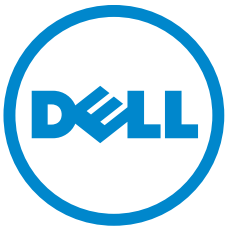
*53-1002110-01*
*02 November 2010*



# PowerConnect B-MLXe

## Diagnostic Reference

Regulatory Model Codes: Brocade MLXe-4, Brocade MLXe-8, Brocade MLXe-16

# Contents

## Chapter 4 Layer 2 Protocol Diagnostics

# About this document

This manual describes troubleshooting and diagnostic commands available in the IronWare command line interface (CLI) for PowerConnect B-MLXe series routing devices.

**NOTE**
Some troubleshooting commands report information about internal hardware settings and registers that is relevant primarily to Dell engineering staff. Consequently, this information is not described in this document.

## Audience

This document is designed for system administrators with a working knowledge of Layer 2 and Layer 3 switching and routing.

If you are using a Dell device, you should be familiar with the following protocols if applicable to your network - IP, RIP, OSPF, BGP, ISIS, IGMP, PIM, MPLS, and VRRP.

### Disclaimer

This manual is provided as a general reference only and is without any warranty of any kind, expressed or implied. When using this manual to troubleshoot Dell products, you assume all risk as to the quality and performance of the diagnostic procedures. Dell assumes no liability for any damages, including general, special, incidental, or consequential damages arising from the use of the diagnostic procedures in this manual (including, but not limited to any loss of profit or savings, loss of data, or failure to successfully troubleshoot network problems).

Diagnostic information may be changed or updated without notice. You are responsible for obtaining newer versions of this manual when they are made available. The procedures in this manual are not intended as a substitute for the expertise of qualified technicians.

Enabling diagnostic commands can seriously degrade system performance. Diagnostic commands are generally intended for use when troubleshooting specific problems while working with qualified service technicians, or in conjunction with calls to Dell technical support. Whenever possible, troubleshoot your system during periods of low network traffic and user activity to preserve system performance.

# How to use this guide

This guide describes many common diagnostic processes for PowerConnect B-MLXe series router. Each chapter contains diagnostic information about a specific segment of your network configuration, for example: Layer 1 Diagnostics, System Diagnostics, or Forwarding Diagnostics. Each topic is described under the following sections, where possible, and when the information is applicable:

- A brief description of the topic
- Show commands related to the topic
- Debug commands related to the topic
- Configuration notes for the topic
- Common diagnostic scenarios

# Supported hardware

The following hardware platforms are supported in this document:

- PowerConnect B-MLXe-4 router
- PowerConnect B-MLXe-8 router
- PowerConnect B-MLXe-16 router

# Document conventions

This section describes text formatting conventions and important notice formats used in this document.

## Text formatting

The narrative-text formatting conventions that are used are as follows:

| | |
|---|---|
| **bold** text | Identifies command names |
| | Identifies the names of user-manipulated GUI elements |
| | Identifies keywords |
| | Identifies text to enter at the GUI or CLI |
| *italic* text | Provides emphasis |
| | Identifies variables |
| | Identifies document titles |
| `code` text | Identifies CLI output |

For readability, command names in the narrative portions of this guide are presented in bold: for example, **show version**.

## Notes, cautions, and danger notices

The following notices and statements are used in this manual. They are listed below in 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.

**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.***

# Related publications

The following guides apply to the PowerConnect B-MLXe:

- The *PowerConnect B-MLXe Configuration Guide* describes how to configure PowerConnect B-MLXe router features, primarily using the CLI.
- The *PowerConnect B-MLXe MIB Reference* describes the Simple Network Management Protocol (SNMP) Information Base (MIB) objects that are supported in Dell devices.

**NOTE**
For the latest edition of this document, which contains the most up-to-date information, see product manuals at support.dell.com.

# Getting technical help or reporting errors

Dell is committed to ensuring that your investment in our products remains cost-effective. If you need assistance, or find errors in the manuals, contact Dell Technical Support.

# Contacting Dell

For customers in the United States, call 800-WWW.DELL (800.999.3355).

**NOTE**
If you do not have an active Internet connection, you can find contact information on your purchase invoice, packing slip, bill, or Dell product catalog.

Dell provides several online and telephone-based support and service options. Availability varies by country and product, and some services may not be available in your area. To contact Dell for sales, technical support, or customer service issues:

1. Visit http://www.support.dell.com.
2. Click your country or region at the bottom of the page. For a full listing of countries and regions, click All.
3. In the Support menu, click All Support.
4. Choose the method of contacting Dell that is convenient for you.

Chapter

# 1

# Using Diagnostic Commands

## How to use debug commands

This chapter describes how to use Dell diagnostic **debug** commands to monitor and troubleshoot PowerConnect B-MLXe series router configurations. Debug commands are accessible from the Privileged EXEC mode in the Multi-Service IronWare command line interface (CLI). Most debug commands can be configured to send output to a destination that you specify.

When enabled, debug commands can noticeably affect system performance. Many debug commands are specifically designed to be used in conjunction with calls to Dell technical support. If you report a problem, the support engineer may ask you to execute one or more of the **debug** commands described in this guide.

**ATTENTION**
Some debug commands report information about internal hardware settings and registers that is relevant primarily to Dell engineering staff. These commands are not described in this document.

The following sections provide basic information about debug commands and how to use them.

### Show commands

Show commands provide information that is extremely helpful for troubleshooting. For most of the environments discussed in this document, related show commands, show command output, and output descriptions are included.

Many debug commands work in conjunction with show commands to generate output for a specific configuration. When contacting Dell Technical Support have the device configuration file and an output capture of show tech-support available.

**show log**

**Syntax: show log**

The **show log** command allows you to view the system log or traps logged on an SNMP trap receiver, Output similar to the following is displayed. This output indicates that one switchover from standby to active has occurred.

```
PowerConnect#show log
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
Buffer logging: level ACDMEINW, 24 messages logged
level code: A=alert C=critical D=debugging M=emergency E=error
I=informational N=notification W=warning
Static Log Buffer:
Sep 28 11:31:25:A:Power Supply 1, 1st left, not installed
Sep 28 11:31:25:A:Power Supply 3, middle left, not installed
Sep 28 11:31:25:A:Power Supply 4, middle right, failed
Sep 28 11:31:25:A:Power Supply 5, 2nd right, not installed
Dynamic Log Buffer (50 lines):
Sep 27 18:06:58:I:Interface ethernet6/2, state up
Sep 27 18:06:57:I:Interface ethernet3/2, state up
Sep 27 15:39:42:I:Interface ethernet3/2, state up
Sep 27 15:39:42:I:Interface ethernet6/2, state up
...
Sep 27 14:23:45:N:Module up in slot 6
Sep 27 14:23:45:N:Module up in slot 3
Sep 27 14:23:27:A:Management module at slot 9 state changed from standby to
active
```

## Brief and detail debug options

When enabled, many debug commands can significantly impact system performance. Many debug commands provide options for brief or detailed reporting. Generating detailed output places an additional burden on system performance, and in many cases the results may be more difficult to interpret than output generated using the brief option. To conserve performance and prevent system disruption, use the brief option whenever possible.

# Generic debug commands

The following generic **debug** commands perform functions related to all debugging actions:

- **debug ?** - Generates a list of debug options.
- **[no] debug all** - Enables or disables all debug functions.
- **show debug -** Shows all enabled debug settings.
- **debug destination** - Allows you to select an output destination; Telnet, SSH, console, or logging (default).

### debug ?

**Syntax: debug ?**

The **debug ?** command generates a list of available debug variables.

**ATTENTION**
Many first-level variables have their own variable subsets. When you enter a **debug** command, the system will indicate that there are additional variables by telling you that you have entered an incomplete command. Add a space and a question mark to your original command to view the additional variables.

```
PowerConnect# debug ip
Incomplete command.
PowerConnect#debug ip ?
access-list          Enable ACL debugging
all                  Enable all debugging
bfd                  Enable BFD debugging
destination          Redirect debug message
dot1x                Debug 802.1X and Events
filters              Enable Filters debugging
gvrp                 Enable gvrp debugging
ip                   Debug trace IP
ipv6                 Debug trace IPv6
isis                 Debug isis
mac                  Enable MAC database debugging
.
.
```

### show debug

**Syntax: show debug**

The **show debug** command displays all enabled debug functions. Output resembles the following, which shows that RSTP and IS-IS debugging are enabled, with the console as the output destination.

```
PowerConnect#show debug
RSTP
        RSTP:  debugging is on
Debug message destination:  Console
INTEGRATED IS-IS :
          IS-IS:  isis debugging is on INTEGRATED IS-IS :
          IS-IS:  isis debugging is on
```

In this example, RSTP and IS-IS debugging are enabled. The output destination is the console.

### debug all

**Syntax:** [no] **debug all**

This command enables all debug functions, and should *only* be used during a troubleshooting session with a Dell technician. To cancel this setting, enter the **no debug all** command.

```
PowerConnect#debug all
Warning! This may severely impact network performance!
All possible debuggings have been turned on
```

**NOTE**

You may not be able to see the **no debug all** command as you type it. However, if you have typed the command correctly, output will stop as soon as you hit the **Enter** key.

**CAUTION**

**This command generates extensive output and can significantly slow device operation. Use this command with caution. Never use this command during periods of peak network activity.Type no debug all to stop the output.**

### debug destination

**Syntax:** [no] **debug destination** [**console** | **logging** | **telnet** <*num*> | **ssh** <*num*>]

This command allows you to specify a destination for debugging output. The default is the system console, but you can redirect output to a Syslog buffer, or a Telnet or SSH session. The following parameters are available for this command:

- **console** - Directs output to the system console.
- **logging** - Directs output to the Syslog buffer and to the Syslog server (default).
- **telnet** - Directs debugging output to a specified Telnet session (a number from 1 through 5).
- **ssh** - Directs debugging output to a specified SSH session (a number from 1 through 5).

#### Example

To send debug output to a Telnet session, first determine your session number using the **show who** command:

```
PowerConnect#show who
```

You should see output similar to the following. For purposes of this example, the relevant Telnet session has been highlighted.

```
Console connections:
        established, monitor enabled
        1 minutes 57 seconds in idle
Telnet connections (inbound):
 1      closed
 2      established, client ip address 10.55.1.128, user is <your login>
        you are connecting to this session
        15 seconds in idle
 3      closed
 4      closed
 5      closed
Telnet connection (outbound):
 6      closed
SSH connections:
 1      closed
 2      closed
 3      closed
 4      closed
```

This example indicates that you are connected through Telnet session 2. Redirect the debug output to your Telnet session by entering:

```
PowerConnect# debug destination telnet 2
```

## Disabling debug commands

When activated, most debug commands instruct the system to collect specific information about router configurations and activity. In all cases, adding **no** in front of the command disables the debug function.

Chapter

# System and System Management Diagnostics 2

This chapter describes many of the common system and system maintenance diagnostic processes for PowerConnect B-MLXe series router.

## Basic system information

Basic system troubleshooting includes the verification of software images and their locations, and monitoring hardware components such as fans and power supplies. The following sections describe how to display information, and what to look for when troubleshooting your hardware and system software.

### System hardware show commands

**show chassis**

**Syntax:** show chassis

The **show chassis** command displays information about your PowerConnect B-MLXe chassis, including power supplies, fan status and operating speeds, and temperature readings for all installed modules (temperatures are, by default, polled every 60 seconds). The following example shows output from the **show chassis** command.

```
PowerConnect> show chassis
*** NetIron MLXe CHASSIS ***

---POWERS ---
Power 1: Installed (Failed or Disconnected)
Power 2: Installed (Failed or Disconnected)
Power 3 (30351200 - AC 1200W): Installed (OK)
Power 4 (30351200 - AC 1200W): Installed (OK)
Total power budget for chassis = 2400W
Total power budget for LPs      = 2049W
Slot Power-On Priority and Power Usage:
Slot4 pri=1 module type=NI-mlxe-10Gx2 2-port 10GbE Module power usage=165W

--- FANS ---
Right fan tray (fan 1): Status = OK, Speed = MED (75%)
Right fan tray (fan 2): Status = OK, Speed = MED (75%)
Right fan tray (fan 3): Status = OK, Speed = MED (75%)
Right fan tray (fan 4): Status = OK, Speed = MED (75%)

--- TEMPERATURE READINGS ---
Active Mgmt Module: 38.0C 52.375C
Standby Mgmt Module: 35.250C
SNM1: 30.0C
SNM2: 27.5C
SNM3: 30.0C
LP2 Sensor1: 38.0C
LP2 Sensor2: 53.0C
LP3 Sensor1: 33.5C
LP3 Sensor2: 40.750C
LP4 Sensor1: 38.5C
LP4 Sensor2: 46.500C
LP4 Sensor3: UNUSED
Temperature Monitoring Poll Period is 60 seconds
```

For more information about how to troubleshoot hardware issues, see "Power supplies, fans, and temperature" on page 42.

## System software show commands

### show version

**Syntax: show version**

Most boot issues occur because incorrect or incompatible images have been downloaded. The **show version** command displays all versions that are currently loaded, as shown in this example:

```
PowerConnect(config)# show version
HW: NetIron MLXe Router
Backplane (Serial #: Not Exist,  Part #: Not Exist)
NI-X-SF Switch Fabric Module 1 (Serial #: PR29050242,  Part #: 31523-100A)
FE 1: Type 00000,  Version 0
FE 3: Type 00000,  Version 0
NI-X-SF Switch Fabric Module 2 (Serial #: PR29050246,  Part #: 31523-100A)
FE 1: Type 00000,  Version 0
FE 3: Type 00000,  Version 0
NI-X-SF Switch Fabric Module 3 (Serial #: PR30050270,  Part #: 31523-100A)
FE 1: Type 00000,  Version 0
FE 3: Type 00000,  Version 0
==========================================================================
SL M1: NI-MLXe-MR Management Module Active (Serial #: SA12061726, Part #:
31524-100A):
Boot     : Version 3.5.0T165 Copyright (c) 1996-2009 Brocade Communications,
Inc.
Compiled on Jul 10 2009 at 19:13:56 labeled as xmprm03500
 (424484 bytes) from boot flash
Monitor  : Version 3.5.0aT165 Copyright (c) 1996-2009 Brocade Communications,
Inc.
Compiled on Jul 30 2009 at 17:35:22 labeled as xmb03500a
 (424748 bytes) from code flash
IronWare : Version 3.5.0cT163 Copyright (c) 1996-2009 Brocade Communications,
Inc.
Compiled on Sep 17 2009 at 01:00:12 labeled as mlxe03500c
 (5840562 bytes) from Primary
Board ID : 00 MBRIDGE Revision : 18
916 MHz Power PC processor  (version 8003/0101) 166 MHz bus
512 KB Boot Flash (AM29LV040B), 32 MB Code Flash (MT28F128J3)
1024 MB DRAM
Active Management uptime is 21 days 10 hours 44 minutes 44 seconds
==========================================================================
SL M2: NI-MLXe-MR Management Module Standby (Serial #: SA11060307, Part #:
31524-100A):
Boot     : Version 3.5.0T165 Copyright (c) 1996-2009 Brocade Communications,
Inc.
Compiled on Jul 10 2009 at 19:13:56 labeled as xmprm03500
 (424484 bytes) from boot flash
Monitor  : Version 3.5.0aT165 Copyright (c) 1996-2009 Brocade Communications,
Inc.
Compiled on Jul 30 2009 at 17:35:22 labeled as xmb03500a
 (424748 bytes) from code flash
IronWare : Version 3.5.0cT163 Copyright (c) 1996-2009 Brocade Communications,
Inc.
Compiled on Sep 17 2009 at 01:00:12 labeled as mlxe03500c
 (5840562 bytes) from Primary
Board ID : 00 MBRIDGE Revision : 18
916 MHz Power PC processor  (version 8003/0101) 166 MHz bus
512 KB Boot Flash (AM29LV040B), 32 MB Code Flash (MT28F128J3)
1024 MB DRAM
Standby Management uptime is 19 days 14 hours 4 minutes 45 seconds
==========================================================================
=======
```

(continued on next page)

```
(continued from previous page)
==========================================================================
SL 3: NI-MLXe-1Gx24-SFP 24-port 1GbE/100FX Module (Serial #: SA23060375, Part #:
31570-103A)
Boot     : Version 3.3.0gT175 Copyright (c) 1996-2009 Brocade Communications,
Inc.
Compiled on Aug 29 2009 at 12:12:02 labeled as xmlprm03300g
 (336122 bytes) from boot flash
Monitor  : Version 3.3.0gT175 Copyright (c) 1996-2009 Brocade Communications,
Inc.
Compiled on Aug 29 2009 at 12:12:46 labeled as xmlb03300g
 (659473 bytes) from code flash
IronWare : Version 3.3.0gT177 Copyright (c) 1996-2009 Brocade Communications,
Inc.
Compiled on Aug 29 2009 at 18:37:36 labeled as xmlp03300g
 (2410342 bytes) from Primary
FPGA versions:
Valid PBIF Version = 2.18, Build Time = 7/21/2009 12:21:0

Valid XPP Version = 2.25, Build Time = 8/2/2009 10:33:0

BCM5695GMAC 0
BCM5695GMAC 1
BCM5695GMAC 2
BCM5695GMAC 3
666 MHz MPC 8541 (version 8020/0020) 333 MHz bus
512 KB Boot Flash (AM29LV040B), 16 MB Code Flash (MT28F640J3)
1024 MB DRAM, 8 KB SRAM, 0 Bytes BRAM
PPCR0: 1024K entries CAM, 16384K PRAM, 2048K AGE RAM
LP Slot 3 uptime is 3 hours 11 minutes 50 seconds
SL 5: NI-MLXe-10Gx4 4-port 10GbE Module (Serial #: pr32050022, Part #:
31546-100A)
Boot     : Version 3.3.0gT175 Copyright (c) 1996-2009 Brocade Communications,
Inc.
Compiled on Aug 29 2009 at 12:12:02 labeled as xmlprm03300g
 (336122 bytes) from boot flash
Monitor  : Version 3.3.0gT175 Copyright (c) 1996-2009 Brocade Communications,
Inc.
Compiled on Aug 29 2009 at 12:12:46 labeled as xmlb03300g
 (659473 bytes) from code flash
IronWare : Version 3.3.0gT177 Copyright (c) 1996-2009 Brocade Communications,
Inc.
Compiled on Aug 29 2009 at 18:37:36 labeled as xmlp03300g
 (2410342 bytes) from Primary
FPGA versions:
Valid PBIF Version = 2.18, Build Time = 7/21/2009 12:21:0

Valid XPP Version = 2.25, Build Time = 8/2/2009 10:33:0

Valid XGMAC Version = 0.11, Build Time = 10/11/2009 12:45:0

BCM5673X10GMAC 0
BCM5673X10GMAC 1
BCM5673X10GMAC 2
BCM5673X10GMAC 3
666 MHz MPC 8541 (version 8020/0020) 333 MHz bus
512 KB Boot Flash (AM29LV040B), 16 MB Code Flash (MT28F640J3)
1024 MB DRAM, 8 KB SRAM, 0 Bytes BRAM
PPCR0: 1024K entries CAM, 16384K PRAM, 2048K AGE RAM
```

### show flash

**Syntax:** show flash

This command displays the images that have been copied onto flash memory.

```
PowerConnect# show flash
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Active Management Module (Right Slot)
Code Flash - Type MT28F128J3, Size 32 MB
  o IronWare Image (Primary)
    Version 3.5.0T163, Size 5819609 bytes, Check Sum ec17
    Compiled on Jun 18 2009 at 07:15:12 labeled as mlxe03500b201
  o LP Kernel Image (Monitor for LP Image Type 0)
    Version 3.5.0T175, Size 386693 bytes, Check Sum 5ff6
    Compiled on May 31 2009 at 14:42:56 labeled as xmlb03500b155
  o LP IronWare Image (Primary for LP Image Type 0)
    Version 3.5.0T177, Size 3128223 bytes, Check Sum f07b
    Compiled on Jun 18 2009 at 07:49:48 labeled as xmlp03500b201
  o Monitor Image
    Version 3.5.0T165, Size 424045 bytes, Check Sum 66f0
    Compiled on May 31 2009 at 14:41:14 labeled as xmb03500b155
  o Startup Configuration
    Size 12466 bytes, Check Sum 1bb2
    Modified on 14:01:37 Pacific Mon Jun 18 2009
Boot Flash - Type AM29LV040B, Size 512 KB
  o Boot Image
    Version 3.5.0T165, Size 424038 bytes, Check Sum f1e9
    Compiled on May 31 2009 at 14:42:00 labeled as xmprm03500b155
Standby Management Module (Left Slot)
Code Flash: Type MT28F128J3, Size 32 MB
  o IronWare Image (Primary)
    Version 3.5.0T163, Size 5819609 bytes, Check Sum ec17
    Compiled on Jun 18 2009 at 07:15:12 labeled as mlxe03500b201
  o LP Kernel Image (Monitor for LP Image Type 0)
    Version 3.5.0T175, Size 386693 bytes, Check Sum 5ff6
    Compiled on May 31 2009 at 14:42:56 labeled as xmlb03500b155
  o LP IronWare Image (Primary for LP Image Type 0)
    Version 3.5.0T177, Size 3128223 bytes, Check Sum f07b
    Compiled on Jun 18 2009 at 07:49:48 labeled as xmlp03500b201
  o Monitor Image
    Version 3.5.0T165, Size 424045 bytes, Check Sum 66f0
    Compiled on May 31 2009 at 14:41:14 labeled as xmb03500b155
  o Startup Configuration
    Size 12466 bytes, Check Sum 1bb2
    Modified on 14:01:38 Pacific Mon Jun 18 2009
Boot Flash: Type AM29LV040B, Size 512 KB
  o Boot Image
    Version 3.5.0T165, Size 424038 bytes, Check Sum f1e9
.
```

### show who

**Syntax: show who**

The **show who** command displays information about users who are logged in to a Telnet connection, including privilege levels, as shown in the following example.

```
PowerConnect#show who
Console connections:
       established
```

```
       3 days 17 hours 31 minutes 27 seconds in idle
Telnet server status: Enabled
Telnet connections (inbound):
1      established, client ip address 10.53.1.65, privilege super-user
       you are connecting to this session
2      closed
3      closed
4      closed
5      closed
Telnet connections (outbound):
6      established, server ip address 10.47.2.200, from Telnet session 1
       4 seconds in idle
7      closed
8      closed
9      closed
10     closed
SSH server status: Enabled
SSH connections:
1      closed
.......
```

## System debug commands

### debug system trace

**Syntax:** [no] **debug system trace**

This command performs a system debugging trace.

### debug trace-l2 events

**Syntax:** [no] **debug trace-l2 events**

This command displays information about Layer 2 Trace protocol events.

## Common diagnostic scenarios

System issues are rare. However, some problem sources can include:

- Software versions are not compatible.
- Line modules or switch fabric modules are not functioning properly.
- Environmental conditions, such as temperatures that are above or below operating thresholds, which may affect operation of hardware components.

If you are experiencing system issues, contact Dell Technical Support for help in troubleshooting your system.

# TCAM partitioning and usage

Ternary Content Addressable Memory (TCAM) is a component of Dell devices that facilitates hardware forwarding. As packets flow through the Dell device from a given source to a given destination, the management processor records forwarding information about the flow in TCAM entries. A TCAM entry generally contains next-hop information, such as the outgoing port, the MAC address of the next-hop router, VLAN tag, and so on. Once the Dell device has this information in TCAM, packets with the same source and destination can be forwarded by hardware, bypassing the management processor and speeding up forwarding time.

TCAM entries can contain Layer 2, Layer 3, or Layer 4 information. Each type of TCAM entry has its own format:

- Layer 2 TCAM entries contain destination MAC information and deal with 802.1p (priority), and VLAN information
- Layer 3 TCAM entries contain destination IP information
- Layer 4 TCAM entries contain destination IP, destination TCP/UDP port, source IP, and source TCP/UDP port information.

When a Dell device is initialized, the software partitions the available TCAM into segments for Layer 2, Layer 3, or Layer 4 information. The percentage of TCAM devoted to each type of TCAM entry is determined by the profile.

Table 1 shows the TCAM sizes and functions for NetIron MLXe devices.

**TABLE 1** TCAM sizes and support

| TCAM | NetIron MLXe | Supports |
|---|---|---|
| TCAM 0/TCAM 1 | 18 Mb | IPv4, IPv6, MAC DA, MAC SA, L3VPN routes (uplink and endpoints) IPv4 for RPF |
| TCAM 2 | 9 Mb | Inbound IPv4 ACL, inbound IPv6 ACL, inbound Layer 2 ACL, MAC SA, MAC DA, Layer 3VPN routes (uplinks and endpoints) VPLS DA, VPLS SA (uplinks and endpoints), multicast |
| TCAM 3 | 9 Mb | outbound IPv4 ACL, outbound IPv6 ACL, outbound Layer 2 ACL only |

## TCAM show commands

### show cam ifl

**Syntax: show cam ifl** *<slotnum>/<portnum>*

This command displays CAM IFL (Internal Forwarding Lookup) information for a specified slot/port. Output resembles the following:

```
PowerConnect# show cam ifl 7/7
Slot Index  Port  Outer VLAN Inner VLAN PRAM  IFL ID IPV4/V6| (Hex)
(Hex)         Routing
7    0081fe9 7/4   4000       0          181fe9 131071 1/1
7    0081fea 7/3   4000       0          181fea 131071 1/1
7    0081feb 7/2   4000       0          181feb 131071 1/1
7    0081fec 7/1   4000       0          181fec 131071 1/1
7    0081fed 7/8   607        0          181fed 131071 1/1
7    0081fee 7/7   607        0          181fee 131071 1/1
7    0081fef 7/8   606        0          181fef 131071 1/1
```

```
7    0081ff0 7/7   606        0          181ff0 131071 1/1
7    0081ff1 7/8   605        0          181ff1 131071 1/1
7    0081ff2 7/7   605        0          181ff2 131071 1/1
7    0081ff3 7/8   604        0          181ff3 131071 1/1
7    0081ff4 7/7   604        0          181ff4 131071 1/1
7    0081ff5 7/8   603        0          181ff5 131071 1/1
7    0081ff6 7/7   603        0          181ff6 131071 1/1
7    0081ff7 7/8   602        0          181ff7 131070 1/1
7    0081ff8 7/7   602        0          181ff8 131070 1/1
7    0081ff9 7/8   601        0          181ff9 131071 1/1
7    0081ffa 7/7   601        0          181ffa 131071 1/1
Slot Index   Port  Outer VLAN Inner VLAN PRAM   IFL ID IPV4/V6
(Hex)                         (Hex)         Routing
7    0081ffb 7/8   6          0          181ffb 131071 1/1
7    0081ffc 7/7   6          0          181ffc 131071 1/1
7    0081ffd 7/19  4000       0          181ffd 131071 1/1
7    0081ffe 7/14  100        0          181ffe 131071 1/1
7    0081fff 7/18  4000       0          181fff 131071 1/1
```

### show cam-partition

**Syntax: show cam-partition** [**brief** | **slot** | **usage** | output modifiers]

The following parameters are available for this command:

- **brief** - Displays a brief summary of partition information.
- **slot** - Displays partition information for a specific slot.
- **usage** - Displays brief partition usage information.

The following examples show output from this command using these parameters.

The **show cam-partition brief** command displays TCAM information per partition and sub partition in three formats: raw size, user size, and reserved size, as shown in this example:

```
PowerConnect# show cam-partition brief

CAM partitioning profile: default

Slot 1 XPP20SP 0:
# of CAM device               = 4
Total CAM Size                = 917504 entries (63Mbits)

IP: Raw Size 524288, User Size 524288(0 reserved)
  Subpartition  0: Raw Size 12288, User Size 12288, (0 reserved)
  Subpartition  1: Raw Size 468107, User Size 468107, (0 reserved)
  Subpartition  2: Raw Size 37335, User Size 37335, (0 reserved)
  Subpartition  3: Raw Size 5140, User Size 5140, (0 reserved)
  Subpartition  4: Raw Size 778, User Size 778, (0 reserved)
IPv6: Raw Size 131072, User Size 65536(0 reserved)
  Subpartition  0: Raw Size 12288, User Size 6144, (0 reserved)
  Subpartition  1: Raw Size 107496, User Size 53748, (0 reserved)
  Subpartition  2: Raw Size 9332, User Size 4666, (0 reserved)
  Subpartition  3: Raw Size 1284, User Size 642, (0 reserved)
  Subpartition  4: Raw Size 384, User Size 192, (0 reserved)
IP VPN Raw Size 131072, User Size 131072(0 reserved)
  Subpartition  0: Raw Size 2048, User Size 2048, (0 reserved)
  Subpartition  1: Raw Size 116886, User Size 116886, (0 reserved)
  Subpartition  2: Raw Size 9333, User Size 9333, (0 reserved)
  Subpartition  3: Raw Size 1285, User Size 1285, (0 reserved)
  Subpartition  4: Raw Size 384, User Size 384, (0 reserved)
MAC: Raw Size 131072, User Size 131072(0 reserved)
  Subpartition  0: Raw Size 10, User Size 10, (0 reserved)
  Subpartition  1: Raw Size 32, User Size 32, (0 reserved)
  Subpartition  2: Raw Size 131030, User Size 131030, (0 reserved)
Session: Raw Size 98304, User Size 49152(0 reserved)
  Subpartition  0: Raw Size 79872, User Size 39936, (0 reserved)
  Subpartition  1: Raw Size 2048, User Size 1024, (0 reserved)
  Subpartition  2: Raw Size 16384, User Size 8192, (0 reserved)
IPv6 Session: Raw Size 32768, User Size 4096(0 reserved)
  Subpartition  0: Raw Size 15872, User Size 1984, (0 reserved)
  Subpartition  1: Raw Size 512, User Size 64, (0 reserved)
  Subpartition  2: Raw Size 16384, User Size 2048, (0 reserved)
Out Session: Raw Size 196608, User Size 98304(49152 reserved)
Out IPv6 Session: Raw Size 65536, User Size 8192(4096 reserved)

Slot 1 XPP20SP 0:
Slot 3 XPP20SP 0:
# of CAM device               = 4
Total CAM Size                = 917504 entries (63Mbits)
```

The **show cam-partition usage** command displays the amount of TCAM being used and how much is available, as shown in the following example.

**NOTE**
The display has been shortened for brevity.

```
PowerConnect# show cam-partition usage
CAM partitioning profile: default

Slot 1 XPP20SP 0:
Slot 1 XPP20SP 0:
           [IP]524288(size), 518257(free), 01.15%(used)
        :SNet 0: 12288(size),  12269(free), 00.15%(used)
        :SNet 1:468107(size), 462099(free), 01.28%(used)
        :SNet 2: 37335(size),  37332(free), 00.00%(used)
        :SNet 3:  5140(size),   5140(free), 00.00%(used)
        :SNet 4:   778(size),    778(free), 00.00%(used)
        .
         [IPV6] 65536(size),  65534(free), 00.00%(used)
        :SNet 0:  6144(size),   6144(free), 00.00%(used)
        :SNet 1: 53748(size),  53748(free), 00.00%(used)
        :SNet 2:  4666(size),   4666(free), 00.00%(used)
        :SNet 3:   642(size),    642(free), 00.00%(used)
        :SNet 4:   192(size),    192(free), 00.00%(used)
        .
        [IP VPN]131072(size), 131072(free), 00.00%(used)
        :SNet 0:  2048(size),   2048(free), 00.00%(used)
        :SNet 1:116886(size), 116886(free), 00.00%(used)
        :SNet 2:  9333(size),   9333(free), 00.00%(used)
        :SNet 3:  1285(size),   1285(free), 00.00%(used)
        :SNet 4:   384(size),    384(free), 00.00%(used)
        .
          [MAC]131072(size), 131067(free), 00.00%(used)
   :Forwarding:131030(size), 131025(free), 00.00%(used)
     :Protocol:    32(size),     32(free), 00.00%(used)
     :Flooding:    10(size),     10(free), 00.00%(used)

      [Session] 49152(size),  49152(free), 00.00%(used)
  :IP Multicast:  8192(size),   8192(free), 00.00%(used)
   :Receive ACL:  1024(size),   1024(free), 00.00%(used)
      :Rule ACL: 39936(size),  39936(free), 00.00%(used)

  [IPV6 Session]  4096(size),   4096(free), 00.00%(used)
  :IP Multicast:  2048(size),   2048(free), 00.00%(used)
   :Receive ACL:    64(size),     64(free), 00.00%(used)
      :Rule ACL:  1984(size),   1984(free), 00.00%(used)

   [Out Session] 49152(size),  49152(free), 00.00%(used)

[Out V6 Session]  4096(size),   4096(free), 00.00%(used)
------------------------------------------------------------------------
```

## Configuration notes

Keep the following information in mind when you are resetting TCAM partitioning:

- Partition TCAMs to best fit the applications that are running on your device.
- If you do not select a non-default profile, the default profile will be in effect.
- The system must be rebooted for TCAM changes to take effect. Always remember to write to memory before you reboot your system.

- Choose a TCAM profile based on all of the application requirements, not on the maximum available TCAM entries for any specific application. The maximum number of entries will vary for different applications.

### *Maximum TCAM address dependencies*

The PowerConnect B-MLXe router can have up to 16,000 static and dynamic MAC address entries stored in the content-addressable memory (TCAM). The ability of the TCAM to store large numbers of addresses depends on the following factors:

- The number of source MAC address being learned by the TCAM
- The number of destination MAC addresses being forwarded by the TCAM
- The distribution of the MAC address entries across ports. For example, if one port is learning all the source MAC addresses, the available TCAM for that port will be used up. In addition, a large number of MAC address entries in the MAC table could increase CPU use.

### *Supernet TCAM partition sharing*

TCAM allocation is optimized for dynamic allocation of resources to each level. If one level runs out of TCAM resources, it can obtain resources that have been allocated to another level but are unused. This feature applies to IPv4 and Layer 3 VPN routes.

### *Configuring adequate TCAM resources for VPLS CPU protection*

There must be adequate TCAM resources available to use VPLS CPU protection. Each end-point and each uplink port requires a single TCAM entry. In addition, if an end-point is a trunk port, one entry is required for each port in the trunk. To determine the number of entries required for your system, add the number of VPLS end-points, ports within a trunk port used as an end-point, and uplink ports. Use this number with the **system-max hw-flooding** command to configure adequate TCAM resources. For more information and an example of the result of this command, see the *NetIron Series Configuration Guide*.

## Common diagnostic scenarios

When troubleshooting TCAM issues, it is helpful to understand how to determine the most appropriate TCAM settings for your system and to know when a device is running out of TCAM. The following sections describe how to work with TCAM settings.

### *Determining appropriate TCAM settings*

When a Dell device boots, the system automatically sets default TCAM partitions. You can customize TCAM settings to best fit the specific tasks your devices are performing.

The default TCAM settings are the same as the default partition percentage settings. For more information about the default cam partitioning profiles for the PowerConnect B-MLXe devices, refer to the *NetIron Series Configuration Guide*.

### *Changing TCAM partition profiles*

TCAM is partitioned on PowerConnect B-MLXe routers using a variety of profiles that you can select depending on your application. To implement TCAM partition profiles, enter the following command.

```
PowerConnect(config)#cam-partition profile ipv4
```

Syntax: **cam-partition profile** [**ipv4** | **ipv4-ipv6** | **ipv4-vpls** | **ipv4-vpn** | **ipv6** | **l2-metro** | **l2-metro-2** | **mpls-l3vpn** | **mpls-l3vpn-2** | **mpls-vpls** | **mpls-vpls-2** | **mpls-vpn-vpls** | **multi-service**]

You can change the default settings based on your specific needs. Dell provides the following TCAM partitioning profiles for PowerConnect B-MLXe routers:

- **ipv4** - Optimized for IPv4 applications.
- **ipv4-ipv6** - Optimized for IPv4 and IPv6 dual-stack applications.
- **ipv4-vpls** - Optimized for IPv4 and MPLS VPLS applications.
- **ipv4-vpn** - Optimized for IPv4 and MPLS Layer 3 VPN applications.
- **ipv6** - Optimized for IPv6 applications.
- **l2-metro** and **l2-metro-2** - Optimized for Layer 2 Metro applications.
- **mpls-l3vpn** and **mpls-l3vpn-2** - Optimized for MPLS Layer 3 VPN applications.
- **mpls-vpls** and **mpls-vpls-2** - Optimized for MPLS VPLS applications.
- **mpls-vpn-vpls** - Optimized for MPLS Layer 3 and Layer 2 VPN applications.
- **multi-service** - Optimized for Multi-Service applications.

To display the TCAM settings on your router, use the **show cam-partition** command, as described in "show cam-partition" on page 12.

## *Determining if a device is running out of TCAM*

The **show cam-partition-usage** command will tell you if your PowerConnect B-MLXe is running out of TCAM.

## *Displaying a TCAM failure count on a line card*

The following example shows how to display a TCAM failure count in a line card through the rconsole.

```
PowerConnect#rcon 1
Connecting to slave CPU 1/1... (Press Ctrl-Shift-6 X to exit)
rconsole-1/1@LP>en
No password has been assigned yet...
rconsole-1/1@LP#show ip cam
  cam-failure-count   Show stats for IP Route that could not be programmed in
cam
rconsole-1/1@LP#show ip cam-failure-count
RecoveryRequired : 0 RecoveryInProgress 0
Total invalid route count 0
 Number of CAM required count
1     Left 6008 Right 0 Total 6008
2     Left 3 Right 0 Total 3
Total number of routes for all level = 6011
 Number of CAM failure count
Total number of routes for all level = 0
 Number of routes not in CAM
Total number of routes for all level = 0
```

# Managing CPU and memory usage

To achieve maximum performance, it is important to understand CPU usage and memory issues in your PowerConnect B-MLXe router. The following sections discuss how to manage memory and CPU:

## Memory and CPU usage show commands

The first step in determining how your device is using memory and CPU is to get a view of the activity. Several **show** commands display information about CPU usage and CPU task activity. This section lists these commands and provides output examples.

### show tasks

**Syntax: show tasks**

This command displays CPU usage statistics for tasks, as shown in the following example.

```
PowerConnect#show tasks
  Task Name    Pri  State  PC        Stack       Size  CPU Usage(%)  task vid
  -----------  ---  -----  --------  --------  ------  ------------  --------
  idle           0  ready  0000448c  0404dfa0    4096           100         0
  monitor       20  wait   0001493c  0404be10   16384             0         0
  wd            31  wait   0001493c  0452df48    8192             0         0
  flash         17  wait   0001493c  04535f48    8192             0         0
  dbg           30  wait   0001493c  04532ef0   16384             0         0
  boot          17  wait   0001493c  0462ee08   65536             0         0
  main           3  wait   0001493c  20819f38  131072             0         1
  itc            6  wait   0001493c  2081eb30   16384             0         1
  tmr            5  wait   0001493c  20854670   16384             0         1
  ip_rx          5  wait   0001493c  20859f78   16384             0         1
  scp            5  wait   0001493c  20882670   16384             0         1
  console        5  wait   0001493c  2088d660   32768             0         1
  vlan           5  wait   0001493c  20895660   16384             0         1
  mac_mgr        5  wait   0001493c  2089c670   16384             0         1
  mrp            5  wait   0001493c  20ca3670   16384             0         1
  vsrp           5  wait   0001493c  20caa668   16384             0         1
  snms           5  wait   0001493c  20caf670   16384             0         1
  rtm            5  wait   0001493c  20cb8670   16384             0         1
  ip_tx          5  ready  0001493c  20f33670   16384             0         1
  rip            5  wait   0001493c  27629668   16384             0         1
. . .
```

### *Sampling CPU usage*

There are three commands that will show you just how much CPU is being used for each task. The first two commands, issued from the management module monitor, identify a task to be sampled, and a rate at which to sample the task, as shown here:

```
MP-1 Monitor>set sample-task <task name>
MP-1 Monitor>set sample-rate 1000
```

### show sample

**Syntax: show sample**

When **set sample-task** and **set sample-rate** are configured, the **show sample** command samples the CPU for a period of time, and prints stack traces. The resulting information shows you what the CPU is doing, which can be especially helpful during periods of high CPU usage. The maximum number of traces that can be stored is 100. To display the print stack traces during the sampling period, enter the following command.

```
MP-1 Monitor>show sample
```

To stop the sampling, enter the following command.

```
MP-1 Monitor>set sample-rate 0
```

## *CPU memory show commands*

The CPU uses memory buffers to handle IPCs and external packets sent and received by the management processor. Buffer pools can consist of 256 bytes, 512 bytes, 1024 bytes, 1040 bytes, and 2048 bytes. All buffers are allocated from these pools on a best-fit basis. The pBuf table maintains start and end addresses, size, stack trace, and number of references for each allocated buffer.

### show bm

**Syntax:** show bm

You can see a general overview of the CPU buffer health using The **show bm** command displays a general overview of the CPU buffer health, as shown in this example:

```
MP Monitor>show bm
  Size    Total  Free  IN-USE  OUT-OF-BUF   BAD-FREE   BAD-REF   BAD-SIG
  ------------------------------------------------------------------------
  256      1023   1023     0        0           0          0         0
  512      1024   1024     0        0           0          0         0
  1024     1024   1024     0        0           0          0         0
  2048     4048   3906   142        0           0          0         0
  10240    512     512    0         0           0          0 0
```

An overview of system activity can be helpful in troubleshooting issues. Too many buffers IN-USE must be justified or there may be memory leaks. BAD-SIG readings may indicate memory corruptions. BAD-REF readings may indicate improper freeing when buffers are shared.

### show bm-dump-mode

**Syntax: show bm-dump-mode**

Use the **show bm-dump-mode** command to pinpoint offending code that may be responsible for double fees and memory leaks.

**NOTE**
A track state of 0 means that the buffer was allocated before the **show bm-dump-mode** was executed.

### show bm-dump-mode hold

**Syntax:** s**how bm-dump-mode hold**

If a buffer leak is suspected, use the **show bm-dump-mode hold** command to help locate the source of the leak, as shown in this example:

```
MP Monitor>show bm-dump-mode hold
 Buffer-ID     Second  Dir  Hold  Application
43636ac        60602   rx   1     o
4362ecc        60585   tx   1     1
436298c        60585   tx   1     1
435fcec        60585   tx   1     1
436f10c        60585   tx   1     1
436c18c        60585   tx   1     1
436a4ec        60394   tx   1     1
436bc8c        60387   tx   1     1
43769cc        60369   tx   1     1
43747ac        110     rx   1     0
.
.
```

### show bm-overflow

**Syntax:** s**how bm-overflow** [**start** | **stop**]

This command stops or starts bm-overflow monitoring.

This command displays buffer overruns, and is enabled by default. Output from this command resembles the following example, where *non-cpu overflow* indicates that the corruption was due to the monitor code (DMA) and not the application code.

```
MP Monitor>show bm-overflow
DABR Watch disabled, DABR disabled because no overflow detected yet.
Info for first overflow
bufptr = 0x29558800 payload = 0x2955810 sig_ptr = 0x2955ffc sig = 0x1234eeee
pid = 3  len = 1500  ref = 1  appl_ref = 1  pri = 0  flags = 0x0
Buf Alloc Stack:

Call Stack:
<-372cc<-371d4<-36dfc<-357d0<-358d4<-4074,-80e9040<-8024d58<-80257cc<-80251f0<-8
025a94<-80241d4<-80200e8<-801ff7c<-8473d70<-84086ac<-84088b4<-8474474<-43f0
Buf Data:
00 00 00 00 00 04 00 00-00 00 00 11 81 00 00 0a
0a 00 6e 82 ff ff ff ff-ff ff 00 2a ff ff ff ff
ff ff ff ff ff ff ff ff-ff ff ff ff ff ff ff ff
b3 fa 00 00 00 00 00 04-00 00 00 00 00 11 81 00
. . .
Prev Buf Data
prev_bufptr = 0x2955000 prev_payload = 0x2955030 prev_sig_ptr = 0x29557fc prev_sig =
0x1234eeee prev_pid = 3 len = 1968
Prev Buf Alloc Stack:
Prev Buf Data:
00 00 00 00 00 04 00 00-00 00 00 11 81 00 00 0a
0a 00 73 6e ff ff ff ff-ff ff 00 29 ff ff ff ff
ff ff ff ff ff ff ff ff-ff ff ff ff ff ff ff ff
ad 4c 0f 16 00 00 4b 16-00 00 00 00 00 04 81 00
00 0a 0a 00 36 ba 00 00-00 00 00 00 00 00 00 00
. . .
next_bufptr = 0x29560000  next_payload = -x2956032  next_sig_ptr = -x29567fc  next_si
0x1234eeee  next_pid = 3  len = 88
Next Buf Alloc Stack:
Next Buf Data:
00 00 00 00 00 04 00 00-00 00 00 11 81 00 00 0a
0a 00 d9 d6 ff ff ff ff-ff ff 00 33 ff ff ff ff
ff ff ff ff ff ff ff ff-ff ff ff ff ff ff ff ff
35 95 00 00 00 00 00 04-00 00 00 00 00 11 81 00
. . .

Info for non-cpu overflow:
Bufptr = NULL
pre_bufptr = NULL
next_bufptr = NULL

Info for DABR hit:
bufptr = 0x2955800  payload = 0x2955810  sig_ptr = 0x2955ffc  sig = 0x123  pid = 3  le
1500  ref = 0  appl_ref = 0  pri = 0  flags = 0x0
. . .
```

## Configuration notes

Several things can affect the memory in your PowerConnect B-MLXe device. For example:

- Whenever you change the table size for a parameter, device memory is reconfigured. Whenever memory is reconfigured, you must save the change to the startup configuration file, then reload the software for the change to take effect.

- Because BGP4 can handle a very large number of routes, it requires a great deal of memory. In a typical configuration with a single BGP4 neighbor, a BGP4 router may need to hold up to 150,000 routes. Many configurations, especially those involving more than one neighbor, can require the router to hold even more routes. PowerConnect B-MLXe devices provide dynamic memory allocation for BGP4 data by automatically allocating memory when needed to support BGP4 neighbors, routes, and route attribute entries. Dynamic memory allocation is performed automatically by the software and does not require a reload.
- You can allocate memory for more VLANs or virtual routing interfaces. By default, you can configure up to 512 VLANs and virtual routing interfaces on the router. Although this is the default maximum, the PowerConnect B-MLXe router can support up to 4094 VLANs and 4095 virtual routing interfaces. (VLAN IDs 0 and 4095 are reserved.) If many of your VLANs will have an identical configuration, you might want to configure VLAN groups. To learn more about VLAN groups and how to configure them, see the *NetIron Series Configuration Guide*.

# Management module diagnostics

The management modules control PowerConnect B-MLXe hardware components, run networking protocols, and provide the Real Time Operating System (RTOS).

Each chassis requires one management module, and can accept a second module for redundancy that works in conjunction with the active management module. If the active module becomes unavailable, the redundant management module automatically takes over the system operation, minimizing system downtime.

## Running management module diagnostics

To run diagnostics on management modules, perform the following steps:

1. Reload the system and immediately hit bbbb until the system boots into monitor mode.
2. Enter the OS by typing **boot os flash primary.**

   The prompt will change from MP Monitor> to MP OS>.
3. From the MP OS prompt enter **diag burn-in**, as shown in the following example.

```
MP-2 OS>diag burn-in
PCI access                - Passed
88E1145 PHY               - Passed
PCMCIA                    - Passed
M41T11 RTC                - Passed
  Port 0 passed
  Port 1 passed
  Port 2 passed
  Port 3 passed
  Port 4 passed
  Port 5 passed
  Port 6 passed
  Port 7 passed
  Port 8 passed
  Port 9 passed
  Port 10 passed
  Port 11 passed
  Port 12 passed
  Port 13 passed
  Port 14 passed
  Port 15 passed
  Port 16 passed
  Port 17 passed
  Port 18 passed
  Port 19 passed
  Port 23 passed
Dx246 Switch Port Loopback - Passed

 ###- PASS -###
```

4. The following command returns the system to normal operation (system reboot):

```
MP-2 OS>reset
```

```
REBOOT S1: NI-MLXe-1Gx24-SFP 24-port 1GbE/100FX Module CARD_STATE_REBOOT 20   0012.f23d.8500
BOOT  S1: NI-MLXe-1Gx24-SFP 24-port 1GbE/100FX Module CARD_STATE_BOOT  20   0012.f23d.8500
CARD_STATE_UP S1: NI-MLXe-1Gx24-SFP 24-port 1GbE/100FX Module CARD_STATE_SW_LOADED 20
0012.f23d.8500
UP    S1: NI-MLXe-1Gx24-SFP 24-port 1GbE/100FX Module  CARD_STATE_UP    20   0012.f23d.8500
```

After the system reboots, you can display the status of the module using the **show module** command, as shown in the following example:

```
PowerConnect#show module
       Module                              Status      Ports  Starting MAC
M1 (upper): NI-MLXe-MR Management Module     Active
M2 (lower):
F1: NI-X-SF Switch Fabric Module         Active
F2: NI-X-SF Switch Fabric Module         Active
F3: NI-X-SF Switch Fabric Module         Active
F4: NI-X-SF Switch Fabric Module         Active
S1: NI-MLXe-1Gx24-SFP 24-port 1GbE/100FX Module  CARD_STATE_SW_LOADED    20
0012.f23d.8500
S2:
S3: NI-MLXe-1Gx24-SFP 24-port 1GbE/100FX Module  CARD_STATE_UP    20   0012.f23d.8550
S4: Configured as NI-MLXe-1Gx20-SFP 20-port 1GbE/100FX Module
```

# Management modules

Table 2 lists the management modules that are available for PowerConnect B-MLXe router.

**TABLE 2** Management modules

| Part number | Description |
|---|---|
| NI-MLX-MR | PowerConnect B-MLXe management module, 1 GB SDRAM, dual PCMCIA slots, EIA or TIA-232 and 10/100/1000 Ethernet ports for out-of-band management. |

# Management module show commands

### show version

**Syntax:** show version

This command displays information about your management modules. For example:

```
PowerConnect(config)#show version
HW: NetIron MLXe Router
Backplane (Serial #: Not Exist,  Part #: Not Exist)
NI-X-SF Switch Fabric Module 1 (Serial #: PR29050242,  Part #: 31523-100A)
FE 1: Type 00000,  Version 0
FE 3: Type 00000,  Version 0
NI-X-SF Switch Fabric Module 2 (Serial #: PR29050246,  Part #: 31523-100A)
FE 1: Type 00000,  Version 0
FE 3: Type 00000,  Version 0
NI-X-SF Switch Fabric Module 3 (Serial #: PR30050270,  Part #: 31523-100A)
FE 1: Type 00000,  Version 0
FE 3: Type 00000,  Version 0
==========================================================================
SL M1: NI-MLXe-MR Management Module Active (Serial #: PR30050511, Part #: 31524-0
00A):
Boot     : Version 3.5.0T165 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on May 31 2009 at 14:42:00 labeled as xmprm03500b155
 (424038 bytes) from boot flash
Monitor  : Version 3.5.0T165 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on May 31 2009 at 14:41:14 labeled as xmb03500b155
 (424045 bytes) from code flash
IronWare : Version 3.5.0T163 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on Jun 11 2009 at 07:15:58 labeled as mlxe03500b181
 (5816681 bytes) from Primary
Board ID : 00 MBRIDGE Revision : 18
--More--, next page: Space, next line: Return key, quit: Control-c^C
```

### show module

**Syntax:** show module

This command displays the status of the management modules. Enter this command at any CLI level. Output resembles the following example:

```
PowerConnect#show module
Module Status Ports Starting MAC
M1 (left): NI-MLX-MR Management Module Active
M2 (right): NI-MLX-MR Management Module Standby (Ready)
```

Management module status is either active or standby. Standby modules can be in one of the following modes:

- Init – The module is currently initializing as the standby module.
- Ready – The module is ready to take over as the active module, if necessary.
- Wait – The module is awaiting boot information from the active management module.
- Sync – The active module is currently synchronizing files with the standby module.

### show redundancy

**Syntax: show redundancy**

This command displays module switchover activity, as shown in this example:

```
PowerConnect#show redundancy
=== MP Redundancy Settings ===
Default Active Slot = 17
Running-Config Sync Period = 7 seconds
=== MP Redundancy Statistics ===

Current Active Session:
Active Slot = 9, Standby Slot = 10 (Ready State), Switchover Cause = No
Switchover
Start Time = 0-0-17 19:47:39 (Wednesday)

Previous Active Session #1:
Active Slot = 10, Standby Slot = 9, Switchover Cause = Active Rebooted
Start Time = 0-0-17 19:46:9 (Wednesday), End Time = 0-0-17 19:47:39 (Wednesday)

Previous Active Session #2:
Active Slot = 9, Standby Slot = 10, Switchover Cause = Active Rebooted
Start Time = 0-0-17 19:44:14 (Wednesday), End Time = 0-0-17 19:46:9 (Wednesday)
```

### show log

**Syntax:** show log

This command allows you to view the system log or the traps logged on an SNMP trap receiver, as shown in the following example, which indicates that one switchover occurred on the management module in slot 9.

```
PowerConnect#show log
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
Buffer logging: level ACDMEINW, 24 messages logged
level code: A=alert C=critical D=debugging M=emergency E=error
I=informational N=notification W=warning
Static Log Buffer:
Sep 28 11:31:25:A:Power Supply 1, 1st left, not installed
Sep 28 11:31:25:A:Power Supply 3, middle left, not installed
Sep 28 11:31:25:A:Power Supply 4, middle right, failed
Sep 28 11:31:25:A:Power Supply 5, 2nd right, not installed
Dynamic Log Buffer (50 lines):
Sep 27 18:06:58:I:Interface ethernet6/2, state up
Sep 27 18:06:57:I:Interface ethernet3/2, state up
Sep 27 15:39:42:I:Interface ethernet3/2, state up
Sep 27 15:39:42:I:Interface ethernet6/2, state up
...
Sep 27 14:23:45:N:Module up in slot 6
Sep 27 14:23:45:N:Module up in slot 3
Sep 27 14:23:27:A:Management module at slot 9 state changed from standby to
active
```

## Management module debug commands

There are no debug commands that are specific to management modules.

