copyright Statement.....	1
Chapter 2. Preface.....	2
Document conventions.....	2
Chapter 3. About This Guide.....	4
Chapter 4. Multicast Routing Overview.....	5
IP multicast.....	5
Multicast channels.....	6
Multicast addresses.....	6
Multicast routing protocols.....	7
Types of multicast.....	7
Internet standard multicast.....	7
Source-specific multicast.....	7
Supported standards.....	8
RFCs.....	8
MIBs.....	8
Chapter 5. Multicast Configuration.....	9
Before you begin.....	9
Basic PIM-SM multicast configuration.....	9
Configure R1.....	10
Configure R2.....	11
Configure RP.....	12
Chapter 6. Multicast Routing Commands.....	13
clear ip mroute statistics.....	13
clear ipv6 mroute statistics.....	13
interfaces ip multicast ttl-threshold.....	13
interfaces bridge multicast-router.....	14
interfaces_ip_pim_mode.....	15
interfaces_ipv6_pim_mode.....	16
interfaces switch multicast-router.....	17
interfaces switch vif ip igmp.....	17

interfaces switch vif ip igmp immediate-leave.....	19
interfaces switch vif ip igmp join-group.....	19
interfaces switch vif ip igmp querier.....	20
interfaces switch vif ip igmp static-group.....	21
interfaces switch vif ip multicast.....	22
interfaces switch vif ip pim.....	22
interfaces switch vif ip pim mode.....	24
interfaces switch vif ip pim state-refresh.....	25
interfaces switch vif ipv6 pim.....	25
interfaces switch vif ipv6 pim mode.....	26
interfaces switch vif ipv6 pim state-refresh.....	27
monitor command.....	28
monitor protocol multicast.....	29
mtrace.....	30
protocols pim dense-group.....	31
protocols pim dense-group ipv6-address.....	32
protocols multicast ip log.....	33
protocols multicast ipv6 log.....	34
protocols multicast ip log-warning.....	35
protocols multicast ip route-limit.....	36
protocols multicast ip routing.....	37
protocols multicast ipv6 log-warning.....	38
protocols multicast ipv6 route-limit.....	38
protocols multicast ipv6 routing.....	39
reset ip mroute.....	40
reset ipv6 mroute.....	40
service application rule icmpv6 name.....	41
show dataplane multicast ip route forwarding.....	42
show dataplane multicast ip route interface.....	42
show dataplane multicast ip route.....	42
show dataplane multicast ipv6 route forwarding.....	43
show dataplane multicast ipv6 route interface.....	43
show dataplane multicast ipv6 route.....	43

show dataplane multicast ip.....	44
show dataplane multicast ipv6.....	44
show ip mroute.....	45
show ip multicast interface.....	47
show ip multicast mroute.....	47
show ip rpf.....	48
show ipv6 mroute.....	49
show platform dataplane mroute.....	50
show platform dataplane mroute6.....	50
show ipv6 multicast interface.....	51
show ipv6 rpf.....	52
show monitoring protocols multicast.....	52
Chapter 7. Supported Data Plane Interfaces.....	54
Chapter 8. VRF Support.....	55
VRF support for multicast.....	55
Command support for VRF routing instances.....	56
Chapter 9. List of Acronyms.....	60

Chapter 1. Copyright Statement

© 2020 IP Infusion Inc. All Rights Reserved.

This documentation is subject to change without notice. The software described in this document and this documentation are furnished under a license agreement or nondisclosure agreement. The software and documentation may be used or copied only in accordance with the terms of the applicable agreement. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's internal use without the written permission of IP Infusion Inc.

IP Infusion Inc.
3965 Freedom Circle, Suite 200
Santa Clara, CA 95054
+1 408-400-1900

<http://www.ipinfusion.com/>.

For support, questions, or comments via E-mail, contact:

support@ipinfusion.com.

Trademarks:

IP Infusion is a trademark of IP Infusion. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Use of certain software included in this equipment is subject to the IP Infusion, Inc. End User License Agreement at <http://www.ipinfusion.com/license>. By using the equipment, you accept the terms of the End User License Agreement.


Chapter 2. Preface


Document conventions


The document conventions describe text formatting conventions, command syntax conventions, and important notice formats used in this document.


Notes, cautions, and warnings

Notes, cautions, and warning statements may be used in this document. They are listed in the order of increasing severity of potential hazards.

 **Note:** A Note provides a tip, guidance, or advice, emphasizes important information, or provides a reference to related information.

 **Attention:** An Attention statement indicates a stronger note, for example, to alert you when traffic might be interrupted or the device might reboot.

 **CAUTION:** A Caution statement alerts you to situations that can be potentially hazardous to you or cause damage to hardware, firmware, software, or data.

 **DANGER:** A Danger statement indicates conditions or situations that can be potentially lethal or extremely hazardous to you. Safety labels are also attached directly to products to warn of these conditions or situations.

Text formatting conventions

Text formatting conventions such as boldface, italic, or Courier font are used to highlight specific words or phrases.

Format	Description
bold text	Identifies command names. Identifies keywords and operands.
<i>italic text</i>	Identifies emphasis. Identifies variables. Identifies document titles.
<code>Courier font</code>	Identifies CLI output. Identifies command syntax examples.

Command syntax conventions

Bold and italic text identify command syntax components. Delimiters and operators define groupings of parameters and their logical relationships.

Convention	Description
bold text	Identifies command names, keywords, and command options.
<i>italic text</i>	Identifies a variable.
[]	Syntax components displayed within square brackets are optional. Default responses to system prompts are enclosed in square brackets.
{ x y z }	A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options.
x y	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, for example, passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, <i>member</i> [<i>member</i> ...].
\	Indicates a "soft" line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Chapter 3. About This Guide

This guide describes how to configure multicast routing on DANOS-Vyatta edition.

Chapter 4. Multicast Routing Overview

IP multicast

There are four types of transmission over a network:

- Unicast—In unicast transmission, a single sender transmits a message to a single receiver (one-to-one). Unicast transmission is signaled to the network by using an IP address in the unicast range. Unicast transmission is supported in both IPv4 and IPv6.
- Broadcast—In broadcast transmission, a sender transmits a message to all possible destinations (“all-hosts on this subnet”) and all receivers receive a copy of the message (one-to-many). In IPv4, broadcast transmission is signaled to the network by using the “broadcast address” 255.255.255.255 or a directed broadcast address formed from the network prefix but using all binary 1s for the host portion of the address. Broadcast transmission is not supported in IPv6; instead, IPv6 defines the “all-nodes” multicast address.
- Multicast—In multicast transmission, a group of receivers interested in a particular stream forms a group (one-to-many). The sender sends the message from its unicast address to the multicast group address. The network routers are responsible for propagating the message to all interested receivers. Multicast groups are identified using a special range of addresses. In IPv4, the former class D addresses compose the multicast range of addresses. In IPv6, multicast addresses are identified from the prefix FF00::/8.
- Anycast—In anycast transmission, a message from a single sender is sent to the topologically “closest” node in a group of potential receivers (one-to-one-of-many). Anycast is used in network load balancing and in certain network protocols, such as Domain Name System (DNS).

IP multicast uses multicast transmission over the IP infrastructure, transmitting to a multicast address at the IP routing level. Networks using IP multicast deliver content to multiple users in various groups—for example, IP multicast is often used for content delivery such as streaming media and IPTV. IP multicast is described in RFC 1112, *Host Extensions for IP Multicasting*.

Note:

For specific information about the Internet Group Management Protocol (IGMP) and the Multicast Listener Discovery (MLD) protocol, see *IGMP Configuration Guide*.

For specific information about Protocol Independent Multicast (PIM), see *PIM Configuration Guide*.

For specific information about Multicast Source Discovery Protocol (MSDP), see *MSDP Configuration Guide*.

Multicast channels

To receive a particular multicast data stream, hosts join a multicast group. The group is identified by its multicast address. The communication between the host and router for this purpose is managed by using the Internet Group Management Protocol (IGMP).

Hosts send an IGMP Join message to their local multicast router, signaling their intention to join the group (G), as represented by a multicast address. The source device (S) that delivers the content sends the message addressed to the multicast address of G. The multicast “channel” is the combination of the IP address of the content source and multicast address of the group—called an (S, G) pair. For messages for which the source can be any device, the S is replaced by the asterisk wildcard (“*”), which means “any source.”

Multicast addresses

For IPv4, addresses in the range 224.0.0.0 to 239.255.255.255 are reserved for multicasting. In the classful addressing system, this is registered as class D. In the Classless Inter-Domain Routing (CIDR) addressing model, the prefix of this group is 224.0.0.0/4.

Within the IPv4 multicast range, address assignments are specified in RFC 5771, *IANA Guidelines for IPv4 Multicast Address Assignments*. [Table 1: Multicast address assignments](#) lists a summary of these assignments. Note that, of these addresses, the network address 224.0.0.0 is guaranteed not to be assigned to any group.

Table 1. Multicast address assignments

Addresses	Usage
Link Local Scope	
224.0.0.1	All systems on this subnet
224.0.0.2	All routers on this subnet
224.0.0.13	All PIM routers address group
Global Scope	
224.0.1.0 to 238.255.255.255	Allocated for multicast traffic across the Internet. SSM reserves the range 232.0.0.0/8. The addresses 224.0.1.39 and 224.0.1.40 are used for Auto-RP negotiation. You can assign the remaining IP addresses to your ISM applications.
Administrative Scope (AS)	
239.0.0.0/8	Allocated for organizations that own an AS number to multicast across the Internet. The AS number of the organization is embedded in the second and third octets of the multicast IP address. For example, AS64501 is 0x-FBF5, with FB and F5 (or 251 and 245 in decimal) representing the second and third octets of the IP address, respectively. The resulting subnet 233.251.24.0 is globally reserved for AS64201 to use. These addresses are called GLOP addresses.

For IPv6, multicast addressing is specified in RFC 3513, *IP Version 6 Addressing Architecture*. In essence, IPv6 multicast addresses are derived from the FF00::/8 prefix.

Multicast routing protocols

The multicast routing protocols supported by the router are IGMP, Multicast Listener Discovery (MLD), and Protocol Independent Multicast (PIM).

- IGMP controls multicast communication between hosts and multicast routers on IPv4 networks, allowing hosts to manage membership of multicast groups. MLD performs this function on IPv6 networks.
- PIM controls multicast communication between multicast routers, so that they can track packet distribution.

Types of multicast

In IP networks, multicast information is propagated through the use of distribution trees created by the multicast routers. Routers create source trees that provide the shortest path to each destination; these trees are called shortest path trees (SPTs). The source of the multicast message is always the root of an SPT.

Multicast depends on which of two models of multicast is in effect:

- [Internet standard multicast](#)
- [Source-specific multicast](#)

Internet standard multicast

In Internet Standard Multicast (ISM), a receiver can receive messages from any source and from multiple sources. In ISM, receivers are not aware of sources; they express interest in receiving a traffic stream by subscribing to the group of interest (G).

In the ISM model, the multicast router discovers and tracks all multicast sources that are sending messages to an address, maintaining information about all sources, and routing data from all sources to all interested receivers. The router does this by maintaining a state table of (S, G) entries. When the number of sources is large, the burden on the multicast router can become heavy.