## Configuration notes

### *Establishing a management session*

Management sessions may be established through the management port on the active management module. If a switchover occurs, the management port on the original active module shuts down and all open CLI, Web management interface, and IronView Network Manager sessions close. You can open new sessions with the new active module after the switchover, if the new module has a management port.

For example, if you were accessing the Web management interface using a PC connected to the original active management port, you can open a new session if a PC is connected to the management port on the new active module. Open a new session using the same IP address you used before the switchover. (If a switchover occurs, the IP address you configured on the original active module is automatically assumed by the new active module.)

## Common diagnostic scenarios

### *Management module switchover events*

In a ***redundant*** management module configuration, whenever the standby management module assumes the role of active module a ***switchover*** event has occurred. This happens when the active module becomes unavailable (for example, power is lost or a component fails), you perform a manual switchover, or you remove or replace the active management module. When a switchover occurs between the active and standby modules, management sessions, Syslog, and SNMP traps may be affected.

### *When an active module becomes unavailable*

The following events will cause an active module to become unavailable and trigger a switchover:

- An active module experiences a problem significant enough to cause a reset of the module. If this is a software problem, capture the output from the **show tech** command. If the output contains a crash dump, contact Dell Support.
- The active module loses power. This may happen if a power supply fails, or because of a general disruption in power.

Before a switchover occurs, the active module resets itself and sends an interrupt signal to the standby module. The standby module then becomes the active module, allowing the interface modules to continue forwarding traffic.

The new active module also begins to manage the system. When the original active module again becomes available or is replaced, the new active module resumes in the role of standby.

### *Management module error messages*

The following messages are displayed in the event of a switchover event, if there is a problem with the standby management module in a redundant configuration:

```
Warning: Active MP running image is not in its flash or PCMCIA card, image
synchronization is not possible:

Warning: Standby Module is not allowed - put standby MP in reset.
```

### *Switchover Syslog and SNMP traps*

When a switchover occurs, the system sends a Syslog message to the local Syslog buffer and to the Syslog server, if one is configured. The system also sends an SNMP trap to the receiver, if one is configured.

if the system is reset as the result of a switchover to the standby management module, the system sends a warm start message and trap.

## Monitoring management module redundancy

You can monitor the following aspects of management module redundancy:

- The status of the management modules (if a module is the active or standby module).
- The switchover history for the management modules.

For more information, see the *NetIron Series Configuration Guide*.

## Management module LEDs

Management modules contain 6 LEDs that indicate operational status. Once you have installed a management module and powered on the system, you can read the LED indicators on the module faceplate. Table 3 lists these LEDs and describes what to do if an LED shows that the module is not operating properly.

**TABLE 3** Management module LED indicators

| LED | Position | State | Meaning |
|---|---|---|---|
| Slot 1 and Slot 2 | Adjacent to the PCMCIA slot it represents | On or blinking | The software is currently accessing the flash card. |
| | | Off | The software is not accessing a PCMCIA flash card inserted in a slot. If this occurs, you should:<br>• Make sure the flash card is fully seated in the connector.<br>• From a command prompt, type **dir/slot1** or **dir/slot 2** to see if either slot is readable. |
| Active | Lower left | On | The module is functioning as the active management module. |
| | | Off | The module is functioning as the redundant management module. |

**TABLE 3** Management module LED indicators

| LED | Position | State | Meaning |
|---|---|---|---|
| Pwr | Upper left | On | The module is receiving power. |
| | | Off | The module is not receiving power. If this occurs, you should:<br>• Make sure the module is seated properly.<br>• Make sure the power supply is operating properly - the Power LED should be green.<br>• Try the module in another slot or chassis. |
| 10/100/1000 Ethernet port | Above and to right of RJ-45 connector | On (green) | A link is established with the remote port. |
| | | Off | No link is established with the remote port. If this occurs, you should:<br>• Using a straight-through cable to the switch, verify that interface management is enabled and the link is good. |
| 10/100/1000 Ethernet Port | Above and left of RJ-45 connector | On or blinking (yellow) | The port is transmitting and receiving packets. |
| | | Off for extended period | The port is not transmitting or receiving packets. If this occurs, you should:<br>• Verify that the port is enabled and configured properly for auto-negotiation.<br>• Try another port to identify a hardware failure. |

# Interface module diagnostics

To run diagnostics on an interface module, perform the following steps:

**NOTE**
Dell recommends that you remove connections to all ports on the module so the module does not receive traffic while the diagnostics are running.

1. Boot the module into interactive mode by entering the following command:

```
PowerConnect#lp boot sys in 1
PowerConnect#Reset slot 1
Slot 1: booted to Interactive Mode.
```

2. With the module in interactive mode, rconsole to the module by entering the following command:

```
PowerConnect#rcon 1
Remote connection to LP slot 1 established
Press CTRL-X or type 'exit' to disconnect it
LP-1 Monitor>
```

3. Boot the module in the OS mode:

```
LP-1 Monitor>boot os flash primary
LP-1 OS>
LP-1 OS>
```

4. Run the diagnostic. You should see an output similar to the following:

```
LP-1 OS>diag burn
PRAM 0 -- TM DDRII support disabled
XPP PLL Status Register at 0 micro-sec = 0x0000ff0f
XPP PLL Status Register at 2 micro-sec = 0x0000ff0f
XPP PLL Status Register at 4 micro-sec = 0x0000ff0f
XPP PLL Status Register at 6 micro-sec = 0x0000ff0f
XPP PLL Status Register at 8 micro-sec = 0x0000ff0f
XPP PLL Status Register at 10 micro-sec = 0x0000ff0f
XPP PLL Status Register at 12 micro-sec = 0x0000ff0f
XPP PLL Status Register at 14 micro-sec = 0x0000ff0f
XPP PLL Status Register at 16 micro-sec = 0x0000ff0f
XPP PLL Status Register at 18 micro-sec = 0x0000ff0f
XPP PLL Status Register at 20 micro-sec = 0x0000ff0f
XPP PLL Status Register at 22 micro-sec = 0x0000ff0f
XPP PLL Status Register at 24 micro-sec = 0x0000ff0f
XPP PLL Status Register at 26 micro-sec = 0x0000ff0f
XPP PLL Status Register at 28 micro-sec = 0x0000ff0f
XPP Reset Sequence Done -  PLL Status Register = 0x0000ff0f
XPP Reset Sequence Done -  PLL Status Register = 0x0000ff0f
XPP Reset Sequence Done -  PLL Status Register = 0x00c0ffff
XPP: P1 board or higher detected
PASSED
Dev 0 PRAM passed
STATSRAM 0 -- PASSED
Dev 0 STATSRAM passed
TXVLAN Table 0 -- PASSED
Dev 0 TXVLANRAM passed
CAM2PRAM 0 -- PASSED
Dev 0 CAMTOPRAMRAM passed
AGERAM 0 -- Pass 1
Pass 2

AGERAM memory tested for 2097152 entries
PASSED
Dev 0 AGERAM passed
ServTypeTable 0 -- PASSED
Dev 0 SERVTYPETABLERAM passed
TXNEXTHOP TABLE 0 -- PASSED
Dev 0 TXNEXTHOPTABLERAM passed
XPP 0 TOS TABLE -- PASSED
Dev 0 TOSTABLERAM passed
XPP 0 MULTICAST START OFFSET TABLE -- PASSED
Dev 0 MCASTSTARTTABLERAM passed
XPP 0 MULTICAST REPLACEMENT TABLE -- PASSED
Dev 0 MCASTREPLTABLERAM passed
PRAM 1 -- TM DDRII support disabled
XPP PLL Status Register at 0 micro-sec = 0x0000ff0f
XPP PLL Status Register at 2 micro-sec = 0x0000ff0f
XPP PLL Status Register at 4 micro-sec = 0x0000ff0f
XPP PLL Status Register at 6 micro-sec = 0x0000ff0f
XPP PLL Status Register at 8 micro-sec = 0x0000ff0f
XPP PLL Status Register at 10 micro-sec = 0x0000ff0f
XPP PLL Status Register at 12 micro-sec = 0x0000ff0f
XPP PLL Status Register at 14 micro-sec = 0x0000ff0f
XPP PLL Status Register at 16 micro-sec = 0x0000ff0f
XPP PLL Status Register at 18 micro-sec = 0x0000ff0f
XPP PLL Status Register at 20 micro-sec = 0x0000ff0f
XPP PLL Status Register at 22 micro-sec = 0x0000ff0f
XPP PLL Status Register at 24 micro-sec = 0x0000ff0f
XPP PLL Status Register at 26 micro-sec = 0x0000ff0f
```

```
XPP PLL Status Register at 28 micro-sec = 0x0000ff0f
XPP Reset Sequence Done -  PLL Status Register = 0x0000ff0f
XPP Reset Sequence Done -  PLL Status Register = 0x0000ff0f
XPP Reset Sequence Done -  PLL Status Register = 0x00c0ffff
XPP: P1 board or higher detected
```

5. Once the diagnostics are complete, return the interface module to operational status by entering the following commands.

```
PowerConnect#lp boot sys flash pri 1
PowerConnect#reset slot 1
```

6. Reconnect the ports you disconnected prior to running the tests.

## Interface modules

Table 4 lists the interface modules that are available for PowerConnect B-MLXe router.

**TABLE 4** Interface modules

| Part number | Description |
|---|---|
| NI-MLX-1Gx48-T-A | 48-port 10/100/1000Base-T mini RJ-21 interface module with IPv4, IPv6, and MPLS hardware support. |
| NI-MLX-10Gx8-M | 8-port 10 Gbps Ethernet (M) module with IPv4, IPv6, and MPLS hardware support - requires SFP+ optics, high-speed fabric modules, and high speed fans NIBI-16-FAN-EXH-A on 16-slot routers. Supported for PowerConnect B-MLXe device only. |
| NI-MLX-10Gx8-D | 8-port 10 Gbps Ethernet (D) module with IPv4 and IPv6 hardware support - requires SFP+ optics, high-speed fabric modules, and high speed fans (NIBI-16-FAN-EXH-A) on 16-slot routers. Supported for PowerConnect B-MLXe device only. |
| NI-MLX-10Gx4 | 4-port 10 Gbps Ethernet module with XFP optical interfaces for wire-speed performance |
| NI-MLX-1Gx24-GC | 24-port 1 Gbps Ethernet copper module with RJ45 interfaces for wire-speed performance |
| NI-MLX-1Gx24-GF | 24-port 1Gbps fiber module for wire-speed performance and 20% increased in port density |

## Interface module show commands

### show media

**Syntax:** show media

This command displays information about the media installed in ports, as shown in this example, which provides Type, Vendor, Part Number, Version and Serial number of the SFP or XFP device installed in a port. If no SFP or XFP device is installed in a port, the Type field displays N/A, the Vendor field is empty, and the other fields display Unknown.

### show optics

**Syntax:** show optics

This command displays optics information for XFP and SFP ports, including temperature, transmit power, receive power, transmit bias and current:

```
PowerConnect#show optics 4
Port Temperature Tx Power Rx Power Tx Bias Current
+----+-----------+----------+------------+-------------------+
4/1 30.8242 C -001.8822 dBm -002.5908 dBm 41.790 mA
Normal Normal Normal Normal
4/2 31.7070 C -001.4116 dBm -006.4092 dBm 41.976 mA
Normal Normal Normal Normal
4/3 30.1835 C -000.5794 dBm 0.000 mA
Normal Low-Alarm Normal Low-Alarm
4/4 0.0000 C 0.000 mA
Normal Normal Normal Normal
```

### show tm-voq-stat

**Syntax: show tm-voq-stat src_port pos** <*source-port*> **dst_port pos** | **ethernet** <*destination-port*> <*priority*>

This command displays traffic management statistics for interface modules.

### show tm statistics

**Syntax: show tm statistics** <*slot number*>

You can monitor traffic manager statistics to ensure that traffic load balancing is working properly. The following example shows traffic statistics for an interface module identified by its slot number:

```
PowerConnect#show tm statistics slot 4
--------- Ports 4/1 - 4/20 ---------
Ingress Counters:
Total Ingress Pkt Count: 22
EnQue Pkt Count: 0
EnQue Byte Count: 0
DeQue Pkt Count: 0
DeQue Byte Count: 0
TotalQue Discard Pkt Count: 0
TotalQue Discard Byte Count: 0
Oldest Discard Pkt Count: 0
Oldest Discard Byte Count: 0
Egress Counters:
EnQue Pkt Count: 0
EnQue Byte Count: 0
Discard Pkt Count: 0
Discard Byte Count: 0
```

## *Clearing traffic management statistics*

To clear traffic management statistics, enter the following command.

```
clear tm-voq-stat src_port dst_port
```

For more information about other commands that can be useful for displaying information about interface modules, see the *NetIron Series Configuration Guide*. These commands include:

```
clear tm-voq-stat src_port dst_port [priority]
show tm-voq-stat src_port multicast
clear tm-voq-stat src_port multicast
show tm-voq-stat   src_port   cpu-queue
```

```
clear tm-voq-stat   src_port   cpu-queue
```

### show version

**Syntax: show version**

This command displays information about your system, including all installed interface modules as shown in the following example:

```
PowerConnect# show version
Monitor  : Version 3.5.0B6T165 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on Jul  9 2009 at 18:46:18 labeled as xmb03500B6
 (424489 bytes) from code flash
IronWare : Version 3.5.0T163 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on Jul 19 2009 at 07:17:58 labeled as mlxe03500b279
 (5836796 bytes) from Primary
Board ID : 00 MBRIDGE Revision : 18
916 MHz Power PC processor  (version 8003/0101) 166 MHz bus
512 KB Boot Flash (AM29LV040B), 32 MB Code Flash (MT28F128J3)
2048 MB DRAM
Standby Management uptime is 1 days 52 minutes 24 seconds
==========================================================================
SL 3: NI-MLXe-1Gx24-GC 24-port 10/100/1000 Copper Module (Serial #: SA33061518, Part #:
35555-200A)
Boot     : Version 3.5.0B6T175 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on Jul  9 2009 at 18:48:28 labeled as xmlb03500B6
 (387059 bytes) from code flash
Compiled on Jul  9 2009 at 18:47:52 labeled as xmlprm03500B6
 (387074 bytes) from boot flash
Monitor  : Version 3.5.0B6T175 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on Jul  9 2009 at 18:48:28 labeled as xmlb03500B6
 (387059 bytes) from code flash
IronWare : Version 3.5.0T177 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on Jul 19 2009 at 07:50:18 labeled as xmlp03500b279
 (3163145 bytes) from Primary

FPGA versions:
Valid PBIF Version = 2.13, Build Time = 2/14/2009 11:55:00
Valid XPP Version = 3.05, Build Time = 5/14/2009 11:12:00
BCM5695GMAC 0
BCM5695GMAC 1
BCM5695GMAC 2
BCM5695GMAC 3
666 MHz MPC 8541 (version 8020/0020) 333 MHz bus
512 KB Boot Flash (AM29LV040B), 16 MB Code Flash (MT28F640J3)
1024 MB DRAM, 8 KB SRAM, 0 Bytes BRAM
PPCR0: 1024K entries CAM, 16384K PRAM, 2048K AGE RAM
LP Slot 3 uptime is 1 days 52 minutes 23 seconds
```

## Interface module debug commands

There are no **debug** commands specific to interface modules.

## Configuration notes

Common diagnostic situations involving interface modules can include:

- The module is not receiving power.

- A link has not been successfully established.
- A port is not transmitting or receiving packets.
- Packets fail to enter an ingress queue on traffic manager - This can happen because the queue has reached its maximum length, or Weighted Random Early Detection (WRED) is enabled to prevent an output queue from ever filling to capacity.

### *Reading interface module LEDs*

Interface module front panels commonly contain two LEDs for each port to indicate the operational status of that port. Diagnostic readings based on these LEDs are similar for most interface modules. Table 5 describes the LEDs for a 10 Gigabit Ethernet module and lists the steps to take if there is a problem. Although this table lists information for a specific module, this information can be used to troubleshoot most interface modules.

**TABLE 5** 10 Gigabit Ethernet module LEDs

| LED | Position | State | Meaning |
|---|---|---|---|
| Link | Left of each Ethernet port | On | A link is established with the remote port. |
| | | Off | A link is not established with the remote port. If this occurs you should check GBIC and cable connections. |
| Active | Left of each Ethernet port | On | The port is transmitting and receiving packets. |
| | | Off | The port is not transmitting or receiving packets. If this occurs you should:<br>• Issue a **show interface <slot/port>** command. Determine from the output whether the port is up and the port is transmitting and receiving.<br>• If there is no link, perform the following steps:<br>1 Swap the XFP on one end of the connection with a known good XFP.<br>2 Swap the XFP on the other end of the connection with a known good XFP.<br>3 Clean the optical connections with a fiber cleaning cloth.<br>4 Use a known good fiber connector.<br>5 Take an optical reading: first enable optical monitoring, then issue a **show optic** command for the module. The output shows power readings. |

For detailed information about the interface modules supported on PowerConnect B-MLXe devices, refer to the *NetIron Series Configuration Guide*.

## Common diagnostic scenarios

### *Troubleshooting switchovers and interface module synchronization*

Interface modules continue to send traffic even when a switchover occurs between management modules. After a switchover event, interface modules send updates to the new active management module, which verifies that the interface modules are synchronized.

If the new active management module becomes out of sync with an interface module, information on the interface module may be overwritten, which can cause an interruption of traffic forwarding. This should only occur if there is a Layer 3 topology change elsewhere in the network during the management switchover. PowerConnect B-MLXe routers support Layer 3 hitless failover with graceful restart for high availability routing in protocols such as BGP and OSPF. With these high availability features enabled, when a router experiences a failover or restart, forwarding disruptions are minimized, and route flapping is diminished to provide continuous service during a switchover or restart event.

# IPC diagnostics

Inter-Processor Communication (IPC) is a set of techniques that manage the exchange of data between two or more processors running on one or more computers connected by a network. IPC methods include message passing, synchronization, shared memory, and remote procedure calls (RPCs).

## IPC show commands

### ipc show dy-sync

**Syntax:** ipc show dy-sync

This command displays dynamic IPC sync statistics. Output resembles the following:

```
PowerConnect#ipc show dy-sync
ARP table sync_type=2, enabled=1
b_cast: serial # 000004DA, packets 1242, msg 1242, hello=160454
b_cast buf=02962862, index=20, msg_i=0
u_cast  1, msg # 1, dropped (alloc failure)=0
u_specific reply 0, u_specific miss 0
10/1:1
The above is slot/cpu u_cast pkt#
DAI table sync_type=30, enabled=1
b_cast: serial # 00000001, packets 1, msg 2, hello=160454
b_cast buf=0298B862, index=20, msg_i=0
u_cast  1, msg # 2, dropped (alloc failure)=0
u_specific reply 0, u_specific miss 0
10/1:1
The above is slot/cpu u_cast pkt#
Label to VRF ta sync_type=25, enabled=1
b_cast: serial # 00000000, packets 0, msg 0, hello=160454
b_cast buf=0296A862, index=20, msg_i=0
u_cast  1, msg # 0, dropped (alloc failure)=0
u_specific reply 0, u_specific miss 0
10/1:1
The above is slot/cpu u_cast pkt#
ND6 neighbor ta sync_type=12, enabled=1
b_cast: serial # 000005D5, packets 1493, msg 1493, hello=160454
b_cast buf=0299E862, index=20, msg_i=0
u_cast  2, msg # 1, dropped (alloc failure)=0
u_specific reply 0, u_specific miss 0
10/1:2
The above is slot/cpu u_cast pkt#
NHT sync_type=23, enabled=1
b_cast: serial # 00000000, packets 0, msg 0, hello=160454
b_cast buf=02969862, index=20, msg_i=0
u_cast  118, msg # 4096, dropped (alloc failure)=0
u_specific reply 0, u_specific miss 0

10/1:118
The above is slot/cpu u_cast pkt#
...
```

### ipc show names

**Syntax:** ipc show names

This command displays IPC message type names, with numbers that correspond to the **show statistics** output. The following display is a result of the **ipc show names** command.

```
PowerConnect#ipc show names
Names of registered IPC message types:
2 IPC reset message
3 IPC startup message
5 IPC Test message
8 File Tx End
12 File Tx Request
13 LP Card Boot
14 LP Card SW Loaded
15 LP Card Oper Info
16 LP Ports Oper Info
17 LP Port Oper Info
18 LP Stripe Sync Send Done
19 LP Stripe Sync Status
31 LP Stripe Sync Loss
32 Slot Info
33 Module Reboot
40 MP Red Standby Boot Info
41 MP Red Standby SW Loaded
45 MP Red Cmd
49 MP Red Standby Info
50 MP SW Upgrade Info
72 HAL Set Port Value
74 HAL Set Port Values
76 HAL Set Ports Value
78 HAL Set Port Active Mac Address
80 HAL Set Port Active Mac Addresses
82 HAL Set Port Data
84 HAL Set Module Value
86 SET LP PROM
93 HAL Add Fid Portmask
95 HAL Delete Card
97 HAL Add Card
104 ACL Mirror Status from LP
106 Trunk query response
110 LP MAC table sync
113 LP MAC free
115 LP MAC SA learn
116 LP MAC DA learn
120 Port security
143 VLAN LP ACK
161 VSRP Update Session
180 AS_PATH REQ
182 IPV6 NEIGHBOR REQUEST
190 LP Mcast Main
191 L2 Mcast msg
192 L2 Mcast6 msg
202 ARP LP REQ
208 AuthReq from LP
209 LP dot1ag Main210 L4 STATUS FROM LP
211 POS Status Update
214 LP CLI show stat Value
```

```
 216 SYNC RTC req
 217 SysLog LP message
 219 LP Temperature Res
 232 VPLS MAC SYNC
 233 IPC Text Msg
 235 Port Stat Req
 236 Port Stat Sampling
 237 IPC_MSGTYPE_MP_SYNC_REQ_0
 238 IPC_MSGTYPE_MP_SYNC_REQ_1
 242 IPC_MSGTYPE_MP_SYNC_REQ_5
 243 Snmp mp sync message
 246 IPC_MSGTYPE_MP_SYNC_REQ_9
 278 IPC_MSGTYPE_MP_SYNC_ACK
 279 BGP STATUS FROM MP
 280 OSPF STATE FROM MP
 284 10g wan phy alarm stat
 287 BFD
 288 BFD
 289 TM logs
```

### ipc show statistics

**Syntax:** ipc show statistics

This command displays IPC statistics, as shown in this example:

```
PowerConnect#ipc show statistics
----------- CPU and Reliable IPC Status (Slot/Cpu) ----------
 9/0 ALIVE   Expected rx seq:  914  Next tx seq:  507  In tx seq:    0
---------------------- Message count ----------------------
 UnrelRxPkt=494902   RelRxPkt=20371    UnrelTxPkt=394404   RelTxPkt=1531
 UnrelRxMsg=767517   RelRxMsg=20371    UnrelTxMsg=394405   RelTxMsg=3205

----MsgType:Count (* instead of : means unregistered type)---
 Reliable RX   :   16:1         17:17       182:2871     201:1
                  202:1251     217:1        236:16227    287:1
 Unreliable RX :    1:81261      2:1          3:1         13:1
                   14:1         15:2         16:1         17:1
                   33:1         72:1         76:2         78:2
                   80:2        113:3        115:3741     116:271709
                  198:813      199:405828   215:87       219:4058
                  289:1
  Reliable TX   :   24:1         28:9         29:1         98:1
                  100:1        103:2        107:1        108:4
                  109:8        110:3027     112:3        114:4
                  117:1        118:1        122:1        124:1
                  125:2        126:1        129:5        134:1
                  144:1        147:5        155:1        162:1
                  163:1        164:2        165:1        167:1
                  170:1        171:17       173:1        177:1
                  178:13       179:1        183:1        185:1
                  186:30       187:8        189:1        194:11
                  201:7        203:1        205:1        227:2
                  229:1        286:18       290:1
```

```
Unreliable TX :    1:90757      2:1         21:1         22:10
                    30:2         34:1         71:1         75:2
                    77:2         79:2        110:2        195:2773
                   196:133952   198:162291   199:132      215:63
                   216:46       218:4058     243:309

-------------------- IPC Debug counters ---------------------
 BufAllocFail  =0        BufFreeFail   =0        BadChecksum   =0
 TxTooBigPkt   =0        TxBadMsgType  =0        ItcSendFail   =0
 RxTooBigPkt   =0        RxBadMsgType  =0        CardDownEvent =0
 TxBadContent  =0        TxBadFid      =0        TxEmptyFidMask=0
 TxRelSemLock  =0        TxUnrelSemLock=0        TxRelQFullErr =0
 DelayTxRelErr =0        DelayTimerErr =0        DelayFlushErr =0
 DelyTxRelQFul =0        RxBadFid      =0        RxNoCallback  =0
 RxBadAckAddr  =0        TxUnrelErr    =0        TxRelErr      =0
 SendBufError  =0        BufAlreadyFree=0        Retransmits   =0
 YieldSendBuf   =0        YieldGetBuf   =0        RxBadContent  =0

 QCountMinusErr=0        RxQIdxErr     =0        LastBadRxQIdx =0
 LastRelRxMsgFr=9        LastRelRxQidx =913      LastRelRxBlkAd=15990396
 RelRxQFull    =0        LastRelRxQFul =0        RelRxOutOfWin =1
 RelRxAlrdyVald=0        RelRxFrags    =0        LastAckTxSlot =9
 LastAckTxSeq  =914      LastAckRxSeq  =507      LastAckRxSlot =9
 RxAckNotInWin =0        RxAckAlrdyAckd=0        RxAckInvdBlk  =0
 RxAckNotInRang=0        RetransBadQIdx=0        LastBadTxQIdx =0
 LastBadTxQIdxQ=0        LastBadTxQidxW=0        LastBadTxQIdxA=0
 CurrentRetranQ=0        FirstRetranWin=0        FirstRetranBlk=0
 FirstRetranBuf=0        RxAckForNActQ =0
```

## IPC debug commands

### debug ip ipc

**Syntax:** [no] **debug ip ipc**

This command generates information about inter-processor communication (IPC) activity in IP modules. The output generally includes a record of IPC messages sent and received, and any errors or warnings related to IP IPC. Several examples follow.

The following example indicates that the system is unable to obtain enough buffer space to send IPC messages to line cards.

```
PowerConnect#debug ip ipc
error - ip_ipc_icmp_config_info. system out of buffer
error - ip_ipc_ve_mac_addr system out of buffer
error - ip_ipc_vport_mask_info system out of buffer
error - ip_ipc_clear_cache_msg system out of buffer
error - ip_ipc_config_info. system out of buffer
error - ip_ipc_mcast_info. system out of buffer
```

The following message indicates that the routing table manager (RTM) is sending initialize data to line card 1, cpu 1.

```
RTM: ipc sync init data to (1,1), max routes 100
```

In the following example, Layer 3 port configuration information (state, mtu, redirect, encap, etc.) is being set for the specified port.

```
IP/IPC: set port data, port 1/10, type:2 value 1
```

In the following example, an IP address for a given port (1/10) is being sent to the line cards.

```
IP/IPC: set port address, port 1/10, add1, addr 10.10.10.1
```

In the following example, forwarding information is being sent to the line cards.

```
IP/IPC: send port table, FID_ID 10
```

In the following example, tunnel forwarding information is being sent to the line cards.

```
IP/IPC: send tunnel table, FID_ID 10
```

In the following example, tunnel forwarding information is being sent to the line cards.

```
IP/IPC: send tunnel config, size 56
```

In the following example, DHCP configuration information is being sent to the line cards.

```
IP/IPC: send port dhcp index, FID_ID 10
```

In the following example, a DHCP list with 200 entries is being sent to the line cards.

```
IP/IPC: send dhcp list, size 200
```

## Common diagnostic scenarios

If errored counts are continuously incrementing at a fast pace, or if the “In tx seq:” value continues to increase, contact Dell Technical Support.

# Switch fabric modules

Switch fabric modules switch user packets from one interface module in a chassis to another. Switch fabric modules are hot swappable.

## Switch fabric fault monitoring

The Switch Fabric Fault Monitoring feature lets you display information about the current status of links between the Switch Fabric Modules (SFMs) and Interface modules and sends log messages to the console regarding the “UP” or “DOWN” status of the Switch Fabric Modules.

## Switch fabric show commands

**show sfm-links all**

**Syntax: show sfm-links** [<*sfm-number*> | **all**]

- <*sfm-number*> - Specifies an SFM for which you want to display link information.
- **all** - Displays link information for all SFMs in the chassis.

**NOTE**

If the number of non-operational links falls below the minimum threshold, you will see this warning:

```
WARN: LP 3 has 8 links up, less than minimum to guarantee line rate traffic
forwarding
```

This warning is displayed to inform users that the line rate traffic will not be maintained.

The **show sfm-links all** command displays information about the current status of links between the SFMs and Interface modules in PowerConnect B-MLXe devices. Each line in the output represents a link between an SFM and an Interface module.

```
PowerConnect#show sfm-links all
SFM#/FE# | FE link# | LP#/TM# | TM link# | link state
----------+-----------+---------+-----------+---------
2 / 1 | 32 | 3 / 1 | 13 | UP
2 / 1 | 31 | 3 / 2 | 01 | UP
2 / 1 | 11 | 3 / 1 | 01 | UP
2 / 1 | 12 | 3 / 2 | 13 | UP
2 / 3 | 32 | 3 / 1 | 19 | UP
2 / 3 | 31 | 3 / 2 | 07 | UP
2 / 3 | 11 | 3 / 1 | 07 | UP
2 / 3 | 12 | 3 / 2 | 19 | UP
3 / 1 | 32 | 3 / 1 | 16 | UP
3 / 1 | 31 | 3 / 2 | 04 | UP
3 / 1 | 11 | 3 / 1 | 04 | UP
3 / 1 | 12 | 3 / 2 | 16 | UP
3 / 3 | 32 | 3 / 1 | 22 | UP
3 / 3 | 31 | 3 / 2 | 10 | UP
3 / 3 | 11 | 3 / 1 | 10 | UP
3 / 3 | 12 | 3 / 2 | 22 | UP
WARN: LP 3 has 8 links up, less than minimum to guarantee line rate traffic
forwarding
```

### show version

**Syntax:** show version

This command displays information about the switch fabric modules installed in your router chassis, as shown in the following example:

```
PowerConnect(config)#show version
HW: PowerConnect MLXe Router
Backplane (Serial #: Not Exist,  Part #: Not Exist)
NI-X-SF Switch Fabric Module 1 (Serial #: PR29050242,  Part #: 31523-100A)
FE 1: Type 00000,  Version 0
FE 3: Type 00000,  Version 0
NI-X-SF Switch Fabric Module 2 (Serial #: PR29050246,  Part #: 31523-100A)
FE 1: Type 00000,  Version 0
FE 3: Type 00000,  Version 0
NI-X-SF Switch Fabric Module 3 (Serial #: PR30050270,  Part #: 31523-100A)
FE 1: Type 00000,  Version 0
FE 3: Type 00000,  Version 0
.
.
.
```

**show sfm-utilization**

**Syntax: show sfm-utilization** [<*sfm-number*> | **all**]

- <*sfm-number*> - Specifies the SFM for which to display the utilization information.
- **all** - Displays utilization information for all SFMs in the chassis.

The **show sfm-utilization** command displays bandwidth usage on all SFMs on the device, as shown in the following example:

```
PowerConnect#show sfm-utilization all
SFM#2
----------+-----------+---------+-----------+---------
last 1 second utilization = 0.4%
last 5 seconds utilization = 0.3%
last 1 minute utilization = 0.1%
last 5 minutes utilization = 0.0%
SFM#3
----------+-----------+---------+-----------+---------
last 1 second utilization = 0.4%
last 5 seconds utilization = 0.4%
last 1 minute utilization = 0.1%
last 5 minutes utilization = 0.0%
last 5 minutes utilization = 0.0%
```

To display bandwidth usage for a single SFM, enter the **show sfm-utilization** <sfm number> command:

```
PowerConnect#show sfm-utilization 2
SFM#2
----------+-----------+---------+-----------+---------
last 1 second utilization = 0.4%
last 5 seconds utilization = 0.3%
last 1 minute utilization = 0.1%
last 5 minutes utilization = 0.0%
```

## Switch fabric debug commands

There are no switch fabric-specific **debug** commands.

## Common diagnostic scenarios

The following common scenarios may occur with switch fabric modules:

- The switch fabric module is not receiving power
- If the switch fabric module is inactive (not switching packets) the links are down. Use the **show switch fabric all** command to identify which links are down.

### *The switch fabric module is not receiving power*

The switch module front panel includes two LEDs, one labeled Pwr (power) and one labeled Active.

- If the Pwr LED is off for an extended period of time, the switch fabric module is not receiving power. Check the power connection and the power supply. Make sure the module is seated properly in the backplane.
- If the Pwr LED is on, but the Active LED is off, the module is not in active mode and cannot switch packets. In this case, see the following section.

### *The switch fabric module is unable to switch packets*

If the switch fabric module is receiving power, but is still unable to switch packets, disable it and then re-enable it to determine if the problem re-occurs. You may need to perform these steps on each switch module in your device to determine which one is faulty. If the problem still appears, contact Dell Technical Support for further assistance.

## Power supplies, fans, and temperature

Information about power supplies, fans, and temperature readings are sent to the static buffer log. New messages replace older ones, so only the most recent message is displayed. For example, only the most recent temperature warning message is shown in the log. The static buffer is not configurable, and static buffer messages do not appear in dynamic buffer logs.

**NOTE**
Always cover empty chassis slots with the slot panels that shipped with your device. Operating the device with exposed empty slots can cause the system to overheat.

For more information about replacing power supplies and fan assemblies, see the *NetIron MLXe Series Installation and Basic Configuration Guide.*

## Power supply, fan, and temperature show commands

### show log

**Syntax:** show log

You can view power supply, fan, and temperature information using the **show log** command. Output resembles the following:

```
PowerConnect#show log
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 82 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning

Static Log Buffer:
Jun  4 09:43:01:A:System: AC Power Supply 4 , 1st from left, Installed (O
Jun  4 09:43:01:A:System: AC Power Supply 4 , 2nd from left, Installed (O

Dynamic Log Buffer (1000 lines):
Jun  4 17:25:30:I:Security: telnet login by debra from src IP 10.55.1.103 to PRI VILEGED
EXEC mode
Jun  4 17:20:54:I:Security: ssh login from src IP 10.47.6.8 to USER EXEC mode
Jun  4 17:20:34:I:Security: ssh logout from src IP 10.47.6.8 from USER EXEC mode
Jun  4 17:20:31:I:Security: ssh login from src IP 10.47.6.8 to USER EXEC mode
Jun  4 14:20:42:I:Security: telnet logout by debra from src IP 10.55.1.103 from
USER EXEC mode
Jun  4 14:05:06:I:Security: telnet login by debra from src IP 10.55.1.103 to PRI
VILEGED EXEC mode
Jun  4 12:29:14:W:    Latched low RX Power warning, port 4/1
Jun  4 12:29:14:A:    Latched low RX Power alarm, port 4/1
Jun  4 12:24:15:I:System: Interface ethernet 4/1, state up
Jun  4 12:24:14:W:    Latched low RX Power warning, port 4/1
Jun  4 12:24:14:A:    Latched low RX Power alarm, port 4/1
Jun  4 12:24:05:I:System: Interface ethernet 4/1, state down - link down
Jun  4 12:24:02:I:System: Interface ethernet 4/1, state up
Jun  4 12:23:54:I:System: Interface ethernet 4/1, state down - link down
.
```

### show fan-threshold

**Syntax:** show fan-threshold

This command displays the current settings of temperature thresholds and fan speeds, as shown in the following example:

```
PowerConnect#show fan-threshold
=== Thermal Sensor Control Block (THERMAL_SENSOR_TEST_RULE_MP) ===
Fan Speed Low: -1 - 60
Fan Speed Med: 57 - 70
Fan Speed Med-Hi: 67 - 80
Fan Speed Hi: 77 - 85
state = 0 (FAN_STATE_LOW)
max_ts_shut_off_count = 3
shut_off_count = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
=== Thermal Sensor Control Block (THERMAL_SENSOR_TEST_RULE_SNM) ===
Fan Speed Low: -1 - 30
Fan Speed Med: 27 - 40
Fan Speed Med-Hi: 37 - 50
Fan Speed Hi: 47 - 75
state = 1 (FAN_STATE_MED)
max_ts_shut_off_count = 3
shut_off_count = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
=== Thermal Sensor Control Block (THERMAL_SENSOR_TEST_RULE_LP) ===
Fan Speed Low: -1 - 50
Fan Speed Med: 46 - 55
Fan Speed Med-Hi: 51 - 60
Fan Speed Hi: 56 - 95
state = 0 (FAN_STATE_LOW)
max_ts_shut_off_count = 3
shut_off_count = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
=== Thermal Sensor Control Block (THERMAL_SENSOR_TEST_RULE_LP_XPP) ===
Fan Speed Low: -1 - 50
Fan Speed Med: 45 - 65
Fan Speed Med-Hi: 60 - 75
Fan Speed Hi: 70 - 113
state = 1 (FAN_STATE_MED)
max_ts_shut_off_count = 3
shut_off_count = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
=== Thermal Sensor Control Block (THERMAL_SENSOR_TEST_RULE_STANDBY_MP) ===
Fan Speed Low: -1 - 60
Fan Speed Med: 57 - 70
Fan Speed Med-Hi: 67 - 80
Fan Speed Hi: 77 - 85
state = 0 (FAN_STATE_LOW)
max_ts_shut_off_count = 3
shut_off_count = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
=== Thermal Sensor Control Block (THERMAL_SENSOR_TEST_RULE_MP_CPU) ===
Fan Speed Low: -1 - 60
Fan Speed Med: 57 - 70
Fan Speed Med-Hi: 67 - 80
Fan Speed Hi: 77 - 95
state = 0 (FAN_STATE_LOW)
max_ts_shut_off_count = 3
shut_off_count = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
=== Thermal Sensor Control Block (THERMAL_SENSOR_TEST_RULE_STANDBY_MP_CPU) ===
Fan Speed Low: -1 - 60
Fan Speed Med: 57 - 70
Fan Speed Med-Hi: 67 - 80
Fan Speed Hi: 77 - 95
```

**NOTE**

For optimum performance and up-time, it is recommended that you install multiple power supplies for redundancy. See the *PowerConnect B-MLXe Hardware Installation Guide* for more information.

### show chassis

**Syntax:** show chassis

The **show chassis** command also displays power supply and fan information, as well as temperature readings for the chassis:

```
PowerConnect#show chassis
*** NetIron MLXe-8 CHASSIS ***

---POWERS ---
Power 1: Installed (Failed or Disconnected)
Power 2: Installed (Failed or Disconnected)
Power 3 (30351200 - AC 1200W): Installed (OK)
Power 4 (30351200 - AC 1200W): Installed (OK)
Total power budget for chassis = 2400W
Total power budget for LPs      = 2049W
Slot Power-On Priority and Power Usage:

--- FANS ---
Right fan tray (fan 1): Status = OK, Speed = MED (75%)
Right fan tray (fan 2): Status = OK, Speed = MED (75%)
Right fan tray (fan 3): Status = OK, Speed = MED (75%)
Right fan tray (fan 4): Status = OK, Speed = MED (75%)

--- TEMPERATURE READINGS ---
Active Mgmt Module: 30.250C 44.125C
SNM1: 25.5C
SNM2: 23.0C
SNM3: 25.5C
LP2 Sensor1: 35.0C
LP2 Sensor2: 50.375C
LP3 Sensor1: 30.5C
LP3 Sensor2: 38.500C
LP4 Sensor1: 35.0C
LP4 Sensor2: 43.875C
LP4 Sensor3: UNUSED
Temperature Monitoring Poll Period is 60 seconds

--- MISC INFO ---
Backplane EEPROM MAC Address:  000c.dbe2.ca00
```

### show temperature

**Syntax:** show temperature

The **show temperature** command displays temperature readings for each interface module. The temperature is polled every 60 seconds. See the output of the show fan threshold command for the thermal_sensor_test_rule_lp_xpp reading. If this reading goes above 75 degrees, it exceeds medium-high and changes to high. It will not return to medium-high until the temperature returns below the low mark for high, which is 70 degrees.

```
PowerConnect#show temperature
SLOT #:       CARD TYPE:     SENSOR #    TEMPERATURE (C):
     17        ACTIVE MG        1            27.500C
     17        ACTIVE MG        2            48.125C
=======================================================
     1           LP            1            33.0C
     1           LP            2            47.625C
     3           LP            1            35.0C
     3           LP            2            60.250C
     5           LP            1            41.5C
     5           LP            2            57.0C
     5           LP            3            UNUSED
     5           LP            4            41.0C
     5           LP            5            57.250C
     5           LP            6            UNUSED

SNM #:      TEMPERATURE (C):
    1             28.0C
    2             23.5C
    3             31.5C
    4             22.5C
```

## Configuration notes

There are several cautions and warnings that you should pay attention to when installing or replacing power supplies. Refer to the *PowerConnect B-MLXe Configuration Guide* for more information.

## Common diagnostic scenarios

- Power supply is not providing power - check all power connections, and replace faulty power supply if necessary. Refer to the *PowerConnect B-MLXe Configuration Guide* for more information.
- Fans are not receiving power - check all power connections, and replace faulty power supply if necessary. Refer to the *PowerConnect B-MLXe Configuration Guide* for more information.
- Temperature is outside normal operating range. See the following section.

### *What to do if the temperature is outside normal operating range*

If the device detects temperatures outside the normal range, depending on the severity of the reading, it will automatically do one of the following:

- Leave the fan speed as is.
- Increase the fan speed.
- Decrease the fan speed.
- Shut down a module to prevent damage after the first warning.
- Generate a Syslog message and an SNMP trap.

If none of these measures resolves the problem, you should perform the following steps:

1. Shut down the device immediately.
2. Inspect all fans for damage or failure.
3. Inspect electrical connections to the fans.
4. Replace any component that has been damaged by excessive temperature.

The normal operating temperature, humidity, and altitude specifications for PowerConnect B-MLXe router are:

- Operating Temperature: 32° – 104° F (0° – 40° C).
- Relative Humidity: 5 to 90%, @ 104° F (40° C), non-condensing.
- Operating Altitude: 0 – 10,000 ft (0 – 3048 meters).

## Replacing the air filter and fans

The air filter should be replaced every three months and the fans every five years. For information about replacing the air filter and other fan components, see the *NetIron MLXe Series Installation and Basic Configuration Guide*.

All fan components are hot-swappable, and can be removed and replaced without powering down the system.

# Fiber optic modules

The most common problems with fiber optic modules are caused by dirty connectors. Optical cables that are contaminated in any way (with dust, hand oil, etc.), may degrade the optic eye pattern. Some of the symptoms that may be experienced are:

- Port appears to not function (either no link or unstable link)
- CRC Errors
- Port flapping
- Packet loss

Before inserting the fiber cable into the fiber optic transceiver, ensure that is it free of dust by cleaning the end.

It is very important that the end of an optical cable is clean when using any data rate. However, it is *critical* that cable ends are clean when using 10 Gigabit data rates.

This should be the first step in troubleshooting symptoms such as those stated above. ALWAYS ensure that the optical cables are cleaned.

**NOTE**
When not using a fiber optic module port connector, replace the protective cover to prevent dust or dirt from contaminating the connector.

## Fiber optic show commands

### show version

**Syntax:** show version

This command displays information about optic modules installed in your PowerConnect B-MLXe router. Optics information resembles the output segment shown here:

```
PowerConnect# show version
System Mode: MLX
Chassis: NetIron 8-slot (Serial #:       GOLD,  Part #: 35549-000C)
NI-X-SF  Switch Fabric Module 1 (Serial #: PR23050271,  Part #: 31523-100A)
FE 1: Type fe200,  Version 2
FE 3: Type fe200,  Version 2
NI-X-SF  Switch Fabric Module 2 (Serial #: SA21091164,  Part #: 35523-302A)
FE 1: Type fe200,  Version 2
FE 3: Type fe200,  Version 2
NI-X-SF  Switch Fabric Module 3 (Serial #: SA21091204,  Part #: 35523-302A)
FE 1: Type fe200,  Version 2
FE 3: Type fe200,  Version 2
======================================================================
SL M2: NI-MLX-MR Management Module Active (Serial #: SA21091472, Part #: 35524-103C):
Boot     : Version 5.1.0T165 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on Aug 11 2010 at 14:06:58 labeled as xmprm05100
 (524038 bytes) from boot flash
Monitor  : Version 5.1.0T165 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on Aug 11 2010 at 14:06:30 labeled as xmb05100
 (524053 bytes) from code flash
IronWare : Version 5.1.0T163 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on Aug 28 2010 at 04:23:16 labeled as mlxe05100b312
 (6985918 bytes) from Primary
Board ID : 00 MBRIDGE Revision : 32
916 MHz Power PC processor 7447A (version 8003/0101) 166 MHz bus
512 KB Boot Flash (AM29LV040B), 32 MB Code Flash (MT28F128J3)
1024 MB DRAM
Active Management uptime is 6 minutes 21 seconds
======================================================================
SL M1: NI-MLX-MR Management Module Standby (Serial #: SA21091421, Part #: 35524-103C):
Boot     : Version 5.1.0T165 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on Aug 11 2010 at 14:06:58 labeled as xmprm05100
 (524038 bytes) from boot flash
Monitor  : Version 5.1.0T165 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on Aug 11 2010 at 14:06:30 labeled as xmb05100
 (524053 bytes) from code flash
IronWare : Version 5.1.0T163 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on Aug 28 2010 at 04:23:16 labeled as mlxe05100b312
 (6985918 bytes) from Primary
Board ID : 00 MBRIDGE Revision : 32
916 MHz Power PC processor 7447A (version 8003/0101) 166 MHz bus
512 KB Boot Flash (AM29LV040B), 32 MB Code Flash (MT28F128J3)
1024 MB DRAM
Standby Management uptime is 5 minutes 33 seconds
======================================================================
```

```
==========================================================================
SL 1: BR-MLX-1GCx24-X 24-port 10/100/1000Base-T Copper Module (Serial #:   Not Exist, Part #:
Not Exist)
License: MLX (LID: ÿÿÿÿÿÿÿÿÿÿÿ)
Boot     : Version 5.1.0T175 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on Aug 11 2010 at 14:07:20 labeled as xmlprm05100
 (492544 bytes) from boot flash
Monitor  : Version 5.1.0T175 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on Aug 11 2010 at 14:07:42 labeled as xmlb05100
 (493244 bytes) from code flash
IronWare : Version 5.1.0T177 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on Aug 28 2010 at 04:28:44 labeled as xmlp05100b312
 (4949977 bytes) from Primary
FPGA versions:
Valid PBIF Version = 3.24, Build Time = 8/4/2010 14:57:00

Valid XPP Version = 6.03, Build Time = 2/18/2010 16:38:00

Valid STATS Version = 0.08, Build Time = 2/18/2010 16:30:00

BCM56512GMAC 0
BCM56512GMAC 1
666 MHz MPC 8541 (version 8020/0020) 333 MHz bus
512 KB Boot Flash (MX29LV040C), 16 MB Code Flash (MT28F128J3)
1024 MB DRAM, 8 KB SRAM, 286331153 Bytes BRAM
PPCR0: 1024K entries CAM, 16384K PRAM, 2048K AGE RAM
LP Slot 1 uptime is 5 minutes 36 seconds
==========================================================================
SL 3: BR-MLX-10Gx4-X 4-port 10GbE Module (Serial #: BMY0319F00J, Part #: 60-1001875-07)
License: MLX (LID: doaFIGOhFFl)
Boot     : Version 5.1.0T175 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on Aug 11 2010 at 14:07:20 labeled as xmlprm05100
 (492544 bytes) from boot flash
Monitor  : Version 5.1.0T175 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on Aug 11 2010 at 14:07:42 labeled as xmlb05100
 (493244 bytes) from code flash
IronWare : Version 5.1.0T177 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on Aug 28 2010 at 04:28:44 labeled as xmlp05100b312
 (4949977 bytes) from Primary
FPGA versions:
Valid PBIF Version = 3.22, Build Time = 2/5/2010 14:43:00

Valid XPP Version = 6.04, Build Time = 2/3/2010 14:39:00

Valid XGMAC Version = 0.13, Build Time = 2/3/2010 14:42:00

X10G2MAC 0
X10G2MAC 1
666 MHz MPC 8541 (version 8020/0020) 333 MHz bus
512 KB Boot Flash (MX29LV040C), 16 MB Code Flash (MT28F128J3)
1024 MB DRAM, 8 KB SRAM, 286331153 Bytes BRAM
PPCR0: 1024K entries CAM, 16384K PRAM, 2048K AGE RAM
PPCR1: 1024K entries CAM, 16384K PRAM, 2048K AGE RAM
LP Slot 3 uptime is 5 minutes 38 seconds
==========================================================================
```

```
SL 4: NI-MLX-1Gx48-T 48-port 10/100/1000Base-T MRJ21 Module (Serial #: SA05091472, Part #:
35663-20EA)
Boot     : Version 5.1.0T175 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on Aug 11 2010 at 14:07:20 labeled as xmlprm05100
 (492544 bytes) from boot flash
Monitor  : Version 5.1.0T175 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on Aug 11 2010 at 14:07:42 labeled as xmlb05100
 (493244 bytes) from code flash
IronWare : Version 5.1.0T177 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on Aug 28 2010 at 04:28:44 labeled as xmlp05100b312
 (4949977 bytes) from Primary
FPGA versions:
Valid PBIF Version = 3.24, Build Time = 8/4/2010 14:57:00

Valid XPP Version = 6.03, Build Time = 2/18/2010 16:38:00

Valid STATS Version = 0.08, Build Time = 2/18/2010 16:30:00

BCM56502GMAC 0
BCM56502GMAC 1
666 MHz MPC 8541 (version 8020/0020) 333 MHz bus
512 KB Boot Flash (AM29LV040B), 16 MB Code Flash (MT28F640J3)
1024 MB DRAM, 8 KB SRAM, 0 Bytes BRAM
PPCR0: 768K entries CAM, 8192K PRAM, 2048K AGE RAM
PPCR1: 768K entries CAM, 8192K PRAM, 2048K AGE RAM
LP Slot 4 uptime is 5 minutes 42 seconds
================================================================================
SL 6: NI-MLX-10Gx4 4-port 10GbE Module (Serial #: SA12090950, Part #: 35600-202D)
Boot     : Version 5.1.0T175 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on Aug 11 2010 at 14:07:20 labeled as xmlprm05100
 (492544 bytes) from boot flash
Monitor  : Version 5.1.0T175 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on Aug 11 2010 at 14:07:42 labeled as xmlb05100
 (493244 bytes) from code flash
IronWare : Version 5.1.0T177 Copyright (c) 1996-2009 Brocade Communications, Inc.
Compiled on Aug 28 2010 at 04:28:44 labeled as xmlp05100b312
 (4949977 bytes) from Primary
FPGA versions:
Valid PBIF Version = 3.22, Build Time = 2/5/2010 14:43:00

Valid XPP Version = 6.04, Build Time = 2/3/2010 14:39:00

Valid XGMAC Version = 0.13, Build Time = 2/3/2010 14:42:00

X10G2MAC 0
X10G2MAC 1
666 MHz MPC 8541 (version 8020/0020) 333 MHz bus
512 KB Boot Flash (AM29LV040B), 16 MB Code Flash (MT28F640J3)
512 MB DRAM, 8 KB SRAM, 286331153 Bytes BRAM
PPCR0: 768K entries CAM, 8192K PRAM, 2048K AGE RAM
PPCR1: 768K entries CAM, 8192K PRAM, 2048K AGE RAM
LP Slot 6 uptime is 5 minutes 44 seconds
================================================================================
All show version done
```

## Fiber optic debug commands

There are no fiber-specific **debug** commands.

## Configuration notes

Before installing or removing fiber optic modules, refer to the precautions and follow the instructions in the *NetIron MLXe Series Installation and Basic Configuration Guide*.

# Testing network connectivity

After you cable the fiber optic modules, you can test connectivity to other network devices by pinging those devices. You also can trace routes.

## Pinging an IP address

To verify that a PowerConnect B-MLXe Series router can reach another device through the network, enter a command such as the following at any level of the CLI on the PowerConnect B-MLXe Series router:

```
PowerConnect>ping 192.33.4.7
```

**Syntax:** **ping** <*ip addr*> | <*hostname*> [**source** <*ip addr*>] [**count** <*num*>] [**timeout** <*msec*>] [**ttl** <*num*>] [**size** <*byte*>] [**quiet**] [**numeric**] [**no-fragment**] [*verify*] [**data** <*1-to-4 byte hex*>] [**brief**]

**NOTE**
If you address the ping to the IP broadcast address, the device lists the first four responses to the ping.

## Tracing a route

To determine the path through which the router can reach another network device, enter a command similar to the following at any level of the CLI on the PowerConnect B-MLXe Series router:

```
PowerConnect>traceroute 192.33.4.7
```

**Syntax:** **traceroute** <*host-ip-addr*> [**maxttl** <*value*>] [**minttl** <*value*>] [**numeric**] [**timeout** <*value*>] [**source-ip** <*ip addr*>]

The CLI displays trace route information for each hop as soon as the information is received. Traceroute requests display all responses to a given TTL. In addition, if there are multiple equal-cost routes to the destination, the Dell device displays up to three responses by default.

Chapter

# Layer 1 Diagnostics 3

This chapter describes common Layer 1 diagnostic procedures for PowerConnect B-MLXe routers. In general, Layer 1 issues are related to hardware, the most common being physical connectivity problems, including:

- Faulty ports
- Faulty cables
- Faulty hardware
- Input and output errors
- Cyclic redundancy check (CRC) errors
- Excessive or late collisions
- Overruns
- Output buffer failures

## Ethernet diagnostics

The following sections describe how to troubleshoot Layer 1 issues for Ethernet interfaces.

### Ethernet autonegotiation

10BaseT, 100BaseTX, and 1000BaseT all use an RJ-45 connector, which creates the potential for connecting electrically-incompatible components to each other, and which can cause network disruption. The IEEE® developed autonegotiation to eliminate the possibility of dissimilar technologies interfering with each other.

Autonegotiation uses electrical pulses generated by a device over a 10 Mbps, 100 Mbps, or 1000 Mbps link (when the link is not sending or receiving data) to detect the presence of a connection to another device. These unipolar, positive-only pulses have a duration of 100 ns, and are generated in trains of a maximum of 33 pulses. These pulse trains are referred to as fast link pulse (FLP) bursts. The time interval between the start of each burst is 16 ms, with a tolerance of 8 ms.

#### *Autonegotiation and 10-BaseT*

An FLP burst is not recognized as valid by a 10BASE-T device receiving it from an autonegotiation device. The 10BASE-T device will detect a failure of the link. To avoid this, when the autonegotiation device receives the 10BASE-T pulses, it automatically switches to 10BASE-T half-duplex mode. If the 10BASE-T device is operating in full-duplex mode, a duplex mismatch can occur.

##### Duplex mismatches

A duplex mismatch can occur between devices when:

- One device is manually set to half duplex and one device is manually set to full duplex

- One device is set to autonegotiation and one device is manually set to full duplex

Duplex mismatches are difficult to diagnose because the network still appears to be working. Simple tests, such as **ping,** report a valid connection even though network performance can be much slower than normal.

When one device operates in full duplex while the other one operates in half duplex, the connection works at a very low speed if both devices attempt to send frames at the same time. This is because a full duplex device may transmit data while it is receiving, but if the other device is working in half duplex, it cannot receive data while it is sending.) The half-duplex device senses a collision and attempts to resend the frame it was sending. Depending on timing, the half duplex device may sense a late collision, which it will interpret as a hard error and will not attempt to resend the frame. At the other end, the full duplex device does not detect a collision and does not resend the frame, even if the half-duplex device has already discarded it as corrupted by collision.

This packet loss happens when both devices are transmitting at the same time, and may happen even when the link is used, from the user's perspective, in one direction only. A TCP stream requires that all packets sent be acknowledged by the receiving device, even if actual data is sent in one direction only. Packet collisions may occur with acknowledgement packets traveling in the other direction.

Because the full duplex device does not expect incoming frames to be truncated by collision detection, the device reports Frame Check Sequence (FCS) errors. The combination of late collisions reported at the half-duplex end, and FCS errors reported by the full duplex end, can indicate a duplex mismatch.

#### Priority

Upon receipt of the technology abilities of the other device, both devices decide the best possible mode of operation supported (each device chooses the mode that is topmost on this list. The priority among modes specified in the 2002 edition of 802.3 is as follows:

- 1000BASE-T full duplex
- 1000BASE-T half duplex
- 100BASE-T2 full duplex
- 100BASE-TX full duplex
- 100BASE-T2 half duplex
- 100BASE-T4
- 100BASE-TX half duplex
- 10BASE-T full duplex
- 10BASE-T half duplex

Currently, all network equipment manufacturers recommend using autonegotiation on all access ports. On rare occasions where autonegotiation may fail, you may still need to force settings.

## *Unidirectional Link Detection*

Unidirectional Link Detection (UDLD) monitors a link between two Dell devices and brings the ports on each side of the link down if the link fails at any point between the two devices. This feature is useful for links that are individual ports and for trunk links.

Ports with UDLD enabled exchange proprietary health-check packets once every second (the keepalive interval). If a port does not receive a health-check packet from the port at the other end of the link within the keepalive interval, the port waits for two more intervals. If the port still does not receive a health-check packet after waiting for three intervals, the port concludes that the link has failed and takes the port down.

**NOTE**
UDLD is supported only on Ethernet ports.

### *Configuring a UDLD holddown threshold*

You can configure a UDLD holddown threshold to prevent network flapping that affects ring topologies. If a port fails after a specified number of times, UDLD brings the port down. A port is considered to have failed if it consistently loses UDLD packets during the specified duration. The port would then be considered as a Blocked port and UDLD considers the port state as "down". A Syslog message is generated to indicate a UDLD holddown is in effect on that port. The port remains disabled until it is manually restored. To display UDLD holddown information, see "show link-keepalive ethernet" on page 58.

## Ethernet show commands

This section describes the show commands that display information about Ethernet interfaces and UDLD.

#### show interface brief

**Syntax: show interface brief**

This command displays a summary of the information provided in the **show interface** command. Output resembles the following:

```
PowerConnect#show interface brief
Port Link State Dupl Speed Trunk Tag Priori MAC Name
1/1 Up Lk-disable None None None No level0 00e0.52a9.bb00
1/2 Down None None None None No level0 00e0.52a9.bb01
1/3 Down None None None None No level0 00e0.52a9.bb02
1/4 Down None None None None No level0 00e0.52a9.bb03
```

### show interface ethernet

**Syntax: show interface ethernet** <*slotnum* | *portnum*>

This command displays information about a specific Ethernet interface, as shown in the following example:

```
PowerConnect#show interface ethernet 1/1
GigabitEthernet2/1 is disabled, line protocol is down, link keepalive is enabled
Hardware is GigabitEthernet, address is 000c.dbe2.5900 (bia 000c.dbe2.5900)
Configured speed 1Gbit, actual unknown, configured duplex fdx, actual unknown
Configured mdi mode AUTO, actual unknown
Member of 2 L2 VLANs, port is tagged, port state is Disabled
STP configured to ON, Priority is level7, flow control enabled
Force-DSCP disabled
mirror disabled, monitor disabled
Not member of any active trunks
Not member of any configured trunks
No port name
MTU 1522 bytes, encapsulation ethernet
300 second input rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
300 second output rate: 0 bits/sec, 0 packets/sec, 0.00% utilization
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 multicasts, 0 unicasts
0 input errors, 0 CRC, 0 frame, 0 ignored
0 runts, 0 giants, DMA received 0 packets
0 packets output, 0 bytes, 0 underruns
Transmitted 0 broadcasts, 0 multicasts, 0 unicasts
0 output errors, 0 collisions, DMA transmitted 0 packets
```

### show interface status

**Syntax:** show interface status

This command displays the status of a single interface. Output resembles the following:

```
PowerConnect#show interface status ethernet 1/1
Information of Eth 1/1
Basic information:
Port type: 100TX
Mac address: 00-01-F4-88-F5-21
Configuration:
Name:
Port admin: Up
Speed-duplex: Auto
Capabilities: 10half, 10full, 100half, 100full
Broadcast storm: Enabled
Broadcast storm limit: 500 packets/second
Flow control: Disabled
LACP: Disabled
Port security: Disabled
Max MAC count: 0
Port security action: None
Current status:
Link status: Down
Operation speed-duplex: 100full
flow control type: none
```

### show interfaces counters

**Syntax: show interfaces counters** [**detailed** | **brief** <*ethernet id*>]

This command displays either detailed or brief statistics for an interface. The following example is a result of the **detailed** option for Ethernet port 1/7:

```
PowerConnect#show interfaces counters detailed ethernet 1/7
Ethernet 1/ 7
Iftable stats:
Octets input: 30658, Octets output: 196550
Unicast input: 6, Unicast output: 5
Discard input: 0, Discard output: 0
Error input: 0, Error output: 0
Unknown protos input: 0, QLen output: 0
Extended iftable stats:
Multi-cast input: 0, Multi-cast output: 3064
Broadcast input: 262, Broadcast output: 1
Ether-like stats:
Alignment errors: 0, FCS errors: 0
Single Collision frames: 0, Multiple collision frames: 0
SQE Test errors: 0, Deferred transmissions: 0
Late collisions: 0, Excessive collisions: 0
Internal mac transmit errors: 0, Internal mac receive errors: 0
Frame too longs: 0, Carrier sense errors: 0
Symbol errors: 0
RMON stats:
Drop events: 0, Octets: 227208, Packets: 3338
Broadcast pkts: 263, Multi-cast pkts: 3064
Undersize pkts: 0, Oversize pkts: 0
Fragments: 0, Jabbers: 0
CRC align errors: 0, Collisions: 0
Packet size <= 64 octets: 3150, Packet size 65 to 127 octets: 139
Packet size 128 to 255 octets: 49, Packet size 256 to 511 octets: 0
Packet size 512 to 1023 octets: 0, Packet size 1024 to 1518 octets: 01023 octets: 0, Packet
size 1024 to 1518 octets: 0
```

The following example is a result of output from the **brief** option:

```
PowerConnect#show interfaces counters brief
Ethernet Packets Collision Errors
Port [Receive Transmit] [Receive] [Input Output]
1/ 1 0 0 0 0 0
1/ 2 0 0 0 0 0
1/ 3 0 0 0 0 0
1/ 4 0 0 0 0 0
1/ 5 0 0 0 0 0
1/ 6 0 0 0 0 0
1/ 7 0 0 0 0 0
1/ 8 0 0 0 0 0
1/ 9 0 0 0 0 0
1/10 0 0 0 0 0
1/11 0 0 0 0 0
1/12 0 0 0 0 0
1/13 0 0 0 0 0
1/14 0 0 0 0 0
1/15 0 0 0 0 0
1/16 0 0 0 0 0
1/17 0 0 0 0 0
1/18 0 0 0 0 0
1/19 0 0 0 0 0
1/20 0 0 0 0 0
1/21 0 0 0 0 0
1/22 0 0 0 0 0
1/23 27 819 0 3 0
1/24 0 0 0 0 0
```

### show link-keepalive ethernet

**Syntax: show link-keepalive ethernet**

See the following command description and example.

### show udld etherlink

**Syntax: show udld etherlink**

To display UDLD information for all Ethernet ports, enter either the **show link-keepalive ethernet** command or the **show udld etherlink** command. For example:

```
PowerConnect#show link-keepalive ethernet
Total link-keepalive enabled ports: 4
Keepalive Retries: 3 Keepalive Interval: 1 Sec.
Port Physical Link Logical Link State
4/1 up up FORWARDING
4/2 up up FORWARDING
4/3 down down DISABLED
4/4 up down DISABLED
```

## Ethernet interface debug commands

There are no specific Ethernet interface **debug** commands.

## Common diagnostic scenarios

The following issues can occur with Ethernet interfaces:

- Faulty hardware

  Whenever you encounter a connection problem, check for faulty hardware. Replace cables, try another port, and check all cable connections. If you find a faulty port, contact Dell Technical Support for assistance.

- Link failures

  Link failures can be due either to a failure of the transmission medium, or of the devices at each end of a connection. Be sure to check all of the hardware involved in the link, including cables and ports.

- CSMA /CD

  The Carrier Sense Multiple Access (CSMA) with Collision Detection (CD) protocol controls access to shared Ethernet media. A switched network (e.g. Fast Ethernet) may use a full duplex mode with access to the full link speed between directly connected NICs, switch-to-NIC cables, or switch-to-switch cables.

- CRC errors

  The Cyclic Redundancy Check (CRC) length specifies whether the CRC portion of each frame transmitted on the interface is 16 bits or 32 bits long. The default is 32 bits.

  A CRC alignment error is generated when the total number of packets received is from 64 – 1518 octets, but contained either a bad FCS with an integral number of octets (FCS Error) or a bad FCS with a non-integral number of octets (Alignment Error).

- Run

  Any received packet that is less than 64 bytes is illegal, and is called a runt. In most cases, runts arise from a collision, and although they indicate an illegal reception, they may occur on correctly functioning networks. The receiving Dell device discards all runt frames.

- Giants

  Any received packet that is greater than the maximum frame size is called a giant. In theory, the jabber control circuit in the transceiver should prevent any node from generating such a frame, but certain failures in the physical layer may also give rise to oversized Ethernet frames. Like runts, giants are discarded by the receiving Dell device.

- Misaligned frames

  Any frame which does not contain an integral number of received octets (bytes) is also illegal. A receiver has no way of knowing which bits are legal, and how to compute the CRC-32 of the frame. Such frames are therefore also discarded by the Ethernet receiver.

- Old software versions.

  Feature issues are often caused because the device is running an old version of the software. Dell recommends that you always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# Link fault signaling

Link fault signaling (LFS) is a physical layer protocol that enables communication on a link between two 10 Gigabit Ethernet devices. When LFS is configured on a 10 Gigabit Ethernet port, the port can detect and report fault conditions on transmit and receive ports. PowerConnect B-MLXe IronWare software supports LFS among all 10 Gigabit Ethernet devices, including LFS support between First and Second generation devices. LFS is disabled by default on 10 Gbps interfaces.

When LFS is enabled on an interface, the following Syslog messages are generated whenever the interface goes up or down, or when the TX or RX fiber is removed from both sides of a link where LFS is configured.

```
interface ethernet1/1, state down - link down
interface ethernet1/1, state up
```

After the fiber is installed, the Link and Activity LEDs light up when traffic is flowing across the link.

**NOTE**
LFS is *disabled* by default on 10 Gbps interfaces. It is *enabled* by default (and cannot be disabled) on 1 Gpbs interfaces.

## LFS show commands

### show link-fault-signaling

**Syntax: show link-fault-signaling**

This command displays the LFS state configured on ports.

```
PowerConnect#show
link-fault-signaling
Global Remote Fault : OFF
PORT #: REMOTE FAULT:
PORT 1/1: OFF
PORT 1/2: OFF
PORT 1/3: OFF
PORT 1/4: ON
PORT 1/5: OFF
PORT 1/6: OFF
PORT 1/7: OFF
PORT 1/8: OFF
PORT 1/9: OFF
```

## LFS debug commands

There are no specific link fault signaling **debug** commands.

## Remote fault notification

For fiber optic connections, you can optionally configure a transmit port to notify the receive port on the remote device whenever the transmit port becomes disabled. When you enable the Remote Fault Notification (RFN) feature, the transmit port notifies the remote port whenever the fiber cable is either physically disconnected or has failed. When this occurs and the feature is enabled, the device disables the link and turns OFF both LEDs associated with the ports.

By default, RFN is disabled. In this case, if the transmit port becomes physically disabled or fails, the link still appears as though it is enabled and the LEDs for both ports remain ON.

Link fault signaling

Chapter

# Layer 2 Protocol Diagnostics

4

This chapter describes Layer 2 troubleshooting and diagnostic processes for PowerConnect B-MLXe routers.

**CAUTION**

**Enabling diagnostic commands may degrade system performance. These commands are best used to troubleshoot specific problems while working with qualified Dell service technicians. Whenever possible, troubleshoot your system during periods of low network traffic and user activity to preserve system performance.**

## MAC address learning

In MAC address learning, the source MAC address of each received packet is stored so that future packets destined for that address can be forwarded only to the interface where that address is located. (Packets destined for unrecognized addresses are forwarded out every bridge interface.) MAC address learning, defined in IEEE 802.1 standard, helps minimize traffic on the attached LANs.

### Address Resolution Protocol

Routers use Address Resolution Protocol (ARP) to learn the MAC addresses of devices on the network. The router sends an ARP request that contains the IP address of a device, and receives the MAC address for that device in an ARP reply. These *dynamically* learned entries are stored in the ARP cache. You can also manually configure MAC addresses, which are called *static* entries.

A *static* ARP entry in the ARP cache resembles the following:

```
Index   IP Address          MAC Address     Port
  1     207.95.6.111        0800.093b.d210  1/1
```

A *dynamic* entry in the ARP cache resembles the following:

```
IP Address     MAC Address     Type                    Age   Port
1              207.95.6.102    0800.5afc.ea21 Dynamic   0     6
ARP age
```

The ARP age is the amount of time the device keeps a learned MAC address in the ARP cache. The device resets the timer to zero each time the ARP entry is refreshed and removes the entry if the timer reaches the ARP age. The ARP age default is 10 minutes.

### *Changing the ARP aging period*

When the switch places an entry in the ARP cache, it also starts an aging timer for the entry. The aging timer ensures that the ARP cache does not retain learned entries that are no longer valid. An entry can become invalid when the device with the MAC address of the entry is no longer on the network.

ARP age affects dynamic entries only. The default ARP age is 10 minutes. You cannot change the ARP age on Layer 2 switches. If you set the ARP age to zero, aging is disabled and entries do not age out.

### *Proxy ARP*

Routers uses Proxy ARP to answer ARP requests for a host, by replying with the router MAC address instead of the host address.

## MAC address learning show commands

The following commands display information about the MAC address table.

**show mac-address**

**Syntax: show mac-address**

This command displays the MAC address table, which contains MAC addresses learned from other devices, or added using the **static-mac-address** command. This table does not contain the MAC addresses of the Dell device ports.

```
PowerConnect#show mac-address
Total entries from all ports = 75
MAC Port Age CamF CIDX0 CIDX1 CIDX2 CIDX3 CIDX4 CIDX5
0000.0300.0000 10 17293 00H 0 0 0 0 0 0
0060.089f.8086 1 12 0bH 23 15 0 6 0 0
0060.9709.914b 16 2130 00H 0 0 0 0 0 0
00a0.249a.0163 16 130 00H 0 0 0 0 0 0
0060.979d.41a5 11 475 00H 0 0 0 0 0 0
00a0.24c5.01d1 11 0 0cH 0 0 20 14 0 0
0060.979d.41df 11 570 00H 0 0 0 0 0 0
0060.9759.4226 16 240 00H 0 0 0 0 0 0
0060.9759.4235 16 130 00H 0 0 0 0 0 0
0800.208f.725b 2 135 00H 0 0 0 0 0 0
0060.9759.4264 16 0 0aH 0 14 0 21 0 0
00a0.24c5.02a1 16 15 09H 5 0 0 33 0 0
....
```

**NOTE**

The information displayed in columns with headings, CamF, and CIDX0 through CIDX5, is not relevant for day-to-day management of the device. This information is used by engineering and technical support staff for debug purposes. Contact Dell Technical support for more information.

### show mac vpls

**Syntax: show mac vpls** <*mac-address*> <*vpls-id*> <*starting-entry*> <*number-of-entries*>

- <*vpls-id*> - Specifies the VPLS ID for which database entries are displayed.
- <*starting-entry*> - Specifies the point in the database from which the entries are displayed. Entering 0 causes entries to be displayed from the start; entering 200 causes the first 200 entries to be skipped; and so on.
- <*number-of-entries*> - Specifies the number of database entries to be displayed from the <*starting-entry*>.

This command displays the contents of the VPLS MAC database, which stores entries associating MAC addresses with VC LSPs. This example displays the VPLS MAC database on the management processor:

```
PowerConnect#show mac vpls
Total VPLS mac entries in the table: 10 (Local: 5, Remote: 5)
VPLS MAC Address L/R Port Vlan/Peer Age
==== =========== === ==== ========= ===
1 0016.0100.1601 R 5/1 3.3.3.3 0
1 0010.0100.1003 L 5/3 2 0
1 0016.0100.1603 R 5/1 3.3.3.3 0
1 0010.0100.1005 L 5/3 2 0
1 0010.0100.1002 L 5/3 2 0
1 0016.0100.1605 R 5/1 3.3.3.3 0
1 0016.0100.1602 R 5/1 3.3.3.3 0
1 0010.0100.1004 L 5/3 2 0
1 0010.0100.1001 L 5/3 2 0
1 0016.0100.1604 R 5/1 3.3.3.3
```

To display a specific entry in the VPLS MAC database on the management processor, enter the following command.

```
PowerConnect#show mac vpls 1 0016.0100.1601
VPLS: 1 MAC: 0016.0100.1601 Age: 0
Remote MAC Port: ethe 5/1 Peer: 3.3.3.3
Trunk slot mask: 00000000
```

## MAC address learning debug commands

### debug ip arp

**Syntax:** [**no**] **debug ip arp** [**event** | **ipc** | **itc** | **packet**]

- **event** - Displays information about ARP events.
- **ipc** - Displays information about ARP IPC messages.
- **itc** - Displays information about ARP ITC messages.
- **packet** - Displays information about ARP packets.

This command displays information about Address Resolution Protocol (ARP) transactions.The **debug ip arp** command enables debugging for all ARP variables, or you can enable the options individually:

### debug ip arp event

**Syntax:** [no] **debug ip arp event**

This command displays information about ARP events, and is useful in determining whether the router is sending and receiving ARP requests. Output is similar to the following, which shows send and receive activity for ARP packets:

```
PowerConnect#debug ip arp event
IP/ARP: sent request for 206.223.143.22
IP/ARP: sent packet src 206.223.143.16 000cdbe2b000: dst 206.223.143.22
000000000000: Port 1062
IP/ARP: sent request for 206.223.143.3
IP/ARP: sent packet src 206.223.143.16 000cdbe2b000: dst 206.223.143.3
000000000000: Port 1062
IP/ARP: sent request for 69.28.144.206
IP/ARP: sent packet src 69.28.144.205 000cdbe2b000: dst 69.28.144.206
000000000000: Port 843
IP/ARP: Received arp request from Lp for dest 69.28.144.206 Port: 843 Router: 1
IP/ARP: sent request for 69.28.181.171
IP/ARP: sent packet src 69.28.181.161 000cdbe2b000: dst 69.28.181.171
000000000000: Port 753
```

### debug ip arp ipc

**Syntax:** [no] **debug ip arp ipc**

This command generates information about ARP interprocess communication (IPC) activity. Output resembles the following:

```
PowerConnect#debug ip arp ipc
IP/ARP: Received arp request from Lp for dest 68.142.108.94 Port: 1049 Router: 1
IP/ARP: Received arp request from Lp for dest 68.142.106.82 Port: 848 Router: 1
IP/ARP: Received arp request from Lp for dest 68.142.108.94 Port: 1049 Router: 1
IP/ARP: Received arp request from Lp for dest 68.142.106.82 Port: 848 Router: 1
IP/ARP: Received arp request from Lp for dest 68.142.108.94 Port: 1049 Router: 1
IP/ARP: Received arp request from Lp for dest 69.28.144.206 Port: 843 Router: 1
IP/ARP: Received arp request from Lp for dest 68.142.78.18 Port:846 Router: 1
IP/ARP: Received arp request from Lp for dest 69.28.144.206Port: 843 Router: 1
IP/ARP: Received arp request from Lp for dest 68.142.78.18 Port: 846 Router: 1
```

### debug ip arp itc

**Syntax:** [no] **debug ip arp itc**

This command generates information about ARP inter-task communications (ITC) activity, which is communications between processing tasks concerning activity or configuration changes. Output resembles the following, which indicates that a static IP address was added.

```
PowerConnect#debug ip arp itc
ARP: itc debugging is on
PowerConnect#config t
PowerConnect(config)#arp 19 6.1.1.19 1001.2001.3019 ethernet 1/2
IP/ARP: Add static arp for Addr: 6.1.1.19 Mac: 100120013019 Port: 1
Vrf_index: 0 Add: 1
```

### debug ip arp packet

**Syntax:** [no] **debug ip arp packet**

This command displays information about ARP packet activity. Output resembles the following, which indicates that the source router is polling routers 68.142.106.98, 69.28.181.122, 206.223.143.27 and 68.142.108.94 to learn their MAC addresses and add them to the source router ARP table:

```
PowerConnect#debug ip arp packet
IP/ARP: sent request for 68.142.106.98
IP/ARP: sent packet src 68.142.106.97 000cdbe2b000: dst 68.142.106.98
000000000000: Port 114
IP/ARP: sent request for 69.28.181.122
IP/ARP: sent packet src 69.28.181.121 000cdbe2b000: dst 69.28.181.122
000000000000: Port 1045
IP/ARP: sent request for 69.28.181.172
IP/ARP: sent packet src 69.28.181.161 000cdbe2b000: dst 69.28.181.172
000000000000: Port 753
IP/ARP: sent request for 206.223.143.27
IP/ARP: sent packet src 206.223.143.16 000cdbe2b000: dst 206.223.143.27
000000000000: Port 1062
IP/ARP: sent request for 68.142.106.82
IP/ARP: sent packet src 68.142.106.81 000cdbe2b000: dst 68.142.106.82
000000000000: Port 848
IP/ARP: sent request for 68.142.108.94
```

### debug mac

**Syntax:** [no] **debug mac** [**action** | **cam** | **error** | **info** | **learning** | **mport** | **port security**]

- **action** - Displays information about the MAC database.
- **cam** - Displays Layer 2 CAM settings.
- **error** - Displays MAC table management error messages.
- **info** - Displays MAC table management information.
- **learning** - Displays MAC database learning information.
- **mport** - Displays Multiport MAC event messages.
- **port security** - Displays MAC table management port security messages.

The **debug mac** commands generate information about MAC address databases, actions, settings, MAC table management, and MAC learning.

### debug mac action

**Syntax:** [no] **debug mac action**

This command displays actions related to the MAC database. Output resembles the following:

```
PowerConnect#debug mac action
info - mac_static_flush() -  execution
info - mac_pms_flush() - execution
info - mac_static_flush() -  execution
MAC ACTION - Normal SPECIFIC FLUSH
Ports: ethe 2/2 to 2/3
Vlans: 1
MAC ACTION - Premature ALL_SYSTEM FLUSH
MAC ACTION - Normal SPECIFIC FLUSH
Ports: All Ports
Vlans: All VLANs
```

### debug mac error

**Syntax:** [no] **debug mac error**

This command displays major errors related to MAC learning. Output resembles the following, which indicates that there is not enough buffer space, which results in a queue overflow.

```
PowerConnect#debug mac error
error - macmgr_ipc_distribute_table. system out of ipc buffers info -
macmgr_ipc_learn_sa_entry. Not learning. port not in forwarding state info -
macmgr_ipc_free_sa_entry. Queue overflowing error - macmgr_ipc_sync_change_age.
system out of buffer
```

### debug mac info

**Syntax:** [no] **debug mac info**

This command generates information about MAC database activity, such as flushing and table distribution. Output resembles the following:

```
PowerConnect#debug mac info
info - macmgr_ipc_flush_entry. Send flush.
info - macmgr_ipc_distribute_table. Distributing. mac table to slot 4
```

### debug mac learning

**Syntax:** [no] **debug mac learning**

This command generates information about MAC address table learning. Output resembles the following:

**CAUTION**

**The debug mac learning command may generate large amounts of output and degrade system performance. Use this command with caution.**

```
PowerConnect#debug mac learning
learning: debugging is on
info - macmgr_ipc_learn_da_entry. Learn DA 00e0.5200.0000 IPC received
info - macmgr_ipc_learn_sa_entry. Learn SA IPC received Mar 20 23:58:30
mac address 00000200 00000001. Port 42 vlan 100 Mar 20 23:58:30 info -
macmgr_ipc_sync_add_entry. Send add entry for0000020000000001 port 42 vlan 512.
info - macmgr_ipc_learn_da_entry. Learn DA 0034.5678.9123 IPC received
info - macmgr_ipc_learn_da_entry. Learn DA 0034.5678.9123 IPC received
```

### debug ip icmp

**Syntax:** [no] **debug ip icmp** [**events** | **external_loop** | **internal_loop** | **packets** | **port_loop**]

- **events** - Displays information about ICMP events.
- **external_loop** - Displays ICMP external loop activity.
- **internal_loop** - Displays ICMP internal loop activity.
- **packets** - Displays information about ICMP packets.
- **port_loop** - Displays port loop activity.

The **debug ip icmp** commands display information about Internal Control Message Protocol (ICMP) transactions. These commands are useful in determining if a router is sending or receiving ICMP messages, and for troubleshooting end-to-end connections.

### debug ip icmp events

**Syntax:** [no] **debug ip icmp events**

This command generates information about ICMP events such as sent and received echo (ping) requests, destination-unreachable messages, and redirect messages. Output resembles the following:

```
PowerConnect#debug ip icmp events
ICMP: rcvd echo request packet of length 40 from 1.1.1.2
ICMP: send echo request packet of length 60 to 1.1.1.2
```

### debug ip icmp packets

**Syntax:** [no] **debug ip icmp packets**

This command generates information about ICMP packets. Output resembles the following:

```
PowerConnect#debug ip icmp packets
ICMP:dst (1.2.3.4), src (0.0.0.0) echo request type
ICMP: Received message from 10.102.50.254 to 10.47.16.33 port mgmt1 type 11 size
36
ICMP: rxed error message from 10.102.50.254:May 23 16:11:32  original destination
10.47.16.33 ICMP Time Exceeded
IP/ICMP: rxed message: size: 36
ICMP: Received message from 67.98.68.129 to 10.47.16.33 port mgmt1 type 11 size
36
ICMP: rxed error message from 67.98.68.129:May 23 16:11:33 original destination
10.47.16.33
ICMP Time Exceeded
```

**debug ipv6 icmp**

**Syntax:** [no] **debug ipv6 icmp**

This command generates information about Internet Control Message Protocol (ICMP) activity, such as sending or receiving ICMP requests, responses, ICMP error messages, and redirected ICMP packets. Output resembles the following:

```
PowerConnect# debug ipv6 icmp
ICMPv6: Sending Echo Request to 3000:1::6, length 24
ICMPv6: Received Echo Request from 3000:1::6, length 24
```

## Configuration notes

- Enabling port-priority changes the source MAC address of all ARP packets to a virtual MAC address.
- The location of the **static-mac-address** command in the CLI depends on whether you configure port-based VLANs on the device. If the device does not have more than one port-based VLAN (VLAN 1, which is the default VLAN that contains all the ports), the **static-mac-address** command is at the global CONFIG level of the CLI. If the device has more than one port-based VLAN, then the **static-mac-address** command is not available at the global CONFIG level. In this case, the command is available at the configuration level for each port-based VLAN.
- ARP is enabled by default and cannot be disabled.
- The ARP request broadcast is a MAC broadcast, which means it goes only to devices that are directly attached to the switch. A MAC broadcast is not routed to other networks. However, some routers, including Dell Layer 3 Switches, can be configured to reply to ARP requests from one network on behalf of devices on another network.
- If the router receives an ARP request packet it cannot deliver to the final destination because of the ARP timeout, and no ARP response is received, the router sends an ICMP Host Unreachable message to the source.

## Common diagnostic scenarios

### *Troubleshooting VPLS*

**Monitoring** – Active monitoring of MAC addresses learned at each site can help determine when the VPLS is in a non-working state. Use the **show mac vpls** command at each site and verify that the local and remote MACs match for each VPLS instance. If you see any network event on the path, you should check the state of the VPLS instances that traverse that path.

**Troubleshooting** – If you discover a problem, you should collect the following information before doing a restoration of the services. Please alert Dell Technical Support, so engineering can be involved in the data capture.

- End nodes (where the VPLS instance terminates) at both ends:

  Management Processor commands:

  ```
  Show mac vpls
  Show mpls summary
  Show mpls vpls detail
  Show mpls lsp detail
  Show mpls route
  Show mpls rsvp session detail
  ```

```
Show mpls stat vpls (few times)
Show mpls debug vpls remote
Show mpls debug vpls local
Show mpls debug next
```

Line Processor commands (slots 1, 2, and 4 on fr3.sjc, all others 1-3)

```
Rconsole <slot>
Show mpls vpls
Show mac vpls
Show mpls vpls counters
Clear mpls vpls counters
Show mpls vpls counters
Show mpls next-hop
Show mpls tunnel
Show mpls vpls local
Show mpls vpls remote
```

- Transit nodes, which identify the path each VPLS instance uses to traverse the network. Perform a traceroute between the end nodes in each direction to verify which routers are providing transit for the affected VPLS instances.

  Management Processor commands:

  ```
  Show mpls summary
  Show mpls rsvp session detail
  Show mpls stat label (few times)
  ```

  Line Processor commands (slots 1 and 2 on transit routers):

  ```
  Rconsole <slot>
  Show mpls lsp_xc
  Show mpls next-hop
  ```

### Recovery

The fastest way to recover is to remove the MPLS configuration from the end nodes and rebuild it. This causes the end nodes to re-signal and build the tunnel, which updates the transit nodes with the correct information.

If you are still unable to correct the problem, try these additional steps:

- Disable or enable LSPs.
- Delete LSPs and then reconfigure them.
- Delete MPLS interface and then reconfigure it.
- Delete VPLS peer configuration and then reconfigure it.
- Delete VPLS instance and then reconfigure it.

# 802.1Q-in-Q tagging

802.1Q-in-Q tagging allows you to configure 802.1q tag-types on a group of ports (such as trunk ports) so that there are two identical 802.1q tags (802.1Q-in-Q tagging) on a single device. This feature improves SAV interoperability between Dell devices and other vendor devices that support the 802.1q tag-types, but may be less flexible about the tag-types they accept.

802.1q tag-type translation allows you to configure an 802.1q tag-type per port, for finer granularity when configuring multiple 802.1q tag-types on a single device. Per-port tag-type configuration allows tag types to be translated from one port to the next on tagged interfaces.

## 802.1Q-in-Q debug commands

There are no specific 802.1Q-in-Q debug commands.

## Configuration notes

- PowerConnect B-MLXe routers support per-port tag-type configuration. Consequently, each port can have its own tag-type setting.
- The default tag-type for a port is 8100.
- PowerConnect B-MLXe routers support 802.1Q-in-Q tagging where the inner and outer tag can have the same tag-type values, or different tag-type values. This feature maximizes interoperability with third-party devices.
- For tag-type translation, If the specified port is part of a multi-slot trunk, the device automatically applies the tag-type to all of the ports that are part of that multi-slot trunk.
- If you do not specify a port or range of ports, the 802.1q tag-type applies to all Ethernet ports on the device.

## Common diagnostic scenarios

- CRC errors can indicate Q-in-Q issues.
- Q-in-Q issues may be caused by a faulty module (hardware problem).
- Old software versions may cause Q-in-Q issues.

  Feature issues are often be caused because the device is running an old version of the software. Dell recommends that you always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# Super Aggregated VLANs

A Super Aggregated VLAN (SAV) contains multiple VLANs. This feature allows you to construct Layer 2 paths and channels. A path contains multiple channels, each of which is a dedicated circuit between two end points. The two devices at the end points of the channel appear to each other to be directly attached, yet the network that connects them remains transparent.

You can aggregate up to 4090 VLANs inside a SAV, for a total VLAN capacity on one router of 16,728,100 channels (4090 * 4090). Since devices connected through the channel are not visible to devices in other channels, each client has a private link to the other side of the channel. SAVs are useful for applications such as Virtual Private Network (VPN) or Transparent LAN Services (TLS) where clients need a private, dedicated Ethernet connection that can reach its subnet transparently across multiple networks. SAV allows point-to-point and point-to-multipoint connections.

## SAV show commands

### show vlan

**Syntax: show vlan**

This command displays information about configured VLANs, as shown in this example:

```
PowerConnect#show vlan
Configured PORT-VLAN entries: 3
Maximum PORT-VLAN entries: 4090
Default PORT-VLAN id: 1
PORT-VLAN 1, Name DEFAULT-VLAN, Priority Level0
L2 protocols: NONE
Untagged Ports: ethernet 2/1 to 2/20 ethernet 3/1 to 3/20 ethernet
PORT-VLAN 2, Name [None], Priority Level0
L2 protocols: NONE
ip-protocol VLAN, Dynamic port disabled
Name: basic
PORT-VLAN 1001, Name [None], Priority Level0
L2 protocols: MRP
Tagged Ports: ethernet 3/1 ethernet 3/12 to 3/13 ethernet 3/20
Bytes received: 6000
```

### show vlan ethernet

**Syntax: show vlan ethernet** <*slot/port*>

This command, with a <*slot/port*> entry, displays VLAN information for the specified port, as shown in this example:

```
PowerConnect#show vlan ethernet
4/1
Port 4/1 is a member of 2 VLANs
```

### show vlan detail

**Syntax: show vlan detail** [<*vlan-id*>]

- <*vlan-id*> - Displays information about a specific VLAN.

This command displays detailed information about VLAN states, port types, and port modes, as well as control protocols configured on the VLAN, as shown in this example:

```
PowerConnect#show vlan detail
Untagged Ports: ethernet 2/1 to 2/20 ethernet 4/4
Tagged Ports: None
Dual-mode Ports: ethernet 3/1 to 3/20jjj ethernet 4/1 to 4/3
Default VLAN: 1
Control VLAN: 4095
VLAN Tag-type: 0x8100
PORT-VLAN 1, Name DEFAULT-VLAN, Priority Level0
----------------------------------------------------------
Port Type Tag-Mode Protocol State
2/1 PHYSICAL UNTAGGED NONE DISABLED
2/2 PHYSICAL UNTAGGED NONE DISABLED
2/3 PHYSICAL UNTAGGED NONE DISABLED
2/4 PHYSICAL UNTAGGED NONE DISABLED
2/5 PHYSICAL UNTAGGED NONE DISABLED
.
. (output edited for brevity)
.
4/1 PHYSICAL UNTAGGED NONE FORWARDING
4/2 PHYSICAL UNTAGGED NONE FORWARDING
4/3 PHYSICAL UNTAGGED NONE FORWARDING
4/4 PHYSICAL UNTAGGED NONE DISABLED
PORT-VLAN 100, Name [None], Priority Level0
----------------------------------------------------------
Port Type Tag-Mode Protocol State
4/1 PHYSICAL TAGGED STP FORWARDING
```

The brief form of this command displays detailed information for a specific VLAN ID, as shown in the following example:

```
PowerConnect(config-vlan-10)#show vlan detail 10
PORT-VLAN 10, Name [None], Priority Level0
----------------------------------------------------------
Port  Type      Tag-Mode  Protocol  State
1/1   PHYSICAL  TAGGED    STP       DISABLED
1/2   PHYSICAL  TAGGED    STP       DISABLED
```

## SAV debug commands

There are no debug commands specific to SAV.

## Configuration notes

- A maximum of 1544 bytes is supported on ports where super-aggregated VLANs are configured. An additional 8 bytes over the untagged port maximum allows for support of two VLAN tags.
- For core devices, you must configure a VLAN tag type (tag ID) that is different than the tag type used on edge devices. If you use the default tag type (8100) on the edge devices, set the tag type on the core devices to another value, such as 9100.

## Common diagnostic scenarios

### *Configuring SAV on a VPLS endpoint*

In an environment that includes a PowerConnect B-MLXe with both VPLS and SAV configured, one VPLS endpoint port is connected to a carrier and will be receiving dual tags, with both tags using tag-type 8100. The outer tag is the carrier's VLAN and the inner tag is from the end user.

The frame structure is **DA | SA | 8100 tag=1000 | 8100 tag=100 | Data** from the carrier side. The port must support multiple VLANs coming in from the carrier.

The other VPLS endpoint goes to the end user with a single tag, and the frame structure is **DA | SA | 8100 tag=100 | Data.**

Use the following configuration on the two endpoints to support dual tags on one side of the VPLS and a single tag on the other side:

**Carrier side endpoint configuration:**

```
Router MPLS
    vpls vpls1000 1000
    vpls-peer 2.2.2.2
    vlan 1000
    tagged ethe 3/1
```

**End user side endpoint configuration:**

```
[Global config]
    tag-type 9100 eth 3/1
Router MPLS
    vpls vpls1000 1000
    vpls-peer 1.1.1.1
    vlan 100
    untagged ethe 3/1
```

This configuration works in the following way:

Packets will ingress the carrier endpoint with dual tags, both with tag-type 8100. The VLAN-ID 1000 will be stripped off the outer tag, and the inner tag customer VLAN ID 100 will be sent over the VPLS as payload. The packet will then egress the end user endpoint with a single tag-type 8100 VLAN-ID of 100.

Packets will ingress the end user endpoint with a single VLAN tag, tag-type 8100, VLAN-ID 100. The port is configured with SAV tag-type 9100 and is also configured as "untagged", so the packets will be accepted and the VLAN-ID 100 will be sent over the VPLS as payload. The packet will then egress the carrier endpoint "tagged" interface, where it will add the outer tag with tag type 8100 VLAN-ID 1000.

- Old software versions

  Feature issues are often caused because the device is running an old version of the software. Dell recommends that you always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# MRP

MRP is a proprietary protocol that prevents Layer 2 loops and provides fast reconvergence in Layer 2 ring topologies. MRP is especially useful in Metropolitan Area Networks (MANs) which typically require more flexibility than STP can deliver.

This section describes how to use debug commands to monitor Metro Ring Protocol (MRP) environments for PowerConnect B-MLXe routers.

## Using MRP diagnostics

When you enable MRP diagnostics, the software tracks RHPs (Ring Health Packets) according to their sequence numbers, and calculates how long it takes an RHP to travel once through the entire ring. When you display the diagnostics, the CLI shows the average round-trip time for the RHPs sent since you enabled diagnostics. The calculated results have a granularity of 1 microsecond.

## Enabling MRP diagnostics

To enable MRP diagnostics for a ring, enter the following command on the master node, at the configuration level for the ring (this command is valid only on the master node):

**diagnostics**

**Syntax: diagnostics**

# MRP show commands

### show metro diagnostics

**Syntax: show metro diagnostics** <*ring id*>

This command displays MRP diagnostics results. In the following example, the results are for metro ring 1:

```
PowerConnect(config-vlan-5-mrp-1)#show metro diagnostics 1
Metro Ring 1
==============
diagnostics results

Ring       Diag       RHP average     Recommended     Recommended
id         state      time(microsec)  hello time(ms)  Prefwing time(ms)

1          enabled    < 126           100             300
Diag frame sent      Diag frame lost
6                           0
```

# MRP debug commands

### debug mrp bpdu

**Syntax:** [**no**] **debug mrp bpdu**

When this command is enabled, you will see an error message whenever a BPDU is lost on the master node (not shown for member nodes).

### debug mrp diagnostics

**Syntax:** [**no**] **debug mrp diagnostics**

This command displays MRP diagnostic information. To activate diagnostic reporting, first enable MRP diagnostics debugging, then display the diagnostic information using the **show debug** command

### debug mrp event

**Syntax:** [**no**] **debug mrp event**

This command displays information about MRP events, as shown in this example:

```
PowerConnect#debug mrp event
        event: debugging is on

PowerConnect#Apr 13 19:05:22 mrp-debug: **state PREFORWARDING for port 2/1 in
ring 1 **
Apr 13 19:05:22 mrp-debug: ** state FORWARDING for port 2/1 in ring 1 **
Apr 13 19:05:22 mrpinfo - port 2/1, up 0
Apr 13 19:05:22 mrp-debug: ** state DISABLED for port 2/1 in ring 1 **
Apr 13 19:05:28 mrpinfo - port 2/1, up 1
Apr 13 19:05:28 mrp-debug: ** state BLOCKING for port 2/1 in ring 1 **
Apr 13 19:05:29 mrp-debug: mrpdiags_receive_packet. rpdu with sequence number
18
184 has been lost. Resetting timers
```

### Configuration notes

MRP can be enabled on port-based VLANs, but cannot be enabled or disabled on protocol-based VLANs.

# Spanning Tree and derivatives

The following sections describe diagnostic procedures for Spanning Tree, and Spanning Tree derivatives, including SSTP, MSTP, RSTP, and SuperSpan.

**NOTE**
Layer 2 protocols such as STP, RSTP, MRP, and VSRP can be enabled on port-based VLANs, but cannot be enabled or disabled on protocol-based VLANs.

## Spanning Tree Protocol

A control protocol, such as Spanning Tree Protocol (STP), can block one or more ports in a protocol-based VLAN that uses a virtual routing interface to route to other VLANs. For IP protocol and IP subnet VLANs, even though some of the physical ports of the virtual routing interface are blocked, the virtual routing interface can still route as long as at least one port in the protocol-based VLAN is not blocked by STP.

## Single Spanning Tree Protocol

For VLANs where Single Spanning Tree Protocol (SSTP) is enabled, the ports become members of a single spanning tree. For VLANs where SSTP is disabled, ports are excluded from the single spanning tree. VLANs can also be selectively added or removed from the single spanning tree domain.

## RSTP

Reverse Spanning Tree Protocol (RSTP) provides rapid traffic reconvergence for point-to-point links within a few milliseconds (< 500 milliseconds), following the failure of a bridge or bridge port. This reconvergence occurs more rapidly than that provided by STP because convergence in RSTP bridges is based on the explicit handshakes between designated ports and their connected root ports rather than on timer values.

## MSTP

With multiple spanning tree regions, (MSTP), the entire network runs a common instance of RSTP. Within the common instance, one or more VLANs can be individually configured into distinct regions. The entire network runs the Common Spanning Tree (CST) instance and the regions run a local instance, or Internal Spanning Tree (IST). Because the CST treats each IST as a single bridge, ports are blocked to prevent loops that might occur within an IST and also throughout the CST. In addition, MSTP can coexist with individual devices running STP or RSTP in the Common and Internal Spanning Trees instance (CIST). With the exception of the provisions for multiple instances, MSTP operates exactly like RSTP.

## SuperSpan™

SuperSpan is an STP enhancement that allows Service Providers (SPs) to use STP in both SP networks and customer networks. The SP PowerConnect B-MLXe devices are configured to tunnel each customer's STP BPDUs through the SP. From the customer's perspective, the SP network is a loop-free non-blocking device or network. The SP network behaves like a hub in the sense that the necessary blocking occurs in the customer network, not in the SP.

## ST show commands

### show spanning-tree

**Syntax: show spanning-tree**

This command displays the following information (in this instance, for VLAN 10):

```
PowerConnect#show spanning-tree vlan 10
VLAN 10 - STP instance 1
------------------------------------------------------------------
STP Bridge Parameters:
Bridge           Bridge Bridge Bridge Hold Last Topology Topology
Identifier       MaxAge Hello  FwdDly Time Change       Change
hex              sec    sec    sec    sec  sec          cnt
8000000480a04000 20     2      15     1    0            0
RootBridge       RootPath DesignatedBridge Root Max Hel Fwd
Identifier       Cost     Identifier       Port Age lo  Dly
hex                       hex              sec  sec     sec
8000000480a04000 0        8000000480a04000 Root 20  2   15
STP Port Parameters:
Port Prio Path State Designat- Designated       Designated
Num  rity Cost       ed Cost   Root             Bridge
1/3  128  4    DISABLED 0      0000000000000000 0000000000000000
1/13 128  4    DISABLED 0      0000000000000000 0000000000000000
```

## STP debug commands

This section describes the debug commands for STP, MSTP, and RSTP PowerConnect B-MLXe router environments.

**CAUTION**

**Enabling diagnostic commands may degrade system performance. These commands are best used to troubleshoot specific problems while working with qualified Dell service technicians. Whenever possible, troubleshoot your system during periods of low network traffic and user activity to preserve system performance.**

### debug spanning-tree

**Syntax:** [no] **debug spanning-tree** [**config-bpdu** | **event** | **port** | **reset** | **show**]

- **config-bpdu** - Displays information about STP BPDU configurations.
- **event** - Displays information about STP non-BPDU events, such as timer, configuration, etc.
- **port** - Displays all STP instances on a specific port.
- **reset** - Resets all STP debugging parameters to the default.
- **show** - Displays information about all spanning tree events (default).

This command generates information about all spanning tree events (by default) or specific events as defined by the variables shown in the syntax line. Output for all events resembles the following:

```
PowerConnect#debug spanning-tree
STP: debugging is on
PowerConnect#
STP: Sending Config BPDU - VLAN 3 Port 2/3
0000 00 00 00 8000000000001c042 00000000
 8000000000001c042 8043 0000 0000 0000 0000
STP: Sending Config BPDU - VLAN 3 Port 2/4
0000 00 00 00 8000000000001c042 00000000
 8000000000001c042 8044 0000 0000 0000 0000
STP: Sending Config BPDU - VLAN 2 Port 2/1
0000 00 00 00 8000000000001c040 00000000
 8000000000001c040 8041 0000 0000 0000 0000
STP: Sending Config BPDU - VLAN 2 Port 2/2
0000 00 00 00 8000000000001c040 00000000
 8000000000001c040 8042 0000 0000 0000 0000
STP: Sending Config BPDU - VLAN 3 Port 2/3
0000 00 00 00 8000000000001c042 00000000
 8000000000001c042 8043 0000 0000 0000 0000
...
```

### debug spanning-tree show

**Syntax:** [no] **debug spanning-tree show**

This command shows the default STP debug configuration:

```
PowerConnect#debug spanning-tree show
STP Debug Parameters
--------------------
STP debugging is ON [Mode: Brief]
NonBpduEvents ConfigBpduEvents TcnBpdusEvents are being tracked
Ports: All
VLANs: All
```

### debug spanning-tree tcn-bpdu

**Syntax:** [no] **debug spanning-tree tcn-bpdu**

This command enables debugging of TCN BPDUs.

### debug spanning-tree verbose

**Syntax:** [no] **debug spanning-tree verbose**

In verbose mode, STP BPDU information is translated into BPDU fields and values, which are easier to read than the default hex output. The verbose form of the output resembles the following:

```
PowerConnect# debug spanning-tree verbose
STP: Sending Config BPDU - VLAN 2 Port 2/2
         protocol-id: 0000
protocol-version: 00
type: 00
flags: 00
root-id: 800000000001c040
path-cost: 00000000
bridge-id: 800000000001c040
port-id: 8042
message-age: 0000
max-age: 0000
hello-time: 0000
forward-delay: 0000
```

By contrast, the default hex output resembles this example:

```
STP: Sending Config BPDU - VLAN 2 Port 2/2
         0000 00 00 00 800000000001c040 00000000
800000000001c040 8042 0000 0000 0000 0000
```

### debug spanning-tree vlan

**Syntax:** [no] **debug spanning-tree vlan**

This command restricts debug output to a select list of VLANs. You may choose to monitor all, one, or a subset of VLANs. For example:

```
PowerConnect#debug spanning-tree vlan 2 3
```

This command enables debugging for VLANs 2 and 3 concurrently.

## MSTP debug commands

### debug mstp bpdu

**Syntax:** [no] **debug mstp bpdu**

This command records and displays MSTP BPDU and MRecord events.

### debug mstp event

**Syntax:** [no] **debug mstp event**

This command displays MSTP state machine events.

### debug mstp mstid

**Syntax:** [no] **debug mstp mstid**

This command displays information for a specific MSTP instance.

### debug mstp port

**Syntax:** [no] **debug mstp port**

This command displays debug information about specific MSTP ports.

**debug mstp show**

**Syntax:** [no] **debug mstp show**

This command displays current MSTP debug parameters.

**debug mstp state**

**Syntax:** [no] **debug mstp state**

This command displays information about MSTP port state events.

**debug mstp verbose**

**Syntax:** [no] **debug mstp verbose**

This command displays MSTP debug information in the verbose mode.

## RSTP debug commands

The **debug rstp** commands are similar in form and function to the spanning-tree commands. For detailed information about these commands, see the related STP command.

## Configuration notes

- Changing the STP state of the primary port in a trunk group affects all ports in the trunk group.
- With RSTP, rapid convergence will not occur on ports connected to shared media devices, such as hubs. To take advantage of the rapid convergence provided by RSTP, make sure to explicitly configure all point-to-point links in a topology.
- When you enable SuperSpan globally, and then create a new VLAN, the new VLAN inherits the global SuperSpan state.
- Enabling diagnostic commands may degrade system performance. These commands are best used to troubleshoot specific problems while working with qualified Dell service technicians. Whenever possible, troubleshoot your system during periods of low network traffic and user activity to preserve system performance.
- In many cases, generic debugging is not useful. If multiple STP instances are configured, it can be difficult to identify content for a specific instance from the extensive output that may be generated. Use the debug spanning-tree port and debug spanning tree vlan commands to define specific instances for debugging.

## Common diagnostic scenarios

- Spanning Tree loops.
- Spanning Tree reacts to topology changes and port flapping.
- Port flapping can trigger a new Spanning Tree learning process.
- Old software versions.

  Feature issues are often caused because the device is running an old version of the software. Dell recommends that you always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# Traps and trap servers

For more information about SNMP traps, and how to configure trap servers, see the *NetIron Series Configuration Guide*, and the *IronWare MIB Reference*.

# LACP trunking

The Link Aggregation Control Protocol (LACP) allows ports on both sides of a redundant link to automatically configure themselves into a trunk link (aggregate link), eliminating the need for manual configuration. LACP has two modes:

- Active mode – When active link aggregation is enabled, the Dell port can exchange standard LACP Protocol Data Unit (LACPDU) messages to negotiate trunk group configuration with the port on the other side of the link. In addition, the Dell port actively sends LACPDU messages on the link to search for a link aggregation partner at the other end of the link, and can initiate an LACPDU exchange to negotiate link aggregation parameters with an appropriately configured remote port.
- Passive mode – In passive link aggregation, the Dell port can exchange LACPDU messages with the port at the remote end of the link, but this port cannot search for a link aggregation port or initiate negotiation of an aggregate link. In passive mode, the port at the remote end of the link must initiate the LACPDU exchange.

When you enable link aggregation on a group of Dell ports, the Dell ports can negotiate with the ports at the remote ends of the links to establish trunk groups.

## Trunk show commands

The following show commands display information about trunking configurations.