Source-specific multicast

In Source-Specific Multicast (SSM), receivers are aware of sources. A receiver explicitly requests a stream from a sender in a group the receiver has joined, using a Join message that specifies the source (an (S, G) Join), and explicitly excludes the use of the wildcard for source—that is, it disallows the (*, G) Join.

Because SSM-enabled hosts track sources of the multicast transmission themselves, multicast routers do not need to discover the multicast sources themselves, and they need to store and maintain only the (*, G) state in the multicast routing table. This greatly reduces the burden on the multicast router.

For IPv4, the destination addresses must be in the range 232.0.0.0/8. For IPv6, the addresses must be in the range FF3x::/96.

SSM is specified in RFC 3569, *An Overview of Source-Specific Multicast (SSM)* and RFC 4607, *Source-Specific Multicast for IP*.

Supported standards

This section presents the following topics:

- [RFCs](#)
- [MIBs](#)

RFCs

The IP Infusion Inc. implementation of multicast routing complies with the following standards:

- RFC 1112: *Host Extensions for IP Multicasting*
- RFC 2236: *Internet Group Management Protocol, Version 2*
- RFC 2710: *Multicast Listener Discovery (MLD) for IPv6*
- RFC 3376: *Internet Group Management Protocol, Version 3*
- RFC 3569: *An Overview of Source-Specific Multicast (SSM)*
- RFC 3810: *Multicast Listener Discovery version 2 (MLDv2) for IPv6*
- RFC 4604: *Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast*
- RFC 4607: *Source-Specific Multicast for IP*

MIBs

The Vyatta implementation of multicast routing supports the following Simple Management Network Protocol (SNMP) management information bases (MIBs).


- IGMP-MIB, RFC2933: *Internet Group Management Protocol MIB*
- IPMROUTE, RFC 2932: *IPv4 Multicast Routing MIB*
- MLD-MIB, RFC 3019: *IP Version 6 Management Information Base for The Multicast Listener Discovery Protocol*

For a list of all MIBs supported on the router, see *Remote Management Configuration Guide*.

Chapter 5. Multicast Configuration

Before you begin


It is assumed that routers depicted in the examples that follow are configured to provide basic connectivity. This configuration includes the host-name, Ethernet, or data plane interfaces, and gateway-address. See *Quick Start Guide* for information on basic system configuration.

 **Note:** In the router, a data plane interface is an abstraction that represents the underlying physical or virtual Ethernet interface of the system. The terms Ethernet interface and data plane interface are synonymous in this guide.

Basic PIM-SM multicast configuration

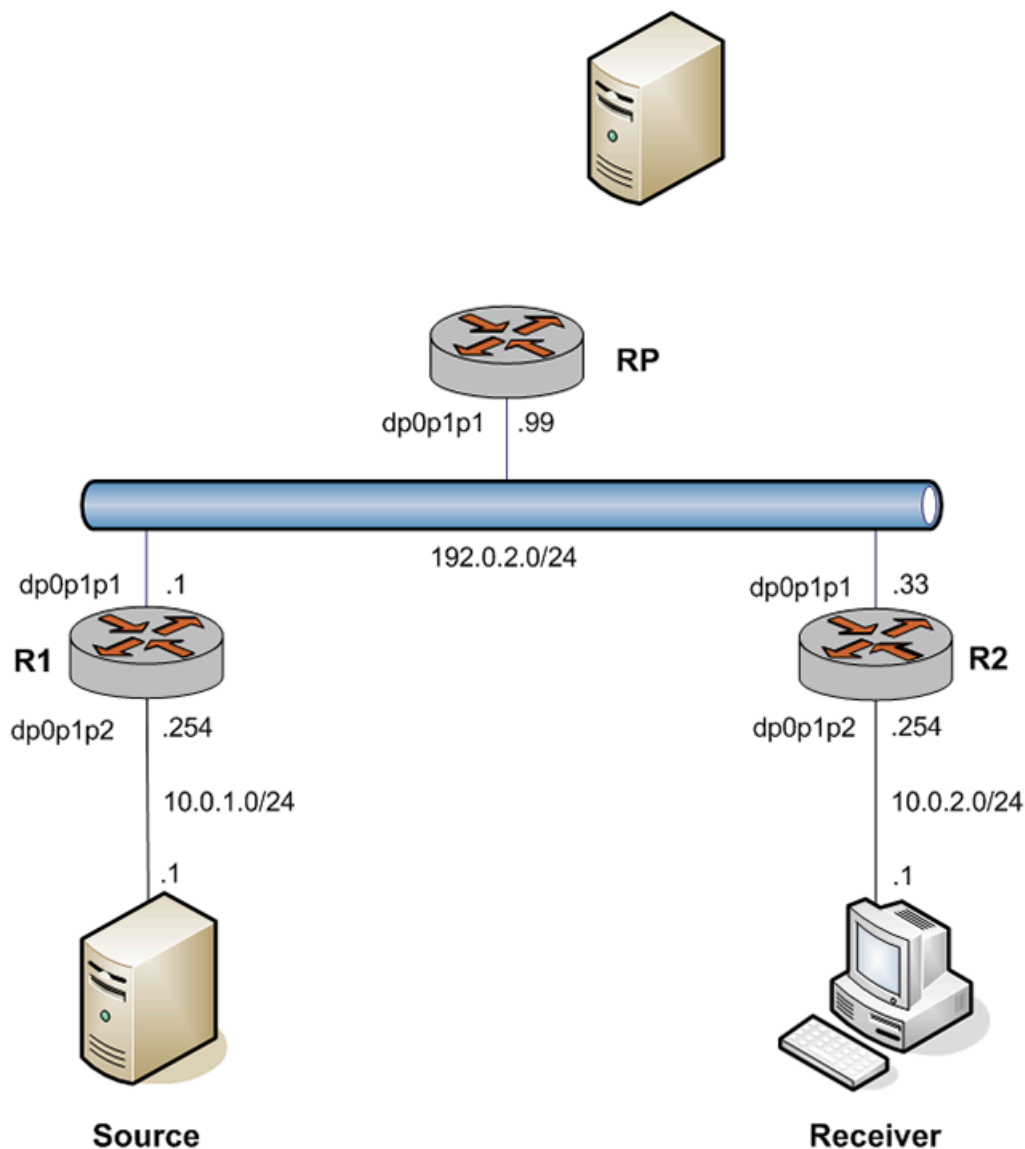
This section presents a configuration example that provides PIM sparse mode multicast capability between the sender of multicast traffic (Source) and a receiver of multicast traffic (Receiver). Three routers are configured for this example: R1, R2 and RP.

R1 and R2 are configured for PIM sparse mode operation and point to RP as the rendezvous point router. RP is configured as the rendezvous point router.

 **Note:** It is assumed that the Sender and Receiver contain software to send and receive multicast traffic.

When you have finished the example, these systems will be configured as shown in the following figure.

Figure 1. Basic PIM-SM multicast setup



Configure R1

The multicast network in the example extends from R1 through the 192.0.2.0/24 LAN segment to R2 and RP. In this example, you configure R1 for PIM sparse mode multicast operation.

[Table 2: Creating a PIM sparse mode configuration on R1](#) creates a PIM sparse mode multicast configuration on R1.

Table 2. Creating a PIM sparse mode configuration on R1

Step	Command
Configure PIM sparse mode on dp0p1p1.	<code>vyatta@R1# set interfaces dataplane dp0p1p1 ip pim mode sparse</code>

Table 2. Creating a PIM sparse mode configuration on R1 (continued)

Step	Command
Commit the configuration.	<pre>vyatta@R1# commit</pre>
View the configuration.	<pre>vyatta@R1# show interfaces dataplane dp0p1p1 ip pim { mode sparse }</pre>
Configure PIM sparse mode on dp0p1p2.	<pre>vyatta@R1# set interfaces dataplane dp0p1p2 ip pim mode sparse</pre>
Commit the configuration.	<pre>vyatta@R1# commit</pre>
View the configuration.	<pre>vyatta@R1# show interfaces dataplane dp0p1p2 ip pim { mode sparse }</pre>
Configure multicast routing.	<pre>vyatta@R1# set protocols multicast ip routing</pre>
Specify the location of the rendezvous point router.	<pre>vyatta@R1# set protocols pim rp-address 192.0.2.99</pre>
Commit the configuration.	<pre>vyatta@R1# commit</pre>
View the configuration.	<pre>vyatta@R1# show protocols multicast { ip { routing { } } } pim { rp-address 192.0.2.99 { } }</pre>

Configure R2

The multicast network in the example extends from R1 through the 192.0.2.0/24 LAN segment to R2 and RP. In this example, you configure R2 for PIM sparse mode multicast operation.

[Table 3: Creating a PIM sparse mode configuration on R2](#) creates a PIM sparse mode multicast configuration on R2.

Table 3. Creating a PIM sparse mode configuration on R2

Step	Command
Configure PIM sparse mode on dp0p1p1.	<pre>vyatta@R2# set interfaces dataplane dp0p1p1 ip pim mode sparse</pre>
Commit the configuration.	<pre>vyatta@R2# commit</pre>
View the configuration.	<pre>vyatta@R2# show interfaces dataplane dp0p1p1 ip pim { mode sparse }</pre>
Configure PIM sparse mode on dp0p1p2.	<pre>vyatta@R2# set interfaces dataplane dp0p1p2 ip pim mode sparse</pre>
Commit the configuration.	<pre>vyatta@R2# commit</pre>

Table 3. Creating a PIM sparse mode configuration on R2 (continued)

Step	Command
View the configuration.	<pre>vyatta@R2# show interfaces dataplane dp0p1p2 ip pim { mode sparse }</pre>
Configure multicast routing.	<pre>vyatta@R2# set protocols multicast ip routing</pre>
Specify the location of the rendezvous point router.	<pre>vyatta@R2# set protocols pim rp-address 192.0.2.99</pre>
Commit the configuration.	<pre>vyatta@R2# commit</pre>
View the configuration.	<pre>vyatta@R2# show protocols multicast { ip { routing { } } } pim { rp-address 192.0.2.99 { } }</pre>

Configure RP

The multicast network in the example extends from R1 through the 192.0.2.0/24 LAN segment to R2 and RP. In this example, you configure RP for PIM sparse mode multicast operation.

[Table 4: Creating a PIM sparse mode configuration on RP](#) creates a PIM sparse mode multicast configuration on RP.

Table 4. Creating a PIM sparse mode configuration on RP

Step	Command
Configure PIM sparse mode on dp0p1p1.	<pre>vyatta@RP# set interfaces dataplane dp0p1p1 ip pim mode sparse</pre>
Commit the configuration.	<pre>vyatta@RP# commit</pre>
View the configuration.	<pre>vyatta@RP# show interfaces dataplane dp0p1p1 ip pim { mode sparse }</pre>
Configure multicast routing.	<pre>vyatta@RP# set protocols multicast ip routing</pre>
Specify the location of the rendezvous point router.	<pre>vyatta@RP# set protocols pim rp-address 192.0.2.99</pre>
Commit the configuration.	<pre>vyatta@RP# commit</pre>
View the configuration.	<pre>vyatta@RP# show protocols multicast { ip { routing { } } } pim { rp-address 192.0.2.99 { } }</pre>

Chapter 6. Multicast Routing Commands

clear ip mroute statistics

Clears IPv4 statistics for multicast routing.

```
clear ip mroute statistics [ group group [ source source ] ]
```

When used with no option, this command clears all IPv4 statistics for multicast routing.

group

Clears statistics for the specified IPv4 multicast group in IPv4 multicast address format.

source

Used in source-specific multicast. Clears statistics for the specified IPv4 multicast source. The format is an IPv4 multicast address.