### show lag

**Syntax: show lag**

This command displays trunk information for the ports managed by the CPU. The server trunk load balancing information is shown in bold type in this example. The number in parentheses indicates how many hash values are assigned to the port. The CPU assigns the hash values evenly to the trunk ports managed by the CPU. In this example, the next time the device needs to assign a hash value, it will assign the value to port 8/4.

```
PowerConnect(config-trunk-14/6,14/8)#show lag
Max number of trunks: 128
Available: 127
Configured number of server trunks: 1

Configured trunks:

Trunk ID: 1
Type: Server
Ports_Configured: 2
Base FID: 0x0400
FID count: 16

Ports         14/6    14/8
Port Names    ve4012  TestCa*
```

```
Port_Status   disable disable

Operational trunks:

Trunk ID: 1
Type: Server
Duplex: None
Speed: None
Tag: Dual
Priority: level0
Active Ports: 0

Ports               14/6   14/8
Link_Status         down   down
```

## Trunk debug commands

There are no debug commands specific to LACP trunking.

## Configuration notes

- You cannot use 802.3ad link aggregation on a port configured as a member of a static trunk group.
- When LACP dynamically adds or changes a trunk group, the **show lag** command displays the trunk as both configured and active. However, the **show running-config** or **write terminal** commands do not contain a trunk command defining the new or changed trunk group.
- You can enable link aggregation on 802.1q tagged ports (ports that belong to more than one port-based VLAN).
- LACP cannot be configured on a VPLS end-point port, and a VPLS end-point cannot be configured on a physical port that has LACP enabled. This restriction is enforced by the CLI. If a VPLS end-point receives any LACP traffic, this traffic will be dropped by the router at ingress. However, if the VPLS has CPU protection enabled, this traffic will be hardware flooded.
- Dell recommends that you disable or remove the cables from the ports you plan to enable for dynamic link aggregation. Doing so prevents the possibility that LACP will use a partial configuration to talk to the other side of a link. A partial configuration does not cause errors, but sometimes requires LACP to be disabled and re-enabled on both sides of the link to ensure that a full configuration is used. It is easier to disable a port or remove its cable first. This applies both for active link aggregation and passive link aggregation.

### *Trunk formation rules*

When troubleshooting trunks, make sure the following rules for trunk formation have been considered:

- Any number of ports between 2 and 20 within the same chassis can be used to configure a trunk port.
- Use the server trunk option when configuring a trunk group that connects to VPLS or VLL end points.
- Ports in a trunk must have the same speed, negotiation mode, and QoS priority or the trunk is rejected.
- All ports configured in a trunk must be configured with the same port attributes.

- Primary port policy applies to all secondary ports. No trunk is rejected.
- The trunk is rejected if any trunk port has mirroring or monitoring configured.
- The trunk is rejected if any trunk port has vlan or inner-vlan translation configured.

### *Layer 2 requirements*

The trunk is rejected if the trunk ports:

- Do not have the same untagged VLAN component.
- Do not share the same SuperSpan Customer ID (CID).
- Do not share the same VLAN membership.
- Do not share the same uplink VLAN membership.
- Do not share the same protocol-VLAN configuration.
- Are configured as marble primary and secondary interfaces.

### *Layer 3 requirements*

The trunk is rejected if any secondary trunk port has any Layer 3 configurations, such as IPv4 or IPv6 addresses, OSPF, RIP, RIPNG, ISIS, etc.

### *Layer 4 (ACL) requirements*

All trunk ports must have the same ACL configurations or the trunk is rejected. You can have a maximum of 128 server trunks and 80 switch trunks.

**NOTE**
These trunking rules are applicable for both statically configured trunks as well as dynamic trunks created using the LACP protocol.

## Common diagnostic scenarios

- When adding a new port to a 4-port LACP trunk, the existing ports go down and come back up in order to establish the fifth port. For existing 5-port trunks, when one port flaps, the other ports in the trunk will not flap.
- LACP trunk links may not operate properly between Dell devices and third-party devices because of a mismatch between the link configurations. If the link is fixed on the third-party side, the link on the Dell device should be trunk. If it is link-aggregated on the third party, then it needs to be the same on the Dell side.
- When adding new ports to a static or dynamic (LACP) trunk, you must reconfigure the LAG group.
- LACP links may not operate properly due to mis-configurations. Contact Dell Technical Support for help with configuration issues.
- Old software versions.

  Feature issues are often caused because the device is running an old version of the software. Dell recommends that you always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# UDLD

Uni-directional Link Detection (UDLD) monitors a link between two routers to quickly detect link failures. UDLD brings the ports on both ends of the link down if the link goes down at any point between the two devices. This feature is useful for individual port links, and for trunk links.

## UDLD show commands

### show link-keepalive

**Syntax: show link-keepalive**

This command displays UDLD information for all ports:

```
PowerConnect(config)#show link-keepalive
Total link-keepalive enabled ports: 4
Keepalive Retries: 5 Keepalive Interval: 1 Sec.
Port     Physical Link     Link-keepalive    Logical Link
4/1      up                up                up
4/2      up                up                up
4/3      down              down              down
4/4      up                down              down
```

### show interface brief

**Syntax: show interface brief**

This command displays concise information If a port is disabled by UDLD, as shown here

```
PowerConnect#show interface brief
Port Link   State       Dupl     Speed    Trunk    Tag    Priori    MAC Name
1/1  Up     LK-DISABLE  None     None     None     No     level0    00e0.52a9.bb00
1/2  Down   None        None     None     None     No     level0    00e0.52a9.bb01
1/3  Down   None        None     None     None     No     level0    00e0.52a9.bb02
1/4  Down   None        None     None     None     No     level0    00e0.52a9.bb03
```

### show link-keepalive ethernet

**Syntax: show link-keepalive ethernet** <*slot/portnum*>

To display detailed UDLD information for a specific port, enter a command similar to the following:

```
PowerConnect#show link-keepalive ethernet 4/1
Current State: up Remote MAC Addr: 00e0.52d2.5100
Local Port: 4/1 Remote Port: 2/1
Local System ID: e0927400 Remote System ID: e0d25100
Packets sent: 254 Packets received: 255
Transitions: 1
```

## UDLD debug commands

There are no debug commands specific to UDLD.

## Clearing UDLD statistics

To clear UDLD statistics, enter the following command.

**clear link-keepalive statistics**

**Syntax:** clear link-keepalive statistics

This command clears the packets sent, packets received, and transitions counters in the **show link-keepalive ethernet** <*slot*>/<*portnum*> display.

## Configuration notes

- UDLD is supported only on Ethernet ports.
- To configure UDLD on a trunk group, you must configure the feature on each port of the group individually.
- Configuring UDLD on the primary port of a trunk group enables the feature on that port only.
- Dynamic trunking is not supported. If you want to configure a trunk group that contains ports where UDLD is enabled, you must remove the UDLD configuration from the ports. After you create the trunk group, you can add the UDLD configuration again.
- When a specified number of port failures occurs, UDLD can bring the port down. Syslog messages are generated when this occurs. You can also keep the UDLD keyword instead of the link-keepalive keyword to configure UDLD and display UDLD information.

### *UDLD messages*

Informational UDLD: Logical link on interface ethernet <*portnum*> is up

Informational UDLD: Logical link on interface ethernet <*portnum*> is down

### *Syslog message*

Once the UDLD holddown threshold for a port is reached, a warning message is generated and written to the system log. The message means that UDLD is blocked on the specified port until the holddown counter is cleared for the port.

## Common diagnostic scenarios

- Port flapping occurs with UDLD configured.

  Port flapping can occur because of a problem with a dirty optic module connection, or other hardware issue.
- UDLD may bring a port down (port blocked) because of a malformed RHR.
- UDLD packets may be sent from a port, but not received on a port.

  This may be a hardware issue with the port or the module.
- UDLD links may not operate properly between Dell devices and third-party devices.

  UDLD is proprietary and not compatible between vendors. Contact Dell Technical Support for assistance.
- UDLD interfaces may flap during a write memory exercise.

CPU usage may be too high. Contact Dell Technical Support for assistance.

- UDLD port numbers on remote devices may be reported by incorrectly (wrong ID number) by the Dell device.

  The remote port number reflects the port ID sent by the remote router or switch and interpreted by the local router. In cases where the local router interprets the port ID differently than the remote, the port ID shown may be incorrect.

- Old software versions

  Feature issues are often caused because the device is running an old version of the software. Dell recommends that you always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# VLAN Translation

VLAN Translation allows traffic from one VLAN to be transported across a different VLAN. Packet VLAN IDs from the original VLAN are changed at the ingress port of the translating VLAN. When the packets reach the egress point on the translating VLAN, the VLAN ID is translated back to the original ID.

This feature is useful for service providers who need to carry traffic from different customers across their network, while preserving the VLAN ID and priority information of the customer network.

## VLAN Translation show commands

### show vlan

**Syntax:** **show vlan** [<*vlan-id*>] [ **begin** <*expression*> | **exclude** <*expression*> | **include** <*expression*>]

This command displays global VLAN Translation information, as shown in the following example:

```
PowerConnect#show vlan
Configured PORT-VLAN entries: 3
Maximum PORT-VLAN entries: 4090
Default PORT-VLAN id: 1
PORT-VLAN 1, Name DEFAULT-VLAN, Priority Level0
L2 protocols: NONE
Untagged Ports: ethernet 2/1 to 2/20 ethernet 3/1 to 3/20 ethernet
PORT-VLAN 2, Name [None], Priority Level0
L2 protocols: NONE
ip-protocol VLAN, Dynamic port disabled
Name: basic
PORT-VLAN 1001, Name [None], Priority Level0
L2 protocols: MRP
Tagged Ports: ethernet 3/1 ethernet 3/12 to 3/13 ethernet 3/20
Bytes received: 6000
```

To display information for a specific VLAN, enter a VLAN ID.

## VLAN Translation debug commands

The debug commands described here apply to the management module processor.

### debug vlan-translation

**Syntax:** [no] **debug vlan-translation** [**aging** | **cam** | **error** | **generic** | **packet**]

- **aging** - Displays aging information for a translation session entry.
- **cam** - Displays information about CAM/PRAM programming.
- **error** - Displays VLAN-translation errors.
- **generic** - Displays generic VLAN-translation information.
- **packet** - Displays information about VLAN-translation packet processing.

This command generates information about VLAN-translation activity.

### debug vlan-translation cam

**Syntax:** debug vlan-translation cam

This command generates information about VLAN CAM activity. Output resembles the following:

```
PowerConnect#debug vlan-translation cam
VLAN-ID-TRANS: Check for CAM optimization
```

This example indicates that a CAM optimization check has been done for a one-to-one VLAN-translation tunnel.

### debug vlan-translation generic

**Syntax:** debug vlan-translation generic

This command generates generic VLAN-translation information. Output resembles the following:

```
PowerConnect#debug vlan-translation generic
VLAN-ID-TRANS: send IPC message VLAN_TRANS_GRP_UPDATE: vtg id 1, entry index 10,
vlan id 100, vtg fid 0x0002, port mask 0, no 0
```

This example indicates that a configuration update change for a VLAN-translation group (vtg 1) has been sent to the line cards.

```
VLAN-ID-TRANS: Allocate FID for port mask 1
```

# STP and RSTP

This section describes how to use Dell diagnostic debug commands to monitor Spanning Tree Protocol (STP) and Rapid Spanning Tree Protocol (RSTP) environments for PowerConnect B-MLXe routers.

**CAUTION**

**Enabling diagnostic commands may degrade system performance. These commands are best used to troubleshoot specific problems while working with qualified Dell service technicians. Whenever possible, troubleshoot your system during periods of low network traffic and user activity to preserve system performance.**

**NOTE**
Since STP and RSTP debug commands are virtually identical, only STP commands are described in detail in this chapter. A list of the debug RSTP commands appears at the end of the chapter.

### debug spanning-tree

**Syntax:** debug spanning-tree

This command generates information about all BPDUs and spanning tree events (by default) or specific events as defined by the variables shown in the syntax line. In many cases, generic debugging is not useful. If multiple STP instances are configured, it can be difficult to identify content for a specific instance from the extensive output that may be generated. Use the **debug spanning-tree port** and **debug spanning tree vlan** commands to define specific instances for debugging.

**Syntax:** [**no**] **debug spanning-tree** [**config-bpdu** | **event** | **port** | **reset** | **show** | **tcn-bpdu** | **verbose** | **vlan**].

- **config-bpdu** - Generates information about STP configuration bridge protocol data units (BPDUs).
- **event** - Generates information about STP non-BPDU events (timer, configuration, etc.).
- **port** - Restricts STP debugging to specific ports.
- **reset** - Resets all STP debugging parameters to default.
- **show** - Displays current STP debug settings.
- **tcn-bpdu** - Enables or disables STP TCN BPDU debugging.
- **verbose** - Enables or disables STP verbose debugging mode.
- **vlan** - Restricts STP debugging to specific VLANs.

### debug spanning-tree

**Syntax: debug spanning-tree**

This command generates information about VLAN-translation activity.

This command generates information about all STP activity and events. Output resembles the following:

```
PowerConnect#debug spanning-tree
STP: debugging is on
PowerConnect#
STP: Sending Config BPDU - VLAN 3 Port 2/3
0000 00 00 00 800000000001c042 00000000
 800000000001c042 8043 0000 0000 0000 0000
STP: Sending Config BPDU - VLAN 3 Port 2/4
0000 00 00 00 800000000001c042 00000000
 800000000001c042 8044 0000 0000 0000 0000
STP: Sending Config BPDU - VLAN 2 Port 2/1
```

### debug spanning-tree config-bpdu

**Syntax: debug spanning-tree config-bpdu**

This command generates information about STP BPDUs. Output resembles the following:

```
PowerConnect#debug spanning-tree config-bpdu
STP: Received Config BPDU - VLAN 10 Port 3/20
        0000 00 00 01 80000012f23d8500 00000000
        80000012f23d8500 8050 0000 1400 0200 0f00
```

```
STP: Received Config BPDU - VLAN 10 Port 3/20
         0000 00 00 01 80000012f23d8500 00000000
        80000012f23d8500 8050 0000 1400 0200 0f00
STP: Received Config BPDU - VLAN 10 Port 3/20
        0000 00 00 01 80000012f23d8500 00000000
        80000012f23d8500 8050 0000 1400 0200 0f00
STP: Received Config BPDU - VLAN 10 Port 3/20
        0000 00 00 01 80000012f23d8500 00000000
        80000012f23d8500 8050 0000 1400 0200 0f00
```

### debug spanning-tree event

**Syntax: debug spanning-tree event**

This command displays information about non-BPDU events, such as timer, configuration, etc. Output resembles the following:

```
PowerConnect#debug spanning-tree event
 STP: LISTENING - VLAN 10 Port 3/20
 STP: LEARNING - VLAN 10 Port 3/20
 STP: Sending TCN BPDU - VLAN 10 Port 3/20
 STP: FORWARDING - VLAN 10 Port 3/20
 STP: TCN ACK Received - VLAN 10 Port 3/20
```

### debug spanning-tree port

**Syntax: debug spanning-tree port** <*slot/port*>

This command displays spanning tree information about a specific port. Output resembles the following:

```
PowerConnect#debug spanning-tree port 1/2
 STP: Sending TCN BPDU - VLAN 10 Port 3/20
 STP: TCN ACK Received - VLAN 10 Port 3/20
```

### debug spanning-tree reset

**Syntax: debug spanning-tree reset**

This command resets all STP debugging parameters to the default. The default mode disables all STP debugs. The show debug command will not longer show STP debug as enabled. This command works in the same way as the **no debug stp** command.

### debug spanning-tree show

**Syntax: debug spanning-tree show**

This command generates information about the STP debug configuration. The following output shows the default configuration:

```
PowerConnect#debug spanning-tree show
STP Debug Parameters
--------------------
STP debugging is ON [Mode: Brief]
NonBpduEvents ConfigBpduEvents TcnBpdusEvents are being tracked
Ports: All
VLANs: All
```

### debug spanning-tree tcn-bpdu

**Syntax:** debug spanning-tree tcn-bpdu

This command displays information about TCN BPDU events. Output resembles the following:

```
PowerConnect#debug spanning-tree tcn-bpdu
 STP: Sending TCN BPDU - VLAN 10 Port 3/20
 STP: TCN ACK Received - VLAN 10 Port 3/20
```

### debug spanning-tree verbose

**Syntax: debug spanning-tree verbose**

In verbose mode, STP BPDU information is translated into BPDU fields and values, which are easier to read than the default hex output. For example, for VLAN2, Ethernet port 2/2, the default hex output resembles the following:

```
PowerConnect# debug spanning-tree verbose
STP: Sending Config BPDU - VLAN 2 Port 2/2
         0000 00 00 00 8000000000001c040 00000000
8000000000001c040 8042 0000 0000 0000 0000
```

By contrast, the verbose form of the output resembles the following:

```
STP: Sending Config BPDU - VLAN 2 Port 2/2
         protocol-id: 0000
protocol-version: 00
type: 00
flags: 00
root-id: 8000000000001c040
path-cost: 00000000
bridge-id: 8000000000001c040
port-id: 8042
message-age: 0000
max-age: 0000
hello-time: 0000
hello-time: 0000
```

### debug spanning-tree vlan

**Syntax:** [**no**] d**ebug spanning-tree vlan** <*vlan id num*> <*vlan id num*>...

This command restricts debug output to a select list of VLANs. You may choose to monitor all, one, or a subset of VLANs. For example, the following output enables debugging for VLANs 2 and 3 concurrently:

```
PowerConnect#debug spanning-tree vlan 2 3
STP: Sending Config BPDU - VLAN 2 Port 2/1
       0000 00 00 00 8000000000001c040 00000000
8000000000001c040 8041 0000 0000 0000 0000
STP: Sending Config BPDU - VLAN 3 Port 2/1
       0000 00 00 00 8000000000001c040 00000000
8000000000001c040 8042 0000 0000 0000 0000
STP: Sending Config BPDU - VLAN 2 Port 2/1
       0000 00 00 00 8000000000001c040 00000000
8000000000001c040 8043 0000 0000 0000 0000
STP: Sending Config BPDU - VLAN 2 Port 2/1
       0000 00 00 00 8000000000001c040 00000000
8000000000001c040 8044 0000 0000 0000 0000
```

# VSRP

VSRP is a proprietary protocol that provides redundancy and sub-second failover in Layer 2 mesh topologies. Based on the Virtual Router Redundancy Protocol Extended (VRRPE), VSRP provides one or more backups for the PowerConnect B-MLXe. If the active PowerConnect B-MLXe becomes unavailable, one of the backups takes over as the active device and continues forwarding traffic for the network.

VSRP provides Layer 2 redundancy, meaning that Layer 2 links are backed up. You can configure VSRP to provide redundancy for Layer 2 only or also for Layer 3.

- Layer 2 only – The Layer 2 links are backed up but specific IP addresses are not backed up.
- Layer 2 and Layer 3 – The Layer 2 links are backed up and a specific IP address is also backed up. Layer 3 VSRP is the same as VRRPE. However, using VSRP provides redundancy at both layers at the same time.

## VSRP show commands

### show log

**Syntax: show log**

This command displays log messages, as shown in this example:

```
PowerConnect#show log
Syslog logging: enabled (0 messages dropped, 1 flushes, 0 overruns)
    Buffer logging: level ACDMEINW, 5 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning

Dynamic Log Buffer (50 lines):
I:VSRP: VLAN 5 VRID 5 - transition to Backup
I:VSRP: VLAN 5 VRID 5 - transition to Master
I:VSRP: VLAN 5 VRID 5 - transition to Master-Confirm
I:VSRP: VLAN 5 VRID 5 - transition to Backup
I:System: Interface ethernet 1/6, state up
```

This log output shows all of the events that have taken place.

### show vsrp

**Syntax:** show vsrp

This command displays VSRP diagnostic information, as shown in the following example:

```
PowerConnect#show vsrp
VLAN 5
 Auth-type no authentication
 VRID 5
 ========
  State           Administrative-status Advertise-backup Preempt-mode
  Initialize      Enabled               Disabled         True

  Parameter        Configured Current    Unit/Formula
  Priority         150        1          (150-0)*(0.0/1.0)
  Hello-interval   1          1          sec/10
  Dead-interval    3          3          sec/10
  Hold-interval    3          3          sec/10
  Initial-ttl      2          2          hops

  Member ports:      ethe 1/6
  Operational ports: None
info - vsrp_set_init() - init for vrid 5
info - vsrp_set_backup() - instance 5 set to backup state
info - vsrp_set_master_confirm() - vrid 5 set to master confirm
```

In this example, VRID 5 is in the initialization state, with a priority of 150. Because the current Master's priority is lower than 150, VRID 5 changes from Backup to Master.

In the section of output shown below, the new Master receives a Hello message with a higher priority (200) and reverts to Backup. In the output, several VSRP PDUs are received, but none with a priority higher than 200, so the new Master remains Master:

```
VSRP PDU received - vlan 5 - port 1/6

ver:2 type:1 vrid:5 pri:200 #ip:0 aut:0
        adv:1 dea:3 hld:3 ttl:2 scl:10 chksum:0x11ecApr 26 16:09:13
event:
==========
info - vsrp_set_backup() - instance 5 set to backup state

VSRP PDU received - vlan 5 - port 1/6

ver:2 type:3 vrid:5 pri:200 #ip:0 aut:0
        adv:1 dea:3 hld:3 ttl:2 scl:10 chksum:0x0fecApr 26 16:09:14
====================================
```

## VSRP debug commands

This section describes how to use Dell debug commands to monitor Virtual Switch Redundancy environments.

### debug vsrp

**Syntax:** [no] **debug vsrp** [**all** | **aware** | **error** | **events** | **packets** | **show** | **state** | **verbose** | **vlan**]

- **all** - Displays information about VSRP events.
- **aware** - Displays information about VSRP aware.
- **error** - Displays VSRP errors.
- **events** - Displays VSRP events.
- **packets** - Displays information about VSRP packets.
- **show** - Displays VSRP debug parameters.
- **state** - Displays VSRP state changes.
- **verbose** - Displays VSRP information in verbose mode.
- **vlan** - Displays VSRP debug information for a specific VLAN.

This command generates information about VSRP activity. All VSRP routers go to backup state first when they are reloaded. Output is similar to the following, where router 1 goes to master-confirm and master state. For Layer 3 VSRP, the master sends out the gratuitous ARP for the virtual router IP address, then sends out the hello packet as Layer 2 VSRP.

```
PowerConnect# debug vsrp all
VSRP debugging is setup for all attributes
PowerConnect# debug vsrp
   VSRP: debugging is on

May 17 11:06:18 VSRP: VLAN VLAN: 100, VRID 1, State BACKUP
May 17 11:06:25 VSRP: Packet received on port 1/6, vlan VLAN: 100
   ver:2 type:1 vrid:1 pri:111 #1p:1 aut:0
   adv:1 dea:3 hld:3 ttl:2 scl:10 chksum:0xa93c 102.53.5.1
May 17 11:06:35 VSRP: Packet received on port 1/6, vlan VLAN: 100
   ver:2 type:1 vrid:1 pri:111 #ip:1 aut:0
   adv:1 dea:3 hld:3 ttl:2 scl:10 chksum:0xa93c 102.53.5.1
May 17 11:06:42 VSRP: VLAN VLAN:100, VRID 1, State MASTER-CONFIRM
May 17 11:06:42 VSRP: vlan VLAN:10, VRID 10, state MASTER
   ver:2 type:1 vrid:10 pri:110 #ip:1 aut:0
   adv:1 dea:3 hld:3 ttl:2 scl:10 chksum:0xaa3c 102.53.5.1
   ver:2 type:1 vrid:10 pri:110 #ip:1 aut:0
   adv:1 dea:3 hld:3 ttl:3 scl:10 chksum:0xaa3c 192.52.5.1
May 17 11:07:04 VSRP: vlan VLAN:100, VRID 1 - send ARP requrest for ip 192.53.5.1
   ver:2 type:1 vrid:10 pri:110 #ip aut:0
```

## Configuration notes

- The **delay-link-event** command delays the sending of port up/down events to Layer 2 protocols. If VSRP is enabled on the port, the ownership won't change until the port status has remained up or down for the configured amount of time to ensure that minor transient states of a link do not unintentionally cause a disruptive topology change in the network.

- VSRP is the same as VRRPE. However, VSRP provides redundancy at both layers at the same time. The PowerConnect B-MLXe support Layer 2 and Layer 3 redundancy. You can configure a PowerConnect B-MLXe for either Layer 2 only or Layer 2 and Layer 3. To configure for Layer 3, specify the IP address where the backup will occur.
- To provide Layer 3 redundancy only, disable VSRP and use VRRPE.
- When you configure VSRP, make sure each non-VSRP Dell device has a separate link to each of the VSRP devices.

## Common diagnostic scenarios

- Layer 2 VSRP packets may be mis-labelled and have incorrect checksums.

  This occurs because Dell uses the same packet format for Layer 2 VSRP and Layer 3 VSRP for consistency. The Layer 3 and Layer 4 fields in the packet are not used as they do not have any relevance in Layer 2 forwarding and processing. When an Layer 2 VSRP packet is received by the Dell switch, it is parsed, interpreted as a VSRP packet and processed. The Layer 2 VSRP packet is not a routed IP packet. It is a Layer 2 switched packet which is using ether type of 0x800 (IP). No router can route this packet as the destination MAC of Layer 2 VSRP does not belong to any router's MAC address. So this packet only has relevance within the VLAN.
- VSRP may not work correctly if the host route table is not pointing to the correct default gateway.
- A Layer 2 hitless upgrade causes MRP and VSRP ports to flap.

  VSRP was formerly not supported for hitless software upgrades. The MRP port state change log messages are normal as and are generated by the new active management module, which now defines the roles of each port. This issue occurred because of old software.
- Old software versions.

  Feature issues are often caused because the device is running an old version of the software. Dell recommends that you always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

Chapter

# Layer 3 Protocol Diagnostics 5

This chapter describes the diagnostic commands for PowerConnect B-MLXe Layer 3 protocol environments.

## BFD

BFD quickly detects the failure of a forwarding path by confirming that the next hop router is alive. Without BFD, failure detection can take from 3 to 30 seconds and may cause unacceptable levels of packet loss.

### BFD show commands

#### show bfd

**Syntax: show bfd**

This command displays information about BFD activity, as shown in the following example:

```
PowerConnect#show bfd
 BFD State: ENABLED Version: 1
 Current Registered Protocols: bgp/1 ospf/0  ospf6/0 bgp/0
 All Sessions: Current: 4 Maximum Allowed: 100 Maximum Exceeded Count: 0
 LP Sessions: Maximum Allowed on LP: 40  Maximum Exceeded Count for LPs: 0
   LP Sessions LP Sessions LP Sessions LP Sessions
   1  0        2  2        3  2        4  0
   5  0        6  0        7  0        8  0
   9  0        10 0        11 0        12 0
   13 0        14 0        15 0        16 0
BFD Enabled ports count: 2
 Port       MinTx       MinRx       Mult Sessions
 eth 2/1    100         100         3    2
 pos 3/1    100         100         3    2
```

#### show bfd application

**Syntax: show bfd application**

This command displays information about BFD applications, as shown in this example:

```
PowerConnect#show bfd application
Registered Protocols Count: 3
Protocol  VRFID  Parameter
ospf      0      1
ospf6     0      0
isis_task 0      0
bgp       0      0
```

### show bfd neighbor

**Syntax: show bfd neighbor [interface ethernet <*slotnum*>/<*portum*> | interface pos <*slotnum*>/<*portum*> | interface ve <*port-num*>]**

- **interface ethernet** - Displays BFD neighbor information for the specified Ethernet interface.
- **interface pos** - Displays POS information for a specific interface, slot, and port.
- **interface ve** - Displays BFD neighbor information for the specified virtual interface.

When the **show bfd application** command is enabled, the **show bfd neighbor** command displays output similar to the following:

```
PowerConnect#show bfd neighbor
Total Entries:7 R:RXRemote(Y:Yes/N:No)H:Hop(S:Single/M:Multi)
NeighborAddress   state  Interface  Holddown   Interval   R/H
51.3.0.1          UP     ve 30      600000     200000     Y/S
31.31.31.14       UP     ve 30      25000000   5000000    Y/M
51.4.0.1          UP     ve 40      600000     200000     Y/S
51.1.0.1          UP     ve 10      300000     100000     Y/S
11.11.11.14       UP     ve 10      25000000   5000000    Y/M
```

### show bfd neighbor detail

**Syntax: show bfd neighbor detail [<*IP-address*> | <*IPv6-address*>]**

To display BFD neighbor information in the detailed format, use the following command.

```
PowerConnect#show bfd neighbor detail
Total number of Neighbor entries: 2
NeighborAddress                         State   Interface Holddown  Interval  RH
13.1.1.1                                UP      eth 4/1   1800000   600000    1
  Registered Protocols(Protocol/VRFID): isis_task/0 ospf/0
  Local: Disc: 1, Diag: 0, Demand: 0 Poll: 0
         MinTxInterval: 500000, MinRxInterval: 500000, Multiplier: 3
  Remote: Disc: 684, Diag: 0, Demand: 0 Poll: 0
          MinTxInterval: 600000, MinRxInterval: 600000, Multiplier: 3
  Stats: RX: 4617 TX: 5189 SessionUpCount: 1 at SysUpTime: 0:0:41:49.325
  Session Uptime: 0:0:40:54.400, LastSessionDownTimestamp: 0:0:0:0.0
NeighborAddress                         State   Interface Holddown  Interval  RH
fe80::202:17ff:fe6e:c41d                DOWN    eth 4/1   0         1000000   0
  Registered Protocols(Protocol/VRFID): ospf6/0
  Local: Disc: 2, Diag: 0, Demand: 0 Poll: 0
         MinTxInterval: 500000, MinRxInterval: 500000, Multiplier: 3
  Remote: Disc: 0, Diag: 0, Demand: 0 Poll: 0
          MinTxInterval: 0, MinRxInterval: 0, Multiplier: 0
  Stats: RX: 0 TX: 3278 SessionUpCount: 0 at SysUpTime: 0:0:41:49.325
  Session Uptime: 0:0:0:0.0, LastSessionDownTimestamp: 0:0:0:0.0
```

## Clearing BFD neighbor sessions

You can clear all BFD neighbor sessions or a specified BFD neighbor session using the following command.

### clear bfd neighbor

**Syntax: clear bfd neighbor** [<*IP-address*> | <*IPv6-address*>]

- <*IP-address*> - Specifies the IPv4 address of a neighbor whose BFD session you want to clear.
- <*IPv6-address*> - Specifies the IPv6 address of a neighbor whose BFD session you want to clear.

Executing this command without specifying an IP or IPv6 address clears the sessions of all BFD neighbors.

## BFD debug commands

This section describes how to use Dell diagnostic debug commands to monitor BFD environments.

### debug bfd

**Syntax: [no] debug bfd [application | hitless-upgrade | ipc-error | ipc-event | itc]**

- **application** - Displays information about BFD applications.
- **hitless-upgrade** - Displays information about hitless upgrade events.
- **ipc-error** - Displays information about interprocessing communication (IPC) errors.
- **ipc-event** - Displays information about IPC events.
- **itc** - Displays information about BFD inter-task communication (ITC) activity.

### debug bfd application

**Syntax: [no] debug bfd application**

This command generates information about BFD application activity. When this command is enabled, output resembles the following example, which shows that the OSPF application was disabled, then enabled.

```
PowerConnect# debug bfd application
        application:  debugging is on
PowerConnect# config t
PowerConnect(config)#no router ospf
BFD/APP: Application: AppId: ospf App Subid: 0 DeRegistered with BFD
PowerConnect(config)#router ospf
BFD/APP: Application: AppId: ospf App Subid: 0 Registered with BFD
PowerConnect(config-ospf-router)#show bfd neighbor
Total number of Neighbor entries: 1
NeighborAddress                         State   Interface Holddown  Interval  RH
23.3.3.2                                UP      ve 100    1200000   400000    1
```

### debug bfd hitless-upgrade

**Syntax: [no] debug bfd hitless-upgrade**

This command displays information about BFD hitless upgrade events, as shown in the following example:

```
PowerConnect# debug bfd hitless-upgrade
MP switchover done, clearing all session without graceful restart app.
bfd_mp_delete_all_app_sessions_with_no_graceful_restart() called
BFD: LP Hitless Upgrade started

Resetting LP 1...
Resetting LP 2...
Resetting LP 3...
Resetting LP 4...
BFD/IPC: saving of IPC message during LP Upgrade started
BFD/IPC: Saving of IPC message during LP Upgrade finished.
BFD/IPC: sending saved ipc to LP
bfd_mp_add_all_app_sessions_with_no_graceful_restart called
BFD: LP Hitless Upgrade finished
```

### debug bfd ipc-error

**Syntax: [no] debug bfd ipc-error**

This command generates information about BFD interprocess communication (IPC) errors. The following example shows output from a session with **debug ip ospf bfd**, **debug bfd ipc-event**, and **debug ipc-error** enabled:

```
PowerConnect#debug ip ospf bfd
         OSPF:  BFD events debugging is on
PowerConnect#debug bfd itc
        itc:  debugging is on
PowerConnect#debug bfd ipc-event
        ipc-event:  debugging is on
PowerConnect#show bfd neighbor
Dec 19 14:51:37 BFD/IPC: Sending MP Request for all sessions information to LP 2.
Dec 19 14:51:37 BFD/IPC: Received LP Response for all sessions information from LP
2.
Total number of Neighbor entries: 2
NeighborAddress                         State   Interface Holddown  Interval  RH
12.2.2.2                                UP      eth 2/2   300000    100000    1
fe80::20c:dbff:fee2:b529                UP      eth 2/2   300000    100000    1
PowerConnect#clear ip ospf nei 12.2.2.2
SYSLOG: Dec 19 14:51:58:<13>R1, OSPF: nbr state changed, rid 1.1.1.1, nbr addr
12.2.2.2, nbr rid 2.2.2.2, state down
SYSLOG: Dec 19 14:51:58:<13>R1, OSPF: interface state changed, rid 1.1.1.1, intf
addr 12.2.2.1, state designated router
BFD/ITC: Received Delete Session Request from App:ospf for Port:eth
2/2 Neighbor:12.2.2.2
BFD: ipc set session admin down(0->2), for session 22
BFD/IPC: Received session parameter change for session=22
BFD: ipc delete session (0->2), for session 22
BFD/ITC: Received Create Session Request from App:ospf for Port:eth 2/2
Neighbor:12.2.2.2
BFD: ipc create session (0->2), for session 76
SYSLOG: Dec 19 14:52:01:<13>R1, OSPF: interface state changed, rid 1.1.1.1, intf
addr 12.2.2.1, state backup designated router
BFD/IPC: Received session parameter change for session=76
BFD/IPC: Received session state change notification for session=76
OSPF: ITC Session State Change Notification rxd for nbr 12.2.2.2
SYSLOG: Dec 19 14:52:01:<13>R1, OSPF: nbr state changed, rid 1.1.1.1, nbr addr
12.2.2.2, nbr rid 2.2.2.2, state full
PowerConnect#show bfd neighbor
BFD/IPC: Sending MP Request for all sessions information to LP 2.
BFD/IPC: Received LP Response for all sessions information from LP 2.
Total number of Neighbor entries: 2
NeighborAddress                         State   Interface Holddown  Interval  RH
fe80::20c:dbff:fee2:b529                UP      eth 2/2   300000    100000    1
12.2.2.2                                UP      eth 2/2   300000    100000    1
```

### debug ip ospf bfd, debug bfd ipc-event, debug bfd itc

**Syntax: [no] debug ip ospf bfd**

**Syntax: [no] debug bfd ipc-event**

**Syntax: [no] debug bfd itc-event**

These commands generate information about IPC and ITC errors. The previous example shows output from a session with these three commands enabled.

**debug bfd itc**

**Syntax: [no] debug bfd itc**

This command displays information about BFD ITC activity. Output resembles that displayed in the example for **debug ip bfd ipc-error**, and is similar to that shown here:

```
PowerConnect#debug bfd itc
        itc:  debugging is on
PowerConnect#show bfd neighbor
BFD/ITC: Received Delete Session Request from App:ospf for Port:eth 2/2
Neighbor:12.2.2.2
BFD/ITC: Received Create Session Request from App:ospf for Port:eth 2/2
Neighbor:12.2.2.2
OSPF: ITC Session State Change Notification rxd for nbr 12.2.2.2
```

## Configuration notes

- BFD session establishment on an interface doesn't start until 90 seconds after the interface comes up. The reason for this delay is to ensure that the link isn't effected by unstable link conditions which could cause BFD to flap. This delay time is not user configurable.
- BFD supports multi-slot trunks in cases where all BFD packets are transmitted only on a single path, which does not change unless the trunk active membership changes. BFD is not be supported on multi-slot trunks where per-packet switching is used (where the path taken by the BFD packets varies).
- The BFD Control Message is a UDP message with destination port 3784.
- When you configure BFD you must set timing and interval parameters.

## Common diagnostic scenarios

- It takes a few minutes for BFD to come up after a partner link comes up over Layer 2.

  This is a expected behavior. The Dell device waits for 90 seconds after a port state changes from down to up before sending the BFD packet out.

- When configuring BFD, one end detects a session but the other end does not.

  The **ip ospf bfd** command must be enabled at the interface level or the **bfd all-interfaces** command must be enabled at the router ospf level.

- Old software versions

  Feature issues are often be caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# BGP

BGP (version 4) is the standard Exterior Gateway Protocol (EGP) used on the Internet to route traffic between Autonomous Systems (AS). BGP maintains a routing table (separate from the main router routing table) of the accessible routes and addresses of AS neighbors.

## BGP show commands

The following sections describe the **show** and **show debug** commands that display BGP information.

### show debug

**Syntax: show debug**

This command can be used at any time to display debug settings for a device. The following example shows debug settings for a device that has only BGP debug settings enabled:

```
PowerConnect#show debug
Debug message destination: console
IP Routing
       BGP:bgp debugging is on
       BGP:events debugging is on
       BGP:keepalives debugging is on
       BGP(RED):bgp debugging is on
       BGP(RED):events debugging is on
       BGP(RED):keepalives debugging is on
```

### show ip bgp debug

**Syntax: show ip bgp debug [memory | network | nexthop | profiling | reset-profiling | route-table | variables |out-policy]**

This command displays information about BGP network configurations, including next hops, available memory, profile data, and internal variables. For more specific output, use one of the variables in the syntax line.

- **memory** - Displays BGP memory pool.
- **network** - Displays information about BGP networks.
- **nexthop** - Displays BGP next-hop configurations.
- **profiling** - Collects profile data on BGP functions.
- **reset-profiling** - Resets the profile data collection setting.
- **route-table** - Displays routing table information.
- **variables** -Displays BGP internal variables.
- **out-policy** - Displays out bound peer policies.
- Output modifiers (search filters).

Output from the **show ip bgp debug** command resembles the following:

```
PowerConnect#show ip bgp debug
BGP Debug Information
Pid Size  Address   Total   Used    Free    NoMem  Errors  #_pools p_unit
0   8     021da799  0       0       0       0      0       0       2000
1   16    021da7c5  0       0       0       0      0       0       2000
2   24    021da7f1  0       0       0       0      0       0       2000
3   32    021da81d  910     2       908     0      0       1       800
4   48    021da849  630     1       629     0      0       1       400
5   64    021da875  0       0       0       0      0       0       200
6   96    021da8a1  0       0       0       0      0       0       80
7   128   021da8cd  0       0       0       0      0       0       40
8   256   021da8f9  31      2       29      0      0       1       20
9   48    021da925  5041    5       5036    0      0       1       4000
10  44    021da951  10666   5       10661   0      0       1       8000
11  86    021da97d  5688    7       5681    0      0       1       4000
12  92    021da9a9  5333    4       5329    0      0       1       4000
Total Memory Use for Route and Attributes Tables   : 1871568
  Memory Block Not Available Count   : 0
  Bad Memory Pool ID Count           : 0
  TCP buffers                        : 2048 0 0 0
  BGP Tx Parameters                 : 5 30 3
  BGP route update time counter     : sched: N 0 (100)
  BGP route update count            : 0 (0) last:
                 event              : (1:1) 0.0.0.0/0
  BGP io semaphore take 5, yield 1, 0
  Max timer process: l-0 s-0 (0), io: 0 0 us
  io_rx_yield_time 0x021d9480, 2
  1 sec timer value: 0, 0 TB
  MP active: 1, standby up 0
  Graceful_restart: enable 0, restart time 120, stale-routes 360, purge 600
    Restarted 0, fwd 0, restart_up_time_count[0] 0

BGP inbound/outbound policy caching Enabled

BGP internal debug trace flags:
bgp class data structure is clean
```

### show ip bgp debug memory

**Syntax: show ip bgp debug memory [check-as-path | check-free | dump-used]**

- **check-as-path** - Check AS paths
- **check-free** - Check free pool entry
- **dump-used** - Dump used pool entry

This command displays information about the internal sizes of various BGP entities. It shows a summary of how many chunks were allocated and freed for each pool, as the example that follows illustrates. The pool number from this output can be provided as the <*pool-id*> argument to the **show ip bgp debug dump** command.

```
PowerConnect#show ip bgp debug memory
BGP_CLASS_VRF: 26367, BGP_PEER_CLASS: 179124, BGP_CONFIGURATION_CLASS: 4135
BGP_AS_PATH_ENTRY: 94, BGP_IPV6_AS_PATH_ENTRY: 106, BGP_AS_PATH_SEGMENTS: 18
BGP_NLRI_ENTRY: 86, BGP_PATRICIA_KEY_ADDRESS: 16, BGP_PATRICIA_NODE: 48
BGP_RIB_OUT_NLRI_ENTRY: 44, BGP_RIB_OUT_HOLDER: 28, BGP_WITHDRAWN_ROUTE_ENTRY: 42
BGP_NEXTHOP_ADDRESS: 16, BGP_NEXTHOP_ENTRY: 199, BGP_DAMPING_NLRI_ENTRY: 36
BGP_ROUTE_DAMPING_REUSE_LIST: 2052, BGP_ROUTE_DAMPING_BLOCK: 3522,
BGP_DAMPENING:37374
BGP pool 0 allocated 1 chunk freed 0 chunk
BGP pool 1 allocated 0 chunk freed 0 chunk
```

```
BGP pool 2 allocated 0 chunk freed 0 chunk
BGP pool 3 allocated 1 chunk freed 0 chunk
BGP pool 4 allocated 1 chunk freed 0 chunk
BGP pool 5 allocated 1 chunk freed 0 chunk
BGP pool 6 allocated 0 chunk freed 0 chunk
BGP pool 7 allocated 0 chunk freed 0 chunk
BGP pool 8 allocated 0 chunk freed 0 chunk
BGP pool 9 allocated 1 chunk freed 0 chunk
BGP pool 10 allocated 1 chunk freed 0 chunk
BGP pool 11 allocated 1 chunk freed 0 chunk
BGP pool 12 allocated 1 chunk freed 0 chunk
BGP pool 13 allocated 0 chunk freed 0 chunk
BGP pool 14 allocated 0 chunk freed 0 chunk
BGP pool 15 allocated 1 chunk freed 0 chunk
BGP pool 16 allocated 0 chunk freed 0 chunk
BGP pool 17 allocated 0 chunk freed 0 chunk
```

**show ip bgp debug memory check-free**

**Syntax: show ip bgp debug memory check-free <***pool id number***>**

This command shows the number of free entries in the memory pool. Output is similar to the following (in this case, memory pool 1):

```
PowerConnect#show ip bgp debug memory check-free 1
Number of free entry in the pool(1) = 0, expected 0
```

**show ip bgp debug memory dump**

**Syntax: show ip bgp debug memory dump <***pool-id***>**

This command displays (in page mode) all memory chunks that are held by a pool (either fully used or still with free entries), as the following example illustrates:

```
PowerConnect#show ip bgp debug memory dump 10
     Pool 10: memory chunk that had free entry:
chunk addr: 2827c000, total entry 2520, free 2405, used 115
     Pool 10: memory chunk that had all entries in use:
```

The pool ID can be obtained from the output of the **show ip bgp debug memory** command.

**show ip bgp debug memory dump-used**

**Syntax: show ip bgp debug memory dump-used <***pool id number***>**

This command displays pool ID numbers that have been deleted.

**show ip bgp debug network**

**Syntax: show ip bgp debug network [***X:X::X:X* **|** *A.B.C.D/A.B.C.D/L***]**

- *X:X::X:X* - IPv6 network address
- *A.B.C.D or A.B.C.D/L* - IP network address

This command displays internal BGP information about network command-related data structures. Output resembles the following (shown for an IP network address):

```
PowerConnect#show ip bgp debug network 2.2.2.2/32
BGP: network 2.2.2.2/32 found
(x05236ebc, 00000000, 00000000) 2.2.2.2/32
   weight:32768 back_door:0 imported:0
   route-map:<>  sptr:x00000000
   next_hop:0.0.0.0 med:0 type:0
```

### show ip bgp debug nexthop

**Syntax: show ip bgp debug nexthop** [*X:X::X:X* | *A.B.C.D or A.B.C.D/L* | *unreachable*]

- *X:X::X:X* - IPv6 network address
- *A.B.C.D or A.B.C.D/L* - IP network address
- *unreachable* - Unreachable nexthops

To see nexthop debug information for a specific IP address, enter:

```
PowerConnect#show ip bgp debug nexthop 60.0.0.2
60.0.0.2: (addr:62 0x25aca004) path:1, nh:60.0.0.2 1/1(0),type:0, sub:0,0,resolve
schema: 0
```

To display information about unreachable nexthops, enter the following command.

```
PowerConnect#show ip bgp debug nexthop unreachable
NEXT_HOP list[0]: 0x09a58900,
```

### show ip bgp debug profiling

**Syntax: show ip bgp debug profiling**

This command displays BGP profiling information as shown in the following example:

```
PowerConnect#show ip bgp debug profiling
BGP Profiling Data:
Send                            TOTAL=0                 FREQ=0          AVG=0
SendRibOut                      TOTAL=0                 FREQ=0          AVG=0
SendRibOutAddAS                 TOTAL=0                 FREQ=0          AVG=0
SendRibOutAddNLRI               TOTAL=0                 FREQ=0          AVG=0
Recv                            TOTAL=0                 FREQ=0          AVG=0
RecvASAlloc                     TOTAL=0                 FREQ=0          AVG=0
RecvUpd                         TOTAL=0                 FREQ=0          AVG=0
RecvAschk                       TOTAL=0                 FREQ=0          AVG=0
RecvAsloop                      TOTAL=0                 FREQ=0          AVG=0
RecvAsadd1                      TOTAL=0                 FREQ=0          AVG=0
RecvAsadd2                      TOTAL=0                 FREQ=0          AVG=0
RecvUpdnlri                     TOTAL=0                 FREQ=0          AVG=0
RecvUpdvpnv4nlri                TOTAL=0                 FREQ=0          AVG=0
RecvVpnv4copy                   TOTAL=0                 FREQ=0          AVG=0
RecvVpnv4alloc                  TOTAL=0                 FREQ=0          AVG=0
RecvVpnv4ribin                  TOTAL=0                 FREQ=0          AVG=0
RecvVpnv4process                TOTAL=0                 FREQ=0          AVG=0
RecvVpnv4processImp             TOTAL=0                 FREQ=0          AVG=0
RecvVpnv4processnewas           TOTAL=0                 FREQ=0          AVG=0
RecvVpnv4processasadd           TOTAL=0                 FREQ=0          AVG=0
RecvRibinTree2                  TOTAL=0                 FREQ=0          AVG=0
RecvRibinTree3                  TOTAL=0                 FREQ=0          AVG=0
RecvRibinTree4                  TOTAL=0                 FREQ=0          AVG=0
AllocAPool1                     TOTAL=0                 FREQ=0          AVG=0
AllocAPool2                     TOTAL=0                 FREQ=0          AVG=0
AllocPoolDymalloc               TOTAL=219               FREQ=3          AVG=73
AllocAPoolChain                 TOTAL=5592              FREQ=3          AVG=1864
vpnv4_ribout_add_nlri            TOTAL=0                FREQ=0          AVG=0
vpnv4_ribout_allocate_label      TOTAL=0                FREQ=0          AVG=0
vpnv4_ribout_run_policy          TOTAL=0                FREQ=0          AVG=0
vpnv4_ribout_rem_withd_rt        TOTAL=0                FREQ=0          AVG=0
rp_match_total                  TOTAL=0                 FREQ=0          AVG=0
rp_set_total                    TOTAL=0                 FREQ=0          AVG=0
rp_match_access                 TOTAL=0                 FREQ=0          AVG=0
bgp_check_update                TOTAL=0                 FREQ=0          AVG=0
Update_Chk_Nexthop              TOTAL=0                 FREQ=0          AVG=0
Update_Add_Routes               TOTAL=0                 FREQ=0          AVG=0
Update_Tail                     TOTAL=0                 FREQ=0          AVG=0
Chk_NextHop_Change_Def          TOTAL=0                 FREQ=0          AVG=0
Chk_NextHop_Change_Hash         TOTAL=0                 FREQ=0          AVG=0
Chk_NextHop_Change_Hash         TOTAL=0                 FREQ=0          AVG=0
Chk_NextHop_Change_Lookup       TOTAL=0                 FREQ=0          AVG=0
Chk_NextHop_Change_Process      TOTAL=0                 FREQ=0          AVG=0
Revert_Idle_State_Delete_All    TOTAL=0                 FREQ=0          AVG=0
Revert_Idle_State_Tail          TOTAL=0                 FREQ=0          AVG=0
RIB_in_delete_all_nlris_from_p   TOTAL=0                FREQ=0          AVG=0
Timer_Add_Routes                TOTAL=0                 FREQ=0          AVG=0
Timer_Delete_All_Nlri           TOTAL=0                 FREQ=0          AVG=0
Add_routes                      TOTAL=0                 FREQ=0          AVG=0
Add_routes_cbk_nlri_list        TOTAL=0                 FREQ=0          AVG=0
Add_routes_cbk_update_ip        TOTAL=0                 FREQ=0          AVG=0
check_and_update_bgp_route      TOTAL=0                 FREQ=0          AVG=0
```

### show ip bgp debug route-table

**Syntax: show ip bgp debug route-table**

This command displays the Network Layer Reachability Information (NLRI) count in the BGP route-table, as shown in this example:

```
PowerConnect#show ip bgp debug route-table
There are 7 NLRIs in BGP Route Table, time 0 ms
```

### show ip bgp debug variables

**Syntax: show ip bgp debug variables**

This command displays detailed BGP information about internal flags, statistics, and states. Output resembles the following:

```
PowerConnect#show ip bgp debug variables
safi:0, &bgp:04d2dfdc, enabled:1, operational:1, dbg_mem=&021da72e/0, curr_afi:0
io_process_running:0, io_process_next_peer_number=1
in_long_loops 0, clear_all 0, timer 00000000, count 0
timer_enabled:1, timer_next_peer_number:0, 1s timer 1, short timer 1
scheduler id:1:1, ip:0.0.0.0/0, time=16977
bgp_tcb:021cae08 (0x0001001f, 0), tick_cnt=17, seconds=691
bgp_tcb6:0001002f (0x00000000, 0x04d2e0c4)
*peer:021d9546, *peer_group:021da4f2, RIB_in_root_node:0bb373b4
Maximum Peer Index Number:2, check_nexthops:0 0
router_id:3.3.3.3, configured:0, cluster_id:0.0.0.0, configured:0
route_is_router_reflector:0, client_to_client_reflection:1
networks:x021cdbe9, aggregate:x021cdc0d
default_metric:4294967294, local_preference:100, keep_alive:60, hold_time:180
originate_default:0, originated:0
distance:20 200 200, fast_external_fallover=0
nexthop recur0, en_def:0, readvertise:1, auto_sum:0, synch:0
always_compare_med:0, compare_med_with_empty_aspath: 0, redistribute_ibgp:0,
local_network_check_time_count:1
nexthop_cache_hit_count:6, nexthop_cache_miss_count:2
system memory:536870912, total_allocated:1964468, bgp_defined_quota:2147483648
import map:"", export map:""
nexthop_lb_interface:<no-such-port>, nexthop_lb_addr:0.0.0.0
```

### show ip bgp debug out-policy

**Syntax: show ip bgp debug out-policy**

This command displays out bound peer policies. The output resembles the following:

```
PowerConnect#show ip bgp debug out-policy
BGP(vrf 0/safi 0) outbound policy entries: 13
Outbound Policy Group: 0x34731a00 (Hash 0), ID: 1, Drop 1, Use Count: 0, Staring:
0, Update: 0
Ribout Group: 0x34831000, ID: 1, Type: -1, Peer Count: 0, Mask: 0x00000000 (0),
ribout: 0, withdrawn: 0
Outbound Policy Group: 0x347a5600 (Hash 0), ID: 9, Drop 0, Use Count: 1, Staring:
42, Update: 0
Ribout Group: 0x35563000, ID: 2, Type: 1, Peer Count: 1, Mask: 0x00000004 (2),
ribout: 2102, withdrawn: 0
Outbound Policy Group: 0x355e9e00 (Hash 31), ID: 5, Drop 0, Use Count: 6, Staring:
18, Update: 0
Ribout Group: 0x35615000, ID: 6, Type: 2, Peer Count: 6, Mask: 0x0000003f (5),
ribout: 175020, withdrawn: 0
routemap: map10
```

```
Outbound Policy Group: 0x355e9200 (Hash 110), ID: 2, Drop 0, Use Count: 6,
Staring: 0, Update: 0
Ribout Group: 0x355f1000, ID: 3, Type: 2, Peer Count: 6, Mask: 0x0000003f (5),
ribout: 175020, withdrawn: 0
routemap: map0
Outbound Policy Group: 0x347a5400 (Hash 111), ID: 8, Drop 0, Use Count: 6,
Staring: 36, Update: 0
Ribout Group: 0x35627000, ID: 9, Type: 2, Peer Count: 6, Mask: 0x0000003f (5),
ribout: 175020, withdrawn: 0
routemap: map1
```

### show ip bgp debug out-policy peer-list

**Syntax: show ip bgp debug out-policy peer-list**

This command displays the peer grouping. The NetIron device groups BGP peers together based on their outbound policies. To reduce RIB-out memory usage, the device then groups the peers within an outbound policy group according to their RIB-out routes. Peers in a group have the same Policy-ID and Group-ID

RIB-out peer grouping is not shared between different VRFs or address families, and is not supported for VPNV4 or Layer 2 VPN peers.

```
PowerConnect# show ip bgp debug out-policy peer-list(ipv4 unicast)
BGP sorted peers on outbound policy (safi=0)
Index ->   PeerIdx    Peer Address  Policy-ID   Group-ID Vrf-idx
0          0          100.0.100.2   2           3        0
1          1          100.0.101.2   2           3        0
2          2          100.0.102.2   2           3        0
3          3          100.0.103.2   2           3        0
4          4          100.0.104.2   2           3        0
5          5          100.0.105.2   2           3        0
6          18         102.0.100.2   3           4        0
7          19         102.0.101.2   3           4        0
8          20         102.0.102.2   3           4        0
.....
```

## BGP debug commands

### debug ip bgp

**Syntax: [no] debug ip bgp [all-vrfs | *<vrf-name>* ] [dampening | event | general | graceful-restart | keepalives | updates [rx | tx] | route-selection]**

- **all-vrfs** - Displays information for all virtual routing and forwarding events.
- <vrf-name> - Displays information for a specific VRF event.
- **dampening** - Displays BGP dampening activity.
- **event** - Displays BGP event information.
- **general** - Displays BGP common debugs.
- **graceful-restart** - Displays information about graceful-restart events.
- **keepalives** - Displays keepalive activity.
- **updates** [**rx** | **tx**] - Displays BGP rx, tx, or rx and tx update messages about debug processing.
- **route-selection** - Displays BGP route selection debug information.