Operational mode

Use this command to clear statistics for multicast routing.

clear ipv6 mroute statistics

Clears IPv6 statistics for multicast routing.

```
clear ipv6 mroute statistics [ group group [ source source ] ]
```

When used with no option, this command clears all IPv6 statistics for multicast routing.

group group

Clears statistics for the specified IPv6 multicast group. The format is an IPv6 multicast address.

source source

Used in source-specific multicast. Clears statistics for the specified IPv6 multicast source. The format is an IPv6 multicast address.

Operational mode

Use this command to clear IPv6 statistics for multicast routing.

interfaces ip multicast ttl-threshold

Sets the time-to-live (TTL) threshold for multicast packets.

```
set interfaces interface ip multicast ttl-threshold ttl
```

```
delete interfaces interface ip multicast ttl-threshold
show interfaces interface ip multicast ttl-threshold
```

The TTL for multicast packets is 0. The default value 0 means that all multicast packets are forwarded out to the interface.

interface

The type keyword and identifier of an interface. For detailed keywords and arguments that can be specified as interfaces, refer to [Supported Dataplane Interfaces](#).

ttl

The TTL value is the hop count. The range is 0 through 255. The default is 0.

Configuration mode

```
interfaces interface {
  ip {
    multicast {
      ttl-threshold ttl
    }
  }
}
```

Use this command to configure the TTL threshold for multicast packets being forwarded from the specified interface. Only multicast packets with a TTL value greater than the threshold are forwarded.

Use the `set` form of this command to configure the multicast TTL value.

Use the `delete` form of this command to restore the default TTL value for multicast.

Use the `show` form of this command to show multicast TTL configuration.

interfaces bridge multicast-router

Sets the interfaces bridge to a multicast-router.

```
set interfaces bridge interface-name multicast-router
delete interfaces bridge interface-name multicast-router
show interfaces bridge interface-name multicast-router
```

interface-name

Name of the interfaces.

Configuration mode

```

interfaces {
    bridge <interface-name> {
        multicast-router
    }
}

```

Use this command to define the parameters for mcast router.

Use the `set interfaces bridge <interface-name> multicast-router` form of this command to set mcast router parameters.

Use the `delete interfaces bridge <interface-name> multicast-router` form of this command to remove mcast router parameters.

Use the `show interfaces bridge <interface-name> multicast-router` form of this command to display mcast router configuration.

interfaces_ip_pim_mode

Sets the interfaces instances to a pim mode.

```

set interfaces interface-name ip pim mode [mode-name | sparse-dense | sparse-dense-passive]

```

```

delete interfaces interface-name ip pim mode [mode-name | sparse-dense | sparse-dense-passive]

```

```

show interfaces interface-name ip pim mode [sparse-dense | sparse-dense-passive]

```

interface-name

Name of the interfaces.

Configuration mode

```

interfaces <interface-name>{
    ip {
        pim {
            mode {
                sparse-dense
                sparse-dense-passive
            }
        }
    }
}

```

Use this command to define the parameters for mcast of PIM mode.

Use the `set interfaces interface-name ip pim mode [sparse-dense | sparse-dense-passive]` form of this command to set PIM parameters.

Use the `delete interfaces interface-name ip pim mode [sparse-dense | sparse-dense-passive]` form of this command to remove PIM mode parameters.

Use the `show interfaces interface-name ip pim mode [sparse-dense | sparse-dense-passive]` form of this command to display PIM mode configuration.

interfaces_ipv6_pim_mode

Sets the interfaces instances to a pim mode.

```
set interfaces interface-name ipv6 pim mode [mode-name | sparse-dense | sparse-dense-passive]
```

```
delete interfaces interface-name ipv6 pim mode [mode-name | sparse-dense | sparse-dense-passive]
```

```
show interfaces interface-name ipv6 pim mode [sparse-dense | sparse-dense-passive]
```

interface-name

Name of the interfaces.

Configuration mode

```
interfaces <interface-name>{
    ipv6 {
        pim {
            mode {
                sparse-dense
                sparse-dense-passive
            }
        }
    }
}
```

Use this command to define the parameters for mcast of PIM mode.

Use the `set interfaces interface-name ipv6 pim mode [sparse-dense | sparse-dense-passive]` form of this command to set PIM parameters.

Use the `delete interfaces interface-name ipv6 pim mode [sparse-dense | sparse-dense-passive]` form of this command to remove PIM mode parameters.

Use the `show interfaces interface-name ipv6 pim mode [sparse-dense | sparse-dense-passive]` form of this command to display PIM mode configuration.

interfaces switch multicast-router

Sets the interfaces switch to a multicast-router.

```
set interfaces switch name multicast-router
```

```
delete interfaces switch name multicast-router
```

```
show interfaces switch name multicast-router
```

name

Name of the switch.

Configuration mode

```
interfaces {
    switch <name> {
        multicast-router
    }
}
```

Use this command to define the parameters for mcast router.

Use the `set interfaces switch <name> multicast-router` form of this command to set mcast router parameters.

Use the `delete interfaces switch <name> multicast-router` form of this command to remove mcast router parameters.

Use the `show interfaces switch <name> multicast-router` form of this command to display mcast router configuration.

interfaces switch vif ip igmp

Sets the interface instances to a igmp.

```
set interface switch namevifnumberipigmp [access-groupvalue | enforce-router-alert | limit-exception value | limit value|offlink|query-intervalvalue|query-max-response-timevalue|robustness-variablevalue|startup-query-countvalue|startup-query-intervalvalue|versionvalue]
```

```
delete interface switch namevifnumberipigmp [access-groupvalue | enforce-router-alert | limit-exception value | limit value|offlink|query-intervalvalue|query-max-response-timevalue|robustness-variablevalue|startup-query-countvalue|startup-query-intervalvalue|versionvalue]
```

```
show interface switch namevifnumberipigmp [access-groupvalue | enforce-router-alert | limit-exception value | limit value|offlink|query-intervalvalue|query-max-response-timevalue|robustness-variablevalue|startup-query-countvalue|startup-query-intervalvalue|versionvalue]
```

Configuration mode

```

interface {
    switch <name> {
        vif <number> {
            ip {
                igmp {
                    access-group <value>
                    enforce-router-alert
                    last-member-query-count
                    <value>
                    last-member-query-interval <value>
                    <value>
                    <value>
                    <value>
                    <value>
                    <value>
                    <value>
                    <value>
                    limit-exception <value>
                    limit <value>
                    offlink
                    query-interval <value>
                    query-max-response-time
                    <value>
                    robustness-variable
                    <value>
                    startup-query-count
                    <value>
                    startup-query-interval
                    <value>
                    version <value>
                }
            }
        }
    }
}

```

Use this command to define the parameters for mcast of igmp.

Use the `set interfaces switch <name> vif <number> ip igmp [access-group <value> | enforce-router-alert | last-member-query-count <value> | last-member-query-interval <value> | limit-exception <value> | limit <value> | offlink | query-interval <value> | query-max-response-time <value> | robustness-variable <value> | startup-query-count <value> | startup-query-interval <value> | version <value>]` form of this command to set igmp parameters.

Use the `delete interfaces switch <name> vif <number> ip igmp [access-group <value> | enforce-router-alert | last-member-query-count <value> | last-member-query-interval <value> | limit-exception <value> | limit <value> | offlink | query-interval <value> | query-max-response-time <value> | robustness-variable <value> | startup-query-count <value> | startup-query-interval <value> | version <value>]` form of this command to remove igmp parameters.

Use the `show interfaces switch <name> vif <number> ip igmp [access-group <value> | enforce-router-alert | last-member-query-count <value> | last-member-query-interval`

<value> | limit-exception <value> | limit <value> | offlink | query-interval <value> | query-max-response-time <value> | robustness-variable <value> | startup-query-count <value> | startup-query-interval <value> | version <value>] form of this command to display igmp configuration.

interfaces switch vif ip igmp immediate-leave

Sets the interface instances to a igmp immediate-leave.

```
set interface switch name vif number ip igmp immediate-leave group-list value
```

```
delete interface switch name vif number ip igmp immediate-leave group-list value
```

```
set interface switch name vif number ip igmp immediate-leave group-list value
```

Configuration mode

```
interface {
    switch <name> {
        vif <number> {
            ip {
                igmp {
                    immediate-leave {
                        group-list <value>
                    }
                }
            }
        }
    }
}
```

Use this command to define the parameters for mcast of igmp.

Use the `set interfaces switch <name> vif <number> ip igmp immediate-leave group-list <value>` form of this command to set igmp parameters.

Use the `delete interfaces switch <name> vif <number> ip igmp immediate-leave group-list <value>` form of this command to remove igmp parameters.

Use the `show interfaces switch <name> vif <number> ip igmp immediate-leave group-list <value>` form of this command to display igmp configuration.

interfaces switch vif ip igmp join-group

Sets the interface instances to a igmp join-group.

```
set interface switch name vif number ip igmp join-group multicast-group source value
```

```
delete interface switch name vif number ip igmp join-group multicast-group source value
set interface switch name vif number ip igmp join-group multicast-group source value
```

Configuration mode

```
interface {
    switch <name> {
        vif <number> {
            ip {
                igmp {
                    join-group {
                        source
                    }
                }
            }
        }
    }
}
```

Use this command to define the parameters for mcast of igmp.

Use the `set interfaces switch <name> vif <number> ip igmp join-group source <value>` form of this command to set igmp parameters.

Use the `delete interfaces switch <name> vif <number> ip igmp join-group source <value>` form of this command to remove igmp parameters.

Use the `show interfaces switch <name> vif <number> ip igmp join-group source <value>` form of this command to display igmp configuration.

interfaces switch vif ip igmp querier

Sets the interface instances to a igmp querier.

```
set interface switch name vif number ip igmp querier [forced | querier-timeout value]
delete interface switch name vif number ip igmp querier [forced | querier-timeout value]
show interface switch name vif number ip igmp querier [forced | querier-timeout value]
```

Configuration mode

```
interface {
    switch <name> {
        vif <number> {
            ip {
                igmp {
```



```

                                querier {
                                    forced
                                    querier-timeout
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

Use this command to define the parameters for mcast of igmp.

Use the `set interfaces switch <name> vif <number> ip igmp querier forced querier-timeout <value>` form of this command to set igmp parameters.

Use the `delete interfaces switch <name> vif <number> ip igmp querier forced querier-timeout <value>` form of this command to remove igmp parameters.

Use the `show interfaces switch <name> vif <number> ip igmp querier forced querier-timeout <value>` form of this command to display igmp configuration.

interfaces switch vif ip igmp static-group

Sets the interface instances to a igmp static-group.

```
set interface switch namevifnumberipigmpstatic-groupmulticast-groupsourcevalue
```

```
delete interface switch namevifnumberipigmpstatic-groupmulticast-groupsourcevalue
```

```
set interface switch namevifnumberipigmpstatic-groupmulticast-groupsourcevalue
```

Configuration mode

```

interface {
    switch <name> {
        vif <number> {
            ip {
                igmp {
                    static-group {
                        source
                    }
                }
            }
        }
    }
}

```

Use this command to define the parameters for mcast of igmp.

Use the `set interfaces switch <name> vif <number> ip igmp static-group source <value>` form of this command to set igmp parameters.

Use the `delete interfaces switch <name> vif <number> ip igmp static-group source <value>` form of this command to remove igmp parameters.

Use the `show interfaces switch <name> vif <number> ip igmp static-group source <value>` form of this command to display igmp configuration.

interfaces switch vif ip multicast

Sets the interface instances to a multicast ttl-threshold.

```
set interface switch namevifnumberipmulticastttl-threshold value
```

```
delete interface switch namevifnumberipmulticastttl-threshold value
```

```
show interface switch namevifnumberipmulticastttl-threshold value
```

Configuration mode

```
interface {
    switch <name> {
        vif <number> {
            ip {
                multicast {
                    ttl-threshold
                }
            }
        }
    }
}
```

Use this command to define the parameters for mcast of multicast ttl-threshold.

Use the `set interfaces switch <name> vif <number> ip multicast ttl-threshold <value>` form of this command to set multicast ttl-threshold parameters.

Use the `delete interfaces switch <name> vif <number> ip multicast ttl-threshold <value>` form of this command to remove multicast ttl-threshold parameters.

Use the `show interfaces switch <name> vif <number> ip multicast ttl-threshold <value>` form of this command to display multicast ttl-threshold configuration.

interfaces switch vif ip pim

Sets the interface instances to a pim.

```
set interface switch name vif number ip pim [bsr-border | dr-priority value | exclude-genid | hello-holdtime value | hello-interval value | neighbor-filter value | propagation-delay value | unicast-bsm]
```

```
delete interface switch name vif number ip pim [bsr-border | dr-priority value | exclude-genid | hello-holdtime value | hello-interval value | neighbor-filter value | propagation-delay value | unicast-bsm]
```

```
show interface switch name vif number ip pim [bsr-border | dr-priority value | exclude-genid | hello-holdtime value | hello-interval value | neighbor-filter value | propagation-delay value | unicast-bsm]
```

Configuration mode

```
interface {
    switch <name> {
        vif <number> {
            ip {
                pim {
                    bsr-border
                    dr-priority <value>
                    exclude-genid
                    hello-holdtime <value>
                    hello-interval <value>
                    neighbor-filter <value>
                    propagation-delay <value>
                    unicast-bsm
                }
            }
        }
    }
}
```

Use this command to define the parameters for mcast of pim.

Use the `set interfaces switch <name> vif <number> ip pim [bsr-border | dr-priority <value> | exclude-genid | hello-holdtime <value> | hello-interval <value> | neighbor-filter <value> | propagation-delay <value> | unicast-bsm]` form of this command to set pim parameters.

Use the `delete interfaces switch <name> vif <number> ip pim [bsr-border | dr-priority <value> | exclude-genid | hello-holdtime <value> | hello-interval <value> | neighbor-filter <value> | propagation-delay <value> | unicast-bsm]` form of this command to remove pim parameters.

Use the `show interfaces switch <name> vif <number> ip pim [bsr-border | dr-priority <value> | exclude-genid | hello-holdtime <value> | hello-interval <value> | neighbor-filter <value> | propagation-delay <value> | unicast-bsm]` form of this command to display pim configuration.

interfaces switch vif ip pim mode

Sets the interface instances to a pim mode.

```
set interface switch name vif number ip pim mode [dense | dense-passive | sparse | sparse-dense | sparse-dense-passive | sparse-passive]
```

```
delete interface switch name vif number ip pim mode [dense | dense-passive | sparse | sparse-dense | sparse-dense-passive | sparse-passive]
```

```
show interface switch name vif number ip pim mode [dense | dense-passive | sparse | sparse-dense | sparse-dense-passive | sparse-passive]
```

Configuration mode

```
interface {
    switch <name> {
        vif <number> {
            ip {
                pim {
                    mode {
                        dense
                        dense-passive
                        sparse
                        sparse-dense
                        sparse-dense-passive
                        sparse-passive
                    }
                }
            }
        }
    }
}
```

Use this command to define the parameters for mcast of pim mode.

Use the `set interfaces switch <name> vif <number> ip pim mode [dense | dense-passive | sparse | sparse-dense | sparse-dense-passive | sparse-passive]` form of this command to set pim parameters.

Use the `delete interfaces switch <name> vif <number> ip pim mode [dense | dense-passive | sparse | sparse-dense | sparse-dense-passive | sparse-passive]` form of this command to remove pim parameters.

Use the `show interfaces switch <name> vif <number> ip pim mode [dense | dense-passive | sparse | sparse-dense | sparse-dense-passive | sparse-passive]` form of this command to display pim configuration.

interfaces switch vif ip pim state-refresh

Sets the interface instances to a pim state-refresh.

```
set interface switch namevifnumberippimstate-refresh origination-intervalvalue
```

```
delete interface switch namevifnumberippimstate-refresh origination-intervalvalue
```

```
show interface switch namevifnumberippimstate-refresh origination-intervalvalue
```

Configuration mode

```
interface {
    switch <name> {
        vif <number> {
            ip {
                pim {
                    state-refresh {
                        origination-interval <value>
                    }
                }
            }
        }
    }
}
```

Use this command to define the parameters for mcast of pim state-refresh.

Use the `set interfaces switch <name> vif <number> ip pim state-refresh origination-interval <value>` form of this command to set pim parameters.

Use the `delete interfaces switch <name> vif <number> ip pim state-refresh origination-interval <value>` form of this command to remove pim parameters.

Use the `show interfaces switch <name> vif <number> ip pim state-refresh origination-interval <value>` form of this command to display pim configuration.

interfaces switch vif ipv6 pim

Sets the interface instances to a pim.

```
set interface switch namevifnumberipv6pim[bsr-border | dr-priorityvalue | exclude-genid | hello-holdtimevalue | hello-intervalvalue | neighbor-filtervalue | propagation-delayvalue | unicast-bsm]
```

```
delete interface switch namevifnumberipv6pim[bsr-border | dr-priorityvalue | exclude-genid | hello-holdtimevalue | hello-intervalvalue | neighbor-filtervalue | propagation-delayvalue | unicast-bsm]
```

```
show interface switch name vif number ipv6 pim [bsr-border | dr-
priority value | exclude-genid | hello-holdtime value | hello-interval value | neighbor-
filter value | propagation-delay value | unicast-bsm]
```

Configuration mode

```
interface {
    switch <name> {
        vif <number> {
            ipv6 {
                pim {
                    bsr-border
                    dr-priority <value>
                    exclude-genid
                    hello-holdtime <value>
                    hello-interval <value>
                    neighbor-filter <value>
                    propagation-delay
                    unicast-bsm
                }
            }
        }
    }
}
```

Use this command to define the parameters for mcast of ipv6 pim.

Use the `set interfaces switch <name> vif <number> ipv6 pim [bsr-border | dr-priority <value> | exclude-genid | hello-holdtime <value> | hello-interval <value> | neighbor-filter <value> | propagation-delay <value> | unicast-bsm]` form of this command to set ipv6 pim parameters.

Use the `delete interfaces switch <name> vif <number> ipv6 pim [bsr-border | dr-priority <value> | exclude-genid | hello-holdtime <value> | hello-interval <value> | neighbor-filter <value> | propagation-delay <value> | unicast-bsm]` form of this command to remove ipv6 pim parameters.

Use the `show interfaces switch <name> vif <number> ipv6 pim [bsr-border | dr-priority <value> | exclude-genid | hello-holdtime <value> | hello-interval <value> | neighbor-filter <value> | propagation-delay <value> | unicast-bsm]` form of this command to display ipv6 pim configuration.

interfaces switch vif ipv6 pim mode

Sets the interface instances to a ipv6 pim mode.

```
set interface switch name vif number ipv6 pim mode [dense | dense-
passive | sparse | sparse-dense | sparse-dense-passive | sparse-passive]
```

```
delete interface switch name vif number ipv6 pim mode [dense | dense-  
passive | sparse | sparse-dense | sparse-dense-passive | sparse-passive]
```

```
show interface switch name vif number ipv6 pim mode [dense | dense-  
passive | sparse | sparse-dense | sparse-dense-passive | sparse-passive]
```

Configuration mode

```
interface {
    switch <name> {
        vif <number> {
            ipv6 {
                pim {
                    mode {
                        dense
                        dense-passive
                        sparse
                        sparse-dense
                        sparse-dense-passive
                        sparse-passive
                    }
                }
            }
        }
    }
}
```

Use this command to define the parameters for mcast of ipv6 pim mode.

Use the `set interfaces switch <name> vif <number> ipv6 pim mode [dense | dense-passive | sparse | sparse-dense | sparse-dense-passive | sparse-passive]` form of this command to set ipv6 pim parameters.

Use the `delete interfaces switch <name> vif <number> ipv6 pim mode [dense | dense-passive | sparse | sparse-dense | sparse-dense-passive | sparse-passive]` form of this command to remove ipv6 pim parameters.

Use the `show interfaces switch <name> vif <number> ipv6 pim mode [dense | dense-passive | sparse | sparse-dense | sparse-dense-passive | sparse-passive]` form of this command to display ipv6 pim configuration.

interfaces switch vif ipv6 pim state-refresh

Sets the interface instances to a ipv6 pim state-refresh.

```
set interface switch name vif number ipv6 pim state-refresh origination-interval value
```

```
delete interface switch name vif number ipv6 pim state-refresh origination-  
interval value
```

```
show interface switch name vif number ipv6 pim state-refresh origination-  
interval value
```

Configuration mode

```
interface {  
    switch <name> {  
        vif <number> {  
            ipv6 {  
                pim {  
                    state-refresh {  
  
origination-interval <value>  
  
                }  
            }  
        }  
    }  
}
```

Use this command to define the parameters for mcast of ipv6 pim state-refresh.

Use the `set interfaces switch <name> vif <number> ipv6 pim state-refresh origination-interval <value>` form of this command to set ipv6 pim parameters.

Use the `delete interfaces switch <name> vif <number> ipv6 pim state-refresh origination-interval <value>` form of this command to remove ipv6 pim parameters.

Use the `show interfaces switch <name> vif <number> ipv6 pim state-refresh origination-interval <value>` form of this command to display ipv6 pim configuration.

monitor command

Monitors an `mtrace` command.

```
monitor command mtrace-command
```

```
run monitor command mtrace-command
```

mtrace-command

The `mtrace` command to be monitored. The `mtrace` command must be enclosed in quotation marks.

Operational mode.

Configuration mode

Use this command to display the output of an `mtrace` command. The session stays open and display information is refreshed every two seconds.

Use the `run` form of this command in configuration mode.

monitor protocol multicast

Sets debugging options for multicast routing.