This command enables common BGP debugs to be displayed for all VRFs or for a specific VRF. Output resembles the following example.

```
/**** local-as of peer has changed ****/
PowerConnect# BGP: 10.1.1.2 Rcv TCP connection closed remotely. handle
00000005:0a0132d4, key 0
BGP: 10.1.1.2 remote peer closed TCP connection
BGP: 10.1.1.2 rcv notification: CEASE Message
BGP: 10.1.1.2 BGP connection closed

PowerConnect# BGP: 10.1.1.2 start peer
BGP: 10.1.1.2 Init TCP Connection to peer, local IP 10.1.1.1
BGP: 10.1.1.2 Rcv TCP connection closed remotely. handle 0000000 6:0a0132d4, key 0
BGP: 10.1.1.2 TCP connection failed
BGP: Rcv incoming TCP connection check. handle 00000007:0a0123d4, key 0
BGP: Incoming TCP connection. peer 10.1.1.2 OKed
BGP: Rcv incoming TCP connection UP. handle 00000007:0a0123d4, key 0
BGP: 10.1.1.2 New incoming TCP connection is open, local IP 10.1.1.1
BGP: 10.1.1.2 sending MultiProcotol cap, afi/safi=1/1, length 4
BGP: 10.1.1.2 sending IPEN, holdTime=180 route_refresh=1 cooperative= 1, restart
0/0
BGP: 10.1.1.2 rcv OPEN w/Option parameter length 16, as 2. hold_time 180
BGP: 10.1.1.2 rcv capability 2, len 0
BGP: 10.1.1.2 rcv capability 128, len 0
```

### debug ip bgp route-selection

**Syntax: [no]debug ip bgp [all-vrfs | <*vrf-name*>] route-selection**

This command enables debugging of BGP route selection, for all VRFs, or for a specified VRF. Output resembles the following:

```
PowerConnect# debug ip bgp route-selection
BGP: Clearing install flags for 104:1:0:61::/64
BGP: 21:1::20 Move path to front for 104:1:0:51::/64
BGP: select best route 104:1:0:61::/64 load_share (ibgp 1, ip 1), (ebgp 1, ip 1)
BGP: eligible route 1
BGP: 21:1::20 Best path up 104:1:0:61::/64, install
BGP: 21:1::20 Move path to front for 104:1:0:61::/64
BGP: add bgp routes to IPv6 table 104:1:0:61::/64
BGP: Adding 104:1:0:61::/64 to ipv6 route table, next_hop=21:1::20
BGP: Clearing install flags for 104:1:0:61::/64
BGP: 21:1::20 Move path to front for 104:1:0:51::/64
BGP: select best route 104:1:0:61::/64 load_share (ibgp 1, ip 1), (ebgp 1, ip 1)
BGP: eligible route 1
BGP: 21:1::20 Best path up 104:1:0:61::/64, install
BGP: 21:1::20 Move path to front for 104:1:0:61::/64
BGP: add bgp routes to IPv6 table 104:1:0:61::/64
BGP: Adding 104:1:0:61::/64 to ipv6 route table, next_hop=21:1::20
```

### debug ip bgp dampening

**Syntax: [no] debug ip bgp [all-vrfs |** <*vrf name*>**] dampening**

This command displays information about dampening process configurations, route penalties, durations, restraint and release. Output resembles the following:

```
PowerConnect# debug ip bgp dampening
BGP: dampening debugging is on
BGP: 21:1::20 dampening 100:1::/64 down, penalty=9154 flaps=572
BGP: 21:1::20 dampening 100:1::/64 up, penalty=8640 suppressed
```

In this example, dampening is enforced on route 100:1::/64 due to flap activity, and lifted after a period of time.

### debug ip bgp events

**Syntax: [no] debug ip bgp [all-vrfs |<*vrf-name*>] events**

This command generates information about BGP events, such as connection attempts and keepalive timer activity, as shown in the following example.

```
PowerConnect#debug ip bgp events
   BGP: events debugging is on
BGP: From Peer 192.168.1.2 received Long AS_PATH=
AS_CONFED_SET(4) 1 2 3 AS_CONFED_SEQUENCE(3) 4 AS_SET(1)
5 6 7 AS_SEQ(2) 8 9 attribute length (9) More than configured MAXASLIMIT 7

PowerConnect#clear ip bgp nei all
   BGP: clear all VRF neighbors
   BGP: 10.1.1.2 sending NOTIFICATION Cease (CEASE Message)
   BGP: 10.1.1.2 reset, BGP notification Cease sent
   BGP: 10.1.1.2 sending NOTIFICATION Cease (Administrative Reset)
   BGP: 10.1.1.2 reset, BGP notification Cease sent
   BGP: 10.1.1.2 Peer went to IDLE state (User Reset Peer Session)
   BGP: 10.1.1.2 Peer already in IDLE state, stays in IDLE state.
   BGP: 10.1.1.2 Peer went to ESTABLISHED state
```

The following line of output indicates a four-byte ASN.

```
PowerConnect#Sep  9 18:36:42 BGP: 192.168.1.1 rcv capability 65, len 4
```

### debug ip bgp graceful-restart

**Syntax: [no] debug ip bgp [all-vrfs | <*vrf-name*>] graceful-restart**

Enable this command to receive information about BGP graceful restarts. The graceful restart feature minimizes disruptions in forwarding and route flapping when a router experiences a restart.

### debug ip bgp keepalives

**Syntax: [no] debug ip bgp [all-vrfs |** <*vrf-name*>] **keepalives**

NetIron routers use keepalives to collect information about applications and services. For example, you can configure a keepalive to continually monitor and report on the online status of a resource, such as BGP. Output resembles the following example.

```
PowerConnect#debug ip bgp keepalives
BGP:  keepalives debugging is on
BGP: 69.28.156.234 KEEPALIVE received
BGP: 68.142.72.222 KEEPALIVE received
BGP: 69.28.148.234 KEEPALIVE received
BGP: 68.142.72.222 sending KEEPALIVE
BGP: 69.28.156.234 sending KEEPALIVE
```

### debug ip bgp neighbor

**Syntax: [no] debug ip bgp [all-vrfs |** <*vrf name*>] **neighbor [**<*ipv4-address*>**|**<*ipv6-address*>]

This command specifies the IPv4 or IPv6 neighbor filter for BGP debugging for all VRFs, or for a specified VRF. Only one IPv4 neighbor filter and one IPv6 neighbor filter can be configured on each VRF.

The neighbor filter acts exclusively, which means that wherever an IPv4 neighbor filter is configured, the corresponding IPv6 debugs are not shown. The neighbor filter functions with the **update rx|tx**, **keepalives**, **events,** and **graceful-restart** debug commands. For example, when **debug ip bgp updates** and **debug ip bgp neighbor 21:1::20** are configured, BGP update debugs are displayed only for neighbor 21:1::20.

Output resembles the following:

```
PowerConnect#debug ip bgp neighbor 21:1::20
   BGP:  neighbor 21:1::20 debugging is on
PowerConnect#debug ip bgp updates rx
   BGP:  updates RX debugging is on
PowerConnect#
BGP: 21:1::20 rcv UPDATE 100:1::/64 -- withdrawn
BGP: rcv UPDATE w/attr: Origin=IGP AS_PATH= AS_SEQ(2) 65400 65181 209 7018
NextHop=21:1::20
```

**debug ip bgp ip-prefix**

**Syntax: [no]debug ip bgp [all-vrfs |**<*vrf name*>**] ip-prefix** <*ip prefix/mask*>

**debug ip bgp ipv6-prefix**

**Syntax: [no]debug ip bgp [all-vrfs |**<*vrf name*>**] ipv6-prefix <***ipv6 prefix/mask***>**

**debug ip bgp ip-prefix-list**

**Syntax: [no]debug ip bgp [all-vrfs | <***vrf name***>] ip-prefix-list <***name***>**

**debug ip bgp ipv6-prefix-list**

**Syntax: [no]debug ip bgp [all-vrfs | <***vrf name***>] ipv6-prefix-list <***name***>**

These commands specify the IPv4 or IPv6 prefix filter for BGP debugging information, for all VRFs, or for a specified VRF. Only one IPv4 prefix filter or prefix list and one IPv6 prefix filter or prefix list can be configured on a VRF. Prefix filters and prefix lists cannot be configured simultaneously. IPv6 and IPv4 filters are applied separately. Configuring an IPv4 prefix-list or prefix-filter does not automatically block IPv6 debugging, unlike the neighbor filter. To block IPv6 debugging, AFI/SAFI should be configured along with prefix filter.

**NOTE**
Prefix filtering is not functional during UPDATE RX processing for VPNV4. Prefix filtering is functional with the **update rx/tx**, **route-selection,** and **dampening debug** commands.

**NOTE**
Removing the configured **ip-prefix-list** or **ipv6-prefix-list** also automatically removes the associated filter.

Output from any of these commands resembles the following, where the first example reflects an **ip-prefix**, or a similar **ip-prefix-list**, along with **updates rx**, and AFI/SAFI as IPv4/Unicast:

```
PowerConnect#debug ip bgp ip-prefix 20.1.1.1/24
         BGP:  ip-prefix debugging is on
                         permit 20.1.1.0/24

/**** Route-Addition ****/
PowerConnect#Sep  9 18:38:13 BGP: BGP rcv UPDATE w/attr: Origin=IGP AS_PATH=
AS_SEQ(2) 2 NextHop=10.1.1.2 MED=0
Sep  9 18:38:13 BGP: (0): 10.1.1.2 rcv UPDATE 20.1.1.0/24

/**** Route-Deletion ****/
PowerConnect #Sep  9 18:38:24 BGP: 10.1.1.2 rcv UPDATE 20.1.1.0/24 - withdrawn
```

The next example reflects an **ipv6-prefix** or a similar **ipv6-prefix-list**, along with **updates rx** and AFI/SAFI as IPv6/Unicast:

```
PowerConnect#debug ip bgp ipv6-prefix 200::1/64
         BGP:  ipv6-prefix debugging is on
                         permit 200::/64

/**** Route-Addition ****/
Sep  9 24:01:32 BGP: rcv UPDATE w/attr: Origin=IGP AS_PATH= AS_SEQ(2) 65400 65181
209 7018 NextHop=21:1::20
Sep  9 24:01:32 BGP: (2): 21:1::20 rcv UPDATE 200:1::/64

/**** Route-Deletion ****/
Sep  9 24:01:40 BGP: 21:1::20 rcv UPDATE 200:1::/64 -- withdrawn
```

### debug ip bgp route-map

**Syntax: [no] debug ip bgp [all-vrfs |<*vrf name*>] route-map <*map-name*>**

This command associates an existing route-map filter with BGP debugging. The route-map filter is functional with the **route-selection** and **dampening** debug commands.

**CAUTION**

**This command may degrade performance.**

**NOTE**

Removing the configured route-map automatically removes the associated filter.

**NOTE**

To debug the entire UPDATE RX process until route selection and route updates correspond to the UPDATE, you must enable the **debug updates rx**, and **debug route-selection** commands along with the appropriate filters. In this case, all of the filters (NBR+AFI/SAFI+PREFIX+ROUTE_MAP) are applied, if they are configured.

### debug ip bgp route-updates

**Syntax: [no] debug ip bgp route-updates**

This command shows the routes that have been shared with a neighbor.

The route information includes the four-byte AS4_PATH attribute and the AS_PATH attribute.

An example of the output follows.

```
PowerConnect#debug ip bgp route-updates
Sep  9 18:41:59 BGP: BGP: 192.168.1.1 rcv UPDATE w/attr: Origin=INCOMP AS_PATH=
AS_SEQ(2) 90000 70001 70002 70003 75000 NextHop=192.168.1.5
Sep  9 18:41:59 BGP: BGP: 192.168.1.1 rcv UPDATE w/attr: Origin=INCOMP AS4_PATH=
AS_SEQ(2) 90000 70001 70002 70003 75000 NextHop=192.168.1.5
```

### debug ip bgp updates

**Syntax: [no] debug ip bgp [all-vrfs | <*vrf-name*>] updates [rx | tx]**

This command enables debugging of BGP update message processing for all VRFs, or for a specific VRF. Update debugging is supported in both IPv4 and IPv6 update message processing. If you do not specify either all VRFs, or a specific VRF, the **debug ip bgp updates** command enables debugging for both rx and tx update message processing.

Output is similar to the following:

```
PowerConnect# debug ip bgp updates
PowerConnect#Sep 9 18:38:13 BGP: BGP rcv UPDATE w/attr: Origin=IGP AS_PATH=
AS_SEQ(2) 2 NextHop=10.1.1.2 MED=0
BGP: (0): 10.1.1.2 rcv UPDATE 20.1.1.0/24
BGP: 10.1.1.2 rcv UPDATE 20.1.1.0/24 - withdrawn
BGP: rcv UPDATE w/attr: Origin=IGP AS_PATH= AS_SEQ(2) 65400 65181 209 7018
NextHop=21:1::20
BGP: (2): 21:1::20 rcv UPDATE 200:1::/64
BGP: 21:1::20 rcv UPDATE 200:1::/64 -- withdrawn
```

### *BFD for BGP4 debug commands*

To debug BFD for BGP4, turn on the following debugging commands:

- **debug ip bgp** [**all-vrfs** | **vrf** <*vrf-name*>] **bfd**
- **debug bfd application**
- **debug bfd itc**

Sample output:

```
PowerConnect# Sep 9 18:37:07 BFD:ITC, Received BFD MHOP ITC Create S
ession Request from bgp(0)
Sep 9 18:37:07 BFD: BFD MHOP ITC Create Session Response Sent to bgp(0)
Sep 9 18:37:07 BGP: 6.1.1.2 Peer Received BFD MHOP Create Session Respo
nse ITC message
Sep 9 18:37:07 BFD: BFD MHOP ITC Update Session Negotiated Parameters Request S
ent to bgp(0)
Sep 9 18:37:07 BGP: 6.1.1.2 Peer Received BFD MHOP BFD Session State Ch
ange Notify ITC message
Sep 9 18:37:07 BGP: 6.1.1.2 Peer Received BFD session UP state notification
Sep 9 18:37:07 BGP: 6.1.1.2 Peer BGP-BFD state changed to UP
Sep 9 18:37:07 BGP: 6.1.1.2 Peer Received BFD MHOP Update Session Negot
Sep 9 18:37:14 BFD:ITC, Received BFD MHOP ITC Route Change Indication from bgp(0)
```

## IPv6 ND6 debug commands

**debug ipv6 nd**

**Syntax:** [no] **debug ipv6 nd**

This command enables debugging for all neighbors. To enable debugging for a specific neighbor, enter the **debug ipv6 nd** command followed by the **debug ipv6 address** <*X.X::X.X*> command.

## IPv6 OSPF debug commands

**debug ipv6 ospf**

**Syntax:** [no] **debug ipv6 ospf** [**bfd** | **ism** | **ism-events** | **ism-status** | **lsa** | **lsa-flooding** | **lsa-generation** | **lsa-install** | **lsa-maxage** | **lsa-refresh** | **nsm** | **nsm-events** | **nsm-status** | **packet** | **packet-dd** | **packet-hello** | **packet-lsa-ack** | **packet-lsa-req** | **packet-lsa-update** | **route** | **route-calc-external** | **route-calc-inter-area** | **route-calc-intra-area** | **route-calc-spf** | **route-calc-transit** | **route-install** | **virtual-link**]

- **bfd** - Displays information about OSPFv3 BFD events.
- **ism** - Displays debug information about the ISM.
- **ism-events** - Displays events on the ISM.

- **ism-status** - Displays status of the ISM.
- **lsa** - Displays LSAs.
- **lsa-flooding** - Displays LSA-flooding activity.
- **lsa-generation** - Displays information about LSA generation.
- **lsa-install** - Displays installed LSAs.
- **lsa-maxage** - Displays the maximum aging information for LSAs.
- **lsa-refresh** - Displays LSA refresh information.
- **nsm** - Displays information about the NSM.
- **nsm-events** - Displays event information for the NSM.
- **nsm-status** - Displays NSM status information.
- **packet** -Displays all OSPFv3 packets in rx or tx mode.
- **packet-dd** - Displays all OSPFv3 data description packets in rx or tx mode.
- **packet-hello** - Displays all OSPFv3 hello packets in rx or tx mode.
- **packet-lsa-ack** - Displays all OSPFv3 LSA ACK packets in rx or tx mode.
- **packet-lsa-req** - Displays all OSPFv3 LSA request packets in rx or tx mode.
- **packet-lsa-update** - Displays all OSPFv3 LSA update packets in rx or tx mode.
- **route** - Displays all OSPFv3 routes.
- **route-calc-external** - Displays external route calculations.
- **route-calc-inter-area** - Displays inter-area route calculations.
- **route-calc-intra-area** - Displays intra-area route calculations.
- **route-calc-spf** - Displays SPF route calculation.
- **route-calc-transit** - Displays transit route calculation.
- **route-install** - Displays all OSPFv3 routes installed.
- **virtual-link** - Displays all OSPFv3 virtual links.

The **debug ipv6 ospf** commands display information about OSPF activity, including Interface State Machine (ISM), neighbor state machine (NSM) and LSA data, packets, routes, and virtual links.

### debug ipv6 ospf ism

**Syntax: [no] debug ipv6 ospf ism**

This command generates comprehensive information about OSPF ISM status changes. Output resembles the following:

```
PowerConnect#debug ipv6 ospf ism
OSPFv3 ISM[137]: IntefaceUp
OSPFv3 ISM[137]: Status change Down -> Waiting (Priority > 0)
OSPFv3 ISM[137]: BackupSeen
OSPFv3 ISM[137]: Status change Waiting -> BDR (BackupSeen:DR Election)
OSPFv3 ISM[137]: (dr:0.0.0.0,bdr:0.0.0.0) -> (dr:2.2.2.2,bdr:1.2.3.4)
```

This output indicates a status change for ISM 137, from UP to Down to Waiting. A switch from the designated router (DR) to the backup designated router (BDR) has also occurred.

### debug ipv6 ospf ism-events

**Syntax: [no] debug ipv6 ospf ism-events**

Displays IPv6 OSPF interface state machine (ISM) activity, such as an interface coming up or going down. Output resembles the following:

```
PowerConnect#debug ipv6 ospf ism-events
OSPFv3 ISM[137]: Interfaces
OSPFv3 ISM[137]: BackupSeen goes up
```

### debug ipv6 ospf ism-status

**Syntax: [no] debug ipv6 ospf ism-status**

This command displays IPv6 OSPF ISM status information. Output resembles the following:

```
PowerConnect# debug ipv6 ospf ism-status
OSPFv3 ISM[137]: Status change Down -> Waiting (Priority > 0)
OSPFv3 ISM[137]: Status change Waiting -> BDR (BackupSeen, DR Election)
OSPFv3 ISM[137]: (dr:0.0.0.0,bdr:0.0.0.0) -> (dr:2.2.2.2,bdr 1.2.3.4)
```

This output indicates that ISM 137 has gone down and is waiting for a switch from the designated router (DR) to the backup designated router (BDR).

### debug ipv6 ospf lsa

**Syntax: [no] debug ipv6 ospf lsa**

This command displays information about OSPF LSAs. Output resembles the following:

```
PowerConnect# debug ipv6 ospf lsa
OSPFv3 LSA Update Intra-Area-Prefix (Stub): Checking Interface 137
OSPFv3 LSA Update Intra-Area-Prefix (Stub): Interface 137 is down
OSPFv3 LSA Update Intra-Area-Prefix (Stub): No prefix to advertise for Area
0.0.0.0
OSPFv3 LSA Update Intra-Area-Prefix (Stub): Area 0.0.0.0
OSPFv3 ISM (137): Status change Down -> Waiting (Priority > 0)
OSPFv3 LSA: Create LSA Type :Router id: 0 Advrouter:1.2.3.4
OSPFv3 LSA Update Intra-Area-Prefix (Stub): Checking Interface 137
OSPFv3 LSA Update Intra-Area-Prefix (Stub): Include 3000:1::2/64
OSPFv3 LSA: Create LSA Type :Router Id: 0 Advrouter: 1.2.3.4
OSPFv3 :LSA Update Intra-Area Prefix (Stub): Area 0.0.0.0
OSPFv3 :LSA Update Link: Interface 137
OSPFv3 LSA: Create LSA Type :Link id: 137 Advrouter: 1.2.3.4
```

### debug ipv6 ospf lsa-flooding

**Syntax: [no] debug ipv6 ospf lsa-flooding**

This command displays IPv6 OSPF LSA flooding activity. Output resembles the following:

```
PowerConnect# debug ipv6 ospf lsa-flooding
OSPFV3:LSA: schedule flooding 2.2.2.2
OSPFV3:LSA: schedule flooding 2.2.2.2
OSPFV3:LSA: schedule flooding 2.2.2.2
OSPFV3:LSA: schedule flooding 2.2.2.2
```

### debug ipv6 ospf lsa-generation

**Syntax: [no] debug ipv6 ospf lsa-generation**

This command shows additions or deletions of LSAs from the link state database. Output resembles the following:

```
PowerConnect# debug ipv6 ospf lsa-generation
```

```
OSPFV3 LSA: Create LSA Type :Router Id: 0 Advrouter:1.2.3.4
OSPFV3 LSA: Create LSA Type :IntraPrefix Id: 0 Advrouter: 1.2.3.4
OSPFV3 LSA: Delete LSA Type: Link Id: 137 Advrouter 1.2.3.4
OSPFV3 LSA: Create LSA Header Type: Router Id: 0 Advrouter: 1.2.3.4
OSPFV3 LSA: Create LSA Header Type: Router Id: 0 Advrouter: 2.2.2.2
OSPFV3 LSA: Create LSA Header Type: Router Id: 0 Advrouter: 1.2.3.4
```

### debug ipv6 ospf lsa-install

**Syntax: [no] debug ipv6 ospf lsa-install**

This command generates information about new LSAs which are installed in the link state database. Output resembles the following:

```
PowerConnect# debug ipv6 ospf lsa-install
OSPFv3 LSA: Turnover type: IntraPrefix Lsa Id: 0.0.0.0 Advrouter:1.2.3.4: contents
not changed
OSPFv3 LSA: Turnover type: Router Lsa Id: 0.0.0.0 AdvRouter:1.2.3.4: contents not
changed
OSPFv3 LSA: Turnover type: Router Lsa Id:0.0.0.0 AdvRouer:1.2.3.4: contents
changed
OSPFv3 LSA: Turnover type: IntraPrefix Lsa Id: 0.0.0.0 AdvRouter: 2.2.2.2:
contents changed
```

### debug ipv6 ospf lsa-maxage

**Syntax: [no] debug ipv6 ospf lsa-maxage**

This command identifies LSAs that are removed from the link state database because the router has not received any updates about the LSA in a specified amount of time. Output resembles the following:

```
PowerConnect# debug ipv6 ospf lsa-maxage
OSPFv3 LSA: Premature aging: Type: Interface, ID : 0, AdvRouter 1.2.3.4
OSPFv3 LSA : Premature aging: Type: IntraPrefix, ID : 0, AdvRouter 1.
```

## Configuration notes

- If you configure the PowerConnect B-MLXe router to use a loopback interface to communicate with a BGP4 neighbor, you must also configure a loopback interface on the neighbor, and configure the neighbor to use that loopback interface to communicate with the PowerConnect router.
- PowerConnect B-MLXe routers use the same router ID for both OSPF and BGP4. If the router is already configured for OSPF, you may want to use the router ID that is already in use rather than set a new one. To display the router ID, enter the **show ip** command at any CLI level.
- The command to set the router ID for a specified VRF is the same as the command for the default VRF. The only difference is that when setting it for a specific VRF, the **ip router-id** command is configured within the VRF.
- When setting the router ID, you can specify an IP address that is being used for an interface, but do not specify an IP address that is in use by another device.
- The OSPF Stub Router Advertisement feature is helpful for preventing a loss of traffic during short periods when adjacency failures are detected and traffic is rerouted. With this feature, traffic can be rerouted before an adjacency failure occurs (due to common service interruptions such as a router being shut down for maintenance). This feature is also useful during router startup because it gives the router enough time to build up the routing table before forwarding traffic. This is helpful where BGP is enabled because it takes time for the BGP routing table to converge.

- Under normal operation, restarting a BGP router reconfigures the network. In this situation, routes available through the restarting router are deleted when the router goes down and are then rediscovered and added back to the routing tables when the router comes back up. In a network where routers are regularly restarted, performance may be significantly degraded, limiting the availability of network resources. The BGP Graceful Restart feature dampens the network response and limits route flapping by allowing routes to remain available during a restart. BGP Graceful Restart operates between a router and its peers and must be configured on all devices.
- When you enable BGP4, you do not need to reset the system. The protocol is activated as soon as it is enabled. The router also begins a BGP4 session with a BGP4 neighbor as soon as you add the neighbor.
- The router attempts to establish a BGP4 session with a neighbor as soon as you enter a command specifying the neighbor IP address. To completely configure neighbor parameters before the router can establish a session, you can administratively shut down the neighbor.

## *Disabling BGP4*

If you disable BGP4, the router removes all configuration information for the disabled protocol from the running configuration. To restore the BGP4 configuration, you must reload the software, which loads the configuration from the startup configuration. When you save the configuration to the startup configuration file after disabling the protocol, all configuration information for the disabled protocol is removed from the startup configuration file. You will see a CLI error message similar to the following:

```
PowerConnect(config)#no router bgp
router bgp mode now disabled. All bgp configuration data will be lost when writing
to flash?
```

When you test a BGP4 configuration, and are likely to disable and re-enable the protocol, you might want to make a backup of the startup configuration file that contains the BGP4 configuration information. If you remove the configuration information by saving the configuration after you disable the protocol, you can then restore the configuration by copying the backup to the flash memory.

To disable BGP4 without losing the BGP4 configuration information, remove the local AS (for example, by entering the **no local-as** <*num*> command). In this case, BGP4 retains the other configuration information but is not operational until you again set the local AS.

## *Forwarding disruptions and port flapping*

The BGP Graceful Restart feature supports high-availability routing. With this feature enabled, disruptions in forwarding are minimized and route flapping diminished to provide continuous service during times when a router experiences a restart. For more information about BGP Graceful Restart, see the *NetIron Series Configuration Guide*.

## *Performance degrades during restarts*

BGP Graceful Restart also helps minimize performance degradation during restarts. For more information about BGP Graceful Restart, see the *NetIron Series Configuration Guide*

## Reducing the complexity of BGP configurations

The following information can help to simplify your BGP configurations.

### Confederations

A *confederation* is a BGP4 Autonomous System (AS) that has been subdivided into multiple, smaller ASs. Subdividing an autonomous system into smaller autonomous systems simplifies administration and reduces BGP-related traffic, thus reducing the complexity of the Interior Border Gateway Protocol (IBGP) mesh among the BGP routers in the AS.

### Router reflection

Another way to reduce the complexity of an IBGP mesh is to use route reflection. However, if you want to run different Interior Gateway Protocols (IGPs) within an AS, configure a confederation. You can run a separate IGP within each sub-AS.

For more information on how to configure these features, see the *NetIron Series Configuration Guide*.

## Disabling and restoring BGP4 configuration information

If you disable BGP4, the PowerConnect B-MLXe remove all BGP4 information from the running configuration. To restore the BGP4 configuration, you must reload the software. In addition, when you save the startup configuration file after disabling the protocol, all configuration information for the disabled protocol is removed from the startup configuration file.

The CLI displays a warning message similar to the following:

```
PowerConnect(config)#no router bgp
router bgp mode now disabled. All bgp config data will be lost when writing to
flash!
```

**NOTE**
The Web management interface does not display a warning message.

If you are testing a BGP4 configuration and are likely to disable and re-enable the protocol, you might want to first make a backup copy of the startup configuration file. This way, when you remove the configuration information by disabling the protocol, you can restore it by copying the backup startup configuration file to the flash memory.

To disable BGP4 without losing the configuration information, remove the local AS (for example, by entering the **no local-as** <*num*> command). In this case, BGP4 retains the other configuration information but will not operate until you reset the local AS.

## BGP memory considerations

BGP4 handles a very large number of routes and therefore requires a substantial amount of memory. For example, in a typical configuration with just a single BGP4 neighbor, a BGP4 router may need to manage as much as 250,000 routes. Many configurations, especially those involving more than one neighbor, may require the router to retain even more routes. The PowerConnect B-MLXe device routers provide dynamic memory allocation for BGP4 data, automatically allocating memory when needed to support BGP4 neighbors, routes, and route attribute entries. Dynamic memory allocation is performed automatically by the software and does not require a reload.

## Common diagnostic scenarios

- MD5 Authentication Is incorrect between two BGP Peers when a BGP session remains connected.

  Because MD5 authentication occurs at the TCP layer, the **debug ip bgp** commands will not help in this situation. To determine the exact error, issue the **debug ip tcp transactions** command.

- Password Configured on One Side Only

  When a password is configured on only one side of the connection, you will see output resembling the following:

```
PowerConnect#debug ip tcp transactions
TCP: transactions debugging is on
TCP: connected to 10.10.10.1:8102 advertising MSS 1460. MD5 1
TCP: connection from 10.10.10.2:179->10.10.10.1:8102 rcvd, MSS 1460
TCP: 10.10.10.2:179 -> 10.10.10.1:8102: missing MD5 option
TCP: connected to 10.10.10.1:8102 advertising MSS 1460. MD5 1
TCP: connection from 10.10.10.2:179->10.10.10.1:8102 rcvd, MSS 1460
TCP: 10.10.10.2:179 -> 10.10.10.1:8102: missing MD5 option
```

- Password does not match on peers.
- A configuration error message indicates that the next hop cannot be found. This may occur because the IP route to the next hop in the IP routing table is incorrect.
- Old software versions.

  Feature issues are often caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# OSPF

The OSPF protocol uses link-state advertisements (LSA) to update neighboring routers about interfaces and information on those interfaces. OSPF routers maintain identical databases that describes their area topology, helping any individual router to determine the shortest path between itself and any neighboring router.

## OSPF show commands

### show ip ospf config

**Syntax: show ip ospf config**

To display general OSPF configuration information, enter this command at any CLI level:

```
PowerConnect#show ip ospf config
Router OSPF: Enabled
Redistribution: Disabled
Default OSPF Metric: 10
OSPF Redistribution Metric: Type2
OSPF External LSA Limit: 1447047
OSPF Database Overflow Interval: 0
RFC 1583 Compatibility: Enabled
Router id: 192.168.100.1
Interface State Change Trap:                        Enabled
Virtual Interface State Change Trap:                Enabled
Neighbor State Change Trap:                         Enabled
Virtual Neighbor State Change Trap:                 Enabled
Interface Configuration Error Trap:                 Enabled
Virtual Interface Configuration Error Trap:         Enabled
Interface Authentication Failure Trap:              Enabled
Virtual Interface Authentication Failure Trap:      Enabled
Interface Receive Bad Packet Trap:                  Enabled
Virtual Interface Receive Bad Packet Trap:          Enabled
Interface Retransmit Packet Trap:                   Disabled
Virtual Interface Retransmit Packet Trap:           Disabled
Originate LSA Trap:                                 Disabled
Originate MaxAge LSA Trap:                          Disabled
Link State Database Overflow Trap:                  Disabled
Link State Database Approaching Overflow Trap:      Disabled
OSPF Area currently defined:
Area-ID          Area-Type Cost
0                normal    0
OSPF Interfaces currently defined:
Ethernet Interface: 3/1-3/2
ip ospf md5-authentication-key-activation-wait-time 300
ip ospf cost 0
ip ospf area 0
Ethernet Interface: v1
ip ospf md5-authentication-key-activation-wait-time 300
ip ospf cost 0
ip ospf area 0
```

### show tasks

**Syntax: show tasks**

To display CPU usage statistics and other OSPF tasks, enter the **show tasks** command:

```
PowerConnect#snow tasks
Task Name   Pri   State     PC     Stack  Size  CPU Usage(%) task id  task vid
----------  ---  -----  ---------  --------  -----  ---------  ------ -------
       idle 0    ready  00001904  04058fa0  4096    99          0        0
    monitor 20    wait  0000d89c  0404bd80  8192    0           0        0
        int 16    wait  0000d89c  04053f90  16384   0           0        0
      timer 15    wait  0000d89c  04057f90  16384   0           0        0
        dbg 30    wait  0000d89c  0404ff08  8192    0           0        0
      flash 17    wait  0000d89c  0409ff90  8192    0           0        0
         wd 31    wait  0000d89c  0409df80  8192    0           0        0
       boot 17    wait  0000d89c  04203e28  65536   0           0        0
       main 3     wait  0000d89c  2060cf38  65536   0           0        1
        itc 6     wait  0000d89c  20612ae8  16384   0           0        1
        tmr 5     wait  0000d89c  20627628  16384   0           0        1
      ip_rx 5     wait  0000d89c  2062ff48  16384   0           0        1
        scp 5     wait  0000d89c  20635628  16384   0           0        1
    console 5     wait  0000d89c  2063e618  32768   0           0        1
       vlan 5     wait  0000d89c  20648618  16384   0           0        1
    mac_mgr 5     wait  0000d89c  20657628  16384   0           0        1
    mrp_mgr 5     wait  0000d89c  2065c628  16384   0           0        1
       vsrp 5     wait  0000d89c  20663620  16384   0           0        1
       snms 5     wait  0000d89c  20667628  16384   0           0        1
        rtm 5     wait  0000d89c  20674628  16384   0           0        1
       rtm6 5     wait  0000d89c  2068a628  16384   0           0        1
      ip_tx 5    ready  0000d89c  206a9628  16384   0           0        1
        rip 5     wait  0000d89c  20762628  16384   0           0        1
        bgp 5     wait  0000d89c  207e6628  16384   0           0        1
     bgp_io 5     wait  0000d89c  2082ef00  16384   0           0        1
       ospf 5     wait  0000d89c  20832628  16384   1           0        1
ospf_r_calc 5     wait  0000d89c  2089ff10  16384   0           0        1
  isis_task 5     wait  0000d89c  208a3628  16384   0           0        1
   isis_spf 5     wait  0000d89c  208a8f10  16384   0           0        1
      mcast 5     wait  0000d89c  208ac628  16384   0           0        1
       vrrp 5     wait  0000d89c  208b4628  16384   0           0        1
      ripng 5     wait  0000d89c  208b9628  16384   0           0        1
      ospf6 5     wait  0000d89c  208c3628  16384   0           0        1
   ospf6_rt 5     wait  0000d89c  208c7f08  16384   0           0        1
     mcast6 5     wait  0000d89c  208cb628  16384   0           0        1
         l4 5     wait  0000d89c  208cf620  16384   0           0        1
        stp 5     wait  0000d89c  209a7620  16384   0           0        1
       snmp 5     wait  0000d89c  209c3628  32768   0           0        1
       rmon 5     wait  0000d89c  209cc628  32768   0           0        1
        web 5     wait  0000d89c  209d6628  32768   0           0        1
       lacp 5     wait  0000d89c  209da628  16384   0           0        1
      dot1x 5     wait  0000d89c  209e0620  16384   0           0        1
  hw_access 5     wait  0000d89c  209e6628  16384   0           0        1
```

### show ip ospf area

**Syntax: show ip ospf area [<*area-id*>] | [<*num*>]**

- <*area-id*> - Shows information for the specified area.
- <*num*> - Corresponds to the entry number you enter. The entry number identifies the entry's position in the area table.

To display OSPF area information, enter the **show ip ospf area** command at any CLI level:

```
PowerConnect#show ip ospf area
Indx  Area       Type   Cost  SPFR  ABR  ASBR  LSA  Chksum(Hex)
1   0.0.0.0      normal  0    1      0     0    1    0000781f
2  192.147.60.0  normal  0    1      0     0    1    0000fee6
3  192.147.80.0  stub    1    1      0     0    2    000181cd
```

### show ip ospf neighbor

**Syntax: show ip ospf neighbor [router-id <*ip-addr*> | <*num*> | extensive]**

- **router-id** <*ip-addr*> - Displays only the neighbor entries for the specified router.
- <*num*> - Displays only the entry in the specified index position in the neighbor table. For example, if you enter "1", only the first entry in the table is displayed.
- **extensive** - Displays detailed information about the neighbor.

To display OSPF neighbor information, enter the following command at any CLI level:

```
PowerConnect#show ip ospf neighbor
Port  Address         Pri  State      Neigh Address   Neigh ID     Ev Op Cnt
v10   10.1.10.1       1    FULL/DR    10.1.10.2       10.65.12.1    5  2   0
v11   10.1.11.1       1    FULL/DR    10.1.11.2       10.65.12.1    5  2   0
v12   10.1.12.1       1    FULL/DR    10.1.12.2       10.65.12.1    5  2   0
v13   10.1.13.1       1    FULL/DR    10.1.13.2       10.65.12.1    5  2   0
v14   10.1.14.1       1    FULL/DR    10.1.14.2       10.65.12.1    5  2   0
```

### show ip ospf interface

**Syntax: show ip ospf interface [<*ip-addr*>]**

- <*ip-addr*> - The specified IP address.

This command displays OSPF interface information, including the IP address, the OSPF state, link ID, and cost, which appear in the following example:

```
PowerConnect#show ip ospf interface 192.168.1.1
Ethernet 2/1,OSPF enabled
     IP Address 192.168.1.1, Area 0
     OSPF state ptr2ptr, Pri 1, Cost 1, Options 2,  Type pt-2-pt Events 1
     Timers(sec): Transit 1, Retrans 5, Hello 10, Dead 40
     DR:  Router ID 0.0.0.0           Interface Address 0.0.0.0
     BDR: Router ID 0.0.0.0           Interface Address 0.0.0.0
     Neighbor Count = 0, Adjacent Neighbor Count= 1
     Neighbor: 2.2.2.2
     Authentication-Key:None
     MD5 Authentication: Key None, Key-Id None, Auth-change-wait-time 300
```

### show ip ospf interface brief

**Syntax: show ip ospf interface brief**

This command displays the OSPF database information in brief format, as shown in the following example:

```
PowerConnect#show ip ospf interface brief
Number of Interfaces is 1
Interface Area IP Addr/Mask Cost State Nbrs(F/C)
eth 1/2    0    16.1.1.2/24   1     down  0/0
```

### show ip ospf routes

**Syntax: show ip ospf routes [<*ip-addr*>]**

- <*ip-addr*> - Specifies a destination IP address for which to display route entries.

This command displays OSPF route information, as shown in the following example:

```
PowerConnect#show ip ospf route
OSPF Area 0x00000000 ASBR Routes 1:
      Destination     Mask            Path_Cost Type2_Cost Path_Type
      10.65.12.1      255.255.255.255 1         0          Intra
      Adv_Router      Link_State      Dest_Type State      Tag       Flags
      10.65.12.1      10.65.12.1      Asbr      Valid      0          6000
      Paths Out_Port  Next_Hop        Type      State
      1     v49       10.1.49.2       OSPF      21 01
      2     v12       10.1.12.2       OSPF      21 01
      3     v11       10.1.11.2       OSPF      21 01
      4     v10       10.1.10.2       OSPF      00 00
OSPF Area 0x00000041 ASBR Routes 1:
      Destination     Mask            Path_Cost Type2_Cost Path_Type
      10.65.12.1      255.255.255.255 1         0          Intra
      Adv_Router      Link_State      Dest_Type State      Tag       Flags
      10.65.12.1      10.65.12.1      Asbr      Valid      0          6000
      Paths Out_Port  Next_Hop        Type      State
      1     v204      10.65.5.251     OSPF      21 01
      2     v201      10.65.2.251     OSPF      20 d1
      3     v202      10.65.3.251     OSPF      20 cd
      4     v205      10.65.6.251     OSPF      00 00
OSPF Area Summary Routes 1:
      Destination     Mask            Path_Cost Type2_Cost Path_Type
      10.65.0.0       255.255.0.0     0         0          Inter
      Adv_Router      Link_State      Dest_Type State      Tag       Flags
      10.1.10.1       0.0.0.0         Network   Valid      0          0000
      Paths Out_Port  Next_Hop        Type      State
      1     1/1       0.0.0.0         DIRECT    00 00
OSPF Regular Routes 208:
      Destination     Mask            Path_Cost Type2_Cost Path_Type
      10.1.10.0       255.255.255.252 1         0          Intra
      Adv_Router      Link_State      Dest_Type State      Tag       Flags
      10.1.10.1       10.1.10.2       Network   Valid      0          0000
      Paths Out_Port  Next_Hop        Type      State
      1     v10       0.0.0.0         OSPF      00 00
      Destination     Mask            Path_Cost Type2_Cost Path_Type
      10.1.11.0       255.255.255.252 1         0          Intra
      Adv_Router      Link_State      Dest_Type State      Tag       Flags
      10.1.10.1       10.1.11.2       Network   Valid      0          0000
      Paths Out_Port  Next_Hop        Type      State
      1     v11       0.0.0.0         OSPF      00 00
```

### show ip ospf redistribute route

**Syntax: show ip ospf redistribute route [<*ip-addr*> <*ip-mask*>]**

This command displays routes that have been redistributed into OSPF:

```
PowerConnect#show ip ospf redistribute route
  4.3.0.0 255.255.0.0 static
  3.1.0.0 255.255.0.0 static
  10.11.61.0 255.255.255.0 connected
  4.1.0.0 255.255.0.0 static
```

In this example, four routes have been redistributed. Three of the routes were redistributed from static IP routes and one was redistributed from a directly-connected IP route.

To display route redistribution for a specific IP address and mask, enter this command as shown in the following example:

```
PowerConnect#show ip ospf redistribute route 3.1.0.0 255.255.0.0
  3.1.0.0 255.255.0.0 static
```

### show ip ospf database

**Syntax: show ip ospf database**

This command displays the OSPF database, as shown in this example:

```
PowerConnect#show ip ospf database
Graceful Link States
Area  Interface  Adv Rtr  Age Seq(Hex) Prd Rsn  Nbr Intf IP
0     eth 1/2    2.2.2.2  7   80000001 60  SW   6.1.1.2

Router Link States
Index AreaID             Type LS ID      Adv Rtr     Seq(Hex)   Age  Cksum
SyncState
1     0                  Rtr  2.2.2.2    2.2.2.2     80000003   93   0xac6c
Done
2     0                  Rtr  1.1.1.1    1.1.1.1     80000005   92   0x699e
Done
3     0                  Net  16.1.1.2   2.2.2.2     80000002   93   0xbd73
Done
4     0                  OpAr 1.0.0.3    1.1.1.1     80000005   83   0x48e7
Done
5     0                  OpAr 1.0.0.2    2.2.2.2     80000006   80   0x50da
Done
6     111.111.111.111    Rtr  1.1.1.1    1.1.1.1     80000004   142  0x0a38
Done
7     111.111.111.111    Summ 1.1.1.1    1.1.1.1     80000001   147  0x292b
Done
8     111.111.111.111    OpAr 1.0.0.2    1.1.1.1     80000002   179  0x063f
Done
Type-5 AS External Link States
Index Age  LS ID     Router   Netmask   Metric    Flag  Fwd Address
1     147  9.9.1.13  1.1.1.1  ffffffff  0000000a  0000  0.0.0.0
2     147  9.9.1.26  1.1.1.1  ffffffff  0000000a  0000  0.0.0.0
```

### show ip ospf database external-link-state

**Syntax: show ip ospf database external-link-state [advertise <*num*> | extensive | link-state-id <*ip-addr*> | router-id <*ip-addr*> | sequence-number <*num(Hex)*>]**

- **advertise** <num> - Displays the hexadecimal data in the specified LSA packet. <*num*> identifies the LSA packet by its position in the router's External LSA table. Enter the **show ip ospf external-link-state** command to display the table.
- **extensive** - Displays the LSAs in decrypted format.

**NOTE**
The **extensive** option displays the entire database, and cannot be used in combination with other display options.

- **link-state-id** <*ip-addr*> - Displays the External LSAs for the LSA source for the specified IP address.
- **router-id** <*ip-addr*> - Shows the External LSAs for the specified OSPF router.
- **sequence-number** <*num(Hex)*> - Displays the External LSA entries for the specified hexadecimal LSA sequence number.

This command displays external link state information, as shown in this example:

```
PowerConnect#show ip ospf database external-link-state
Index Aging  LS ID           Router          Netmask  Metric    Flag  Fwd Address
SyncState
1     591    10.65.13.0      10.65.12.1      ffffff00 8000000a  0000  0.0.0.0
Done
2     591    10.65.16.0      10.65.12.1      ffffff00 8000000a  0000  0.0.0.0
Done
3     591    10.65.14.0      10.65.12.1      ffffff00 8000000a  0000  0.0.0.0
Done
4     591    10.65.17.0      10.65.12.1      ffffff00 8000000a  0000  0.0.0.0
Done
5     592    10.65.12.0      10.65.12.1      ffffff00 8000000a  0000  0.0.0.0
Done
6     592    10.65.15.0      10.65.12.1      ffffff00 8000000a  0000  0.0.0.0
Done
7     592    10.65.18.0      10.65.12.1      ffffff00 8000000a  0000  0.0.0.0
Done
```

### show ip ospf database database-summary

**Syntax: show ip ospf database database-summary**

This command displays database summary information, as shown in this example:

```
PowerConnect#show ip ospf database database-summary
Area ID      Router  Network Sum-Net  Sum-ASBR  NSSA-Ext  Opq-Area  Subtotal
0.0.0.0      104      184     19       42        0         0         349
AS External                                                         308
Total        104       184    19       42        0         0         657
```

### show ip ospf database link-state

**Syntax: show ip ospf database link-state [advertise <*num*> | asbr [<*ip-addr*>] [adv-router <*ip-addr*>] | extensive | link-state-id <*ip-addr*> | network [<*ip-addr*>] [adv-router <*ip-addr*>] | nssa [<*ip-addr*>] [adv-router <*ip-addr*>] | router [<*ip-addr*>] [adv-router <*ip-addr*>] | router-id <*ip-addr*> | self-originate l sequence-number <*num(Hex)*> | summary [<*ip-addr*>] [adv-router <*ip-addr*>]]**

- **advertise** <*num*> - Displays the hexadecimal data in the specified LSA packet. The <*num*> parameter identifies the LSA packet by its position in the router LSA table. To determine an LSA packet's position in the table, enter the **show ip ospf link-state** command to display the table.
- **asbr** - Shows ASBR LSAs.
- **extensive** - Displays the LSAs in decrypted format.

  **NOTE**
  The **extensive** option displays the entire database, and cannot be used in combination with other display options.

- **link-state-id** <*ip-addr*> - Displays the LSAs for the LSA source for the specified IP address.
- **network** <*ip-addr*> - Shows network LSAs.
- **nssa** <*ip-addr*> - Shows NSSA LSAs.
- **router** <*ip-addr*> - Displays LSAs by router link.
- **router-id** <*ip-addr*> - Shows the LSAs for the specified OSPF router.

- **sequence-number** <*num(Hex)*> - Displays the LSA entries for the specified hexadecimal LSA sequence number.
- **self-originate** - Shows self-originated LSAs.
- **sequence-number** <*num(Hex)*> - Displays LSAs for the specified sequence number
- **summary** - Shows summary information.

This command displays database link state information, as shown in this example:

```
PowerConnect#show ip ospf database link-state
Index Area ID        Type   LS ID           Adv Rtr          Seq(Hex) Age  Cksum
1     0              Rtr  10.1.10.1       10.1.10.1       800060ef 3    0x4be2
2     0              Rtr  10.65.12.1      10.65.12.1      80005264 6    0xc870
3     0              Net  10.1.64.2       10.65.12.1      8000008c 1088 0x06b7
4     0              Net  10.1.167.2      10.65.12.1      80000093 1809 0x86c8
5     0              Net  10.1.14.2       10.65.12.1      8000008c 1088 0x2ec1
6     0              Net  10.1.117.2      10.65.12.1      8000008c 1087 0xbccb
7     0              Net  10.1.67.2       10.65.12.1      8000008c 1088 0xe4d5
8     0              Net  10.1.170.2      10.65.12.1      80000073 604  0xa5c6
9     0              Net  10.1.17.2       10.65.12.1      8000008c 1088 0x0ddf
10    0              Net  10.1.120.2      10.65.12.1      8000008c 1087 0x9be9
11    0              Net  10.1.70.2       10.65.12.1      8000008c 1088 0xc3f3
12    0              Net  10.1.173.2      10.65.12.1      80000017 1087 0x3d88
13    0              Net  10.1.20.2       10.65.12.1      8000008c 1088 0xebfd
14    0              Net  10.1.123.2      10.65.12.1      8000008c 1087 0x7a08
15    0              Net  10.1.73.2       10.65.12.1      8000008c 1088 0xa212
16    0              Net  10.1.176.2      10.65.12.1      80000025 1087 0xffb4
17    0              Net  10.1.23.2       10.65.12.1      8000008c 1088 0xca1c
18    0              Net  10.1.126.2      10.65.12.1      8000008c 1087 0x5926
```

### show ip ospf border-routers

**Syntax: show ip ospf border-routers [<*ip-addr*>]**

- <*ip-addr*> - Displays the ABR and ASBR entries for the specified IP address.

This command displays OSPF ABR and ASBR information, as shown in the following example:

```
PowerConnect#show ip ospf border-routers
      router ID       router type next hop router  outgoing interface  Area
1     10.65.12.1      ABR         10.1.49.2        v49                 0
1     10.65.12.1      ASBR        10.1.49.2        v49                 0
1     10.65.12.1      ABR         10.65.2.251      v201                65
1     10.65.12.1      ASBR        10.65.2.251      v201                65
```

### show ip ospf trap

**Syntax: show ip ospf trap**

All traps are enabled by default when you enable OSPF. To display the state of each OSPF trap, enter the following command at any CLI level:

```
PowerConnect#show ip ospf trap
Interface State Change Trap:                      Enabled
Virtual Interface State Change Trap:              Enabled
Neighbor State Change Trap:                       Enabled
Virtual Neighbor State Change Trap:               Enabled
Interface Configuration Error Trap:               Enabled
Virtual Interface Configuration Error Trap:       Enabled
Interface Authentication Failure Trap:            Enabled
Virtual Interface Authentication Failure Trap:    Enabled
Interface Receive Bad Packet Trap:                Enabled
Virtual Interface Receive Bad Packet Trap:        Enabled
Interface Retransmit Packet Trap:                 Disabled
Virtual Interface Retransmit Packet Trap:         Disabled
Originate LSA Trap:                               Disabled
Originate MaxAge LSA Trap:                        Disabled
Link State Database Overflow Trap:                Disabled
Link State Database Approaching Overflow Trap:    Disabled
```

### show ip ospf interface

**Syntax: show ip ospf interface [<*ip-addr*>]**

- <*ip-addr*> - Specifies an IP address.