```
monitor protocol multicast [ [ background { start | stop } ] | { enable |
disable } { ip | ipv6 } [ event | fib-msg | mrib-msg | mrt | mtrace | mtrace-
detail | nsm-msg | register-msg | stats | vif ] ]
```

Multicast debugging is disabled.

background

Performs debugging operations in the background.

start

Starts debugging in the background.

stop

Stops debugging in the background.

enable

Enables the specified debugging option.

disable

Disables the specified debugging option.

ip

Specifies IPv4 multicast debugging.

ipv6

Specifies IPv6 multicast debugging.

event

Displays debugging messages for multicast events.

fib-msg

Reports all forwarding information base messages.

mrib-msg

Reports all multicast routing information base messages.

mrt

Displays debugging messages for multicast routes.

mtrace

Displays multicast traceroute debugging.

mtrace-detail

Displays detailed multicast traceroute debugging.

nsm-msg

Reports all Network Services Module messages.

register-msg

Reports all Protocol Independent Multicast (PIM) register messages.

stats

Displays debugging messages for multicast statistics.

vif

Displays debugging messages for multicast-enabled vif interfaces.

Operational mode

Use this command to enable or disable debugging for multicast and to configure multicast debugging options.

The following example starts debugging in the background for IPv4 multicast events.

```
vyatta@vyatta:~$monitor protocol multicast enable ip event
```

The following example disables all IPv6 multicast debugging.

```
vyatta@vyatta:~$monitor protocol multicast disable ipv6
```

mtrace

Displays the route that packets take from a multicast source.

```
set mtrace host [ destination addr [ group group [ detail ] | detail ] | group
group [ destination addr [ detail ] | detail ] | detail ]
```

Output shows routes from the source host to the host on which the command is entered.

host

The host that is the source for the trace. The host is either a name (if DNS is being used on the network) or an IPv4 or IPv6 unicast address.

addr

The host that is the destination for the trace. The host is either an IPv4 or IPv6 unicast address.

group

The multicast group to trace. The format is an IPv4 or IPv6 multicast address.

detail

Displays detailed output. This output includes IP multicast packet-rate and packet-loss information.

Operational mode

Use this command to show the path from a source to a receiver in a multicast network.

The following example displays a trace from 10.14.0.1 through 10.12.0.2 using group 226.0.0.1.

```
vyatta@vyatta:~$mtrace 10.14.0.1 destination 10.12.0.2 group 226.0.0.1
Mtrace from 10.14.0.1 to 10.12.0.2 via group 226.0.0.1
Querying full reverse path...
  0 ? (10.12.0.2)
 -1 ? (10.12.0.2) PIM thresh^ 1 RPF interface
 -2 ? (10.12.0.1) PIM thresh^ 1
 -3 ? (10.14.0.1)
Round trip time 1 ms
```

The following example displays a detailed trace from 10.14.0.1 through 10.12.0.2 using group 226.0.0.1.

```
vyatta@vyatta:~$mtrace 10.14.0.1 destination 10.12.0.2 group 226.0.0.1
detail
Mtrace from 10.14.0.1 to 10.12.0.2 via group 226.0.0.1
Round trip time 0 ms

Waiting to accumulate statistics... Results after 10 seconds:
```

Source	Response Dest	Packet Statistics For	Only For Traffic
10.14.0.1	10.12.0.2	All Multicast Traffic	From 10.14.0.1
v	___/ rtt 0 ms	Lost/Sent = Pct Rate	To 226.0.0.1
10.14.0.1			
10.12.0.1			
v	^ ttl 1	0	0 pps
pps			
10.12.0.2		RPF interface	
v	___ ttl 2		
10.12.0.2	10.12.0.2		
Receiver	Query Source		

protocols pim dense-group

Sets the protocol instances to a PIM mode.

```
set protocols pim dense-group ipv4-address
```

```
delete protocols pim dense-group ipv4-address
```

```
show protocols pim dense-group ipv4-address
```

Configuration mode

```
protocols {
```

```

    pim {
        dense-group <ipv4-address>
    }
}

```

Use this command to define the parameters for mcast of PIM mode.

Use the `set protocols pim dense-group <x.x.x.x>` form of this command to set PIM parameters.

Use the `delete protocols pim dense-group <x.x.x.x>` form of this command to remove PIM mode parameters.

Use the `show protocols pim dense-group <x.x.x.x>` form of this command to display PIM mode configuration.

protocols pim dense-group ipv6-address

Sets the protocol instances to a PIM mode.

```
set protocols pim dense-group ipv6-address
```

```
delete protocols pim dense-group ipv6-address
```

```
show protocols pim dense-group ipv6-address
```

Configuration mode

```

protocols {
    pim {
        dense-group <ipv6-address>
    }
}

```

Use this command to define the parameters for mcast of PIM mode.

Use the `set protocols pim dense-group <h:h:h:h:h:h:h:h>` form of this command to set PIM parameters.

Use the `delete protocols pim dense-group <h:h:h:h:h:h:h:h>` form of this command to remove PIM mode parameters.

Use the `show protocols pim dense-group <h:h:h:h:h:h:h:h>` form of this command to display PIM mode configuration.

protocols multicast ip log

Enables IPv4 MRIB logs.

```
set protocols multicast ip log { all | event | fib-msg | mrrib-msg | mrt | mtrace | mtrace-
detail | nsm-msg | register-msg | stats | vif }
```

```
delete protocols multicast ip log { all | event | fib-msg | mrrib-msg | mrt | mtrace | mtrace-
detail | nsm-msg | register-msg | stats | vif }
```

```
show protocols multicast ip log { all | event | fib-msg | mrrib-msg | mrt | mtrace | mtrace-
detail | nsm-msg | register-msg | stats | vif }
```

None

all

Enables MRIB debugging for all multicast protocol parameters.

event

Enables MRIB debugging for all multicast events.

fib-msg

Enables MRIB debugging for only multicast FIB events.

mrrib-msg

Enables MRIB debugging for only multicast RIB events.

mrt

Enables MRIB debugging for multicast routes.

mtrace

Enables MRIB debugging for multicast trace routes.

mtrace-detail

Enables MRIB debugging for multicast traceroute messages.

nsm-msg

Enables MRIB debugging for multicast NSM messages.

register-msg

Enables MRIB debugging for multicast PIM register messages.

stats

Enables MRIB debugging for multicast statistics.

vif

Enables MRIB debugging for multicast virtual interfaces.

Configuration mode

```
protocols {
  multicast ip {
    log {
```

```

    all
    event
    fib-msg
    mrib-msg
    mrt
    mtrace
    mtrace-detail
    nsm-msg
    register-msg
    stats
    vif
  }
}

```

Use the `set` form of this command to enable IPv4 multicast routing information base (MRIB) logs.

Use the `delete` form of this command to remove IPv4 MRIB logs.

Use the `show` form of this command to view IPv4 MRIB logs.

protocols multicast ipv6 log

Enables IPv6 MRIB logs.

```

set protocols multicast ipv6 log { all|event|fib-msg|mrib-msg|mrt|mtrace|mtrace-
detail|nsm-msg|register-msg|stats|vif}

```

```

delete protocols multicast ipv6 log { all|event|fib-msg|mrib-
msg|mrt|mtrace|mtrace-detail|nsm-msg|register-msg|stats|vif}

```

```

show protocols multicast ipv6 log { all|event|fib-msg|mrib-msg|mrt|mtrace|mtrace-
detail|nsm-msg|register-msg|stats|vif}

```

None

all

Enables MRIB debugging for all multicast protocol parameters.

event

Enables MRIB debugging for all multicast events.

fib-msg

Enables MRIB debugging for only multicast FIB events.

mrib-msg

Enables MRIB debugging for only multicast RIB events.

mrt

Enables MRIB debugging for multicast routes.

mtrace

Enables MRIB debugging for multicast trace routes.

mtrace-detail

Enables MRIB debugging for multicast traceroute messages.

nsm-msg

Enables MRIB debugging for multicast NSM messages.

register-msg

Enables MRIB debugging for multicast PIM register messages.

stats

Enables MRIB debugging for multicast statistics.

vif

Enables MRIB debugging for multicast virtual interfaces.

Configuration mode

```
protocols {
multicast ipv6 {
    log {
        all
        event
        fib-msg
        mrib-msg
        mrt
        mtrace
        mtrace-detail
        nsm-msg
        register-msg
        stats
        vif
    }
}
}
```

Use the `set` form of this command to enable IPv6 multicast routing information base (MRIB) logs.

Use the `delete` form of this command to remove IPv6 MRIB logs.

Use the `show` form of this command to view IPv6 MRIB logs.

protocols multicast ip log-warning

Sets a warning threshold for IPv4 multicast routes.

```
set protocols multicast ip log-warning warning-limit
```

```
delete protocols multicast ip log-warning
```

```
show protocols multicast ip log-warning
```

If this value is not configured, the system issues a warning only when the maximum number of routes has been reached.

warning-limit

The number of IPv4 multicast routes that causes the system to generate a warning. The value configured for this argument must not exceed the value set for the maximum route limit that is configured by using the `protocols multicast ip route-limit route-limit` command. The range is 1 through 2147483647.

Configuration mode

```
protocols {
  multicast {
    ip {
      log-warning warning-limit
    }
  }
}
```

Use this command to configure the warning threshold for IPv4 routes in the multicast routing table. When this limit is exceeded, the system continues to generate a message until the maximum number of multicast routes is reached that is configured by using the `protocols multicast ip route-limit route-limit` command.

Use the `set` form of this command to configure the warning threshold for IPv4 multicast routes.

Use the `delete` form of this command to restore the default IPv4 warning threshold.

Use the `show` form of this command to show the IPv4 warning threshold configuration.

protocols multicast ip route-limit

Sets the maximum number of IPv4 routes that can be added to the multicast routing table.

```
set protocols multicast ip route-limit route-limit
```

```
delete protocols multicast ip route-limit
```

```
show protocols multicast ip route-limit
```

The maximum number of IPv4 multicast routes is 2,147,483,647.

route-limit

The maximum number of IPv4 routes that can be added to the multicast routing table. The value configured for this argument must be greater than or equal to the log warning limit that is configured by using the `protocols multicast ip log-warning warning-limit` command. The range is 1 through 2147483647.

Configuration mode

```

protocols {
  multicast {
    ip {
      route-limit route-limit
    }
  }
}

```

Use this command to limit the number of IPv4 routes that can be added to the multicast routing table.

Use the `set` form of this command to specify the IPv4 multicast route limit.

Use the `delete` form of this command to restore the default IPv4 multicast route limit.

Use the `show` form of this command to show the IPv4 multicast route limit configuration.

protocols multicast ip routing

Enables IPv4 multicast routing.

```
set protocols multicast ip routing
```

```
delete protocols multicast ip routing
```

```
show protocols multicast ip routing
```

IPv4 multicast routing is disabled.

Configuration mode

```

protocols {
  multicast {
    ip {
      routing {
      }
    }
  }
}

```

Use this command to enable the system to use multicast routing protocols for IPv4 traffic.

When this configuration node is deleted, multicast routing protocols such as the Internet Group Management Protocol (IGMP) and Protocol Independent Multicast (PIM) do not work for IPv4 traffic, even if configured.

Use the `set` form of this command to enable multicast routing.

Use the `delete` form of this command to remove the multicast routing configuration.

Use the `show` form of this command to display the multicast routing configuration.

protocols multicast ipv6 log-warning

Sets the warning threshold for IPv6 multicast routes.

```
set protocols multicast ipv6 log-warning warning-limit
```

```
delete protocols multicast ipv6 log-warning
```

```
show protocols multicast ipv6 log-warning
```

If this value is not configured, the system issues a warning only when the maximum number of routes has been reached.

warning-limit

The number of IPv6 multicast routes that causes the system to generate a warning. The value configured for this argument must not exceed the value set for the maximum route limit that is configured by using the `protocols multicast ipv6 route-limit route-limit` command. The range is 1 through 2147483647.

Configuration mode

```
protocols {
  multicast {
    ipv6 {
      log-warning warning-limit
    }
  }
}
```

Use this command to configure the warning threshold for IPv6 routes in the multicast routing table. When this limit is exceeded, the system continues to generate a message until the maximum number of multicast routes is reached that is configured by using the `protocols multicast ipv6 route-limit route-limit` command.

Use the `set` form of this command to configure the IPv6 warning threshold.

Use the `delete` form of this command to restore the default IPv6 warning threshold.

Use the `show` form of this command to show the IPv6 warning threshold configuration.

protocols multicast ipv6 route-limit

Sets the maximum number of IPv6 routes that can be added to the multicast routing table.

```
set protocols multicast ipv6 route-limit route-limit
```

```
delete protocols multicast ipv6 route-limit
```

```
show protocols multicast ipv6 route-limit
```

The maximum number of IPv6 multicast routes is 2,147,483,647.

route-limit

The maximum number of IPv6 routes that can be added to the multicast routing table. The value configured for this argument must be greater than or equal to the log warning limit that is configured by using the `protocols multicast ip log-warning warning-limit` command. The range is 1 through 2147483647.

Configuration mode

```
protocols {
  multicast {
    ip {
      route-limit route-limit
    }
  }
}
```

Use this command to limit the number of IPv6 routes that can be added to the multicast routing table.

Use the `set` form of this command to specify the IPv6 multicast route limit.

Use the `delete` form of this command to restore the default IPv6 multicast route limit.

Use the `show` form of this command to show the IPv6 multicast route limit configuration.

protocols multicast ipv6 routing

Enables IPv6 multicast routing.

```
set protocols multicast ipv6 routing
```

```
delete protocols multicast ipv6 routing
```

```
show protocols multicast ipv6 routing
```

IPv6 multicast routing is disabled.

Configuration mode

```
protocols {
  multicast {
    ipv6 {
      routing {
```

```

}
}
}
}
}
}
}
}
}
}

```

Use this command to enable the system to use multicast routing protocols for IPv6 traffic.

When this configuration is deleted, multicast routing protocols such as the Multicast Listener Discovery (MLD) and PIM do not work for IPv6 traffic, even if configured.

Use the `set` form of this command to enable IPv6 multicast routing.

Use the `delete` form of this command to remove IPv6 multicast routing configuration.

Use the `show` form of this command to display the IPv6 multicast routing configuration.

reset ip mroute

Removes IPv4 entries from the multicast routing information base of the specified group.

```
reset ip mroute [ group group [ source source ] ]
```

When used with no option, this command deletes all routes from the multicast routing information base.

group

The IPv4 multicast group in IPv4 multicast address format. Routes are removed for the specified group.

source

Used in source-specific multicast. Removes routes for the specified IPv4 multicast source. The format is an IPv4 multicast address.

Operational mode

Use this command to remove IPv4 routes from the multicast routing and forwarding information bases.

Each multicast routing protocol has a distinct command for clearing multicast routes from the routing table for the protocol.

reset ipv6 mroute

Removes IPv6 entries from the multicast routing information base.

```
reset ipv6 mroute [ group group [ source source ] ]
```

When used with no option, this command deletes all routes from the multicast routing information base.

group group

Removes IPv6 routes for the specified multicast group. The format is an IPv6 multicast address.

source source

Used in source-specific multicast. Shows multicast routes for the specified IPv6 multicast source. The format is an IPv6 address.

Operational mode

Use this command to remove IPv6 routes from the multicast routing and forwarding information bases. Each multicast routing protocol has a distinct command for clearing multicast routes from the routing table for the protocol.

service application rule icmpv6 name

Sets the service application rules to a icmpv6.

```
set service application rule rule-numbericmpv6name[multicast-listener-done
|multicast-listener-query|multicast-listener-report]
```

```
delete service application rule rule-numbericmpv6name[multicast-listener-done
|multicast-listener-query|multicast-listener-report]
```

```
show service application rule rule-numbericmpv6name[multicast-listener-done
|multicast-listener-query|multicast-listener-report]
```

Configuration mode

```
service {
    application {
        rule <number> {
            icmpv6 {
                name {
                    multicast-listener-done
                    dr-priority <value>
                    multicast-listener-query
                    hello-holdtime <value>
                }
            }
        }
    }
}
```

Use this command to define the parameters for mcast of icmpv6.

Use the `set service application rule <rule-number> icmpv6 name [multicast-listener-done | multicast-listener-query | multicast-listener-report]` form of this command to set icmpv6 parameters.

Use the `delete service application rule <rule-number> icmpv6 name [multicast-listener-done | multicast-listener-query | multicast-listener-report]` form of this command to remove icmpv6 parameters.

Use the `show service application rule <rule-number> icmpv6 name [multicast-listener-done | multicast-listener-query | multicast-listener-report]` form of this command to display icmpv6 configuration.

show dataplane multicast ip route forwarding

Displays information about dataplane multicast ip forwarding instances.

```
show dataplane multicast ip forwarding [routing-instance <name>]
```

Operational mode

Use the `show dataplane multicast ip forwarding [routing-instance <name>]` command to display information about the multicast ip routing-instance.

The `show dataplane multicast ip forwarding [routing-instance <name>]` command displays the following information.

show dataplane multicast ip route interface

Displays information about dataplane multicast ip interface instances.

```
show dataplane multicast ip interface [routing-instance <name>]
```

Operational mode

Use the `show dataplane multicast ip interface [routing-instance <name>]` command to display information about the multicast ip routing-instance.

The `show dataplane multicast ip interface [routing-instance <name>]` command displays the following information.

show dataplane multicast ip route

Displays information about dataplane multicast ip route instances.

```
show dataplane multicast ip route [routing-instance <name>]
```

Operational mode

Use the `show dataplane multicast ip route [routing-instance <name>]` command to display information about the multicast ip routing-instance.

The `show dataplane multicast ip route [routing-instance <name>]` command displays the following information.

show dataplane multicast ipv6 route forwarding

Displays information about dataplane multicast ip forwarding instances.

```
show dataplane multicast ipv6 forwarding [routing-instance <name>]
```

Operational mode

Use the `show dataplane multicast ipv6 forwarding [routing-instance <name>]` command to display information about the multicast ip routing-instance.

The `show dataplane multicast ipv6 forwarding [routing-instance <name>]` command displays the following information.

show dataplane multicast ipv6 route interface

Displays information about dataplane multicast ip interface instances.

```
show dataplane multicast ipv6 interface [routing-instance <name>]
```

Operational mode

Use the `show dataplane multicast ipv6 interface [routing-instance <name>]` command to display information about the multicast ip routing-instance.

The `show dataplane multicast ipv6 interface [routing-instance <name>]` command displays the following information.

show dataplane multicast ipv6 route

Displays information about dataplane multicast ip route instances.

```
show dataplane multicast ipv6 route [routing-instance <name>]
```

Operational mode

Use the `show dataplane multicast ipv6 route [routing-instance <name>]` command to display information about the multicast ip routing-instance.

The `show dataplane multicast ipv6 route [routing-instance <name>]` command displays the following information.

show dataplane multicast ip

Displays dataplane multicast IPv4 forwarding statistics, interfaces, routes and summary of routing instances.

```
show dataplane multicast ip [ [ routing-instancename ] | route [ routing-
instancename ] | forwarding [ routing-instancename ] | interface [ routing-
instancename ] ]
```

When used with no option, this command shows dataplane multicast IPv4 summary.

routing-instance

Mandatory. The type keyword and identifier of a routing instance.

route

Mandatory. The type keyword and identifier of a route.

forwarding

Mandatory. The type keyword and identifier of forwarding statistics.

interface

Mandatory. The type keyword and identifier of an interface.

Operational mode

Use this command to show dataplane multicast IPv4 forwarding statistics, interfaces, routes and summary of routing instances.

show dataplane multicast ipv6

Displays dataplane multicast IPv4 forwarding statistics, interfaces, routes and summary of routing instances.

```
show dataplane multicast ipv6 [ [ routing-instancename ] | route [ routing-
instancename ] | forwarding [ routing-instancename ] | interface [ routing-
instancename ] ]
```

When used with no option, this command shows dataplane multicast IPv6summary.

routing-instance

Mandatory. The type keyword and identifier of a routing instance.

route

Mandatory. The type keyword and identifier of a route.

forwarding

Mandatory. The type keyword and identifier of forwarding statistics.

interface

Mandatory. The type keyword and identifier of an interface.

Operational mode

Use this command to show dataplane multicast IPv6 forwarding statistics, interfaces, routes and summary of routing instances.

show ip mroute

Displays the IPv4 multicast routing table.

```
show ip mroute [ group group [ source source ] ] [ dense | sparse | count | summary ]
```

When used with no option, this command displays information for the complete IPv4 multicast routing table.

group *group*

Shows IPv4 multicast routes for the specified multicast group. The format is an IPv4 multicast address.

source *source*

Used in source-specific multicast. Shows multicast routes for the specified IPv4 multicast source. The format is an IPv4 multicast address.

dense

Shows dense-mode IPv4 multicast routes.

sparse

Shows sparse-mode IPv4 multicast routes.

count

Shows IPv4 multicast-route and packet-count information.

summary

Shows abbreviated IPv4 multicast route information.

Operational mode

Use this command to display the IPv4 multicast routing table.

The following example shows how to display an IPv4 multicast routing table.

```
vyatta@vyatta:~$show ip mroute

IP Multicast Routing Table
Flags: I - Immediate Stat, T - Timed Stat, F - Forwarder installed
Timers: Uptime/Stat Expiry
Interface State: Interface (TTL)
```

```
(10.10.1.52, 224.0.1.3), uptime 00:00:31, stat expires 00:02:59
Owner PIM-SM, Flags: TF
Incoming interface: wm0
Outgoing interface list:
wm1 (1)
vyatta@vyatta:~$
```

The following example shows how to display routes for multicast group 224.0.1.3 and source 10.10.1.52.

```
vyatta@vyatta:~$show ip mroute group 224.0.1.3 source 10.10.1.52
IP Multicast Routing Table
Flags: I - Immediate Stat, T - Timed Stat, F - Forwarder installed
Timers: Uptime/Stat Expiry
Interface State: Interface (TTL)
(10.10.1.52, 224.0.1.3), uptime 00:03:24, stat expires 00:01:28
Owner PIM-SM, Flags: TF
Incoming interface: wm0
Outgoing interface list:
wm1 (1)vyatta@vyatta:~$
```

The following example shows how to display packet counts for multicast routes.