This command displays OSPF point-to-point information, as shown in the following example:

```
PowerConnect#show ip ospf interface 192.168.1.1
Ethernet 2/1,OSPF enabled
IP Address 192.168.1.1, Area 0
OSPF state ptr2ptr, Pri 1, Cost 1, Options 2, Type pt-2-pt Events 1
Timers(sec): Transit 1, Retrans 5, Hello 10, Dead 40
DR: Router ID 0.0.0.0 Interface Address 0.0.0.0
BDR: Router ID 0.0.0.0 Interface Address 0.0.0.0
Neighbor Count = 0, Adjacent Neighbor Count= 1
Neighbor: 2.2.2.2
Authentication-Key:None
MD5 Authentication: Key None, Key-Id None, Auth-change-wait-time 300
```

### show ip ospf debug misc

**Syntax: show ip ospf debug misc**

This command displays miscellaneous OSPF information, including router counts and SPF calculations. Output resembles the following:

```
PowerConnect#show ip ospf debug misc
Type-5 Forwarding Addr Count :0
Imported Route Count :        1
External Route Flap Count :   0
NSSA Route Flap Count :       0
External Lsa Count :          1
NSSA Lsa Count :              0
```

```
OSPF Recalc Statistics:
        phase_number: 0, area_id: 0xffffffff, next_chunk: 0x00000000
        duration(50ms): 0
MAX_AGE EXT lsa count 0, total EXT lsa count 1
```

### show ip ospf debug graceful-restart

**Syntax: show ip ospf debug graceful-restart**

This command displays information about OSPF graceful-restart events. Output resembles the following:

```
PowerConnect#show ip ospf debug graceful-restart
MP active: 1, standby up 0, nbr(1 0), vi(0, 0)
OSPF graceful-restart: enable 0, helper 1, timer 120/0, count 0, restarting 0
OSPF graceful-restart helper:
     Neighbor             ID      Area   Interface State Grace Helper Time
     11.1.1.1           1.1.1.1     0      1/1      8       0      0 0
OSPF graceful-restart LSA:
Area Interface         ID      Type Age   Max Seq        Interface Option
```

### show run

**Syntax: show run**

This command displays OSPF virtual neighbor and virtual link information, as shown in this example:

```
PowerConnect#show run
Current configuration:
ver V2.2.1T143
module 1 rx-bi-1g-24-port-fiber
module 2 rx-bi-10g-4-port
module 6 rx-bi-10g-4-port
module 7 rx-bi-1g-24-port-copper
!
no spanning-tree
!
vlan 1 name DEFAULT-VLAN
!
clock summer-time
clock timezone us Pacific
hostname R11-RX8
router ospf
 area 2
 area 1
 area 1 virtual-link 131.1.1.10
```

### show ip ospf virtual neighbor

**Syntax: show ip ospf virtual neighbor [<*num*>]**

- <*num*> - Displays the table beginning at the specified entry number.

This command displays OSPF virtual neighbor information, as shown in this example (which relates to the previous configuration):

```
PowerConnect#show ip ospf virtual neighbor
Indx Transit Area    Router ID       Neighbor address options
1    1               131.1.1.10      135.14.1.10      2
     Port   Address         state           events          count
\     6/2    27.11.1.27      FULL            5               0
```

### show ip ospf virtual link

**Syntax: show ip ospf virtual link [<*num*>]**

When you enter a numeric value for <*num*>, this command displays the table beginning at the specified entry number.

This command displays OSPF virtual link information. The output below represents the virtual links for the previous configuration.

```
PowerConnect#show ip ospf virtual link
Indx Transit Area     Router ID        Transit(sec) Retrans(sec) Hello(sec)
1    1                131.1.1.10       1            5            10
     Dead(sec)        events           state        Authentication-Key
     40               1                ptr2ptr      None
     MD5 Authentication-Key:       None
     MD5 Authentication-Key-Id:    None
     MD5 Authentication-Key-Activation-Wait-Time:      300
```

### show ip ospf neighbors

**Syntax: show ip ospf neighbors**

This command displays OSPF Graceful Restart information for OSPF neighbors as shown in this example:

```
PowerConnect#show ip ospf neighbors
Port  Address        Pri State       Neigh Address   Neigh ID        Ev Opt Cnt
2/7   50.50.50.10    0   FULL/OTHER  50.50.50.1      10.10.10.30     21 66  0
  < in graceful restart state,  helping 1, timer 60 sec >
```

### show ip ospf database grace-link-state

**Syntax: show ip ospf database grace-link-state**

This command displays Type 9 Graceful LSAs, as shown in this example:

```
PowerConnect#show ip ospf database grace-link-state
Graceful Link States
Area  Interface  Adv Rtr  Age Seq(Hex) Prd Rsn  Nbr Intf IP
0     eth 1/2    2.2.2.2  7   80000001 60  SW   6.1.1.2
```

### show ip ospf

**Syntax: show ip ospf**

This command displays the current OSPF Router Advertisement configuration. The bold text in the following example is displayed for an OSPF Router Advertisement configuration.

```
PowerConnect#show ip ospf
OSPF Version                  Version 2
Router Id                     10.10.10.10
ASBR Status                   No
ABR Status                    No         (0)
Redistribute Ext Routes from
External LSA Counter          5
External LSA Checksum Sum     0002460e
Originate New LSA Counter     5
Rx New LSA Counter            8
External LSA Limit            14447047
Database Overflow Interval    0
Database Overflow State :     NOT OVERFLOWED
RFC 1583 Compatibility :      Enabled
Originating router-LSAs with maximum metric
   Condition: Always Current State: Active
   Link Type: PTP STUB TRANSIT
```

### show ip ospf debug

**Syntax: show ip ospf debug**

This command generates descriptive information about OSPF activity, as shown in the following example:

```
PowerConnect#show ip ospf debug
External LSA Counter          1
Timer enable 1, 1s counter 5, ticks/sec 10, currtime 1167, md5_seq 0
sptr_area_list 0x0b10002c, import_routes 1
build_routing_table 0, is pending 0, ospf_spf_pending_list_fwd 0
route_calculation_in_progress 0
ospf->ospf_schedule_build_wait_time 0
SPF build timers:  last end 896, scheduled 821, init 863, run 821
route_calc_process_take_semaphore 6
process_redis_events 0, ospf_flush_cache_for_new_route 0
originate_ext_lsa_counts 2, ospfOriginateNewLsas 20
*ospf->of_max_one_second_timer_value 0, ospf->of_one_second_timer_value 0
*ospf->of_max_redis_timer_value 0, ospf->of_redis_timer_value 0
ospf->of_max_r_cal_time_value 0 0 0 0 0 0 0 0
of_max_change_ext_route_fwd_addr_time_value
ospf->of_max_originate_external_lsa_time_value 0
ospf->of_max_retransmit_lsa_timer_value 0, ospf->of_retransmit_lsa_timer_value 0
ospf->of_max_retransmit_db_timer_value 0, ospf->of_retransmit_db_timer_value 0
*of_max_neighbor_retransmit_queue_count 0, ospf->ospf_retransmit_queue_count 0
*of_max_retransmit_queue_count_exceed 0
ospf->set_one_shot_timer 0, ospf->ospf_one_shot_timer_token 0
*ospf->of_max_one_shot_timer_value 0, ospf->of_one_shot_timer_value 0
*ospf->of_max_one_shot_timer_count_long 0,
ospf->of_max_one_shot_timer_count_short 0
ospf->of_max_flood_refresh_lsa_timer_value 0, ospf-
>of_flood_refresh_lsa_timer_value 0
total ospf->of_flood_refresh_lsa_func_count 0, current flood_lsa_count 20
ospf->ospf_out_of_memory_for_send_packet 0
```

```
ospf->of_rtm_add_redis_count 1,ospf->of_rtm_add_redis_added_count 1,
invalid_count 0
ospf->of_rtm_del_redis_count 0, ospf->of_rtm_del_redis_deleted_count 0
of_rtm_clear_count 0,  of_rtm_clear_all_count 0, of_rtm_default_count 0
number_of_routes_imported 1, ospf->of_rtm_modify_redis_count 0
*of_max_process_adver_time_value 0, of_process_adver_time_value 0
*ospf->of_max_find_lsa_time_value 0, ospf->of_find_lsa_time_value 0
*ospf->of_max_cleaup_database_time_value 0, ospf->of_max_find_database_time_value
0
*ospf->of_max_find_ls_request_time_value 0, ospf->of_max_count_on_finding_lsa 0
msg_q_length = 0 msg_q_high_mark = 1
```

### show ip ospf debug memory

**Syntax: show ip ospf debug memory**

This command displays information about OSPF memory pools. Output resembles the following:

```
PowerConnect#show ip ospf debug memory
OSPF Memory Use 1302832
Pid SBlock TBlocks UBlocks FBlocks EBlocks SAddress CAddress
0   0      0       0       0       0       00000000 00000000
1   40     2000    30      1970    0       09207010 09207470
2   56     4000    25      3975    0       0921a8a0 0921ae18
3   132    32      10      22      0       07ab7b48 07ab8070
4   260    16      2       14      0       07ab8bd8 07ab8bd8
5   516    32      4       28      0       07ab9c28 07aba438
6   1504   32      0       32      0       092513b0 092513b0
7   4290   16      1       15      0       0925cfc0 0925e082
8   53571  16      3       13      0       0b100028 0b1273f1
9   0      0       0       0       0       00000000 00000000
Total Memory blocks allocated 75
Mega Memory List
Pool Id = 1, Total Mega blocks = 1 Errors = 0
Pool Id = 2, Total Mega blocks = 1 Errors = 0
Pool Id = 3, Total Mega blocks = 1 Errors = 0
Pool Id = 4, Total Mega blocks = 1 Errors = 0
Pool Id = 5, Total Mega blocks = 1 Errors = 0
Pool Id = 6, Total Mega blocks = 1 Errors = 0
Pool Id = 7, Total Mega blocks = 1 Errors = 0
Pool Id = 8, Total Mega blocks = 1 Errors = 0
OSPF Main Routing Table: 078dd444
node_count 6, top 0x078dd5b4, default_valid 0, default_route 0xffffffff
Table private pool:
init#=4096 unit_s=36 total=4096 in_use=5 fail=0 upper=no-limit min_mem=0
    UBlocks  EBlocks Total   PType
0   0      0      0       OSPF_MEMORY_POOL_ANY
1   3      0      14      OSPF_MEMORY_POOL_ROUTER_LINK_ADVERTISEMENT
2   1      0      2       OSPF_MEMORY_POOL_NETWORK_LINK_ADVERTISEMENT
3   15     0      17      OSPF_MEMORY_POOL_SUMMARY_LINK_ADVERTISEMENT
4   1      0      2       OSPF_MEMORY_POOL_EXTERNAL_LINK_ADVERTISEMENT
5   0      0      0       OSPF_MEMORY_POOL_GRACE_LINK_ADVERTISEMENT
6   5      0      9       OSPF_MEMORY_POOL_OPAQUE_AREA_ADVERTISEMENT
7   0      0      1       OSPF_MEMORY_POOL_LS_DATABASE_SUMMARY
8   0      0      10      OSPF_MEMORY_POOL_LS_DATABASE_NODE
9   0      0      30      OSPF_MEMORY_POOL_SHORTEST_PATH_NODE
Total Memory blocks allocated 75
Mega Memory List
Pool Id = 1, Total Mega blocks = 1 Errors = 0
Pool Id = 2, Total Mega blocks = 1 Errors = 0
Pool Id = 3, Total Mega blocks = 1 Errors = 0
Pool Id = 4, Total Mega blocks = 1 Errors = 0
Pool Id = 5, Total Mega blocks = 1 Errors = 0
Pool Id = 6, Total Mega blocks = 1 Errors = 0
```

```
Pool Id = 7, Total Mega blocks = 1 Errors = 0
Pool Id = 8, Total Mega blocks = 1 Errors = 0
OSPF Main Routing Table: 078dd444
node_count 6, top 0x078dd5b4, default_valid 0, default_route 0xffffffff
init#=4096 unit_s=36 total=4096 in_use=5 fail=0 upper=no-limit min_mem=0
UBlocks  EBlocks Total   PType
0   0      0      0       OSPF_MEMORY_POOL_ANY
1   3      0      14      OSPF_MEMORY_POOL_ROUTER_LINK_ADVERTISEMENT
2   1      0      2       OSPF_MEMORY_POOL_NETWORK_LINK_ADVERTISEMENT
3   15     0      17      OSPF_MEMORY_POOL_SUMMARY_LINK_ADVERTISEMENT
4   1      0      2       OSPF_MEMORY_POOL_EXTERNAL_LINK_ADVERTISEMENT
5   0      0      0       OSPF_MEMORY_POOL_GRACE_LINK_ADVERTISEMENT
6   5      0      9       OSPF_MEMORY_POOL_OPAQUE_AREA_ADVERTISEMENT
7   0      0      1       OSPF_MEMORY_POOL_LS_DATABASE_SUMMARY
8   0      0      10      OSPF_MEMORY_POOL_LS_DATABASE_NODE
9   0      0      30      OSPF_MEMORY_POOL_SHORTEST_PATH_NODE
10  8      0      8       OSPF_MEMORY_POOL_OSPF_ROUTE_INFO
11  6      0      6       OSPF_MEMORY_POOL_OSPF_MAIN_ROUTE_ENTRY
12  0      0      0       OSPF_MEMORY_POOL_OSPF_SUMMARY_ROUTE_ENTRY
13  0      0      0       OSPF_MEMORY_POOL_OSPF_EXT_SUMMARY_ROUTE_ENTRY
14  1      0      1       OSPF_MEMORY_POOL_OSPF_ABR_ROUTE_ENTRY
15  1      0      1       OSPF_MEMORY_POOL_OSPF_ASBR_ROUTE_ENTRY
16  1      0      1       OSPF_MEMORY_POOL_EXTERNAL_ROUTE
17  0      0      0       OSPF_MEMORY_POOL_ADVERTISEMENT_NODE
18  25     0      35      OSPF_MEMORY_POOL_LS_DATABASE_ENTRY
19  0      0      2       OSPF_MEMORY_POOL_LS_REQUEST
20  0      0      4       OSPF_MEMORY_POOL_LS_HEADER_QUEUE
21  0      0      5       OSPF_MEMORY_POOL_NEIGHBOR_LIST
22  0      0      0       OSPF_MEMORY_POOL_TRANSIT_AREA_ENTRY
23  0      0      29      OSPF_MEMORY_POOL_NEXT_HOP_BLOCK
24  0      0      0       OSPF_MEMORY_POOL_HOSTS
25  0      0      0       OSPF_MEMORY_POOL_ADDRESS_RANGE_LIST
26  1      0      1       OSPF_MEMORY_POOL_NEIGHBOR
27  4      0      4       OSPF_MEMORY_POOL_INTERFACE
28  0      0      1       OSPF_MEMORY_POOL_OSPF_HEADER
29  3      0      3       OSPF_MEMORY_POOL_AREA
Total Memory blocks allocated 75
```

### show ip ospf debug misc

**Syntax: show ip ospf debug misc**

This command displays miscellaneous OSPF information, including router counts and SPF calculations, as shown in the following example:

```
PowerConnect#show ip ospf debug misc
Type-5 Forwarding Addr Count :0
Imported Route Count :        1
External Route Flap Count :   0
NSSA Route Flap Count :       0
External Lsa Count :          1
NSSA Lsa Count :              0
OSPF Recalc Statistics:
        phase_number: 0, area_id: 0xffffffff, next_chunk: 0x00000000
        duration(50ms): 0
MAX_AGE EXT lsa count 0, total EXT lsa count 1
```

**show ip ospf debug graceful-restart**

**Syntax: show ip ospf debug graceful-restart**

This command displays information about OSPF graceful-restart events, as shown in the following example:

```
PowerConnect#show ip ospf debug graceful-restart
MP active: 1, standby up 0, nbr (1 0), vi (0, 0)
OSPF graceful-restart: enable 0, helper 1, timer 120/0, count 0, restarting 0
OSPF graceful-restart helper:
     Neighbor             ID      Area   Interface State Grace Helper Time
   11.1.1.1           1.1.1.1     0      1/1      8       0      0 0
OSPF graceful-restart LSA:
Area Interface         ID      Type Age   Max Seq        Interface Option
```

### *Clearing OSPF neighbors*

You can clear all OSPF neighbors or a specified OSPF neighbor using the following command.

**clear ip ospf neighbor**

**Syntax: clear ip ospf neighbor [all |** <*ip-address*>**]**

- **all** - Clears all of the OSPF neighbors on the router.
- <*ip-address*> - Clears a specific OSPF neighbor.

## OSPF debug commands

The following section describes the OSPF debug commands and shows examples of output from these commands.

**debug ip ospf**

**Syntax: [no] debug ip ospf [***A.B.C.D* **| adj | all-vrfs | bfd | error | events | flood | graceful-restart | log-debug_message | log-empty-lsa | lsa-generation | max-metric | packet | retransmission | route | sham-link | shortcuts | spf | vrf]**

- *A.B.C.D* - Displays OSPF information for a specific IP address.
- **adj** - Displays information about IP OSPF adjacencies.
- **all-vrfs** - Displays OSPF information specific to all-vrfs.
- **bfd** - Displays information about OSPF BFD events.
- **error** - Displays IP OSPF errors.
- **events** - Displays IP OSPF events.
- **flood** - Displays IP OSPF flood information.
- **graceful-restart** - Displays information about graceful restarts.
- **log-debug_message** - Displays log-debug messages.
- **log-empty-lsa** - Displays information about empty link state advertisements (LSAs).
- **lsa-generation** - Displays information about LSAs.
- **max-metric** - Displays max-metric configuration.
- **packet** - Displays IP OSPF packet information.

- **retransmission** - Displays IP OSPF retransmission information.
- **route** - Displays information about IP OSPF routes.
- **sham-link** - Displays information about sham-link configuration.
- **shortcuts** - Displays information about OSPF shortcuts for IP over MPLS.
- **spf** - Displays IP OSPF SPF information.
- **vrf** - Displays OSPF information for the specified VRF.

### debug ip ospf

**Syntax: [no] debug ip ospf** *<A.B.C.D>*

This command generates OSPF debugging information about a specific neighbor. Output indicates state transitions, hello packets received, LSA acknowledgements received, LSA processing and flooding information and database descriptions, similar to the following:

```
PowerConnect#debug ip ospf 11.1.1.1
OSPF: rvcd
11.1.1.1
OSPF: Neighbor 11.1.1.1, int 1/1, state FULL processing event HELLO_RECEIVED
hello from 11.1.1.1 area 0 on interface 11.1.1.2, state DR, DR 11.1.1.2, BDR
OSPF: Neighbor 11.1.1.1, int 1/1, state FULL processing event ADJACENCY_OK
OSPF: rcv lsa ack from neighbor 11.1.1.1, state FULL
OSPF: rcv LSA ack from 11.1.1.1, type 10, id 1.0.0.7,seq 0x80000009,adv 2.2.2.2,
age 1
```

### debug ip ospf adj

**Syntax: [no] debug ip ospf adj**

This command displays information about OSPF adjacencies and authentication, including designated router (DR) and backup designated router (BDR) elections, sent and received hello packets, neighbor state transitions, and database description information. Output resembles the following:

```
PowerConnect#debug ip ospf adj
OSPF:  adjacency events debugging is on
OSPF: rvcd hello from 11.1.1.1 area 0 on interface 11.1.1.2, state DR, DR 0.0.0.0,
BDR 0.0.0.0
OSPF: Neighbor 11.1.1.1, int 1/1,  state DOWN processing event HELLO_RECEIVED
OSPF: Neighbor 11.1.1.1 state changed from Down to Initializing - event
HELLO_RECEIVED, intf-type 1
OSPF: Neighbor 11.1.1.1, int 1/1,  state INITIALIZING processing event ONE_WAY
OSPF: send hello on area 0 interface 2.2.2.2
OSPF: send hello on area 0 interface 11.1.1.2
OSPF: Neighbor 11.1.1.1, int 1/1,  state INITIALIZING processing event
TWO_WAY_RECEIVED
OSPF: establish_adjacency with 11.1.1.1
OSPF: DR/BDR election for 11.1.1.2 on 1/1
OSPF: Run interface 11.1.1.2 DR elect, state changed to DR from DR
OSPF: interface (11.1.1.2) state = INTERFACE_DESIGNATED_ROUTER
OSPF: Neighbor 11.1.1.1, int 1/1,  state EXCHANGE_START processing event
ADJACENCY_OK
OSPF: 11.1.1.2 Flushing Network LSA if needed as we are not DR anymore, state old
5, new 5
OSPF: elect BDR(backup designated router): Router ID 1.1.1.1 IP interface 11.1.1.1
OSPF: elect DR(designated router): Router ID 2.2.2.2, IP interface 11.1.1.2
OSPF: Neighbor 11.1.1.1 state changed from Initializing to ExStart - event
TWO_WAY_RECEIVED, intf-type 1
```

**debug ip ospf all-vrfs**

**Syntax: [no] debug ip ospf all-vrfs**

This command enables OSPF debugging for all VPN routing and forwarding activity. Output is similar to that of the **debug ip ospf** commands.

**debug ip ospf bfd**

**Syntax: [no] debug ip ospf bfd**

This command displays information about OSPF BFD events.

**debug ip ospf error**

**Syntax: [no] debug ip ospf error**

This command reports the receipt of OSPF packets with errors, or mismatches between Hello packet options.

**CAUTION**

**If the router receives too many packets with errors, substantial output may be generated and severely affect system performance. To prevent a disruption of system activity, use this command only when network traffic levels are low.**

**debug ip ospf events**

**Syntax: [no] debug ip ospf events**

This command displays information about internal OSPF events related to configuration or interaction with the standby management processor and interface state transitions. Output resembles the following:

```
PowerConnect#debug ip ospf events
OSPF: Interface 1/1 (11.1.1.2) state Down processing event Interface Up
OSPF: interface 11.1.1.2 up, state changed to WAITING from Down
OSPF: Interface 1/1 (11.1.1.2) state Waiting processing event Neighbor Change
OSPF: Interface 1/1 (11.1.1.2) state Waiting processing event Backup Seen
```

**debug ip ospf flood**

**Syntax: [no] debug ip ospf flood**

This command displays information about LSA flooding activity. Output resembles the following:

```
PowerConnect#debug ip ospf flood
OSPF:  flooding debugging is on
OSPF: flood LSA Type:1 AdvRtr:2.2.2.2 Age:0 LsId:2.2.2.2
OSPF: flood advertisement throughout a specific area = 00000001
OSPF: flood LSA Type:3 AdvRtr:2.2.2.2 Age:0 LsId:0.0.0.0
OSPF: flood advertisement throughout a specific area = 00000001
OSPF: flood LSA Type:3 AdvRtr:2.2.2.2 Age:0 LsId:0.0.0.0
OSPF: flood advertisement throughout a specific area = 00000003
OSPF: flood LSA Type:3 AdvRtr:2.2.2.2 Age:0 LsId:2.2.2.2
```

**debug ip ospf graceful-restart**

**Syntax: [no] debug ip ospf graceful-restart**

Enable this command to receive information about OSPF graceful restart events, including restart phases, graceful LSA transmit and receive activity and Syslog messages, as shown in this example:

```
PowerConnect#debug ip ospf graceful-restart
Restart Router:
PowerConnectrw_mbridge_isr is called (cause = 00ff0002)
rw_isr_active_mp_lost() called
MP Manufacture Info:
=== Manufacturing Information ====
Board Class : 01 (Mgmt)
Foundry Assembly Part Number : 31524-000A
Chassis Type: NI-MLXe (ac)
Foundry Serial Number : PR30050521
Date of Manufacture : 255-255-2225
Bench Test Status : UNKNOWN
Burn-in Test Status : UNKNOWN
Manufacturing Deviation : ÿÿÿÿÿÿ
RMA Number : ÿÿÿÿÿÿÿÿÿÿ
PCB Revision : ÿÿÿÿ
MFG Test : ÿÿÿÿÿÿÿÿÿ
g_slot_presence_mask = 00000189, g_snm_presence_mask = 00000007
End Time: my_slot = 18, active_mp_slot = 18, standby_mp_slot = 17
rw_mbridge_isr is called (cause = 00fc0005)
rw_isr_present is called:
OLD: prw = 000000fc, fan = 00000000, lp = 0000fe76, snm = 00080000, mnm = 00000000
NEW: prw = 000000fc, fan = 00000000, lp = 0000fe76, snm = 00080000, mnm = 00000000
rw_isr_power is called (snapshot = 000000fc)
RW_MBRIDGE_CARD_PRESENT_REG = 0008fe76
RW_MBRIDGE_CARD_POWER_OFF_REG = 00070189
SNM1 presence detected, powering it on
SNM2 presence detected, powering it on
SNM3 presence detected, powering it on
Power on SNM1: Writing 00070189 to RW_MBRIDGE_CARD_POWER_OFF_REG
Power on SNM2: Writing 00070189 to RW_MBRIDGE_CARD_POWER_OFF_REG
Power on SNM3: Writing 00070189 to RW_MBRIDGE_CARD_POWER_OFF_REG
MGMT board temp is: 35.500C  52.625C
Power Supply 1 is Installed (OK)
Power Supply 2 is Installed (OK)
Power Supply 3 is Not Installed (FAILED)
Power Supply 4 is Not Installed (FAILED)
Power Supply 5 is Not Installed (FAILED)
Power Supply 6 is Not Installed (FAILED)
Power Supply 7 is Not Installed (FAILED)
Power Supply 8 is Not Installed (FAILED)
Write 00070189 to RW_MBRIDGE_CARD_POWER_OFF_REG
Module is up in slot 1
Module is up in slot 4
Module is up in slot 8
Module is up in slot 9
All Modules Are Up (4 total)
SYSLOG: Feb 1 23:59:19:<13>MLXe1, System: Module up in slot 4
SYSLOG: Feb 1 23:59:19:<13>MLXe1, System: Module up in slot 8
SYSLOG: Feb1 23:59:19:<13>MLXe1, System: Module up in slot 9

SYSLOG: Feb 1 23:59:19:<13>MLXe1, System: Module up in slot 1
SYSLOG: Feb 1 23:59:19:<14>MLXe1, System: Interface ethernet1/1, state up
SYSLOG: Feb 1 23:59:19:<14>MLXe1, System: Interface ethernet1/9, state up
SYSLOG: Feb 1 23:59:19:<14>MLXe1, System: Interface ethernet1/12, state up
SYSLOG: Feb 1 23:59:19:<14>MLXe1, System: Interface ethernet9/1, state up
SYSLOG: Feb 1 23:59:19:<9>MLXe1, System: Management module at slot 18 state
changed from standby to active
OSPF: switchover handoff done
OSPF: send_grace_ls to 224.0.0.5, intf addr 1.1.1.1, age 0, auth 0
```

```
OSPF: send_grace_ls to 224.0.0.5, intf addr 60.0.0.1, age 0, auth 0
OSPF: send_grace_ls to 224.0.0.5, intf addr 11.0.0.1, age 0, auth 0
OSPF: send_grace_ls to 224.0.0.5, intf addr 50.0.0.1, age 0, auth 0
OSPF: send_grace_ls to 224.0.0.5, intf addr 1.1.1.1, age 0, auth 0
OSPF: send_grace_ls to 224.0.0.5, intf addr 60.0.0.1, age 0, auth 0
OSPF: send_grace_ls to 224.0.0.5, intf addr 11.0.0.1, age 0, auth 0
OSPF: send_grace_ls to 224.0.0.5, intf addr 50.0.0.1, age 0, auth 0
OSPF: send_grace_ls to 224.0.0.5, intf addr 1.1.1.1, age 0, auth 0
OSPF: send_grace_ls to 224.0.0.5, intf addr 60.0.0.1, age 0, auth 0
OSPF: send_grace_ls to 224.0.0.5, intf addr 11.0.0.1, age 0, auth 0
OSPF: send_grace_ls to 224.0.0.5, intf addr 50.0.0.1, age 0, auth 0
OSPF: Graceful Restart setup, waiting for 2 (0) peers

SYSLOG: Feb 1 23:59:20:<14>MLXe1, System: Interface ethernetmgmt1, state up
ipc_send_mp_red_active_boot_info: reboot_needed = 0
Start code flash synchronization to standby MP.
Code flash synchronization to standby MP is done.
OSPF: GR no waiting from neighbor 60.0.0.2, interface state Waiting, DR 60.0.0.2
OSPF: Graceful Restart (1) SPF waiting for 1 (0) neighbors, 116 sec
OSPF: Graceful Restart (1) SPF waiting for 1 (0) neighbors, 115 sec
OSPF: Graceful Restart (1) SPF waiting for 1 (0) neighbors, 115 sec
OSPF: GR no waiting from neighbor 50.0.0.2, interface state Waiting, DR 50.0.0.2
OSPF: Graceful Restart (1) SPF waiting for 1 (0) neighbors, 115 sec
OSPF: Graceful Restart (1) SPF waiting for 1 (0) neighbors, 115 sec
OSPF: Graceful Restart (1) SPF waiting for 1 (0) neighbors, 114 sec
OSPF: Graceful Restart (1) SPF waiting for 1 (0) neighbors, 114 sec
OSPF: Graceful Restart (1) SPF waiting for 1 (0) neighbors, 113 sec
OSPF: Graceful Restart (1) SPF waiting for 1 (0) neighbors, 112 sec
OSPF: Graceful Restart (1) SPF waiting for 1 (0) neighbors, 112 sec
Start running config synchronization to standby MP.

Running config synchronization to standby MP is done.
OSPF: Graceful Restart (1) SPF waiting for 1 (0) neighbors, 112 sec
OSPF: Graceful Restart (1) SPF waiting for 1 (0) neighbors, 112 sec
OSPF: Graceful Restart (1) SPF waiting for 1 (0) neighbors, 111 sec
OSPF: Graceful Restart (1) SPF waiting for 1 (0) neighbors, 111 sec
OSPF: Graceful Restart (1) SPF waiting for 1 (0) neighbors, 110 sec
OSPF: Graceful Restart all none-VI neighbors back to FULL state

OSPF: GR restart phase neighbor connect Done, neighbor 0 (0), abort 0
OSPF: Graceful Restart phase neighbor full Done
OSPF: Graceful Restart phase VI neighbor full done
OSPF: restart_detected = OSPF_RESTART_STATE_SPF_ORIGINATE_LSA
OSPF: Graceful Restart reoriginate router LSA, restart_detected =
OSPF_RESTART_STATE_SPF_ORIGINATE_LSA
OSPF: restart_detected = OSPF_RESTART_STATE_SPF_ORIGINATE_LSA
OSPF: restart_detected = OSPF_RESTART_STATE_SPF_ORIGINATE_LSA
OSPF: Graceful Restart  originated router/network LSAs
OSPF: Graceful restart: start SPF
RTM: switch over done for protocol ospf
RTM: switch over done for ALL protocol
OSPF: Graceful Restart phase originate lsa DONE
OSPF: originate grace LSA, interface 1.1.1.1
OSPF: send_grace_ls to 224.0.0.5,intf addr 1.1.1.1, age 3600, auth 0
OSPF: originate grace LSA, interface 60.0.0.1
OSPF: send_grace_ls to 224.0.0.5,intf addr 60.0.0.1, age 3600, auth 0
OSPF: originate grace LSA, interface 11.0.0.1
OSPF: send_grace_ls to 224.0.0.5,intf addr 11.0.0.1, age 3600, auth 0
OSPF: originate grace LSA, interface 50.0.0.1
OSPF: send_grace_ls to 224.0.0.5,intf addr 50.0.0.1, age 3600, auth 0
```

```
OSPF: Graceful Restart phase flush lsa DONE
Standby MP is ready
OSPF: GR restart phase neighbor connect Done, neighbor 0 (0), abort 0
OSPF: Graceful Restart phase neighbor full Done
OSPF: Graceful Restart phase VI neighbor full done
OSPF: restart_detected = OSPF_RESTART_STATE_SPF_ORIGINATE_LSA
OSPF: Graceful Restart reoriginate router LSA, restart_detected =
OSPF_RESTART_STATE_SPF_ORIGINATE_LSA
OSPF: restart_detected = OSPF_RESTART_STATE_SPF_ORIGINATE_LSA
OSPF: restart_detected = OSPF_RESTART_STATE_SPF_ORIGINATE_LSA
OSPF: Graceful Restart  originated router/network LSAs
OSPF: Graceful restart: start SPF
RTM: switch over done for protocol ospf
RTM: switch over done for ALL protocol
OSPF: Graceful Restart phase originate lsa DONE
OSPF: originate grace LSA, interface 1.1.1.1
OSPF: send_grace_ls to 224.0.0.5,intf addr 1.1.1.1, age 3600, auth 0
OSPF: originate grace LSA, interface 60.0.0.1
OSPF: send_grace_ls to 224.0.0.5,intf addr 60.0.0.1, age 3600, auth 0
OSPF: originate grace LSA, interface 11.0.0.1
OSPF: send_grace_ls to 224.0.0.5,intf addr 11.0.0.1, age 3600, auth 0
OSPF: originate grace LSA, interface 50.0.0.1
OSPF: send_grace_ls to 224.0.0.5,intf addr 50.0.0.1, age 3600, auth 0
OSPF: Graceful Restart phase flush lsa DONE
Standby MP is ready
```

The following example shows output from a graceful restart on a helper router:

```
PowerConnect# Dec 15 17:29:39 OSPF: rcv GRACE LSA from 60.0.0.1, age 0, Adv
1.1.1.1
OSPF:  install new GraceLSA, int 0, neighbor 60.0.0.1, age 0
OSPF: rcv Grace_LSA from 60.0.0.1, area 0
OSPF:  neighbor 60.0.0.1 entering graceful restart state, timer 120, lsa age 0,
max 120, helping 0
OSPF: flood grace LSA, AdvRtr:1.1.1.1,  Age:0
OSPF: rcv GRACE LSA from 60.0.0.1, age 0, Adv 1.1.1.1

SYSLOG: Dec 15 17:29:43:<13>MLXe2, OSPF: nbr state changed, rid 2.2.2.2, nbr addr
60.0.0.1, nbr rid 1.1.1.1, state full
OSPF: rcv GRACE LSA from 60.0.0.1, age 3600, Adv 1.1.1.1
OSPF: LSA flush rcvd Type:9 AdvRtr:1.1.1.1 LsId:3.0.0.0
OSPF:  install new GraceLSA, int 0, neighbor 60.0.0.1, age 3600
OSPF: rcv Grace_LSA from 60.0.0.1, area 0
OSPF:  neighbor 60.0.0.1 exiting graceful restart state, timer 120, lsa age 3600,
max 3600,
helping 0
OSPF: flood grace LSA, AdvRtr:1.1.1.1,  Age:3600
OSPF: age out GraceLSA, from 1.1.1.1, age 3600
OSPF: remove grace LSA, age 3600
```

### debug ip ospf log-debug-message

**Syntax: [no] debug ip ospf log-debug-message**

This command logs instances when large (greater than MTU) LSA update messages are sent or received.

### debug ip ospf log-empty-lsa

**Syntax: [no] debug ip ospf log-empty-lsa**

This command logs instances when empty or truncated LSA update messages are sent or received.

### debug ip ospf lsa-generation

**Syntax: [no] debug ip ospf lsa-generation**

This command generates information about LSAs in output similar to the following example:

```
PowerConnect#debug ip ospf lsa-generation
Jan 15 16:35:25 OSPF: install a new lsa, type 3, ls_id 0.0.0.0, age 0, seq
80000003 area-id 10
Jan 15 16:35:25 OSPF: NSR : Sync node add, type 3, ls_id 0.0.0.0, age 0, seq
80000003
Jan 15 16:35:25 OSPF: originate router LSA, area 0
Jan 15 16:35:25 OSPF: install a new lsa, type 1, ls_id 2.2.2.2, age 0, seq
80000004 area-id 0
Jan 15 16:35:25 OSPF: NSR : Sync node add, type 1, ls_id 2.2.2.2, age 0, seq
80000004
Jan 15 16:35:25 OSPF:NSR Sync ACK received for LSA
Jan 15 16:35:25 OSPF:ls_header.id 0.0.0.0 type 3 ToBesyncedState 2
Jan 15 16:35:25 OSPF:NSR Sync ACK received for LSA
Jan 15 16:35:25 OSPF:ls_header.id 2.2.2.2 type 1 ToBesyncedState 2
Jan 15 16:35:25 OSPF: install a new lsa, type 3, ls_id 37.0.0.0, age 0, seq
80000001 area-id 1
Jan 15 16:35:25 OSPF: NSR : Sync node add, type 3, ls_id 37.0.0.0, age 0, seq
80000001
Jan 15 16:35:25 OSPF:NSR Sync ACK received for LSA
Jan 15 16:35:25 OSPF:ls_header.id 37.0.0.0 type 3 ToBesyncedState 2
Jan 15 16:35:28 OSPF: redistribute into ospf26.0.0.0 with ffffff00 forwarding
address 0.0.0.0
Jan 15 16:35:28 OSPF: originate external lsa26.0.0.0 with ffffff00
Jan 15 16:35:28 OSPF: install a new lsa, type 5, ls_id26.0.0.0, age 0, seq
80000001 area-id 0
Jan 15 16:35:28 OSPF: NSR : Sync node add, type 5, ls_id26.0.0.0, age 0,
seq80000001
Jan 15 16:35:29 OSPF:NSR Sync ACK received for LSA
Jan 15 16:35:29 OSPF:ls_header.id 26.0.0.0 type 5 ToBesyncedState2
```

This output indicates that the sequence number (seq) is a unique identifier for each LSA. When a router initiates an LSA, it includes a sequence number, which is recorded in the link state database of every receiving router. If a router receives an LSA that is already in the database and that has the same sequence number, the received LSA is discarded. If the information is the same but the sequence number is greater, the LSA information and new sequence number are entered into the database and the LSA is flooded. Sequence numbers allow LSA flooding to stop when all routers have received the most recent LSA.

**NOTE**
This command can be enabled on Standby MP as well.

### debug ip ospf packet

**Syntax: [no] debug ip ospf packet**

This command generates information about OSPF packets. Output resembles the following:

```
PowerConnect# debug ip ospf packet
OSPF: recv from:69.28.172.17 Intf:eth 2/1 Hello  L:44 A:2 Rid:69.28.156.234
DR:69.28.172.17 BDR:0.0.0.0
```

This output describes an OSPF packet received from Ethernet interface 2/1 on router 69.28.172.17. Descriptors include: L = packet length, A = authentication, Rid = router ID, DR = designated router ID, and BDR= backup designated router ID.

### debug ip ospf retransmission

**Syntax: [no] debug ip ospf retransmission**

This command generates internal information about OSPF retransmission of LSAs. Output resembles the following:

```
PowerConnect#debug ip ospf retransmission
OSPF: retransmission debugging is on
OSPF: examine each neighbor and add advertisement to the retransmission list if
necessary
OSPF: remove current database copy from all neighbors retransmission lists
```

### debug ip ospf route

**Syntax: [no] debug ip ospf route**

This command generates network-specific information during Dijkstra computation, routing table calculation, and LSA origination. This command is useful for tracking a specific OSPF prefix. Output resembles the following:

```
PowerConnect#debug ip ospf route 1.1.1.1
OSPF: debug ospf route 1.1.1.1
OSPF: Orig summary LSA to area 0, route 1.1.1.1, type 1, prem 1...
OSPF: Originating type 4 summary LSA to area 0, route 1.1.1.1
OSPF: Orig summary LSA to area 1, route 1.1.1.1, type 1, prem 1...
OSPF: Orig summary LSA to area 3, route 1.1.1.1, type 1, prem 1...
OSPF: delete route 1.1.1.1 from rtm 0x053742b0, not_in_main 0
```

### debug ip ospf sham-link

**Syntax: [no] debug ip ospf sham-link**

This command generates information about OSPF sham-links. A sham-link is required between any two VPN sites that belong to the same OSPF area and share an OSPF backdoor link. If no backdoor link exists between the sites, no sham-link is required.

### debug ip ospf shortcuts

**Syntax: [no] debug ip ospf shortcuts**

This command generates information about OSPF shortcuts for IP over MPLS. Output resembles the following:

```
PowerConnect#debug ip ospf shortcuts
OSPF: Clearing OSPF DSPT Route Table, num of entries 5
OSPF: Clearing OSPF DSPT Route Table completed, num of entries 5
```

### debug ip ospf spf

**Syntax: [no] debug ip ospf spf**

This command generates information about OSPF SPF activity including SPF runs and calculations. Output resembles the following:

**debug ip rtm**

**Syntax: [no] debug ip rtm [***A.B.C.D* **| all | errors| nexthop]**

- *A.B.C.D* - Displays RTM information for a specified IPv4 address.
- **all** - Displays all RTM information.
- **errors** - Displays RTM errors.
- **nexthop** - Logs various next-hop related events to the console.

This command displays information about the routing table manager (RTM), including changes in the routing table. With **debug ip rtm** enabled, and using the **show ip route** command, output resembles the following examples for specific routing table activity.

```
PowerConnect#debug ip rtm
IP: rtm debugging is on
PowerConnect#show ip route
Total number of IP routes: 9
Type Codes - B:BGP D:Connected I:ISIS S:Static R:RIP O:OSPF; Cost - Dist/Metric
Destination Gateway Port Cost Type
...
7 11.11.11.0/24 10.10.10.2 eth 1/1 110/10 O2
11.11.11.0/24 10.10.11.2 eth 1/1 110/10 O2
...
RTM: Remove 11.11.11.0/24 (ospf) from rtm
RTM: un-install 11.11.11.0/24 (ospf) in rtm
```

This example indicates that OSPF route 11.11.11.0/24 has been deleted from the route table.

```
RTM: Add 11.11.11.0/24 (ospf) to rtm, path 2
RTM: install 11.11.11.0/24 (ospf) in rtm
```

This example indicates that OSPF route 11.11.11.0/24 has been added to the route table with 2 paths.

```
RTM: pack tree node VRF 0 11.11.11.0/24
```

This example indicates that the change for route 11.11.11.0/24 has been sent to the line card.

```
RTM: Modify 11.11.11.0/24 (ospf) in rtm
RTM: install 11.11.11.0/24 (ospf) in rtm
```

This example indicates that OSPF route 11.11.11.0/24 has been modified and added to the routing table.

## Configuration notes

- If a router is to operate as an ASBR, you must enable the ASBR capability at the system level.
- All router ports must be assigned to one of the defined areas on an OSPF router. When a port is assigned to an area, all corresponding subnets on that port are automatically included in the assignment.
- If a PowerConnect B-MLXe router is to function as a graceful restart router, a secondary management module must be installed. If the router functions as a graceful restart helper router only, a second management module is not necessary.
- If you disable OSPF, the PowerConnect B-MLXe removes all OSPF configuration information from the running configuration. In addition, after disabling OSPF, when you save the configuration to the startup configuration file, all configuration information for OSPF is removed from the startup configuration file.

  The CLI displays a warning message similar to the following:

```
PowerConnect(config-ospf-router)# no router ospf
router ospf mode now disabled. All ospf config data will be lost when writing
to
flash!
```

If you have disabled OSPF but have not yet saved the startup configuration file and reloaded the software, you can restore the configuration information by re-entering the **router ospf** command to enable the protocol. If you have already saved the startup configuration file and reloaded the software, the information is gone.

If you are testing an OSPF configuration and are likely to disable and re-enable OSPF, you may want to make a backup copy of the startup configuration file before you begin. Then, if you remove the configuration information by saving the configuration after disabling the protocol, you can restore the configuration by saving the backup copy of the startup configuration file onto the flash memory.

## Common diagnostic scenarios

- A third party router and a Dell router are both receiving bad OSPF packets.

  This indicates that the carrier may be the source of the corruption.

- Frequent OSPF link flapping events.

  This issue was resolved by upgrading the software version to include the latest patches.

- Old software versions.

  Feature issues are often caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# IS-IS

Intermediate System to Intermediate System (IS-IS) is a link-state interior gateway protocol. IS-IS designates an intermediate system (router) as either a Level 1 or Level 2 router. A Level 1 router routes traffic only within the area where it resides. A Level 2 router routes traffic between areas within a routing domain.

**NOTE**
The PowerConnect B-MLXe device does not support routing of Connectionless-Mode Network Protocol (CLNP) packets. The PowerConnect B-MLXe device uses IS-IS for TCP/IP only.

Detailed IS-IS configuration instructions can be found in the *NetIron Series Configuration Guide*.

## IS-IS show commands

This section describes the **show** commands that display information about IS-IS activity and configurations. Some of these are regular user commands, and some are debug commands.

### show isis

**Syntax: show isis**

This command displays general IPv4 IS-IS information, as shown in this example:

```
PowerConnect#show isis
 IS-IS Routing Protocol Operation State: Enabled
IS-Type: Level-1-2
System ID: 1111.1111.1111
Manual area address(es):47
Level-1-2 Database State: On
Administrative Distance: 115
Maximum Paths: 4
Default redistribution metric: 0
Protocol Routes redistributed into IS-IS: Static
Number of Routes redistributed into IS-IS: 11
Level-1 Auth-mode: None
Level-2 Auth-mode: None
Metric Style Supported for Level-1: Wide
Metric Style Supported for Level-2: Wide
IS-IS Partial SPF Optimizations: Enabled
Timers:
L1 SPF: Max-wait 120s Init-wait 100ms Second-wait 120000ms
L2 SPF: Max-wait 100s Init-wait 100ms Second-wait 100000ms
L1 SPF is not scheduled
L2 SPF is not scheduled
PSPF: Max-wait 120000ms Init-wait 120000ms Second-wait 120000ms
PSPF is not scheduled
 LSP: max-lifetime 1200s, refresh-interval 900s, gen-interval 10s
retransmit-interval 5s, lsp-interval 33ms
SNP: csnp-interval 10s, psnp-interval 2s
Global Hello Padding : Enabled
Global Hello Padding For Point to Point Circuits: Enabled
Ptpt Three Way HandShake Mechanism: Enabled
IS-IS Traffic Engineering Support: Disabled
BFD: Disabled
Interfaces with IPv4 IS-IS configured:
eth 1/1
```

### show isis interface

**Syntax: show isis interface**

This command displays information about IS-IS interfaces, as shown in the following example:

```
PowerConnect#show isis interface
Total number of IS-IS Interfaces: 2
Interface : gre_tnl 1
   Circuit State: UP Circuit Mode: LEVEL-1-2
   Circuit Type : PTP Passive State: FALSE
   Circuit Number: 0x02, MTU: 1497
   Level-1 Metric: 10, Level-1 Priority: 64
   Level-1 Auth-mode: None
   Level-2 Auth-mode: None
   Level-1 Metric: 10, Level-1 Priority: 50
   Level-1 Hello Interval: 10 Level-1 Hello Multiplier: 3
   Level-1 Designated IS: MLXe1-02 Level-1 DIS Changes: 0
   Level-2 Metric: 10, Level-2 Priority: 50
   Level-2 Hello Interval: 10 Level-2 Hello Multiplier: 3
   Level-2 Designated IS: MLX2-02 Level-2 DIS Changes: 0
   Circuit State Changes: 1 Circuit Adjacencies State Changes: 1
   Rejected Adjacencies: 0
   Circuit Authentication L1 failures: 0
   Circuit Authentication L2 failures: 0
   Bad LSPs 0
   Control Messages Sent: 318 Control Messages Received: 229
   IP Enabled: TRUE
   IP Address and Subnet Mask:
    50.50.50.20         255.255.255.0
   IPv6 Enabled: FALSE
```

### show isis neighbor

**Syntax: show isis neighbor**

This command displays information about IS-IS neighbors, as shown in the following example:

```
PowerConnect#show isis neighbor
Total number of IS-IS Neighbors: 3
System Id      Interface  SNPA            State Holdtime Type  Pri    StateChgeTime
0000.0000.0004 eth 6/2    0000.0576.4805  UP    30       ISL1  0   0  :0 :8 :42
0000.0000.0004 eth 6/2    0000.0576.4805  UP    30       ISL2  0   0  :0 :8 :42
MLXe1           gre_tnl 1  0900.2b00.0005  UP    30       PTPT  127 0  :0 :9 :16
```

### show isis debug

**Syntax: show isis debug [adj-options-order |adj-timer | ip-nexthop-set | ipv6-nexthop-set | lsp-list | lsp-timer | memory | nexthops | pent | pent-level-info | pspf-lsp-list | redis | route-info | summary | v6-nexthops | v6route-info]**

- **adj-options-order** - Displays IS-IS adjacency options in self LSP.
- **adj-timer** - Displays IS-IS adjacency hold timers.
- **ip-nexthop-set** - Displays IS-IS IP next-hop information.
- **ipv6-nexthop-set** - Displays IS-IS IPv6 nexthop information.
- **lsp-list** - Displays IS-IS LSP list debugging information.
- **lsp-timer** - Displays IS-IS LSP hold timers.
- **memory** - Displays IS-IS memory debugging information.

- **nexthops** - Displays IS-IS nexthop lists debugging information.
- **pent** - Displays IS-IS SPD path entries.
- **pent-level-info** - Displays integrated IS-IS level information list associated with path entries.
- **pspf-lsp-list** - Displays integrated IS-IS PSPF LSP list.
- **redis** - Displays IS-IS redistribution debugging information.
- **route-info** - Displays integrated IS-IS route information list.
- **summary** - Displays a summary of IS-IS debugging information.
- **v6-nexthops** - Displays debugging information for IS-IS nexthop lists.
- **v6route-info** - Displays the IS-IS route information list.

The output resembles the following:

```
PowerConnect(config-isis-router)#show isis debug
Router-id: 140.140.140.4
Tics: 168833
[SPF: Act:NOT RUNNING Bld:N  Run[N,N]
Code assertions are ON
Ptpt 3way HandShake Enabled
Manual Area Addresses: 57(1) 00.0057(3) 00.0001(3)
Union Area Addresses: 00.0001(3) 00.0057(3) 57(1)
isis.run_pspf=N, Ir_pspf_leve11=N, Ir_pspf_level2=Y
PSPF bucket count: Reached Maximum
L1 SPF bucket count: 0
L2 SPF bucket count: 0
ISIS IPv4 Default Entry 00000000
ISIS IPv6 Default Entry 00000000
PSPF Newevent=Y, Masklevel=0, Timerstate=R, RemainingTime=400
L1-SPF Newevent=N, Masklevel=0, Timerstate=S, RemainingTime=0, FailCount=0
L2-SPF Newevent=N, Masklevel=0, Timerstate=S, RemainingTime=0, FailCount=0
isis.l1default: 0 isis_ip6.ip6_l1default: 0
IPv4 L1 SPF Uses:Native Topology, L2 SPF Uses:Native Topology
IPv6 L1 SPF Uses:Native Topology, L2 SPF Uses:Native Topology
NSR State: Normal
isis.sync_instance.sync_enabled: TRUE
isis.sync_instance.asi.peer_device_status 2/Ready
```

The following examples show output from some of the **show isis debug** commands.

### show isis debug adj-options-order

**Syntax: show isis debug adj-options-order**

This command displays IS-IS adjacency options in self LSP.

```
PowerConnect(config-isis-router)#show isis debug adj-options-order
Level-1 List
 LSP-ID Dut2.00-00
        Metric:  10       IS Dut2.01
 LSP-ID Dut2.01-00
        Metric:  0        IS Dut2.00
Level-2 List
 LSP-ID Dut2.00-00
        Metric:  10       IS Dut2.01
 LSP-ID Dut2.01-00
        Metric:  0        IS Dut2.00
        Metric:  0        IS Dut4.00
```

### show isis debug pspf-lsp-list

**Syntax: show isis debug pspf-lsp list <*level*>**

- <*level*> - Displays information for a specific list level.