```
vyatta@vyatta:~$show ip mroute count
IP Multicast Statistics
Total 1 routes using 132 bytes memory
Route limit/Route threshold: 2147483647/2147483647
Total NOCACHE/WROGVIF/WHOLEPKT recv from fwd: 1/0/0
Total NOCACHE/WROGVIF/WHOLEPKT sent to clients: 1/0/0
Immediate/Timed stat updates sent to clients: 0/0
Reg ACK recv/Reg NACK recv/Reg pkt sent: 0/0/0
Next stats poll: 00:01:10
Forwarding Counts: Pkt count/Byte count, Other Counts: Wrong If pkts
Fwd msg counts: WROGVIF/WHOLEPKT recv
Client msg counts: WROGVIF/WHOLEPKT/Imm Stat/Timed Stat sent
Reg pkt counts: Reg ACK recv/Reg NACK recv/Reg pkt sent
(10.10.1.52, 224.0.1.3), Forwarding: 2/19456, Other: 0
Fwd msg: 0/0, Client msg: 0/0/0/0, Reg: 0/0/0
vyatta@vyatta:~$
```

The following example shows how to display a summary of the multicast routing table.

```
vyatta@vyatta:~$show ip mroute summary
IP Multicast Routing Table
Flags: I - Immediate Stat, T - Timed Stat, F - Forwarder installed
Timers: Uptime/Stat Expiry
Interface State: Interface (TTL)
```

```
(10.10.1.52, 224.0.1.3), 00:01:32/00:03:20, PIM-SM, Flags: TF
vyatta@vyatta:~$
```

show ip multicast interface

Displays information about IPv4 multicast-enabled interfaces.

```
show ip multicast interface [ interface ]
```

When used with no option, this command shows information for all IPv4 multicast-enabled interfaces.

interface

Mandatory. The type keyword and identifier of an interface. For detailed keywords and arguments that can be specified as interfaces, refer to [Supported Dataplane Interfaces](#).

Operational mode

Use this command to show operational information for IPv4 multicast-enabled interfaces.

The following example shows how to display all IPv4 multicast interface information.

```
vyatta@vyatta:~$ show ip multicast

Interface Vif Owner TTL Local Remote Uptime
Idx Module Address Address
wlm0 0 PIM-SM 1 192.168.1.53 0.0.0.0 00:04:26
Register 1 1 192.168.1.53 0.0.0.0 00:04:26
wlm1 2 PIM-SM 1 192.168.10.53 0.0.0.0 00:04:25
vyatta@vyatta:~$
```

The following example shows how to display IPv4 multicast interface information for the wlm0 interface.

```
vyatta@vyatta:~$ show ip multicast interface wlm0

Interface Vif Owner TTL Local Remote Uptime
Idx Module Address Address
wlm0 0 PIM-SM 1 192.168.1.53 0.0.0.0 00:05:17
vyatta@vyatta:~$
```

show ip multicast mrimfo

Displays information about IPv4 multicast neighbors.

```
show ip multicast mrimfo host [ source-addr | interface ]
```

host

The host that is being queried. The host is either a name (if DNS is being used on the network) or an IPv4 unicast address.

source-addr

Optional. The source address used in the mrimfo request. If omitted, the IPv4 address of the interface on which the request is sent is used.

interface

Optional. The source address used in the mrimfo request is the IPv4 address of the interface specified. If omitted, the IPv4 address of the interface on which the request is sent is used. For detailed keywords and arguments that can be specified as interfaces, refer to [Supported Dataplane Interfaces](#).

Operational mode

Use this command to show information about IPv4 multicast neighbors.

The following example shows how to display information about the IPv4 multicast neighbor mbone.phony.dom.net.

```
vyatta@vyatta:~$ show ip multicast mrimfo mbone.phony.dom.net
127.148.176.10 (mbone.phony.dom.net) [version 3.3]:
127.148.176.10 -> 0.0.0.0 (?) [1/1/querier]
127.148.176.10 -> 127.0.8.4 (mbone2.phony.dom.net) [1/45/tunnel]
127.148.176.10 -> 105.1.41.9 (momoney.com) [1/32/tunnel/down]
127.148.176.10 -> 143.192.152.119 (mbone.dipu.edu) [1/32/tunnel]
vyatta@vyatta:~$
```

show ip rpf

Displays Reverse Path Forwarding (RPF) information for a specific IPv4 multicast source address.

```
show ip rpf source
```

source

An IPv4 multicast source address.

Operational mode

Use this command to display the RPF information for a specific IPv4 multicast source address.

The following example shows how to display RPF information for the source address 172.18.92.1.

```
vyatta@vyatta:~$ show ip rpf 172.18.92.1
```

```
RPF Information for 172.18.92.1
RPF interface: dp0p1p1
RPF neighbor: 172.18.93.100
RPF prefix length: 24
RPF distance: 1
RPF mteric: 1
vyatta@vyatta:~$
```

show ipv6 mroute

Displays the IPv6 multicast routing table.

```
show ipv6 mroute [ group group [ source source ] ] [ dense | sparse | count | summary ]
```

When used with no option, this command displays information for the complete IPv6 multicast routing table.

group *group*

Shows IPv6 multicast routes for the specified multicast group. The format is an IPv6 multicast address.

source *source*

Used in source-specific multicast. Shows multicast routes for the specified IPv6 multicast source. The format is an IPv6 multicast address.

dense

Shows dense-mode IPv6 multicast routes.

sparse

Shows sparse-mode IPv6 multicast routes.

count

Shows IPv6 multicast-route and packet-count information.

summary

Shows abbreviated IPv6 multicast route information.

Operational mode

Use this command to display the IPv6 multicast routing table.

The following example shows how to display an IPv6 multicast routing table.

```
vyatta@vyatta:~$show ipv6 mroute

IPv6 Multicast Routing Table
Flags: I - Immediate Stat, T - Timed Stat, F - Forwarder installed
Timers: Uptime/Stat Expiry
Interface State: Interface
```

```
(3ffe:10:10:1::96, ffile::10), uptime 00:00:09, stat expires 00:03:21
Owner PIM-SMv6, Flags: TF
Incoming interface: wm0
Outgoing interface list:
wm1
(3ffe:10:10:1::96, ffile::12), uptime 00:00:02, stat expires 00:03:28
Owner PIM-SMv6, Flags: TF
Incoming interface: wm0
Outgoing interface list:
wm1
vyatta@vyatta:~$
```

show platform dataplane mroute

Displays the IPv4 system platform dataplane multicast routing table.

```
show platform dataplane mroute [ error | no-resource | no-support | not-needed |
partial ]
```

When used with no option, this command displays information for the complete IPv4 multicast routing table.

error

Shows objects that had error while programming in hardware.

no-resource

Shows objects that hardware did not have resource for.

no-support

Shows objects there is no support for.

not-needed

Shows objects that are not needed in hardware.

partial

Shows objects that are partially programmed in hardware.

Operational mode

Use this command to display the IPv4 system platform dataplane multicast routing table.

show platform dataplane mroute6

Displays the IPv6 system platform dataplane multicast routing table.

```
show platform dataplane mroute6 [ error | no-resource | no-support | not-needed
| partial ]
```

When used with no option, this command displays information for the complete IPv6 multicast routing table.

error

Shows objects that had error while programming in hardware.

no-resource

Shows objects that hardware did not have resource for.

no-support

Shows objects there is no support for.

not-needed

Shows objects that are not needed in hardware.

partial

Shows objects that are partially programmed in hardware.

Operational mode

Use this command to display the IPv6 system platform dataplane multicast routing table.

show ipv6 multicast interface

Displays information about IPv6 multicast-enabled interfaces.

```
show ipv6 multicast interface [ interface ]
```

When used with no option, this command shows information for all IPv6 multicast-enabled interfaces.

interface

The type keyword and identifier of an interface. For detailed keywords and arguments that can be specified as interfaces, refer to [Supported Dataplane Interfaces](#).

Operational mode

Use this command to show operational information for IPv6 multicast-enabled interfaces.

The following example shows how to display all IPv6 multicast interface information.

```
vyatta@vyatta:~$show ipv6 multicast

Interface Mif Owner Uptime
Idx Module
wlm0 0 PIM-SMv6 00:17:18
Register 1 00:17:18
wlm1 2 PIM-SMv6 00:17:18
vyatta@vyatta:~$
```

The following example shows how to display all IPv6 multicast interface information for the wlm0 interface.

```
vyatta@vyatta:~$show ipv6 multicast wlm0

Interface Mif Owner Uptime
Idx Module
wlm0 0 PIM-SMv6 00:19:06
vyatta@vyatta:~$
```

show ipv6 rpf

Displays Reverse Path Forwarding (RPF) information for a specific IPv6 multicast source address.

```
show ipv6 rpf source
```

source

An IPv6 multicast source address.

Operational mode

Use this command to display RPF information for a specific IPv6 multicast source address.

The following example shows how to display RPF information for the IPv6 source address 2036::6.

```
vyatta@vyatta:~$show ipv6 rpf 2036::6
RPF Information for 2036::6
  RPF interface: dp0p1p3
  RPF neighbor: fe80::250:56ff:fe9b:5aaf
  RPF prefix length: 64
  RPF distance: 2
  RPF mteric: 110
vyatta@vyatta:~$
```

show monitoring protocols multicast

Shows information about multicast debugging configuration.

```
show monitoring protocols multicast { ip | ipv6 }
```

ip

Shows what IPv4 multicast debugging options are enabled.

ipv6

Shows what IPv6 multicast debugging options are enabled.

Operational mode

Use this command to see what debugging options are currently enabled for IPv4 or IPv6 multicast routing.

The following example shows how to display the current configuration for IPv4 multicast debugging.

```
vyatta@vyatta:~$show monitoring protocols multicast ip

Debugging status:
MRIBv4 event debugging is on
MRIBv4 VIF debugging is on
MRIBv4 route debugging is on
MRIBv4 route statistics debugging is on
MRIBv4 FIB message debugging is on
MRIBv4 PIM Register message debugging is on
MRIBv4 NSM IPC message debugging is on
MRIBv4 MRIB IPC message debugging is on
MRIBv4 traceroute debugging is on
MRIBv4 traceroute detailed debugging is on
vyatta@vyatta:~$
```

Chapter 7. Supported Data Plane Interfaces

The following table shows the syntax and parameters of the supported types of data plane interfaces.

Interface Type	Syntax	Parameters
Data plane	<code>dataplane interface-name</code>	<p><i>interface-name</i>: The name of a data plane interface. Following are the supported formats of the interface name:</p> <ul style="list-style-type: none"> • <code>dp_xpy_pz</code>—The name of a data plane interface, where <ul style="list-style-type: none"> — <code>dp_x</code> specifies the data plane identifier (ID). Currently, only <code>dp0</code> is supported. — <code>py</code> specifies a physical or virtual PCI slot index (for example, <code>p129</code>). — <code>pz</code> specifies a port index (for example, <code>p1</code>). For example, <code>dp0p1p2</code>, <code>dp0p160p1</code>, and <code>dp0p192p1</code>. • <code>dp_xemy</code> —The name of a data plane interface on a LAN-on-motherboard (LOM) device that does not have a PCI slot, where <code>emy</code> specifies an embedded network interface number (typically, a small number). For example, <code>dp0em3</code>. • <code>dp_xsy</code>—The name of a data plane interface in a system in which the BIOS identifies the network interface card to reside in a particular physical or virtual slot <i>y</i>, where <i>y</i> is typically a small number. For example, for the <code>dp0s2</code> interface, the BIOS identifies slot 2 in the system to contain this interface. • <code>dp_xP_npy_pz</code> —The name of a data plane interface on a device that is installed on a secondary PCI bus, where <i>P_n</i> specifies the bus number. You can use this format to name data plane interfaces on large physical devices with multiple PCI buses. For these devices, it is possible to have network interface cards installed on different buses with these cards having the same slot ID. The value of <i>n</i> must be an integer greater than 0. For example, <code>dp0P1p162p1</code> and <code>dp0P2p162p1</code>. • <code>dp0vrrp_N</code> — The name of a VRRP data plane interface, where <i>N</i> is the <i>n</i>th VRRP group that is created. This name is generated by the router when <code>rfc-compatibility</code> mode is used. The first created interface is <code>dp0vrrp1</code>, the second is <code>dp0vrrp2</code>, and so on.
Data plane vif	<code>dataplane interface-name vif vif-id [vlan vlan-id]</code>	<p><i>interface-name</i>: Refer to the preceding description.</p> <p><i>vif-id</i>: A virtual interface ID. The ID ranges from 1 through 4094.</p> <p><i>vlan-id</i>: The VLAN ID of a virtual interface. The ID ranges from 1 through 4094.</p>

Chapter 8. VRF Support

VRF support for multicast

You can configure multicast within a routing instance by using the CLI, NetConf, or SNMP. When you configure multicast on the router without specifying a routing instance, the configuration applies to the default routing instance. To configure multicast for a particular routing instance, specify the instance.

All protocol-specific multicast commands can be applied to specific routing instances. For example, the following commands apply the indicated protocols to the RED routing instance.

```
vyatta@R1# set routing routing-instance RED protocols multicast ...
vyatta@R1# set routing routing-instance RED protocols pim ...
vyatta@R1# set routing routing-instance RED protocols pim6 ...
vyatta@R1# set routing routing-instance RED protocols igmp ...
vyatta@R1# set routing routing-instance RED protocols mld ...
vyatta@R1# set routing routing-instance RED protocols msdp ...
```

Any multicast configuration that is applied to an interface can be referred to under a routing instance to bind it to that instance, as in the following examples. In these examples, the dp0p161p1 interface is bound to the RED routing instance.

```
vyatta@R1# set routing routing-instance RED interfaces dp0p161p1 ip pim
vyatta@R1# set routing routing-instance RED interfaces dp0p161p1 ip igmp
vyatta@R1# set routing routing-instance RED interfaces dp0p161p1 ip
multicast
vyatta@R1# set routing routing-instance RED interfaces dp0p161p1 ipv6 pim
vyatta@R1# set routing routing-instance RED interfaces dp0p161p1 ipv6 mld
```

You can apply routing instances to **show** commands for supported protocols. The following example shows details about IPv6 MLD groups for the RED routing instance. If no routing instance is specified, the command applies to the default routing instance.

```
vyatta@vyatta:~$ show ipv6 mld groups routing-instance RED detail
```

You can apply routing instances to **reset** commands. The following example shows how to clear IP BGP addresses for routing instance RED. If no routing instance is specified, the command applies to the default routing instance.

```
vyatta@vyatta:~$ reset ip bgp routing-instance RED detail
```

Logging by multicast protocols is configured on a per routing instance basis. If no routing instance is specified, the command applies to the default routing instance. The following example shows how to enable all PIM logs in the RED routing instance.

```
vyatta@vyatta:~$ monitor protocol multicast routing-instance RED pim enable
```

Command support for VRF routing instances

VRF allows a router to support multiple routing tables, one for each VRF routing instance. Some commands in this guide support VRF and can be applied to particular routing instances.

Use the guidelines in this section to determine correct syntax when adding VRF routing instances to commands. For more information about VRF, refer to *Basic Routing Configuration Guide*. This guide includes an overview of VRF, VRF configuration examples, information about VRF-specific features, and a list of commands that support VRF routing instances.

Adding a VRF routing instance to a Configuration mode command

For most Configuration mode commands, specify the VRF routing instance at the beginning of a command. Add the appropriate VRF keywords and variable to follow the initial action (**set**, **show**, or **delete**) and before the other keywords and variables in the command.

Configuration mode example: syslog

The following command configures the syslog logging level for the specified syslog host. The command does not include a VRF routing instance, so the command applies to the default routing instance.

```
vyatta@R1# set system syslog host 10.10.10.1 facility all level debug
vyatta@R1# show system syslog
syslog {
  host 10.10.10.1 {
    facility all {
      level debug
    }
  }
}
```

The following example shows the same command with the VRF routing instance (GREEN) added. Notice that **routing routing-instance GREEN** has been inserted between the basic action (**set** in the example) and the rest of the command. Most Configuration mode commands follow this convention.

```
vyatta@R1# set routing routing-instance GREEN system syslog host 10.10.10.1
  facility all level debug
vyatta@R1# show routing
routing {
  routing-instance GREEN {
    system {
      syslog {
        host 11.12.13.2:514 {
          facility all {
```


Operational mode examples without optional parameters

The following command displays dynamic DNS information for the default routing instance.

```
vyatta@vyatta:~$ show dns dynamic status
```

The following command displays the same information for the specified routing instance (GREEN). The command does not have any optional parameters, so the routing instance is specified at the end of the command.

```
vyatta@vyatta:~$ show dns dynamic status routing-instance GREEN
```

Operational mode example with optional parameters

The following command obtains multicast path information for the specified host (10.33.2.5). A routing instance is not specified, so the command applies to the default routing instance.

```
vyatta@vyatta:~$ mtrace 10.33.2.5 detail
```

The following command obtains multicast path information for the specified host (10.33.2.5) and routing instance (GREEN). Notice that the routing instance is specified before the optional **detail** keyword.

```
vyatta@vyatta:~$ mtrace 10.33.2.5 routing-instance GREEN detail
```

Operational mode example output: SNMP

The following SNMP **show** commands display output for routing instances.

```
vyatta@vyatta:~$ show snmp routing-instance
Routing Instance SNMP Agent is Listening on for Incoming Requests:
Routing-Instance          RDID
-----
RED                        5
```

```
vyatta@vyatta:~$ show snmp community-mapping
SNMPv1/v2c Community/Context Mapping:
Community                  Context
-----
commA                       'RED'
commB                       'BLUE'
deva                        'default'
```

```
vyatta@vyatta:~$ show snmp trap-target
SNMPv1/v2c Trap-targets:
Trap-target                Port    Routing-Instance Community
```

```

-----
1.1.1.1          'RED'          'test'

vyatta@vyatta:~$ show snmp v3 trap-target
SNMPv3 Trap-targets:
Trap-target      Port      Protocol Auth Priv Type      EngineID
  Routing-Instance User
-----
2.2.2.2          '162'    'udp'    'md5  'infor
  'BLUE'         'test'

```

Chapter 9. List of Acronyms

Acronym	Description
ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
AH	Authentication Header
AMI	Amazon Machine Image
API	Application Programming Interface
AS	autonomous system
ARP	Address Resolution Protocol
AWS	Amazon Web Services
BGP	Border Gateway Protocol
BIOS	Basic Input Output System
BPDU	Bridge Protocol Data Unit
CA	certificate authority
CCMP	AES in counter mode with CBC-MAC
CHAP	Challenge Handshake Authentication Protocol
CLI	command-line interface
DDNS	dynamic DNS
DHCP	Dynamic Host Configuration Protocol
DHCPv6	Dynamic Host Configuration Protocol version 6
DLCI	data-link connection identifier
DMI	desktop management interface
DMVPN	dynamic multipoint VPN
DMZ	demilitarized zone
DN	distinguished name
DNS	Domain Name System
DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
eBGP	external BGP
EBS	Amazon Elastic Block Storage
EC2	Amazon Elastic Compute Cloud
EGP	Exterior Gateway Protocol
ECMP	equal-cost multipath
ESP	Encapsulating Security Payload
FIB	Forwarding Information Base
FTP	File Transfer Protocol

Acronym	Description
GRE	Generic Routing Encapsulation
HDLCL	High-Level Data Link Control
I/O	Input/Output
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IGMP	Internet Group Management Protocol
IGP	Interior Gateway Protocol
IPS	Intrusion Protection System
IKE	Internet Key Exchange
IP	Internet Protocol
IPOA	IP over ATM
IPsec	IP Security
IPv4	IP Version 4
IPv6	IP Version 6
ISAKMP	Internet Security Association and Key Management Protocol
ISM	Internet Standard Multicast
ISP	Internet Service Provider
KVM	Kernel-Based Virtual Machine
L2TP	Layer 2 Tunneling Protocol
LACP	Link Aggregation Control Protocol
LAN	local area network
LDAP	Lightweight Directory Access Protocol
LLDP	Link Layer Discovery Protocol
MAC	medium access control
mGRE	multipoint GRE
MIB	Management Information Base
MLD	Multicast Listener Discovery
MLPPP	multilink PPP
MRRU	maximum received reconstructed unit
MTU	maximum transmission unit
NAT	Network Address Translation
NBMA	Non-Broadcast Multi-Access
ND	Neighbor Discovery
NHRP	Next Hop Resolution Protocol
NIC	network interface card

Acronym	Description
NTP	Network Time Protocol
OSPF	Open Shortest Path First
OSPFv2	OSPF Version 2
OSPFv3	OSPF Version 3
PAM	Pluggable Authentication Module
PAP	Password Authentication Protocol
PAT	Port Address Translation
PCI	peripheral component interconnect
PIM	Protocol Independent Multicast
PIM-DM	PIM Dense Mode
PIM-SM	PIM Sparse Mode
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPPoA	PPP over ATM
PPPoE	PPP over Ethernet
PPTP	Point-to-Point Tunneling Protocol
PTMU	Path Maximum Transfer Unit
PVC	permanent virtual circuit
QoS	quality of service
RADIUS	Remote Authentication Dial-In User Service
RHEL	Red Hat Enterprise Linux
RIB	Routing Information Base
RIP	Routing Information Protocol
RIPng	RIP next generation
RP	Rendezvous Point
RPF	Reverse Path Forwarding
RSA	Rivest, Shamir, and Adleman
Rx	receive
S3	Amazon Simple Storage Service
SLAAC	Stateless Address Auto-Configuration
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
SONET	Synchronous Optical Network
SPT	Shortest Path Tree
SSH	Secure Shell
SSID	Service Set Identifier

Acronym	Description
SSM	Source-Specific Multicast
STP	Spanning Tree Protocol
TACACS+	Terminal Access Controller Access Control System Plus
TBF	Token Bucket Filter
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
ToS	Type of Service
TSS	TCP Maximum Segment Size
Tx	transmit
UDP	User Datagram Protocol
VHD	virtual hard disk
vif	virtual interface
VLAN	virtual LAN
VPC	Amazon virtual private cloud
VPN	virtual private network
VRRP	Virtual Router Redundancy Protocol
WAN	wide area network
WAP	wireless access point
WPA	Wired Protected Access