This command displays the LSPs in the partial SPF list. The argument "level" is optional. If you do not mention the argument level, the LSPs in both the Level-1 and Level-2 lists are displayed.

```
PowerConnect(config-isis-router)#show isis debug  pspf-lsp-list
 ISIS Level-1 PSPF LSP List
LSP-ID            State      Time-Stamp(Tics)           Last-Trans-Time(Tics)
mu2.00-00         Update     0m0s      (1202772185)      1h10m9s   (0)
mu2.00-01         New        0m0s      (1202772185)     1h10m9s   (0)

ISIS Level-2 PSPF LSP List
LSP-ID             State      Time-Stamp(Tics)           Last-Trans-Time(Tics)
mu2.00-00         Update      0m0s      (1202772185)     1h10m9s   (0)
mu2.00-01         New         0m0s      (1202772185)     1h10m9s   (0)
```

### show isis debug adj-timer

**Syntax: show isis debug adj-timer**

This command is meant for use by Dell developers.

```
PowerConnect#show isis debug adj-timer
Summary:
WheelTimer: cur_time 371080, cur_slot 280, num_slots 400, active_slots 11
        Callbacks[Tmo=0x08acb064,Print=0x08acb1a0]
        Buckets Callbacks[Ins=0x0849f108,Rem=0x0849f178,GetRdy=0x0849f1d0]
        Ready Queue: Empty
        Total(Rdy+Slots):    BuckNodes 11, ElemNodes 14
        Avg/Active Slot:     BuckNodes 1, ElemNodes 1
        Slot HighWaterMark:  BuckNodes 1, ElemNodes 2
```

### show isis debug ip-nexthop-set

**Syntax: show isis debug ip-nexthop-set**

This command displays the IP address of each set of next-hops in all sets of next-hops. The maximum number of hops in a set is eight.

```
PowerConnect#show isis debug ip-nexthop-set
Set 1 with No Of Nexthops 1 Address 277ac1a4 up since 0  :0 :0 :6
   Nexthop IPAddr 30.1.1.3, Circ-id 0(eth 1/1)
Pent List Pointing to this Nexthop Set
   Pent-Id MLXe16.00-00 level-1
   Pent-Id MLXe16.01-00 level-1
```

### show isis debug ipv6-nexthop-set

**Syntax: show isis debug ipv6-nexthop-set**

This command displays the IPv6 next-hop set, as the following example illustrates:

```
PowerConnect#show isis debug ipv6-nexthop-set
Set 1 with No Of Nexthops 1 Address 277ac1e0 up since 0  :0 :0 :43
                Nexthop IPAddr fe80::20c:dbff:fef6:3300, Circ-id 0(eth 1/1)
        Pent List Pointing to this Nexthop Set
                Pent-Id MLXe16.00-00 level-1
                Pent-Id MLXe16.01-00 level-1
```

### show isis debug lsp-list

**Syntax: show isis debug lsp-list**

This command displays the number of instances of certain IS-IS items, such as the number of LSPs in each hash table, the number of partial sequence numbers (PSNPs), and so on.

```
PowerConnect#show isis debug lsp-list
sizeof(LSPI) = 124, LSPI_SIZE = 382, nd_srm_size = 129
LSP Hash L1 Count: 1
LSP Hash L2 Count: 39
LSP Sort L1 Count: 1
LSP Sort L2 Count: 39
LSP PSNP List Count: 0
LSP Tx List Count: 0
LSP Flood Count: 25
```

### show isis debug lsp-timer

**Syntax: show isis debug lsp-timer**

This command is meant for use by Dell developers, and displays information about the LSP-timer, as shown in the following example:

```
PowerConnect#show isis debug lsp-timer
Summary:
WheelTimer: cur_time 371574, cur_slot 374, num_slots 400, active_slots 19
        Callbacks[Tmo=0x08ad8438,Print=0x08ad8498]
        Buckets Callbacks[Ins=0x0849f108,Rem=0x0849f178,GetRdy=0x0849f1d0]
        Ready Queue: Empty
        Total(Rdy+Slots):    BuckNodes 19, ElemNodes 32
        Avg/Active Slot:     BuckNodes 1, ElemNodes 1
        Slot HighWaterMark:  BuckNodes 1, ElemNodes 3
```

### show isis debug memory

**Syntax: show isis debug memory**

This command displays various dimensions of memory. The displays memory usage in bytes. Primarily, you would note any indication of a failure or error. If you noticed errors, then the other items in the display might lead to a part of memory related to the problem.

```
PowerConnect#show isis debug memory
 Total IS-IS Memory In Use: 3050881
 Total P2 Route Memory In Use: 464928
 Total P2 Other Memory In Use: 0
 Total Memory Allocated: 10569
 Total Memory Allocation Failed: 0
 Total Packet Buffer Allocated: 0
```

```
Total Packet Buffer Allocation Failed: 0
Errors in freeing memory (bad addr): 0
Errors in freeing memory (bad pool-id): 0
Maximum Memory IS-IS allowed to use: 104857600
```

### show isis debug memory pool

**Syntax: show isis debug memory pool**

This command displays information about the memory pool.

```
PowerConnect# show isis debug memory pool
Pool# 0 @ 0x0928ef4c
blk_size: 292,  initial_blk_cnt: 60,  exp_blk_cnt: 20
curr_#_of_blks 60,  #_of_sub_pools: 1
#_of_blks_in_use: 3,  #_of_blks_free: 57,  #_of_mem_alloc_failed: 0
total_memory_allocated_for_this_pool: 17524
sptr_memory_list: 0x28ca6370, &0x0928ef70
sptr_sub_pool_list: 0x28ca6000 &0x0928ef74
Pool# 1 @ 0x0928ef78
blk_size: 68,  initial_blk_cnt: 60,  exp_blk_cnt: 20
curr_#_of_blks 60,  #_of_sub_pools: 1
#_of_blks_in_use: 0,  #_of_blks_free: 60,  #_of_mem_alloc_failed: 0
total_memory_allocated_for_this_pool: 4084
sptr_memory_list: 0x28cab004, &0x0928ef9c
sptr_sub_pool_list: 0x28cab000 &0x0928efa0
Pool# 2 @ 0x0928efa4
blk_size: 44,  initial_blk_cnt: 1000,  exp_blk_cnt: 250
curr_#_of_blks 1000,  #_of_sub_pools: 1
#_of_blks_in_use: 1,  #_of_blks_free: 1000,  #_of_mem_alloc_failed: 0
total_memory_allocated_for_this_pool: 44004
sptr_memory_list: 0x28cac004, &0x0928efc8
sptr_sub_pool_list: 0x28cac000 &0x0928efcc
Pool# 3 @ 0x0928efd0
blk_size: 285,  initial_blk_cnt: 60,  exp_blk_cnt: 20
curr_#_of_blks 60,  #_of_sub_pools: 1
#_of_blks_in_use: 1,  #_of_blks_free: 59,  #_of_mem_alloc_failed: 0
total_memory_allocated_for_this_pool: 17104
sptr_memory_list: 0x28cb7121, &0x0928eff4
sptr_sub_pool_list: 0x28cb7000 &0x0928eff8
Pool# 4 @ 0x0928effc
blk_size: 471,  initial_blk_cnt: 256,  exp_blk_cnt: 20
curr_#_of_blks 256,  #_of_sub_pools: 1
#_of_blks_in_use: 4,  #_of_blks_free: 252,  #_of_mem_alloc_failed: 0
total_memory_allocated_for_this_pool: 120580
sptr_memory_list: 0x28cbc760, &0x0928f020
sptr_sub_pool_list: 0x28cbc000 &0x0928f024
Pool# 5 @ 0x0928f028
blk_size: 121,  initial_blk_cnt: 1024,  exp_blk_cnt: 20
curr_#_of_blks 1024,  #_of_sub_pools: 1
#_of_blks_in_use: 0,  #_of_blks_free: 1024,  #_of_mem_alloc_failed: 0
total_memory_allocated_for_this_pool: 123908
sptr_memory_list: 0x28cda004, &0x0928f04c
sptr_sub_pool_list: 0x28cda000 &0x0928f050
.
.
.
```

### show isis debug nexthops

**Syntax: show isis debug nexthops**

This command displays all of the next hops available on the system.

```
PowerConnect#show isis debug nexthops
IS-IS IP Nexthops List:
  Node (2185b00c) -> Nexthop (29124050) ref 3, If 158(eth 4/15) Addr 129.0.0.38
  Node (2185b030) -> Nexthop (291240c0) ref 3, If 146(eth 4/3) Addr 129.0.0.14
  Node (2185b0b4) -> Nexthop (291240e0) ref 3, If 42(eth 1/43) Addr 41.1.6.2
  Node (2185b06c) -> Nexthop (29124090) ref 1, If 14(eth 1/15) Addr 41.1.1.2
  Node (2185b060) -> Nexthop (291240a0) ref 1, If 150(eth 4/7) Addr 129.0.0.30
  Node (2185b03c) -> Nexthop (291240b0) ref 1, If 154(eth 4/11) Addr 129.0.0.18
  Node (2185b0e4) -> Nexthop (29124100) ref 1, If 157(eth 4/14) Addr 129.0.0.34
  Node (2185b1bc) -> Nexthop (29124240) ref 3, If 17(eth 1/18) Addr 41.1.2.2
  Node (2185b1d4) -> Nexthop (29124250) ref 2, If 152(eth 4/9) Addr 129.0.0.25
  Node (2185b1ec) -> Nexthop (29124260) ref 2, If 155(eth 4/12) Addr 129.0.0.46
  Node (2185b3e4) -> Nexthop (29124410) ref 1, If 144(eth 4/1) Addr 129.0.0.21
  Node (2185b288) -> Nexthop (29124480) ref 1, If 147(eth 4/4) Addr 129.0.0.6
  Node (2185b1b0) -> Nexthop (291243e0) ref 3, If 49(eth 2/2) Addr 129.0.0.10
```

### show isis debug pent

**Syntax: show isis debug pent**

This command displays path entries. It displays all the nodes in the topology and the cost to each node from the root node, the preference, flags, and the IPv4 and IPv6 next hop associations, as the following example illustrates:

```
PowerConnect#show isis debug pent
Path Table for Level 1:
Pent 2aa8e008
 Hash-Idx 34 PENT_IS  R2-0-0    cost 0 pref 1  flags 0
                No IP Nexthops associated with this Pent entry
                No IPv6 Nexthops associated with this Pent entry
Path Table for Level 2:
Pent 2aa8e2b0
 Hash-Idx 0 PENT_IS  R4-3-0     cost 61 pref 2  flags 0
        Pent IPv4 Nexthop Set 2c2e1020
                24.3.1.4 eth 3/1
                No IPv6 Nexthops associated with this Pent entry
Pent 2aa8e118
 Hash-Idx 0 PENT_IS  R4-0-0     cost 1 pref 2  flags 0
        Pent IPv4 Nexthop Set 2c2e1020
                24.3.1.4 eth 3/1
                No IPv6 Nexthops associated with this Pent entry
Pent 2aa8e448
 Hash-Idx 0 PENT_IS  R1-3-0     cost 120 pref 2  flags 0
        Pent IPv4 Nexthop Set 2c2e1030
                12.37.1.1 eth 3/7
                No IPv6 Nexthops associated with this Pent entry
```

### show isis debug pent-level-info

**Syntax: show isis debug pent-level-info**

The **show isis debug pent-level-info** command displays the nodes and the IPv4 and IPv6 prefixes that they advertise in their prefixes (along with costs and flags), as the following example illustrates:

```
PowerConnect#show isis debug pent-level-info
 Pent-Id   level metric pref Chg(N-D-MC-PC-IPV4NC-IPV6NC-PSPFC)
 MLXe2.00-00 L2      10     2 (0-0-0-0-0-0-0)
        135.0.0.0/255.255.255.252 cost:306, Pre:2, Up/Down:0 flags:Act Ena   IPv4
        135.0.0.4/255.255.255.252 cost:10, Pre:2, Up/Down:0 flags:Act Ena   IPv4
        129.0.0.36/255.255.255.252 cost:1000, Pre:2, Up/Down:0 flags:Act Ena   IPv
        192.0.0.0/255.255.255.0 cost:10, Pre:2, Up/Down:0 flags:Act Ena   IPv4
        126.0.0.0/255.255.255.0 cost:10, Pre:2, Up/Down:0 flags:Act Ena   IPv4
        41.1.6.0/255.255.255.0 cost:10, Pre:2, Up/Down:0 flags:Act Ena   IPv4
        124.0.0.8/255.255.255.252 cost:10, Pre:2, Up/Down:0 flags:Act Ena   IPv4
        124.0.0.12/255.255.255.252 cost:10, Pre:2, Up/Down:0 flags:Act Ena   IPv4
        129.0.0.12/255.255.255.252 cost:10, Pre:2, Up/Down:0 flags:Act Ena   IPv4
        125.0.0.0/255.255.255.0 cost:10, Pre:2, Up/Down:0 flags:Act Ena   IPv4
        22.22.22.22/255.255.255.255 cost:10, Pre:2, Up/Down:0 flags:Act Ena   IPv4
 GSR.00-00 L2      20     2 (0-0-0-0-0-0-0)
        27.0.1.0/255.255.255.0 cost:10, Pre:2, Up/Down:0 flags:Act Ena   IPv4
        28.1.1.0/255.255.255.0 cost:10, Pre:2, Up/Down:0 flags:Act Ena   IPv4
        9.9.9.9/255.255.255.255 cost:10, Pre:2, Up/Down:0 flags:Act Ena   IPv4
        25.25.25.0/255.255.255.0 cost:10, Pre:2, Up/Down:0 flags:NoAct Ena   IPv4
```

### show isis debug redis

**Syntax: show isis debug redis**

This command shows all routes that have been added, deleted, and so on, from other protocols.

```
PowerConnect#show isis debug redis
ISIS Redistribution Stats:
   Add 20, Modify 0, Del 0, Clear 0, Clear-All 0, Invalid 0
   Invalid-Add 0, Invalid-Mod 0
Prefix Mask level rt_type cost met_type
100.1.10.0/24 3 0 1 1
200.1.2.0/24 3 0 1 1
200.1.5.0/24 3 0 1 1
100.1.2.0/24 3 0 1 1
100.1.5.0/24 3 0 1 1
200.1.8.0/24 3 0 1 1
100.1.8.0/24 3 0 1 1
100.1.11.0/24 3 0 1 1
200.1.3.0/24 3 0 1 1
100.1.3.0/24 3 0 1 1
200.1.6.0/24 3 0 1 1
200.1.9.0/24 3 0 1 1
100.1.6.0/24 3 0 1 1
100.1.9.0/24 3 0 1 1
100.1.1.0/24 3 0 1 1
200.1.4.0/24 3 0 1 1
200.1.7.0/24 3 0 1 1
100.1.4.0/24 3 0 1 1
100.1.7.0/24 3 0 1 1
200.1.10.0/24 3 0 1 1
```

### show isis debug route-info

**Syntax: show isis debug route-info**

This command displays the routes in isis route_table. For each route, it displays the next hops, the cost from the root node, flags, and its parent path entries, as the following example shows.

```
PowerConnect#show isis debug route-info
Total number of IS-IS routes: 47
Destination      Mask             Cost      Type Tag      Flags
9.9.9.9          255.255.255.255  30        L2   00000000 00000208
   Path: 1       Next Hop IP: 41.1.1.2          Interface: 1/15
   Path: 2       Next Hop IP: 129.0.0.30        Interface: 4/7
   Path: 3       Next Hop IP: 129.0.0.18        Interface: 4/11
   Path: 4       Next Hop IP: 129.0.0.34        Interface: 4/14
  Route 27ebd6ae
  Level-1 Info List is empty
  Level-2 Info List
        PENT-V4:GSR.00-00 cost:10, Pre:2, Up/Down:0 flags:Act Ena   IPv4
22.22.22.22      255.255.255.255  20        L2   00000000 00000208
   Path: 1       Next Hop IP: 41.1.6.2          Interface: 1/43
   Path: 2       Next Hop IP: 129.0.0.14        Interface: 4/3
   Path: 3       Next Hop IP: 129.0.0.38        Interface: 4/15
  Route 27ebd28c
  Level-1 Info List is empty
  Level-2 Info List
        PENT-V4:MLXe2.00-00 cost:10, Pre:2, Up/Down:0 flags:Act Ena   IPv4
25.25.25.0       255.255.255.0    20        L2   00000000 00000208
   Path: 1       Next Hop IP: 41.1.1.2          Interface: 1/15
   Path: 2       Next Hop IP: 129.0.0.30        Interface: 4/7
   Path: 3       Next Hop IP: 129.0.0.18        Interface: 4/11
   Path: 4       Next Hop IP: 129.0.0.34        Interface: 4/14
  Route 27ebd064
  Level-1 Info List is empty
  Level-2 Info List
        PENT-V4:MLXe5.00-00 cost:10, Pre:2, Up/Down:0 flags:Act Ena   IPv4
        PENT-V4:GSR.00-00 cost:10, Pre:2, Up/Down:0 flags:NoAct Ena   IPv4
```

### show isis debug summary

**Syntax: show isis debug summary**

This command displays summary information about redistributed routes, such as whether a route is active (UP=yes or no), the metric (cost), the level (L column), and so on.

```
PowerConnect#show isis debug summary
Summary           UP Met L
100.1.0.0      /16 Y   1     3    (e1=11,e2=11,i1=0,i2=0) (l1=12e0fc34/1 l2=12e0fb14/1)
```

### show isis debug v6-nexthops

**Syntax: show isis debug v6-nexthops**

This command displays about IS-IS IPv6 next hops:

```
PowerConnect#show isis debug v6-nexthops
IS-IS IPv6 Nexthops List:
IS-IS IPv6 Null0 Nexthops List:
Node 2185b000 -> Nexthop 2809f000 ref 0, If 65534(drop) Addr ::
IS-IS IPv6 Nexthops Set List:
```

### show isis debug v6route-info

**Syntax: show isis debug v6route-info**

This command displays the routes in isis ipv6 route table. For each route, it displays the next hops, the cost from the root node, flags, and its Parent Path entries, as the following example shows:

```
PowerConnect#show isis debug v6route-info
ISIS IPv6 Routing Table
Total Routes: 1  Level1: 0 Level2: 1 Equal-cost multi-path: 1
Type IPv6 Prefix                Next Hop Router           Interface  Cost
L2   5000:2000::/64             fe80::202:17ff:fe6e:c41c  eth 1/11   0
        PENT:Cisco.00-00 cost:10, Pre:2, Up/Down:0 flags:Act Ena   IPv6
```

### show ip route isis

**Syntax: show ip route isis**

This command displays information about IS-IS routes. For example:

```
PowerConnect#show ip route isis
Type Codes - B:BGP D: Connected I: ISIS S: Static R: RIP O:O SPF; Cost - Dist/Metric
        Destination        Gateway         Port         Cost       Type
1       30.30.30.0/24      50.50.50.10     gre_tnl 1    115/20     IL1
2       100.100.100.0/24   50.50.50.10     gre_tnl 1    115/20     IL1
3       100.100.101.0/24   50.50.50.10     gre_tnl 1    115/20     IL1
4       100.100.102.0/24   50.50.50.10     gre_tnl 1    115/20     IL1
5       100.100.103.0/24   50.50.50.10     gre_tnl 1    115/20     IL1
6       100.100.104.0/24   50.50.50.10     gre_tnl 1    115/20     IL1
7       100.100.105.0/24   50.50.50.10     gre_tnl 1    115/20     IL1
8       100.100.106.0/24   50.50.50.10     gre_tnl 1    115/20     IL1
9       100.100.107.0/24   50.50.50.10     gre_tnl 1    115/20     IL1
```

### show isis name hostname

**Syntax: show isis name hostname**

This command displays IS-IS name mappings, as shown in this example:

```
PowerConnect#show isis name hostname
Total number of entries in IS-IS Hostname Table: 1
   System ID        Hostname            * = local IS
 * bbbb.cccc.dddd   IMR
```

## IS-IS debug commands

### debug isis

**Syntax: [no] debug isis [adj |bfd| error |interface | l1-csnp | l1-hello | l1-lsp | l1-psnp | l2-csnp | l2-hello | l2-lsp | l2-psnp | lsp-flood | lsp-dump | memory | nsr | pp-hello | ppp | pspf | pspf-detail | redistribution | route-table | spf | spf-log | te | trace]**

- **adj** - Displays information about IS-IS adjacencies.
- **bdf** - Displays IS-IS BFD information.
- **error** - Displays IS-IS errors.
- **interface** - Limits the display of IS-IS information to specific interface

- **l1-csnp** - Displays level 1 CSNP PDU information.
- **l1-hello** - Displays level 1 hello PDU information.
- **l1-lsp** - Displays level 1 LSP PDU information.
- **l1-psnp** - Displays level 1 PSNP PDU information.
- **l2-csnp** - Displays level 2 CSNP PDU information.
- **l2-hello** - Displays level 2 hello PDU information.
- **l2-lsp** - Displays level 2 LSP PDU information.
- **l2-psnp** - Displays level 2 PSNP PDU information.
- **lsp-flood** - Displays information about LSP flooding.
- **lsp-dump** - Displays a dump of context-specific LSP contents.
- **memory** - Displays memory information.
- **nsr** - Displays IS-IS Nonstop Routing information.
- **pp-hello** - Displays PP hello PDU information.
- **ppp** - Displays OSI PPP information.
- **pspf** - Displays IS-IS partial SPF information.
- **pspf-detail** - Displays IS-IS partial SPF information details.
- **redistribution** - Displays IS-IS route redistribution.
- **route-table** - Displays IS-IS route table information.
- **spf** - Displays IS-IS SPF information.
- **spf-log** - Displays information about the SPF-log.
- **te** - Displays information about ISIS traffic engineering.
- **trace** - Displays trace information for IS-IS code path.

### debug isis adj

**Syntax: [no] debug isis adj**

This command generates information about IS-IS adjacencies. Output resembles the following:

```
PowerConnect#debug isis adj
ISIS: Clearing all adjacencies on 1/1
ISIS: Deleting PTPT Adj to rtr1 on 1/1 from HT Timer for HT 30 Index 1]
ISIS: L1 DIS change on 1/4 to rtr2-3
ISIS: L2 DIS change on 1/4 to rtr2-3
ISIS: Deleting PTPT Adj to rtr1 on 1/1 [HoldTimer expiry]
ISIS: Deleting PTPT Adj to rtr1 on 1/1 from HT Timer for HT 30 [Index 0]
ISIS: Deleting PTPT Adj to rtr1 on 1/2 [HoldTimer expiry]
ISIS: Deleting PTPT Adj to rtr1 on 1/2 from HT Timer for HT 30 [Index 1]
ISIS: Adding PTPT Adj 0000.0000.0001 on 1/1 to HT Timer for HT 30 [Index 0]
ISIS: Adding PTPT Adj 0000.0000.0000 on 1/2 to HT Timer for HT 7680 [Index 1]
ISIS: L1 DIS change on 1/4 to rtr2-3
ISIS: L2 DIS change on 1/4 to rtr2-3
ISIS: L1 DIS change on 1/4 to rtr2-3
ISIS: Adding PTPT Adj rtr1 on 1/1 to HT Timer for HT 30 [Index 0]
ISIS: Adding PTPT Adj rtr1 on 1/2 to HT Timer for HT 30 [Index 1]
```

### debug isis l1-csnp

**Syntax: [no] debug isis l1-csnp**

This command displays information about level 1 complete sequence number PDUs (CSNPs) sent and received on the device. Output resembles the following:

```
PowerConnect#debug isis l1-csnp
ISIS: Sending L1 CSNP on 2/24, length 1497
ISIS: Received L1 CSNP on 2/24, length 256 from 0004.8026.b337
```

### debug isis l1-hello

**Syntax: [no] debug isis l1-hello**

This command generates information about level 1 hello PDUs sent and received. Output resembles the following:

```
PowerConnect#debug isis l1-hello
ISIS: Received L1 LAN IIH on 2/24, length 256 from 0004.8026.b337
ISIS: Sending L1 LAN IIH on 2/24, length 1497
ISIS: Received L1 LAN IIH on 2/24, length 256 from 0004.8026.b337
```

### debug isis l1-lsp

**Syntax: [no] debug isis l1-lsp**

This command generates information about level 1 link state PDUs (LSPs) sent and received. Output resembles the following, and the description is self-explanatory:

```
PowerConnect#debug isis l1-lsp
ISIS: Sending L1 LSP on 2/24, length 27
ISIS: Received L1 LSP on 2/24, length 256 from 0004.8026.b337
```

### debug isis l1-psnp

**Syntax: [no] debug isis l1-psnp**

This command generates information about level 1 partial sequence number PDUs (PSNPs) sent and received. Output resembles the following, and the description is self-explanatory:

```
PowerConnect#debug isis l1-psnp
ISIS: Received L1 PSNP on 2/24, length 256
ISIS: Received L1 PSNP on 2/24, length 35
```

### debug isis l2-csnp

**Syntax: [no] debug isis l2-csnp**

This command generates information about level 2 CSN PDUs sent and received. Output resembles the following, and the description is self-explanatory (in the following example, source MAC addresses are shown):

```
PowerConnect#debug isis l2-csnp
ISIS: Rcvd L2 CSNP on 2/1, length 906 from fr1.iad.QA.Tes [MAC 000c.dbe3.0c02]
ISIS: Rcvd L2 CSNP on 1/1, length 906
ISIS: Rcvd L2 CSNP on 1/2, length 906 from fr1.sjc.QA.Tes [MAC 000c.dbe3.b629]
ISIS: Rcvd L2 CSNP on v510, length 906
ISIS: Rcvd L2 CSNP on v510, length 906 from fr1.sjc.QA.Tes [MAC 000c.dbe3.b600]
ISIS: Rcvd L2 CSNP on 2/1, length 906
ISIS: Rcvd L2 CSNP on 2/1, length 906 from fr1.iad.QA.Tes [MAC 000c.dbe3.0c02]
ISIS: Rcvd L2 CSNP on 1/1, length 906
```

### debug isis l2-hello

**Syntax: [no] debug isis l2-hello**

This command generates information about level 2 hello PDUs sent and received. Output resembles the following:

```
PowerConnect#debug isis ls-hello
ISIS: Received L2 LAN IIH on 2/24, length 256 from 0004.8026.b337
ISIS: Sending L2 LAN IIH on 2/24, length 1497
ISIS: Received L2 LAN IIH on 2/24, length 256 from 0004.8026.b337
```

### debug isis l2-lsp

**Syntax: [no] debug isis l2-lsp**

This command generates information about level 2 LS PDUs sent and received. Output resembles the following:

```
PowerConnect#debug isis l2-lsp
ISIS: Sending L2 LSP on 2/24, length 27
ISIS: Received L2 LSP on 2/24, length 256 from 0004.8026.b337
```

### debug isis l2-psnp

**Syntax: [no] debug isis l2-psnp**

This command generates information about level 2 PSN PDUs (PSNPs) sent and received. Output resembles the following:

```
PowerConnect#debug isis l2-psnp
ISIS: Received L2 PSNP on 2/24, length 256
ISIS: Received L2 PSNP on 2/24, length 35
```

### debug isis memory

**Syntax: [no] debug isis memory**

This command generates information about IS-IS memory allocations and releases. Output resembles the following:

```
PowerConnect#debug isis memory
ISIS: Memory Allocated for buffer description at 21a54ad8
ISIS: Memory Allocated for packet-buffer at 211e1680
ISIS: Memory Released for buffer descriptor at 21a54ad
ISIS: Memory Allocation for circuit IP address failed
```

### debug isis nsr

**Syntax: [no] debug isis nsr**

This command displays information related to LSP, Neighbor syncing, and NSR state-related information. Output resembles the following:

```
PowerConnect#debug isis nsr
ISIS: Sending L1-LSP MLXe36.01-00 Seq-No 6173 Flags Lsp Update Length 53 to
standby
ISIS: Ack recv for L1 Lsp MLXe36.01-00 Addition from Standby
ISIS: Sending L1 Nbr CES Flags Neighbor Delete to standby
ISIS: Ack recv for L2 Neighbor CES Deletion from Standby
ISIS: Sending L2 Nbr CES Flags Neighbor Update to standby
ISIS: Ack recv for L2 Neighbor CES Addition from Standby
```

### debug isis pp-hello

**Syntax: [no] debug isis pp-hello**

This command displays information about point-to-point Hello PDUs sent and received. Output resembles the following:

```
PowerConnect#debug isis pp-hello
ISIS: Sending PTP IIH on 9/1, length 1492
ISIS: Received PTP IIH on 9/1, length 256
```

### debug isis ppp

**Syntax: [no] debug isis ppp**

This command generates information about OSI PPP packets sent and received. Output resembles the following:

```
PowerConnect#debug isis ppp
ISIS PPP: sending isis packet on pos port 512
ISIS: osicp datainput rx pkt length 1492 on unit 32
ISIS: Received PTP IIH on 9/1, length 256
ISIS: Sending PTP IIH on 9/1, length 1492
ISIS PPP: sending isis packet on pos port 512
```

### debug isis pspf

**Syntax: [no] debug isis pspf**

This command generates information about IS-IS PSPF activity. Output resembles the following:

```
PowerConnect#debug isis pspf
ISIS: Comparing old options against new to detect routes that may have been
removed
ISIS: Checking ISOC_EIPREACH,
ISIS: Checking ISOC_EIPREACH,
ISIS: Comparing new options against old to detect routes that may have been added
ISIS: isis_check_if_partial_spf_needed called
ISIS: isis_identify_and_process_changed_ip_information_in_lsp called
ISIS: Checking ISOC_EREACH
ISIS: Checking ISOC_IREACH
```

### debug isis pspf-detail

**Syntax: [no] debug isis pspf-detail**

This command generates detailed information about IS-IS PSPF activity. Output resembles the following:

```
PowerConnect#debug isis pspf-detail
ISIS: Total Route Calculation Time is 0 milliseconds.
ISIS: PSPF Started for level 2
PENT_IP found id = 50.0.6.0 255.255.255.0 cost 41 pref 2 up=0
PENT_IP found id = 45.0.170.0 255.255.255.0 cost 20 pref 2 up=0
PENT_IP found id = 50.0.12.0 255.255.255.0 cost 50 pref 2 up=0
PENT_IP found id = 50.0.18.0 255.255.255.0 cost 50 pref 2 up=0
PENT_IP found id = 50.0.4.0 255.255.255.0 cost 41 pref 2 up=0
PENT_IP found id = 50.0.10.0 255.255.255.0 cost 50 pref 2 up=0
PENT_IP found id = 50.0.16.0 255.255.255.0 cost 50 pref 2 up=0
PENT_IP found id = 203.131.243.16 255.255.255.252 cost 40 pref 2 up=0
```

### debug isis redistribution

**Syntax: [no] debug isis redistribution**

This command displays route redistribution into or out of IS-IS routes. Output resembles the following:

```
PowerConnect#debug isis redistribution
ISIS: Imported CONNECTED route 30.10.0.3 255.255.0.0
ISIS: Imported CONNECTED route 40.10.0.3 255.255.0.0
ISIS: Added external route 30.10.0.3 255.255.0.0 to L12 LSP
ISIS: Added external route 40.10.0.0 255.255.0.0 to L12 LSP
ISIS: Unimported CONNECTED route 30.10.0.0 255.255 0.0
ISIS: Unimported CONNECTED route 40.10.0.0. 255.255.0.0
ISIS: Deleted external route 30.10.0.0. 255.255.0.0 from L12 LSP
ISIS: Deleted external route 40.10.0.0 255.255.0.0. from L12 LSP
```

This output indicates several redistribution activities, including importing and unimporting connected routes, and adding and deleting external routes.

### debug isis route-table

**Syntax: [no] debug isis route-table**

This command reports changes to the IS-IS route table. Output resembles the following:

```
PowerConnect#debug isis route-table
ISIS: Deleting route 12.10.0.0. 255.255.0.0 level 2
ISIS: Deleting route 11.10.0.0 255.255.0.0 level 2
ISIS: Creating new route for 100.10.0.0 255.255.0.0. level 2 type 1
ISIS: Adding path Next hop = 192.147.201.200 Interface 2/4
ISIS: Creating new route for 12.10.0.0 255.255.0.0 level 2 type 1
```

### debug isis spf

**Syntax: [no] debug isis spf**

This command generates information about SPF calculations made for IS-IS. Output resembles the following:

```
PowerConnect#debug isis spf
ISIS: Running Decision for level 1
ISIS: IsisgetMinTent
ISIS: Finished Decision for level 1
ISIS: Total Route Calculation Time for Level 1 is 0 milliseconds.
```

### debug isis trace

**Syntax: [no] debug isis trace**

This command generates information about internal IS-IS functions. Output resembles the following:

```
PowerConnect#debug isis trace
ISIS:proc_SNPE
ISIS: build_csnp
ISIS: build_csnp
ISIS: sig_description
```

## Configuration notes

None of the IS-IS parameters require a software reload for changes to take effect, and most parameter changes take effect immediately. However, changes for the following parameters take effect only after you disable and then re-enable redistribution:

- Changing the default metric
- Adding, changing, or negating route redistribution parameters

The following sections describe some of the common scenarios you may encounter when using IS-IS configurations.

## Common diagnostic scenarios

- Packets are dropped during authentication mode change

  Changing the authentication mode can cause packets to drop during the transition period, because not all of the routers are reconfigured simultaneously. During such a transition, it can be useful to disable IS-IS authentication checking temporarily until all routers are reconfigured and the network is stable.

  Use the following commands to disable IS-IS authentication checking on a specified interface:

  ```
  PowerConnect(config)#interface ethernet 3/1
  PowerConnect(if-e10000-3/1)#no isis auth-check level-1
  ```
  **Syntax:** [**no**] **isis auth-check** [**level-1** | **level-2**]

  This command enables and disables IS-IS authentication checking. The default is enabled and the **[no]** parameter disables authentication checking.

  The **level-1** parameter specifies that authentication checking is enabled/ disabled for Level 1 Hello packets.

  The **level-2** parameter specifies that authentication checking is enabled/disabled for Level 2 Hello packets.

  **NOTE**
  If either level-1 or level-2 are not specified, the configuration is applied to both level-1 and level-2.

- PowerConnect LSPs are timed-out by neighbor

  The **max-lsp-lifetime** and the **lsp-refresh-interval** must be set so that the LSPs are refreshed before the **max-lsp-lifetime** expires; otherwise, the PowerConnect B-MLXe device's originated LSPs may be timed out by its neighbors.

  To prevent a neighbor from timing-out LSPs, give the lsp-refresh-interval a shorter setting than that of the max-lsp-lifetime interval. The LSP refresh interval is the maximum number of seconds the PowerConnect B-MLXe device waits between sending updated LSPs to its IS-IS neighbors. The interval can be from 1 – 65535 seconds. The default is 900 seconds.

  For example, to change the LSP refresh interval to 20000 seconds, enter a command similar to the following:

  ```
  PowerConnect(config-isis-router)#lsp-refresh-interval 20000
  ```
  **Syntax:** [**no**] **lsp-refresh-interval** <*secs*>

  The <*secs*> parameter specifies the maximum refresh interval and can be from 1 through 65535 seconds. The default is 900 seconds (15 minutes).

- An IS-IS link is experiencing low performance rates

  If an IS-IS link is performing poorly, it may be due to padding that is added the end of a hello packet to make the packet the same size as the maximum length of PDU the PowerConnect B-MLXe device support. PowerConnect B-MLXe device adds this padding by default to the following types of hello packets:

- ES hello (ESH PDU)
- IS hello (ISH PDU)
- IS to IS hello (IIH PDU)

  When padding is enabled, the maximum length of a Hello PDU sent by the PowerConnect B-MLXe device is 1514 bytes.

  If you suspect that padding is affecting the link's performance, you can disable padding globally or on individual interfaces. If you enable or disable padding on an interface, the interface setting overrides the global setting.

  To globally disable padding of IS-IS hello PDUs, enter the following command.

  ```
  PowerConnect(config-isis-router)#no hello padding
  ```

  This command disables all hello PDU padding on the PowerConnect B-MLXe device. To re-enable padding, enter the following command.

  ```
  PowerConnect(config-isis-router)# hello padding
  ```

  **Syntax:** [**no**] **hello padding**

  By default, hello padding is enabled. Enter the **no** form of the command to disable hello padding.

- ISIS adjacent ports are flapping.

  Check for any high CPU condition, or evidence of packet loss. Do a ping test to see if packet loss occurs.

- Using summary addresses to enhance performance

  Summary addresses can enhance performance by reducing the size of the Link State database, reducing the amount of data the PowerConnect B-MLXe device needs to send to its neighbors, and reducing the CPU cycles used for IS-IS.

  When you configure a summary address, the address applies only to Level-2 routes by default. You can specify Level-1 only, Level-2 only, or Level-1 and Level-2 when you configure the address.

  To configure a summary address, enter a command similar to the following:

  ```
  PowerConnect(config-isis-router-ipv4u)#summary-address 192.168.0.0
  255.255.0.0
  ```

  This command configures a summary address for all Level-2 IS-IS route destinations between 192.168.1.0 – 192.168.255.255.

  **Syntax:** [**no**] **summary-address** <*ip-addr*> <*subnet-mask*> [**level-1** | **level-1-2** | **level-2**]

  The <*ip-addr*> <*subnet-mask*> parameters specify the aggregate address. The mask indicates the significant bits in the address. Ones are significant, and zeros allow any value. In the command example above, the mask 255.255.0.0 matches on all addresses that begin with 192.168 and contain any values for the final two octets.

The **level-1 | level-1-2 | level-2** parameter specifies the route types to which the aggregate route applies. The default is **level-2**.

- Old software versions.

  Feature issues are often be caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# VRRP and VRRPE

This section describes how to troubleshoot the IP VRRP and VRRP-Extended environments for PowerConnect B-MLXe routers.

VRRP is an election protocol provides alternate router paths for a host without changing the IP address or MAC address by which the host knows its gateway.

VRRPE is proprietary version of VRRP that overcomes limitations in the standard protocol. It is similar to VRRP, but differs in the following respects:

## VRRP show commands

### show ip vrrp

**Syntax: show ip vrrp [brief | ethernet <*slotnum*>/<*portnum*> | ve <*num*> | vrid <*num*> | statistics]**

- **brief** - Displays IPv4 VRRP summary information. If you do not use this parameter, detailed information is displayed instead.
- **ethernet** <*slotnum*>/<*portnum*> - Specifies an Ethernet port. If you use this parameter, the command displays IPv4 VRRP information only for the specified port.
- **ve** <*num*> - Specifies a virtual interface. If you use this parameter, the command displays IPv4 VRRP information only for the specified virtual interface.
- **vrid** <*num*>- Displays IPv4 VRRP information only for the specified virtual router ID.
- **statistics** - Displays statistics.

### show ip vrrp brief

**Syntax: show ip vrrp brief**

This command displays IPv4 VRRP summary information, as shown in this example:

```
PowerConnect#show ip vrrp brief
Total number of VRRP-Extended routers defined: 4
Inte- VRID Current  P  State  Master IP       Backup IP       Virtual IP
rface      Priority           Address         Address         Address
-------------------------------------------------------------------------------
v10   1    100         Init   Unknown         Unknown         192.168.1.1
v20   1    100         Init   Unknown         Unknown         10.10.20.1
v30   1    100         Init   Unknown         Unknown         10.10.30.1
v100  1    100         Init   Unknown         Unknown         10.10.100.1
```

### show ip vrrp-extended

**Syntax: show ip vrrp-extended [brief | ethernet** <*slotnum*>/<*portnum*> **| ve** <*num*> **| vrid** <*num*>**| statistics]**

- **brief** - Displays IPv4 VRRPE summary information.
- **ethernet** <*slotnum*>/<*portnum*> - Specifies an Ethernet port. If you use this parameter, the command displays IPv4 VRRPE information only for the specified port.
- **ve** <*num*> - Specifies a virtual interface. If you use this parameter, the command displays IPv4 VRRPE information only for the specified virtual interface.
- **vrid** <*num*> - Displays IPv4 VRRPE information only for the specified virtual router ID.
- **statistics** - Displays statistics.

The following example shows output from the **show ip vrrp-extended** command:

```
PowerConnect#show ip vrrp-extended
Total number of VRRP-Extended routers defined: 4
Interface v10
----------------
auth-type no authentication

VRID 1 (index 1)
 interface v10
 state initialize
 administrative-status enabled
 mode non-owner(backup)
 virtual mac 02e0.52e5.cd01
 priority 100
 current priority 100
 track-priority 5
 hello-interval 1 sec
 backup hello-interval 60 sec
 advertise backup disabled
 dead-interval 0 sec
 current dead-interval 0.0 sec
 preempt-mode false
 virtual ip address 192.168.1.1
```

### show ipv6 vrrp

**Syntax: show ipv6 vrrp [brief | ethernet <***slotnum***>/<***portnum***>| statistics | ve** <*num*> **| vrid** <*num*> **]**

- **brief** - Displays virtual router summary information.
- **ethernet** <*slotnum*>/<*portnum*> - Limits the display of IPv6 VRRP information only to the specific Ethernet interface.
- **statistics** - Displays virtual router statistics.
- **ve** <*num*> - Displays IPv6 VRRP information only for the specified virtual interface.
- **vrid** <*num*> - Displays IPv6 VRRP information only for the specified virtual router ID.

### show ipv6 vrrp brief

**Syntax: show ipv6 vrrp brief**

This command displays IPv6 VRRP summary information.

```
PowerConnect#show ip vrrp brief
Total number of VRRP routers defined: 1
Flags Codes - P:Preempt 2:V2 3:V3
Intf   VRID   CurrPrio Flags    State   Master-IPv6- Backup-IPv6- Virtual-IPv6-
                                        Address      Address        Address
------------------------------------------------------------------------------
1/3    13     100        P3     Master     Local      3013::2       fe80::3013:2
```

### show ipv6 vrrp vrid

**Syntax: show ipv6 vrrp vrid** <*num*>

This command displays IPv6 VRRP information only for the specified virtual router ID.

```
PowerConnect#show ipv6 vrrp vrid 13
Interface 1/3
----------------------------
auth-type no authentication VRID 13 (index 1)
interface 1/3
state master
administrative-status enabled
version v3
mode non-owner(backup)
virtual mac 0000.5e00.020d
priority 100
current priority 100
track-priority 1
hello-interval 200 ms
backup hello-interval 60000 ms
advertise backup disabled
dead-interval 700 ms
preempt-mode true
ipv6-address fe80::3011:9
ipv6-address 3011::9
next hello sent in 200 ms
```

### show ipv6 vrrp statistics

**Syntax: show ipv6 vrrp statistics**

This command displays router statistics information.

```
PowerConnect#show ipv6 vrrp statistics
Global IPv6 VRRP statistics
------------------------------
- received vrrp packets with checksum errors = 0
- received vrrp packets with invalid version number = 0
- received vrrp packets with unknown or inactive vrid = 0
Interface 1/3
----------------
VRID 13
- number of transitions to backup state = 1
- number of transitions to master state = 1
- total number of vrrp packets received = 0
. received backup advertisements = 19
. received packets with zero priority = 0
received packets with invalid type = 0
. received packets with invalid authentication type = 0
. received packets with authentication type mismatch = 0
. received packets with authentication failures = 0
. received packets dropped by owner = 0
. received packets with ttl errors = 0
```

```
. received packets with ipv6 address mismatch = 0
. received packets with advertisement interval mismatch = 0
. received packets with invalid length = 0
- total number of vrrp packets sent = 1175
. sent backup advertisements = 0
. sent packets with zero priority = 0
- received neighbor solicitation packets dropped = 0
- received proxy neighbor solicitation packets dropped = 0
- received ipv6 packets dropped = 0
```

### show ipv6 vrrp-extended

**Syntax: show ipv6 vrrp-extended [brief | ethernet <*slotnum*>/<*portnum*> | statistics | ve <*num*> | vrid <*num*> ]**

- **brief** - Displays virtual router summary information.
- **ethernet** <*slotnum*>/<*portnum*> - Limits the display of IPv6 VRRP-E information only to the specific Ethernet interface.
- **statistics** - Displays virtual router statistics.
- **ve** <*num*> - Displays IPv6 VRRP-E information only for the specified virtual interface.
- **vrid** <*num*> - Displays IPv6 VRRP-E information only for the specified virtual router ID.

### show ipv6 vrrp-extended brief

**Syntax: show ipv6 vrrp-extended brief**

This command displays IPv6 VRRP-E summary information.

```
PowerConnect#show ipv6 vrrp-extended brief
Total number of VRRP routers defined: 1
Flags Codes - P:Preempt 2:V2 3:V3
Intf   VRID   CurrPrio   Flags  State  Master-IPv6   Backup-IPv6   Virtual-IPv6
                                      -Address      -Address      -Address
-------------------------------------------------------------
1/3    13     100        P3     Master  Local        3013::2       3013::99
```

### show ipv6 vrrp-extended vrid

**Syntax: show ipv6 vrrp-extended vrid** <*num*>

This command displays IPv6 VRRP-E information only for the specified virtual router ID.

```
PowerConnect#show ipv6 vrrp-extended vrid 13
Interface 1/3
----------------------------
auth-type no authentication VRID 13 (index 1)
interface 1/3
state master
administrative-status enabled
mode non-owner(backup)
virtual mac 02e0.521d.000d
priority 100
current priority 100
track-priority 5
hello-interval 1 sec
backup hello-interval 60 sec
advertise backup enabled
dead-interval 3.1 sec
preempt-mode true
virtual ipv6 address 3013::99
next hello sent in 0.1 sec
```

```
backup router 3013::2 expires in 175.0 sec
```

### show ipv6 vrrp-extended statistics

### Syntax: show ipv6 vrrp-extended statistics

This command displays virtual router statistical information.

```
PowerConnect#show ipv6 vrrp-extended statistics
Global IPv6 VRRP-Extended statistics
------------------------------
- received vrrp-extended packets with checksum errors = 0
- received vrrp-extended packets with invalid version number = 0
- received vrrp-extended packets with unknown or inactive vrid = 0
Interface 1/3
----------------
VRID 13
- number of transitions to backup state = 1
- number of transitions to master state = 1
- total number of vrrp-extended packets received = 0
. received backup advertisements = 19
. received packets with zero priority = 0
. received packets with invalid type = 0
. received packets with invalid authentication type = 0
. received packets with authentication type mismatch = 0
. received packets with authentication failures = 0
. received packets dropped by owner = 0
. received packets with ttl errors = 0
. received packets with ipv6 address mismatch = 0
. received packets with advertisement interval mismatch = 0
. received packets with invalid length = 0
- total number of vrrp-extended packets sent = 1175
. sent backup advertisements = 0
. sent packets with zero priority = 0
- received neighbor solicitation packets dropped = 0
- received proxy neighbor solicitation packets dropped = 0
- received ipv6 packets dropped = 0
```

## Clearing VRRP statistics

To clear IPv4 VRRP statistics, enter the following command at any level of the CLI:

**clear ip vrrp statistics**
**Syntax: clear ip vrrp statistics**

To clear IPv6 VRRP statistics, enter the following command at any level of the CLI:

**clear ipv6 vrrp statistics**
**Syntax: clear ipv6 vrrp statistics**

## Clearing VRRP-E statistics

To clear IPv4 VRRP-E statistics, enter the following command at any level of the CLI:

**clear ip vrrp-extended statistics**
**Syntax: clear ip vrrp-extended statistics**

To clear IPv6 VRRP-E statistics, enter the following command at any level of the CLI:

**clear ipv6 vrrp-extended statistics**
**Syntax: clear ipv6 vrrp-extended statistics**

## VRRP debug commands

**debug ip vrrp**
**Syntax: [no] debug ip vrrp [all | error | ethernet <***slotnum***>/<***portnum***> | events | packets | show | state | ve** <*num*> **| verbose | vrid** <*num*> **]**

- **all** - Displays information about all IPv4 VRRP instances (default).
- **error** - Displays error conditions where a packet is not being processed.
- **ethernet** <*slotnum*>/<*portnum*> - Displays information about IPv4 VRRP instances on a specific physical interface.
- **events** - Displays information about activate and shutdown, port up/port down, timer events, backup VRRP router events, etc.
- **packets** - Displays information about IPv4 VRRP transmitted and received packets, including ARP packets.
- **show** - Shows the current IPv4 VRRP debug settings.
- **state** - Displays information about IPv4 VRRP state changes, such as monitor transitions from master to backup, or vice versa.
- **ve** <*num*> - Displays information about a specific virtual interface.
- **verbose** - Decodes hex output into more easily understood fields and values.
- **vrid** <*num*> - Displays information about a specific virtual router ID.

This command displays information about IPv4 VRRP instances. The default is all instances. Several parameters are available to help isolate a specific IPv4 VRRP instance.

### debug ip vrrp all

**Syntax: [no] debug ip vrrp all**

Enables or disables all IPv4 VRRP debugging settings.

### debug ip vrrp error

**Syntax: [no] debug ip vrrp error**

Displays information about IPv4 VRRP error conditions where a packet is not being processed. This command will isolate header errors, checksum errors, and packet errors that cause a packet to be dropped.

### debug ip vrrp ethernet

**Syntax: [no] debug ip vrrp ethernet <*slotnum*>/<*portnum*>**

Displays information about IPv4 VRRP activity on a specific physical interface.

### debug ip vrrp events

**Syntax: [no] debug ip vrrp events**

Displays information about IPv4 VRRP events, such as activate and shutdown, port up/port down, timer events, backup VRRP router events, etc.

### debug ip vrrp packets

**Syntax: [no] debug ip vrrp packets**

Displays information about all received and transmitted IPv4 VRRP packets, including ARP packets.

### debug ip vrrp show

**Syntax: [no] debug ip vrrp show**

Shows the status of the IPv4 VRRP debugging function. For example:

```
PowerConnect#debug ip vrrp show
VRRP debugging is ON
- All VRIDs on all ports in normal mode.
- Debugging is setup for errors events packets state.
```

### debug ip vrrp state

**Syntax: [no] debug ip vrrp state**

Displays information about IPv4 VRRP state changes, such as transitions from master to backup, or vice versa.

### debug ip vrrp ve

**Syntax: [no] debug ip vrrp ve** <*num*>

Displays information about a specific IPv4 VRRP virtual interface.

### debug ip vrrp verbose

**Syntax: [no] debug ip vrrp verbose**

Sets the debug mode to verbose, which decodes hex output into fields and data that is easier to decipher.

### debug ip vrrp vrid

**Syntax: [no] debug ip vrrp vrid** <*num*>

Displays information about a specific virtual router ID.

### debug ipv6 vrrp

**Syntax: [no] debug ipv6 vrrp [all | error | ethernet <***slotnum***>/<***portnum***> | events | packets | show | state | ve** <*num*>**| verbose | vrid** <*num*>**]**

- **all** - Displays information about all IPv6 VRRP instances.
- **error** - Displays error conditions where a packet is not being processed.
- **ethernet** <*slotnum*>/<*portnum*> - Displays information about IPv6 VRRP instances on a specific physical interface.
- **events** - Displays information about activate and shutdown, port up/port down, timer events, backup VRRP router events, etc.
- **packets** - Displays information about IPv6 VRRP transmitted and received packets, including ARP packets.
- **show** - Shows the current IPv6 VRRP debug settings.
- **state** - Displays information about IPv6 VRRP state changes, such as monitor transitions from master to backup, or vice versa.
- **ve** <*num*> - Displays information about a specific virtual interface.
- **verbose** - Decodes hex output into more easily understood fields and values.
- **vrid** <*num*> - Displays information about a specific virtual router ID.

This command displays information about IPv6 VRRP instances. The default is all instances. Several parameters are available to help isolate a specific IPv6 VRRP instance.

The output resembles the following:

```
PowerConnect#debug ipv6 vrrp
IPV6 VRRP: debugging is on
VRRP6: Port 1/3, VRID 23 - send advertisement
Ver:3 Type:1 Vrid:23 Pri:100 #IP:2 AuthType:0 Adv:100 Chksum:0xd37c
IpAddr: fe80::3013:2 3013::2
VRRP6: Port 1/3, VRID 23 - send advertisement
Ver:3 Type:1 Vrid:23 Pri:100 #IP:2 AuthType:0 Adv:100 Chksum:0xd37c
IpAddr: fe80::3013:2 3013::2
```

### debug ipv6 vrrp all

**Syntax: [no] debug ipv6 vrrp all**

Enables or disables all IPv6 VRRP debugging settings.

### debug ipv6 vrrp error

**Syntax: [no] debug ipv6 vrrp error**

Displays information about IPv6 VRRP error conditions where a packet is not being processed. This command will isolate header errors, checksum errors, and packet errors that cause a packet to be dropped.

### debug ipv6 vrrp ethernet

**Syntax: [no] debug ipv6 vrrp ethernet <***slotnum***>/<***portnum***>**

Displays information about IPv6 VRRP activity on a specific physical interface.

**debug ipv6 vrrp events**

**Syntax: [no] debug ipv6 vrrp events**

Displays information about IPv6 VRRP events, such as activate and shutdown, port up/port down, timer events, backup VRRP router events, etc.

**debug ipv6 vrrp packets**

**Syntax: [no] debug ipv6 vrrp packets**

Displays information about all received and transmitted IPv6 VRRP packets, including ARP packets.

**debug ipv6 vrrp show**

**Syntax: [no] debug ipv6 vrrp show**

Shows the status of the IPv6 VRRP debugging function.

**debug ipv6 vrrp state**

**Syntax: [no] debug ipv6 vrrp state**

Displays information about IPv6 VRRP state changes, such as transitions from master to backup, or vice versa.

**debug ipv6 vrrp ve**

**Syntax: [no] debug ipv6 vrrp ve** <*num*>

Displays information about a specific IPv6 VRRP virtual interface.

**debug ipv6 vrrp verbose**

**Syntax: [no] debug ipv6 vrrp verbose**

Sets the debug mode to verbose, which decodes hex output into fields and data that is easier to decipher.

**debug ipv6 vrrp vrid**

**Syntax: [no] debug ipv6 vrrp vrid** <*num*>

Displays information about a specific virtual router ID.

## Configuration notes and diagnostic scenarios

The following rules apply to VRRPE configurations:

- The interfaces of all routers in a virtual router must be in the same IP subnet.
- The IP address assigned to the virtual router cannot be configured on any of the PowerConnect B-MLXe device devices.
- The address you enter with the **ip-address** command cannot be the same as a real IP address configured on the interface.
- The Hello interval must be set to the same value on all the PowerConnect B-MLXe devices.
- The Dead interval must be set to the same value on all the PowerConnect B-MLXe devices.
- The track priority for a virtual router must be lower than the VRRP-E priority.
- When you configure the Master (Owner), the address you enter with the **ip-address** command must already be configured on the interface.

- When you configure a Backup router, the router interface on which you are configuring the virtual router must have a real IP address that is in the same subnet (but is not the same address) as the address associated with the virtual router by the Owner.
- If you disable VRRP-E, the router removes all information for the disabled protocol from the running configuration. When you save to the startup configuration after disabling the protocol, all information for the disabled protocol is removed from the startup configuration.

**CAUTION**

**You must configure a VRF on an interface before configuring a Virtual Router (VRRPE) on it. If you enable the Virtual Router before you enable the VRF, the Virtual Router configuration will be deleted.**

Chapter

# MPLS Diagnostics 6

This chapter contains diagnostic information for the MPLS configurations.

## MPLS

MPLS debugging commands help the users to diagnose and determine the cause of faults for MPLS related features.

The following sections describe show and debug commands that can be used to obtain information about MPLS activity.

### MPLS debug commands

#### debug mpls?

**Syntax: debug mpls?**

This command displays all the module types of MPLS.

```
PowerConnect# debug mpls ?
all           turns on/off all MPLS debugs
cspf          MPLS cspf debug
error         MPLS error msgs
forwarding    MPLS forwarding debug
ldp           MPLS ldp debug
lmgr          MPLS label manager debug
oam           MPLS OAM debug
routing       MPLS routing (rsir) debug
rsvp          MPLS rsvp debug
vll-local     MPLS VLL-Local debug
```

#### debug mpls

**Syntax: [no] debug mpls**

This command is used to turn on the debugging output, while preserving the debugging configuration. The output resembles the following:

```
PowerConnect# debug mpls
PowerConnect# show debug
MPLS Debug Settings
    MPLS Debug is ON
    Forwarding
        All
```

#### debug mpls all

**Syntax: [no] debug mpls all**

This command is used to turn on all the MPLS debugging configuration.

You can turn on MPLS debugging for all the module and submodule types. For example: the command **debug mpls cspf computation all** will enable debugging for the submodule Computation which belongs to the module CSPF.

```
PowerConnect# debug mpls cspf computation all
PowerConnect# show debug
MPLS Debug Settings
    MPLS Debug is OFF
    Cspf
        Computation
               All
```

You can also enable debugging for all the filter types as shown in the following example.

```
PowerConnect#debug mpls cspf computation lsp name all
PowerConnect#show debug
MPLS Debug Settings
    MPLS Debug is OFF
    Cspf
        Computation
           Lsp:
               Name:
                   All
```

### debug mpls error

**Syntax: [no] debug mpls error**

This command is used to turn on error debugging for all the modules.

```
PowerConnect#debug mpls error
PowerConnect#show debug
MPLS Debug Settings
    MPLS Debug is OFF
    Error
        All
```

Error debugging can be enabled for each module type.
For example, error debugging for CPSF module is enabled as shown in the following example.

```
PowerConnect# debug mpls cspf error
PowerConnect# show debug
MPLS Debug Settings
    MPLS Debug is OFF
    Cspf
        Error
```

### show mpls debug counter

**Syntax: show mpls debug counter [aall | ai3 | ambl | amb | amh | ase | asel | bfd | i3 | ipl | nbase-root | ntl | rcp | rcs | rlm | rldf | rric | rri | rrt | rsip | ipr | rspx | rstc | sck | yads]**

- aall - Displays counters for ACL library.
- ai3 - Displays counters for I3 library.
- amb - Displays counters for MIB manager.
- ambl - Displays counters for MIB library.
- amh - Displays counters for MIB handler.
- ase - Displays counters for System manager.
- asel - Displays counters for System manager lite.

- bfd - Displays counters for BFD stub.
- i3 - Displays counters for I3 stub.
- ipl - Displays counters for IP library.
- ipr - Displays counters for Route (RSIR) stub.
- nbase-root - Displays counters for NBASE-ROOT.
- ntll - Displays counters for NTL library.
- rcp - Displays counters for LDP Path Manager.
- rcs - Displays counters for LDP Session Controller.
- rldf - Displays counters for RLDF Stub.
- rlm - Displays counters for Label manager.
- rri - Displays counters for RSVP.
- rric - Displays counters for RRIC stub.
- rrt - Displays counters for TE-MIB.
- rsip - Displays counters for IP stub.
- rspx - Displays counters for Proxy stub.
- rstc - Displays counters for CPCS LDT.

This command displays the debug counters. The debug counter keeps track of each debug statement access regardless of the match condition.

```
PowerConnect# show mpls debug counter
Counter-mnemonic            Severity    HitCount
PCT_IPR | 2, 0              Audit       1
PCT_IPR | 16, 0             Dev         12
PCT_IPR | 16, 1             Dev         4
PCT_IPR | 16, 20            Dev         1
PCT_ASE | 2, 0              Dev         122
PCT_ASE | 6, 0              Dev         32
PCT_ASE | 24, 0             Audit       13
PCT_ASE | 26, 0             Dev         44
PCT_RLM | 10, 0             Audit       1
PCT_RLM | 54, 0             Audit       1
PCT_RLM | 88, 1             Dev         311
PCT_RLM | 90, 1             Dev         311
PCT_RLM | 91, 1             Dev         311
PCT_RLDF | 33, 8            Dev         1
PCT_RLDF | 48, 0            Dev         311
PCT_RLDF | 49, 75           Dev         304
PCT_RSIP | 13, 0            Audit       2
PCT_RSIP | 23, 0            Dev         26
PCT_RRI | 177, 2            Exception   25
PCT_RRI | 291, 1            Dev         1587
PCT_RRI | 292, 1            Dev         1587
PCT_RRI | 293, 1            Dev         1472
PCT_RRI | 294, 1            Dev         1472
```

### clear mpls debug counters

**Syntax: clear mpls debug counters [aall | ai3 | ambl | amb | amh | ase | asel | bfd | i3 | ipl | nbase-root | ntl | rcp | rcs | rlm | rldf | rric | rri | rrt | rsip | ipr | rspx | rstc | sck | yads]**

This command clears the debug counters.

**NOTE**
The **show mpls debug counter** and **clear mpls debug counters** commands are hidden CLI commands.

## MPLS CSPF debug commands

Constrained Shortest Path First (CSPF) computes the shortest path that fulfills a set of constraints. This means that CSPF runs a shortest path algorithm after pruning any links that violate a given set of constraints, such as minimum bandwidth required per link (also known as bandwidth guaranteed constraint).

### debug mpls cspf

**Syntax: [no] debug mpls cspf [computation |mapping | ted | error | all]**

This debug command displays information about CSPF computations, mapping, TE databases, and errors.

- **computation** - Displays CSPF computation information.
- **mapping** - Displays information about address mappings in the CSPF module.
- **ted** - Displays information about the TE database.
- **error** - Displays CSPF error messages.
- **all** - Displays all debug error messages related to CSPF module.

### debug mpls cspf computation

**Syntax: [no] debug mpls cspf computation [all | detail | lsp]**

- **all** - Displays all debug messages related to CSPF computation.
- **detail** - Displays more detailed information.
- **lsp** - Displays CSPF computation messages for specific LSPs.

This command displays CSPF computation information filtered by more specific criteria. If an LSP does not come up with error code “No path found”, it means that CSPF could not calculate a path for the destination with the specified set of constraints. Enable this debug tracing to know why the path computation failed. Route query related events for LSP including Detour path and Facility Path are traced by enabling this debugging module.

A sample output is as follows:

```
PowerConnect# debug mpls cspf computation
MPLS: CSPF: Unable to find router ID corresponding to destination IP
100.100.100.100 in area 0. LSP not created
MPLS: CSPF: Strict Hop - Locate Link from SrcRtr 20.20.20.20 to DstRtr
100.100.100.100, DstIntfAddr 1.1.1.1
MPLS: CSPF: Strict Hop processing matching link to 100.100.100.100
[O[0]:1.1.2.2:1.1.2.1:20.20.20.20:100.100.100.100]
MPLS: CSPF: Link Constraints - Satisfied:
[O[0]:1.1.2.2:1.1.2.1:20.20.20.20:100.100.100.100]
CSPF: Strict Hop processing matching link to 100.100.100.100
[O[0]:1.1.1.2:1.1.1.1:20.20.20.20:100.100.100.100]
MPLS: CSPF: Link Constraints - Satisfied: [O[0]:1.1.1.2:1.1.1.1:
20.20.20.20:100.100.100.100]
MPLS: CSPF: Strict Hop - Found Link from SrcRtr 20.20.20.20 to DstRtr
100.100.100.100, DstIntfAddr 1.1.1.1
[O[0]:1.1.1.2:1.1.1.1:20.20.20.20:100.100.100.100]
```

```
MPLS: CSPF: Final CSPF route in area 0
          Hop 1: 1.1.1.1, Rtr 100.100.100.100
RSIR: route query for 1.1.1.1/32
RSIR: Route Query success, NH 1.1.1.0 EgressIf e1/1 Ingr 0 Egr 0
RSIR: Route Query success, NH 1.1.1.0 EgressIf e1/1 Ingr 0 Egr 0
MPLS: CSPF: Strict Hop - Locate Link from SrcRtr 20.20.20.20 to DstRtr
100.100.100.100, DstIntfAddr 1.1.2.1
MPLS: CSPF: Strict Hop processing matching link to 100.100.100.1
00 [O[0]:1.1.2.2:1.1.2.1:20.20.20.20:100.100.100.100]
MPLS: CSPF: Link Constraints - Satisfied: [O[0]:1.1.2.2:1.1.2.1:
20.20.20.20:100.100.100.100]
MPLS: CSPF: Strict Hop - Found Link from SrcRtr 20.20.20.20 to DstRtr
100.100.100.100, DstIntfAddr 1.1.2.1 [O[0]:1.1.2.2:1.1.2.1:20.20.20.20:100
.100.100.100]
MPLS: CSPF: Final CSPF route in area 0
          Hop 1: 1.1.2.1, Rtr 100.100.100.100
RSIR: route query for 1.1.2.1/32
RSIR: Route Query success, NH 1.1.2.0 EgressIf e1/2 Ingr 0 Egr 0
RSIR: Route Query success, NH 1.1.2.0 EgressIf e1/2 Ingr 0 Egr 0
```

### debug mpls cspf computation lsp

**Syntax:** [no] **debug mpls cspf computation lsp** [**name** <*name*> | **sess_obj** <*source_ip_address*> <*destination_ip_address*> <*tunnel_id*>]

This command displays CSPF computation information for specific LSPs.

- **name** <*name*> - Limits the display of information to debug messages for the LSP that matches with the specified LSP name.
- **sess_obj** <*source_ip_address*> <*destination_ip_address*> <*tunnel_id*> - Limits the display of information to debug messages for the LSP that matches with the specified session object which includes source IP address, destination IP address, and tunnel ID.

## MPLS forwarding debug commands

### debug mpls forwarding

**Syntax:** [no] **debug mpls forwarding** [**all** | **ldp** | **rsvp** | **resource** | **error**]

- **all** - Displays all debug messages related to MPLS forwarding.
- **ldp** - Displays LDP-related forwarding information.
- **rsvp** - Displays RSVP-related forwarding information.
- **resource** - Displays information about available MPLS resources.
- **error** - Displays MPLS forwarding-related error messages.

MPLS control plane interacts with data forwarding plane through forwarding interface. This debugging command displays RSVP, LDP, and resource usage related information.

Sample output message:

```
PowerConnect# debug mpls forwarding
RLDF: ADD out_cb(0X14319B54), out-s-idx=3, out-int=2, out-lbl=0
lsp_cb=0X1431A220
RLDF: ADD xc_cb(0X14319910), in_cb=0X1308137C, out_cb=0X14319B54,
lsp_cb=0X1431A220
RLDF : Check BW
   Path_TSpec valid: BW = 0 Kb/sec
Resv_TSpec valid: BW = 0 Kb/sec
```

```
Alloc BW[outseg idx: 3]: setup/hold priority 3/7:
Path_TSpec valid: BW = 0 Kb/sec
Resv_TSpec valid: BW = 0 Kb/sec
Allocated BW 0 kbps for priority 0
RLDF: Update XC: lsp_cb 0X00000000, in_cb 0X00000000, out_cb 0X14319B54, xc_cb
0X14319910
RLDF: Update XC: lsp_xc_id 3, in-lbl 0, in-if port_id 65535, out-seg_idx 3, out-if
e1/2
RLDF: update_tnnl_vif_nht_index: Old 65535, new 1
RLDF - tnl 6 goes up
RLDF - tnnl add - For an UP tunnel, the out_seg are the same
RLDF - tnnl add - For an UP tunnel, the out_seg are the same
RLDF - rx'ed DISASSOCIATE_FEC_XC for fec 100.100.100.100
RLDF - tnl 5 goes down
RLDF - tnl 5 deleted from mpls route table and indicated to application
RLDF - LDP Tnnl 5 for fec 100.100.100.100 deleted
RLDF: DEL xc_cb (0X1306BF0C) lsp_cb (0X1306C394)
RLDF: DEL out_cb(0X1306C150), out-s-idx=1, out-int=1, out-lbl=0
lsp_cb=0X1306C394
RLDF: DEL in_cb(0X13081D38), lsp_cb 0X1306C394
RLDF: free lsp_cb=0X1306C394 lsp_xc_index=0X00000001
RLDF: ADD lsp_cb(0X1306C5D8), lsp_xc_idx=0X00000004
RLDF: ADD out_cb(0X1306BF0C), out-s-idx=4, out-int=1, out-lbl=0
lsp_cb=0X1306C5D8
RLDF: Add in_cb(0X13081E9C),lblsp-idx=0, in-if=0,in-lbl=0 lsp_cb 0X1306C5D8
RLDF: ADD xc_cb(0X1306C81C), in_cb=0X13081E9C, out_cb=0X1306BF0C,
lsp_cb=0X1306C5D8
RLDF: Update XC: lsp_cb 0X00000000, in_cb 0X00000000, out_cb 0X1306BF0C, xc_cb
0X1306C81C
RLDF: Update XC: lsp_xc_id 4, in-lbl 0, in-if port_id 65535, out-seg_idx 4, out-if
e1/1
RLDF - rx'ed ASSOCIATE_FEC_XC for fec 100.100.100.100
RLDF - tnl 5 goes up
RLDF - tnl 5 added to mpls route table and indicated to application
RLDF - LDP tnnl 0X00000005 installed for fec 100.100.100.100, outgoing port e1/1
```

### debug mpls forwarding rsvp

**Syntax: [no] debug mpls forwarding rsvp [all | sess_obj** <*source_ip_address*> <*destination_ip_address*> <*tunnel_id*>**]**

This command displays RSVP-related forwarding information filtered by more specific criteria.

- **all** - Displays all debug messages related to RSVP.
- **sess_obj** *source_ip_address*> <*destination_ip_address*> <*tunnel_id*> - limits the display of information to those matching specific RSVP session object which include source IP address, destination IP address and tunnel ID.

### debug mpls forwarding resource

**Syntax: [no] debug mpls forwarding resource**

This command displays information about available MPLS resources. Output resembles the following:

```
PowerConnect# debug mpls forwarding resource
Check BW:
    Path_TSpec valid: BW = 50 Kb/sec
    Resv_TSpec valid: BW = 50 Kb/sec
 Alloc BW[outseg idx: 7]: setup/hold priority 7/0:
```

```
    Path_TSpec valid: BW = 50 Kb/sec
    Resv_TSpec valid: BW = 50 Kb/sec
 Allocated BW 50 kbps for priority 0
```

In this example, (Path_TSpec) TSpec is a collection of QoS parameters in the RSVP path message corresponding to the traffic requirement of the requester or sender. The TSpec is specified in RFC 2210 and is needed to implement RSVP IntServ (IETF integrated services). In this example the requested and reserved bandwidths are both 50 Kbps.

## MPLS routing debug commands

### debug mpls routing

**Syntax:** [no] **debug mpls routing [all | error | interface | prefix]**

- **all** - Displays all debug messages related to MPLS routing.
- **error** - Displays MPLS routing-related error messages.
- **interface** - Limits the messages to specific interfaces. This filter has been provided to capture the event of a particular IP interface indication (or polling) by routing stub module to MPLS.
- **prefix** - Limits the messages to specific prefixes. The purpose of this filter is to trace a particular IP route notification by routing module to MPLS.

This command displays MPLS routing-related information.

The output resembles the following:

```
PowerConnect#debug mpls routing
RSIR skip route add: 1.1.1.0/24, ingr(0), egr(0)
RSIR port state change indication for e1/1: Admin: UP Oper: UP
RSIR IP address indication to RRI for e1/1: addr add 1.1.1.2/24
RSIR IP address indication to RCP for e1/1: addr add 1.1.1.2/24
RSIR IP address indication to RCS for e1/1: addr add 1.1.1.2/24
RSIR route add(RSVP) indication: 1.1.2.0/24, idx 0x0X0A3682E6, sh_cut
0x0X00000000 nh 1.1.2.0, intf e1/2, ingr 0, egr 0, r_flag 0x0X00000000
RSIR skip route add: 1.1.2.0/24, ingr(0), egr(0)
RSIR port state change indication for e1/2: Admin: UP Oper: UP
RSIR IP address indication to RRI for e1/2: addr add 1.1.2.2/24
RSIR IP address indication to RCP for e1/2: addr add 1.1.2.2/24
RSIR IP address indication to RCS for e1/2: addr add 1.1.2.2/24
RSIR route add(RSVP) indication: 1.1.3.0/24, idx 0x0X0A3682F6, sh_cut
0x0X00000000 nh 1.1.3.0, intf e1/3, ingr 0, egr 0, r_flag 0x0X00000000
RSIR skip route add: 1.1.3.0/24, ingr(0), egr(0)
RSIR port state change indication for e1/3: Admin: UP Oper: UP
RSIR IP address indication to RRI for e1/3: addr add 1.1.3.2/24
RSIR IP address indication to RCP for e1/3: addr add 1.1.3.2/24
RSIR IP address indication to RCS for e1/3: addr add 1.1.3.2/24
RSIR port state change indication for e1/4: Admin: UP Oper: UP
```

### debug mpls routing interface

**Syntax:** [no] **debug mpls routing interface [all | ethernet** <*slot/port*> **| pos** <*slot/port*> **| ve** <*index*>**]**

This command display MPLS routing-related information to the interfaces with specific types.

- **all** - Displays all debug messages related to MPLS routing for all the interfaces.
- **ethernet** <*slot/port*> - Limits the messages to specific Ethernet interfaces.
- **pos** <*slot/port*> - Limits the messages to specific POS interfaces.

- **ve** <*index*> - Limits the messages to specific virtual Ethernet interfaces.

**debug mpls routing prefix**

**Syntax: [no] debug mpls routing prefix** <*ip-address*> <*prefix-length*>

This command limits the display of MPLS routing-related information to specific prefixes.

## MPLS RSVP debug commands

**debug mpls rsvp**

**Syntax: [no] debug mpls rsvp [all | error |event | packets | tunnel | session]**

- **all** - Displays all messages related to MPLS RSVP module.
- **error** - Displays debug messages related to MPLS RSVP errors.
- **event** - Displays debug messages related to MPLS RSVP events. This includes those events that are not a session-specific, for example, interface up/down, IP route indication etc.
- **packets** - Displays debug messages related to MPLS RSVP packets.
- **session** - Displays debug messages related to a specific MPLS RSVP session.
- **tunnel** - Displays debug messages related to MPLS RSVP LSP interaction with other modules as virtual interfaces.

**debug mpls rsvp event**

**Syntax: [no] debug mpls rsvp event**

This command displays information about MPLS rsvp events, as shown in the following example:

```
PowerConnect#debug mpls rsvp event
RSVP: kill_tc DEL_IN_SEG Sess 0x0da02020 33.33.33.1(2)<-1.11.11.
RSVP: kill_session 0x0da02020 33.33.33.1(2)<-1.11.11.11
RSVP: make_session 0x0da022f0 10.4.1.2(1)<-1.11.11.11
RSVP: make_PSB 0x0da10fc8 Sess 0x0da022f0 10.4.1.2(1)<-1.11.11.1
RSVP: make_PSB 0x0da107f8 Sess 0x0da022f0 10.4.1.2(1)<-1.11.11.1
RSVP: new_tc QUERY_ROUTE Sess 0x0da022f0 10.4.1.2(1)<-1.11.11.11
RSVP: kill_tc QUERY_ROUTE Sess 0x0da022f0 10.4.1.2(1)<-1.11.11.1
rrr_query_route_rsp: inserting PSB 0x0da10fc8 in unrouted list
RSVP_FRR: rrr_frr_merge_point: Merging PSB not found.
RSVP_FRR: Path Tear on non-merging protected LSP, PSB 0da10fc8.
RSVP_FRR: Tear down PLR detour 0da107f8
RSVP: kill_PSB 0x0da10fc8 Sess 0x0da022f0 10.4.1.2(1)<-1.11.11.1
RSVP_FRR: rrr_frr_merge_point: Merging PSB not found.
RSVP: new_tc DEL_IN_SEG Sess 0x0da022f0 10.4.1.2(1)<-1.11.11.11
RSVP: kill_tc DEL_IN_SEG Sess 0x0da022f0 10.4.1.2(1)<-1.11.11.11
RSVP: kill_session 0x0da022f0 10.4.1.2(1)<-1.11.11.11
RSVP: new_tc TIMER_POP Sess 0x0da03ad8 10.1.1.2(4)<-1.11.11.11
RSVP: kill_tc TIMER_POP Sess 0x0da03ad8 10.1.1.2(4)<-1.11.11.11
RSVP: new_tc TIMER_POP Sess 0x0da02e30 10.3.3.2(5)<-1.11.11.11
RSVP: kill_tc TIMER_POP Sess 0x0da02e30 10.3.3.2(5)<-1.11.11.11
RSVP: new_tc TIMER_POP Sess 0x0da03ad8 10.1.1.2(4)<-1.11.11.11
RSVP: kill_tc TIMER_POP Sess 0x0da03ad8 10.1.1.2(4)<-1.11.11.11
RSVP: new_tc TIMER_POP Sess 0x0da041e0 22.22.22.1(3)<-1.11.11.11
RSVP: kill_tc TIMER_POP Sess 0x0da041e0 22.22.22.1(3)<-1.11.11.1
RCPF RE: fec type 2(1/0) dest 10.4.1.0 nexthop 10.1.1.2: prefix len 24
```

```
dest_inet_pl 4, rt_idx 0, event 2, fec_cb 0x0d902138
RCPF RE: fec cb: ing/egress =  (1/0), rt_index 0, lib_ret 0, state 1 pend_not 0
RCPF RE: fec type 2(1/0) dest 44.44.44.0 nexthop 10.1.1.2: prefix len 24
dest_inet_pl 4, rt_idx 0, event 2, fec_cb 0x0d90f5b8
```

### debug mpls rsvp packets

**Syntax:** **[no] debug mpls rsvp packets [all | detail | count** <*number*>**| direction [send | receive]| pkt_type [path | patherr | resv] | interface | sess_obj** <*source_ip_address*> <*destination_ip_address*> <*tunnel_id*>**]**

This command displays MPLS RSVP packets-related information, which is further filtered by direction, packet type, interface, and session object.

- **all** - Displays all messages related to MPLS RSVP packets.
- **detail** - Displays detailed information about MPLS RSVP packets.
- **count** <*number*> - Limits the display of MPLS RSVP packets to the specified number.
- **direction** - Displays information about MPLS RSVP packets for the specified direction.
- **send** - Displays information about sent MPLS RSVP packets.
- **receive** - Displays information about received MPLS RSVP packets.
- **pkt_type** - Displays information about MPLS RSVP packet types (similar to the **debug rsvp packets detail** command).
- **path** - Displays information about MPLS RSVP packet paths.
- **patherr** - Displays information about MPLS RSVP packet path errors.
- **resv** - Displays information about reserved MPLS RSVP packets.
- **interface** - Displays RSVP packets transmitted or received on an interface.
- **sess_obj** <*source_ip_address*> <*destination_ip_address*> <*tunnel_id*> - Displays information about the MPLS RSVP packets for the specified RSVP session object which includes source IP address, destination IP address, and tunnel ID.

The output resembles the following:

```
PowerConnect#debug mpls rsvp packets
Send Path message: src 20.20.20.20, dst 100.100.100.100 on port 1/1
Dest 100.100.100.100, tunnelId 1, ext tunnelId 20.20.20.20
Receive Resv message: src 1.1.2.1, dst 1.1.2.2 on port 1/2
Dest 100.100.100.100, tunnelId 2, ext tunnelId 20.20.20.20
```

### debug mpls rsvp packets interface

**Syntax:** **[no] debug mpls rsvp packets interface [all | ethernet** <*slot/port*> **| pos** <*slot/port*> **| ve** <*index*>**]**

This command limits the display of MPLS RSVP packets to interfaces with specific types.

- **all** - Displays all MPLS RSVP packets on all interfaces.
- **ethernet** <*slot/port*> - Limits the display of packets to specific Ethernet interfaces.
- **pos** <*slot/port*> -Limits the display of packets to specific POS interfaces.
- **ve** <*index*> - Limits the display of packets to specific virtual Ethernet interfaces.

### debug mpls rsvp session

**Syntax:** **[no] debug mpls rsvp session [all | detail | lsp]**

This command displays MPLS RSVP session-related information.

- **all** - Displays all messages related to a MPLS RSVP session.
- **detail** - Displays messages in detailed version.
- **lsp** - Limits the display of a MPLS RSVP session to specific LSPs.

The output resembles the following:

```
PowerConnect#debug mpls rsvp session
Processing input queue event "RSVP_pkt" for SESS(100.100.100.100
/1/20.20.20.20) Destp 0X142ED82C
Processing input queue event "Path_refresh_tmr_exp" for SESS(100
.100.100.100/1/20.20.20.20) Destp 0X142ED82C
Processing input queue event "Path_refresh_tmr_exp" for SESS(100
.100.100.100/2/20.20.20.20) Destp 0X142ECC80
Processing input queue event "Resv_refresh_tmr_exp" for SESS(100
.100.100.100/2/20.20.20.20) Destp 0X142ECC80
Processing input queue event "RSVP_pkt" for SESS(100.100.100.100
/1/20.20.20.20) Destp 0X142ED82C
Processing input queue event "Resv_refresh_tmr_exp" for SESS(100
.100.100.100/1/20.20.20.20) Destp 0X142ED82C
Processing input queue event "RSVP_pkt" for SESS(100.100.100.100
/2/20.20.20.20) Destp 0X142ECC80
Processing input queue event "Path_refresh_tmr_exp" for SESS(100
.100.100.100/1/20.20.20.20) Destp 0X142ED82C
Processing input queue event "Resv_refresh_tmr_exp" for SESS(100
.100.100.100/2/20.20.20.20) Destp 0X142ECC80
Processing input queue event "RSVP_pkt" for SESS(100.100.100.100
/1/20.20.20.20) Destp 0X142ED82C
Processing input queue event "Path_refresh_tmr_exp" for SESS(100
.100.100.100/2/20.20.20.20) Destp 0X142ECC80
Processing input queue event "Path_refresh_tmr_exp" for SESS(100
.100.100.100/1/20.20.20.20) Destp 0X142ED82C
Processing input queue event "Resv_refresh_tmr_exp" for SESS(100
.100.100.100/2/20.20.20.20) Destp 0X142ECC80
Processing input queue event "RSVP_pkt" for SESS(100.100.100.100
/1/20.20.20.20) Destp 0X142ED82C
Processing input queue event "Resv_refresh_tmr_exp" for SESS(100
.100.100.100/1/20.20.20.20) Destp 0X142ED82C
Processing input queue event "RSVP_pkt" for SESS(100.100.100.100
/2/20.20.20.20) Destp 0X142ECC80
Processing input queue event "Path_refresh_tmr_exp" for SESS(100
.100.100.100/2/20.20.20.20) Destp 0X142ECC80
```

### debug mpls rsvp session lsp

**Syntax:** **[no] debug mpls rsvp session lsp [name** <*name*>**| sess_obj** <*source_ip_address*> <*destination_ip_address*> <*tunnel_id*>**]**

This command displays RSVP session information to specific LSPs.

- **name** <*name*> - Limits the display of information to the LSP that matches with the specified LSP name.
- **sess_obj** <*source_ip_address*> <*destination_ip_address*> <*tunnel_id*> - Limits the display of information to the LSP that matches with the specified LSP session object which includes source IP address, destination IP address, and tunnel ID.

### debug mpls rsvp tunnel

**Syntax:** **[no] debug mpls rsvp tunnel [all | detail | lsp]**

This command displays MPLS RSVP LSP as tunnel interface-related information.

- **all** - Displays all messages related to MPLS RSVP tunnel.
- **detail** - Displays detailed information about RSVP tunnel state transitions for all tunnels and their retries whenever the retry timer is expired.
- **lsp** - Displays RSVP tunnel information to specific LSPs.

Output resembles the following:

```
PowerConnect#debug mpls rsvp tunnel
MPLS: TNNL(test1): try signal LSP
MPLS: TNNL(test1): event = 2(ENABLE_CSPF_OK), change from state 3(PATH_SENT) to
2(ROUTE_FOUND)
MPLS: TNNL(test1): event = 31(SENT_PATH), change from state 2(ROUTE_FOUND) to
3(PATH_SENT)
RSVP_TNNL(test1): Update tunnel_vif_index 2
RSVP_TNNL(test1): Update tunnel oper old 0, new 1
MPLS: TNNL(test1): tnl 2 goes up
MPLS: TNNL(test1): primary(current instance), path path1 up
MPLS: TNNL(test1): activate primary
MPLS: TNNL(test1): tnl 2 added to mpls route table and indicated to application
MPLS: TNNL(test1): notify IP with vif 2 UP notification
```

### debug mpls rsvp tunnel lsp

**Syntax:** **[no] debug mpls rsvp tunnel lsp [name** <*name*>**| sess_obj** <*source_ip_address*> <*destination_ip_address*> <*tunnel_id*>**]**

This command displays RSVP tunnel information to specific LSPs.

- **name** <*name*> - Limits the display of information to debug messages for the specified LSP using LSP name.
- **sess_obj** <*source_ip_address*> <*destination_ip_address*> <*tunnel_id*> - Limits the display of information to debug messages for the specified LSP using LSP session object which includes source IP address, destination IP address and tunnel ID.

## MPLS label manager debug commands

### debug mpls lmgr

**Syntax:** **[no] debug mpls lmgr [all | error | ldp | rsvp]**

This command displays label manager-related information.

- **all** - Displays all debug messages related to label manager.
- **error** - Displays label manager-related error messages.
- **ldp** - Displays label manager information for LDP.
- **rsvp** - Displays label manager information for RSVP.

The output resembles the following:

```
PowerConnect#debug mpls lmgr
LMGR 1 sent LSI_COMMON_LSP_XC SESSION (100.100.100.100, 0X0007, 20.20.20.20) LSP
(0X0001, 20.20.20.20), ref_flags (2,4,0,0,2,0,0), update_flags 0X00000011,
lspxc_flags 0X00000000, rvs_ref flags (0,0,0,0,0,0,0), rvs_update_flags
0X00000000, in_seg_index 0, xc_index 5, out_seg_index 0.
LMGR 1 rcvd LSI_COMMON_LSP_XC SESSION (100.100.100.100, 0X0007, 20.20.20.20) LSP
(0X0001, 20.20.20.20), ref_flags (2,4,0,0,2,0,0), update_flags 0X00000011,
lspxc_flags 0X00000000, rvs_ref flags (0,0,0,0,0,0,0), rvs_update_flags
0X00000000, rc 1, in_seg_index 0, xc_index 5, out_seg_index 0.
```

```
LMGR 1 sent LSI_COMMON_LSP_XC SESSION (100.100.100.100, 0X0007, 20.20.20.20) LSP
(0X0001, 20.20.20.20), ref_flags (1,4,2,2,1,0,2), update_flags 0X0000004C,
lspxc_flags 0X00000000, rvs_ref flags (0,0,0,0,0,0,0), rvs_update_flags
0X00000000, in_seg_index 0, xc_index 5, out_seg_index 4.
LMGR 1 rcvd LSI_COMMON_LSP_XC SESSION (100.100.100.100, 0X0007, 20.20.20.20) LSP
(0X0001, 20.20.20.20), ref_flags (1,4,2,2,1,0,2), update_flags 0X0000004C,
lspxc_flags 0X00000000, rvs_ref flags (0,0,0,0,0,0,0), rvs_update_flags
0X00000000, rc 1, in_seg_index 0, xc_index 5, out_seg_index 4.
```

### debug mpls lmgr rsvp

**Syntax:** [no] **debug mpls lmgr rsvp [all | lsp [name** <*name*> **| sess_obj** <*source_ip_address*> <*destination_ip_address*> <*tunnel_id*>]]

This command displays label manager information for RSVP.

- **all** - Displays all debug messages related to label manager for RSVP.
- **lsp** - Displays label manager information for RSVP to specific LSPs.
  - **name** <*name*> - Limits the display of information to debug messages for the specified LSP using LSP name.
  - **sess_obj** <*source_ip_address*> <*destination_ip_address*> <*tunnel_id*> - Limits the display of information to debug messages for the specified LSP session object which includes source IP address, destination IP address, and tunnel ID.

## MPLS VLL debug commands

### debug mpls vll

**Syntax:** [no] **debug mpls vll**

This command displays information about activity in MPLS Virtual Lease Line (VLL) configurations. In the following example, VLL 5 is turned off, and then turned on again. With **debug mpls vll** enabled, the activity around this event is displayed.

```
PowerConnect# debug mpls vll
MPLS VLL debug turned ON.
PowerConnect#show mpls conf vll-local
PowerConnect(config-mpls)#no vll 5 5
DEBUG VLL: Configured VLL in Tagged-mode(default-mode)
DEBUG_VLL: Trap - Vll event is DOWN event, min-index 0, max-index 0
DEBUG_VLL: Final Index min_index 0, max-index 0
DEBUG VLL: send vll info to LP
DEBUG VLL: MP_VLL_DY_SYNC: Outbound - vll_name = 5, vll_id = 5, vll_port = 3,
vlan_id = 301
DEBUG VLL : remote_vc = 800000, tunnel_label = 3, mode TAGGED-MODE, action DELETE
DEBUG VLL : MP_VLL_DY_SYNC: Inbound - vll_id = 5, vll_port = 3, vlan_id = 301,
VCLabel = 800000, tag_type = 2

DEBUG VLL : ** Tunnel has changed for peer 1.1.1.1
DEBUG VLL :      VLL VC ID 6:  current tnnl = 3, new tnnl = 0
DEBUG VLL-COS: BETTER-TUNNEL Selection: VLL 6 cos 0, LSP:old_cos 0, new_cos 0,
ols_num_vll 6, new_num_vll 0
DEBUG VLL-COS: BETTER-TUNNEL Selection: vll 6, cos 0, new_lsp_qos 0
DEBUG VLL : send vll info to LP
DEBUG VLL : MP_VLL_DY_SYNC: Outbound - vll_name = 6, vll_id = 6, vll_port = 3,
vlan_id = 302
DEBUG VLL : remote_vc = 800001, tunnel_label = 3, mode TAGGED-MODE, action DELETE
DEBUG VLL : MP_VLL_DY_SYNC: Inbound - vll_id = 6, vll_port = 3, vlan_id = 302,
```

```
VCLabel = 800001, tag_type = 2
DEBUG VLL-COS: vll 6 is OPERATIONAL with cos 0
DEBUG VLL : send vll info to LP
DEBUG VLL : MP_VLL_DY_SYNC: Outbound - vll_name = 6, vll_id = 6, vll_port = 3,
vlan_id = 302
DEBUG VLL : remote_vc = 800001, tunnel_label = 3, mode TAGGED-MODE, action ADD
DEBUG VLL : MP_VLL_DY_SYNC: Inbound - vll_id = 6, vll_port = 3, vlan_id = 302,
VCLabel = 800001, tag_type = 2
DEBUG VLL :      VLL VC ID 7:  current tnnl = 3, new tnnl = 2
DEBUG VLL-COS: BETTER-TUNNEL Selection: VLL 7 cos 0, LSP:old_cos 0, new_cos 0,
ols_num_vll 5, new_num_vll 0
DEBUG VLL-COS: BETTER-TUNNEL Selection: vll 7, cos 0, new_lsp_qos 0
DEBUG VLL : send vll info to LP
DEBUG VLL : MP_VLL_DY_SYNC: Outbound - vll_name = 7, vll_id = 7, vll_port = 3,
vlan_id = 303
DEBUG VLL : remote_vc = 800002, tunnel_label = 3, mode TAGGED-MODE, action DELETE
DEBUG VLL : MP_VLL_DY_SYNC: Inbound - vll_id = 7, vll_port = 3, vlan_id = 303,
VCLabel = 800002, tag_type = 2
DEBUG VLL-COS: vll 7 is OPERATIONAL with cos 0
DEBUG VLL : send vll info to LP
DEBUG VLL : MP_VLL_DY_SYNC: Outbound - vll_name = 7, vll_id = 7, vll_port = 3,
vlan_id = 303
DEBUG VLL : remote_vc = 800002, tunnel_label = 3, mode TAGGED-MODE, action ADD
DEBUG VLL : MP_VLL_DY_SYNC: Inbound - vll_id = 7, vll_port = 3, vlan_id = 303,
VCLabel = 800002, tag_type = 2
DEBUG VLL :      VLL VC ID 8:  current tnnl = 3, new tnnl = 1
DEBUG VLL-COS: BETTER-TUNNEL Selection: VLL 8 cos 0, LSP:old_cos 0, new_cos 0,
ols_num_vll 4, new_num_vll 0
DEBUG VLL-COS: BETTER-TUNNEL Selection: vll 8, cos 0, new_lsp_qos 0
DEBUG VLL : send vll info to LP
DEBUG VLL : MP_VLL_DY_SYNC: Outbound - vll_name = 8, vll_id = 8, vll_port = 3,
vlan_id = 304
DEBUG VLL : remote_vc = 800003, tunnel_label = 3, mode TAGGED-MODE, action DELETE
DEBUG VLL : MP_VLL_DY_SYNC: Inbound - vll_id = 8, vll_port = 3, vlan_id = 304,
VCLabel = 800003, tag_type = 2
DEBUG VLL-COS: vll 8 is OPERATIONAL with cos 0
DEBUG VLL : send vll info to LP
DEBUG VLL : MP_VLL_DY_SYNC: Outbound - vll_name = 8, vll_id = 8, vll_port = 3,
vlan_id = 304
DEBUG VLL : remote_vc = 800003, tunnel_label = 3, mode TAGGED-MODE, action ADD
DEBUG VLL : MP_VLL_DY_SYNC: Inbound - vll_id = 8, vll_port = 3, vlan_id = 304,
VCLabel = 800003, tag_type = 2
DEBUG VLL :      VLL VC ID 9:  current tnnl = 3, new tnnl = 0
DEBUG VLL-COS: BETTER-TUNNEL Selection: VLL 9 cos 0, LSP:old_cos 0, new_cos 0,
ols_num_vll 3, new_num_vll 1
DEBUG VLL-COS: BETTER-TUNNEL Selection: vll 9, cos 0, new_lsp_qos 0
DEBUG VLL : send vll info to LP
DEBUG VLL : MP_VLL_DY_SYNC: Outbound - vll_name = 9, vll_id = 9, vll_port = 3,
vlan_id = 305
DEBUG VLL : remote_vc = 800004, tunnel_label = 3, mode TAGGED-MODE, action DELETE
DEBUG VLL : MP_VLL_DY_SYNC: Inbound - vll_id = 9, vll_port = 3, vlan_id = 305,
VCLabel = 800004, tag_type = 2
DEBUG VLL-COS: vll 9 is OPERATIONAL with cos 0
DEBUG VLL : send vll info to LP
DEBUG VLL : MP_VLL_DY_SYNC: Outbound - vll_name = 9, vll_id = 9, vll_port = 3,
vlan_id = 305
DEBUG VLL : remote_vc = 800004, tunnel_label = 3, mode TAGGED-MODE, action ADD
DEBUG VLL : MP_VLL_DY_SYNC: Inbound - vll_id = 9, vll_port = 3, vlan_id = 305,
VCLabel = 800004, tag_type = 2
DEBUG VLL : ** End peer 1.1.1.1
```

```
PowerConnect(config-mpls)# vll 5 5
DEBUG VLL : Configured VLL in Tagged-mode(default-mode)
DEBUG_VLL: VLL ID 5, vll-index 0
PowerConnect(config-mpls-vll-5)# vll-peer 1.1.1.1
PowerConnect(config-mpls-vll-5)# vlan 301
PowerConnect(config-mpls-vll-5-vlan)#  tagged e 1/4
DEBUG VLL : No corresponding vll
DEBUG VLL : No corresponding vll

DEBUG VLL-COS: vll 5 is OPERATIONAL with cos 0
DEBUG VLL : send vll info to LP
DEBUG VLL : MP_VLL_DY_SYNC: Outbound - vll_name = 5, vll_id = 5, vll_port = 3,
vlan_id = 301
DEBUG VLL : remote_vc = 800000, tunnel_label = 3, mode TAGGED-MODE, action ADD
DEBUG VLL : MP_VLL_DY_SYNC: Inbound - vll_id = 5, vll_port = 3, vlan_id = 301,
VCLabel = 800005, tag_type = 2
DEBUG_VLL: Trap - Vll event is UP event, min-index 0, max-index 0
DEBUG_VLL: Final Index min_index 0, max-index 0

PowerConnect(config-mpls-vll-5-vlan)# show mpls vll
Name      VC-ID      Vll-peer        End-point                 State Tunnel-LSP
5         5          1.1.1.1         tag vlan 301  e 1/4       UP    tog2
6         6          1.1.1.1         tag vlan 302  e 1/4       UP    tog
7         7          1.1.1.1         tag vlan 303  e 1/4       UP    tog2
8         8          1.1.1.1         tag vlan 304  e 1/4       UP    tog1
9         9          1.1.1.1         tag vlan 305  e 1/4       UP    tog
```

### debug mpls vll local

**Syntax:** [no] debug mpls vll local

This command displays information about activity in local MPLS Virtual Lease Line (VLL-Local) configurations. In the following example, VLL-local 10 is turned off, and then turned on again. With **debug mpls vll** enabled, the activity around this event is displayed.

```
PowerConnect(config)#router mpls
PowerConnect(config-mpls)#show debug
MPLS Debug Settings

    VLL Local

PowerConnect(config-mpls)#no vll-local 10
 DEBUG VLL LOCAL : MP_VLL_LOCAL_DY_SYNC: name = 10, vll_local_id = 10
                 :port1 = 65535, vlan_id1 = 1 cos1 = 0
                 :port2 = 23, vlan_id2 = 501, cos2 = 0 state = 0
 DEBUG VLL LOCAL : MP_VLL_LOCAL_DY_SYNC: 0x029a6862 1 102
 DEBUG VLL LOCAL : state change: name = 10, vll_local_id = 10, state = 0 oldState
= 1
 DEBUG VLL LOCAL : MP_VLL_LOCAL_DY_SYNC: name = 10, vll_local_id = 10
                 :port1 = 65535, vlan_id1 = 1 cos1 = 0
                 :port2 = 65535, vlan_id2 = 501, cos2 = 0 state = 0
 DEBUG VLL LOCAL : MP_VLL_LOCAL_DY_SYNC: 0x029a6862 2 184
 DEBUG VLL LOCAL : state change: name = 10, vll_local_id = 10, state = 0 oldState
= 0
 DEBUG VLL LOCAL : MP_VLL_LOCAL_DY_SYNC: name = 10, vll_local_id = 10
                 :port1 = 65535, vlan_id1 = 1 cos1 = 0
                 :port2 = 65535, vlan_id2 = 501, cos2 = 0 state = 0
 DEBUG VLL LOCAL : MP_VLL_LOCAL_DY_SYNC: 0x029a6862 3 266

PowerConnect(config-mpls)#vll-local 10
```

```
 DEBUG VLL LOCAL : MP_VLL_LOCAL_DY_SYNC: name = 10, vll_local_id = 1
                 :port1 = 65535, vlan_id1 = 1 cos1 = 0
                 :port2 = 65535, vlan_id2 = 1, cos2 = 0 state = 0
 DEBUG VLL LOCAL : MP_VLL_LOCAL_DY_SYNC: 0x0299f862 1 102
 DEBUG VLL LOCAL : state change: name = 10, vll_local_id = 1, state = 0 oldState =
0

PowerConnect(config-mpls-vll-lo-10)# vlan 501
PowerConnect(config-mpls-vll-lo-10-vlan)#tag e 1/4
DEBUG VLL LOCAL : MP_VLL_LOCAL_DY_SYNC: name = 10, vll_local_id = 1
                 :port1 = 3, vlan_id1 = 501 cos1 = 0
                 :port2 = 65535, vlan_id2 = 1, cos2 = 0 state = 0
DEBUG VLL LOCAL : MP_VLL_LOCAL_DY_SYNC: 0x0299f862 2 184
DEBUG VLL LOCAL : state change: name = 10, vll_local_id = 1, state = 0 oldState =
0

PowerConnect(config-mpls-vll-lo-10-if-e-1/4)#tag e 2/4
DEBUG VLL LOCAL : MP_VLL_LOCAL_DY_SYNC: name = 10, vll_local_id = 1
                 :port1 = 3, vlan_id1 = 501 cos1 = 0
:port2 = 23, vlan_id2 = 501, cos2 = 0 state = 0
DEBUG VLL LOCAL : MP_VLL_LOCAL_DY_SYNC: 0x0298d062 1 102
DEBUG VLL LOCAL : state change: name = 10, vll_local_id = 1, state = 1 oldState =
0
```

# MPLS LDP

Dell devices support the Label Distribution Protocol (LDP) for setting up non-traffic-engineered tunnel LSPs in an MPLS network.

When used to create tunnel LSPs, LDP allows a set of destination IP prefixes (known as a Forwarding Equivalence Class or FEC) to be associated with an LSP. Each label switch router (LSR) establishes a peer relationship with its neighboring LDP-enabled routers and exchanges label mapping information, which is stored in an LDP database.

The result of an LDP configuration is a full mesh of LSPs in an MPLS network, with each LDP-enabled router a potential ingress, transit, or egress LSR, depending on the destination.

The implementation supports the following aspects of LDP.

## MPLS LDP show commands

You can display the following information about LDP:

- The LDP version number, as well as the LSR's LDP identifier and loopback number
- Information about active LDP-created LSPs on the device
- Information about LDP-created tunnel LSPs for which this device is the ingress LER
- The contents of the LDP database
- Information about the LDP session between this LSR and its LDP peers
- Information about the connection between this LSR and its LDP peers
- Information about LDP-enabled Interfaces on the LSR

### show mpls ldp

**Syntax: show mpls ldp**

To display the LDP version number, the LSR's LDP identifier and loopback number, and the LDP hello interval and hold time, enter the **show mpls ldp** command:

```
PowerConnect(config)#show mpls ldp
Label Distribution Protocol version 1
  LSR ID: 2.2.2.2, using Loopback 1 (deleting it will stop LDP)
  Hello interval: Link 5 sec, Targeted 15 sec
  Hold time value sent in Hellos: Link 15 sec, Targeted 45 sec
 Keepalive interval: 6 sec, Hold time multiple: 6 intervals
```

### show mpls ldp path

**Syntax: show mpls ldp path**

Use the **show mpls ldp path** command to display information about active LDP-created LSPs for which the device is an ingress, transit or egress LSR. For example:

```
PowerConnect(config)#show mpls ldp path
Upstr-session(label)     Downstr-session(label, intf)    Destination route
33.3.3.3:0(3)            (egress)                        11.1.1.1/32
22.2.2.2:0(3)            (egress)                        11.1.1.1/32
33.3.3.3:0(1024)         22.2.2.2:0(3, e2/10)            22.2.2.2/32
22.2.2.2:0(1024)         22.2.2.2:0(3, e2/10)            22.2.2.2/32
(ingress)                22.2.2.2:0(3, e2/10)            22.2.2.2/32
33.3.3.3:0(1026)         33.3.3.3:0(3, e2/20)            33.3.3.3/32
22.2.2.2:0(1026)         33.3.3.3:0(3, e2/20)            33.3.3.3/32
(ingress)                33.3.3.3:0(3, e2/20)            33.3.3.3/32
```

**NOTE**
In this context, "upstream" and "downstream" refer to the direction that data traffic flows in an LSP. This is opposite of the direction that labels are distributed using LDP.

### show mpls ldp tunnel

**Syntax: show mpls ldp tunnel**

This command displays information about LDP-created LSPs for which this device is the ingress LER:

```
PowerConnect#show mpls ldp tunnel
                Oper      Tunnel    Outbound
To              State     Intf      Intf
22.2.2.2        UP        tnl0      e3/1
33.3.3.3        UP        tnl1      e3/2
```

### show mpls ldp database

**Syntax: show mpls ldp database**

This command displays the LSR LDP Label Information Base, which contains all the labels that have been learned from each LSR peer, as well as all of the labels it has sent to its LDP peers.

```
PowerConnect#show mpls ldp database
Session 1.1.1.1:0 - 2.2.2.2:0
 Downstream label database:
   Label     Prefix              State
```

```
   3         2.2.2.2/32          Installed
   1104      3.3.3.3/32          Retained
   1106      14.14.14.14/32      Retained
   1107      44.44.44.44/32      Retained
   800005    VC-FEC              Installed
 Upstream label database:
   Label     Prefix
   3         1.1.1.1/32
   1024      2.2.2.2/32
   1026      3.3.3.3/32
   1028      14.14.14.14/32
   1029      44.44.44.44/32
   800005    VC-FEC
   800006    VC-FEC
```

### show mpls ldp session

**Syntax: show mpls ldp session [detail]**

This command displays information about the LDP session between an LSR and its LDP peers, as shown in this example:

```
PowerConnect#show mpls ldp session
Peer LDP ID             State          Adj Used  My Role   Max Hold  Time Left
2.2.2.2:0               Operational    Link      Passive   36        32
3.3.3.3:0               Operational    Link      Passive   36        26
8.8.8.8:0               Operational    Targeted  Passive   36        33
14.14.14.14:0           Operational    Targeted  Passive   36        24
```

To retrieve more detailed information, enter the **show mpls ldp session detail** command:

```
PowerConnect#show mpls ldp session detail
Peer LDP ID: 1.1.1.1:0, Local LDP ID: 2.2.2.2:0, State: Operational
 Adj: Link, Role: Active, Next keepalive: 2 sec, Hold time left: 26 sec
  Keepalive interval: 6 sec, Max hold time: 36 sec
  MD5 Authentication Key: $+b!o
  Neighboring interfaces: p4/1
  TCP connection: 2.2.2.2:9002--1.1.1.1:646, State: ESTABLISHED
  Next-hop addresses received from the peer:
    1.1.1.1  10.1.1.1  11.1.1.1  12.1.1.1  13.1.1.1  40.1.1.1  43.1.1.1
```

### show mpls ldp neighbor

**Syntax: show mpls ldp neighbor**

This command displays information about the connection between this LSR and its LDP-enabled neighbors, as shown in the following example:

```
PowerConnect#show mpls ldp neighbor
Nbr Transport       Interface      Nbr LDP ID          Max Hold  Time Left
1.1.1.1             p4/1           1.1.1.1:0           15        14
5.5.5.5             p3/2           5.5.5.5:0           15        11
4.4.4.4             (targeted)     4.4.4.4:0           15        13
```

### show mpls ldp interface

**Syntax: show mpls ldp interface**

This command displays information about LDP-enabled interfaces on the LSR, as shown in the following example:

```
PowerConnect#show mpls ldp interface
                   Label-space    Nbr           Hello          Next
Interface          ID             Count         Interval       Hello
e4/1               0              1             5              --
(targeted)         1              0             15             --
```

### show mpls ldp peer

**Syntax: show mpls ldp peer [detail]**

This command displays LDP peer information, as shown in this example:

```
PowerConnect#show mpls ldp peer
Peer LDP ID        State          Num-VLL        Num-VPLS-Peer
2.2.2.2:0          Operational    2              0
3.3.3.3:0          Operational    0              0
8.8.8.8:0          Operational    2              0
9.9.9.9:0          Unknown        2              0
14.14.14.14:0      Operational    1              0
```

To display more detailed information about LDP peers, enter the **show mpls ldp peer detail** command, as shown in the following example:

```
 PowerConnect#show mpls ldp peer detail
 Peer LDP ID: 2.2.2.2:0, Local LDP ID: 1.1.1.1:0, State: Operational
 Session Status UP, Entity Idx: 4, Targeted: No, Target Adj Added: Yes
 Num VLL: 2, Num VPLS: 0
 Rcvd VC-FECs:
  From 2.2.2.2: Label: 800001, VC Id: 120, Grp_Id: 0, VC Type: 4
 Peer LDP ID: 8.8.8.8:0, Local LDP ID: 1.1.1.1:0, State: Operational
 Session Status UP, Entity Idx: 2, Targeted: Yes, Target Adj Added: Yes
 Num VLL: 2, Num VPLS: 0
 Rcvd VC-FECs:
  From 8.8.8.8: Label: 16, VC Id: 19, Grp_Id: 0, VC Type: 32773
  From 8.8.8.8: Label: 18, VC Id: 18, Grp_Id: 0, VC Type: 32772
```

### show mpls ldp fec prefix

**Syntax: show mpls ldp fec prefix [<*IP address*> |<*IP address_with_NetMask*>]**

This command displays host address and prefix FECs from the LDP FEC database:

```
PowerConnect#show mpls ldp fec prefix
Total number of prefix FECs: 2
Destination        State     Out-intf    Next-hop       Ingress    Egress
125.125.125.1/32   current   e2/2        90.90.90.20    Yes        No
128.128.128.0/24   current   --          --             No         Yes
```

Use the <*IP address*> |<*IPaddress_with_NetMask*> options for a detailed view of the specified FEC, as shown in the following example:

```
PowerConnect#show mpls ldp fec prefix 125.125.125.1/32
FEC_CB: 0x29391f8c, idx: 1, type: 2, pend_notif: None
  State: current, Ingr: Yes, Egr: No, UM Dist. done: No
  Prefix: 125.125.125.1/32, next_hop: 90.90.90.20, out_if: e2/2

  Downstream mappings:
  Local LDP ID       Peer LDP ID      Label  State     CB
  128.128.128.28:0   125.125.125.1:0  3      Installed 0x29391cb0(-1)
```

### show mpls ldp fec summary

**Syntax: show mpls ldp fec summary**

This command displays LDP FEC summary information.

```
PowerConnect#show mpls ldp fec summary
L3 FEC summary:
Total number of prefix FECs: 2
Total number of VC-FEC type 128: 0
Total number of VC-FEC type 129: 0
Total number of route update processing errors: 0
Total number of VC FEC processing errors: 0
```

### show mpls ldp fec vc

**Syntax: show mpls ldp fec vc <*vc-id*>**

You can display a list of VC FECs from the LDP FEC database as shown in the following example:

```
PowerConnect#show mpls ldp fec vc
Total number of VC FECs: 2
Peer LDP ID      State      VC-ID  VC-Type   FEC-Type   Ingress    Egress
125.125.125.1:0  current    100    4         128        Yes        Yes
125.125.125.1:0  current    1000   5         128        Yes        Yes
```

You can display detailed information about a specific VC by entering the **show mpls ldp fec vc** command with a VC ID as shown in this example:

```
PowerConnect#show mpls ldp fec vc 100
FEC_CB: 0x29391510, idx: 6, type: 128, pend_notif: None
State: current, Ingr: Yes, Egr: Yes, UM Dist. done: Yes
VC-Id: 100, vc-type: 4, grp-id: 0
Local-mtu: 1500, remote-mtu: 1500, MTU enforcement: enabled

Downstream mappings:
Local LDP ID      Peer LDP ID      Label    State     CB
128.128.128.28:0  125.125.125.1:0  800000   Installed 0x29391328(-1)

Upstream mappings:
Local LDP ID      Peer LDP ID      Label    CB
128.128.128.28:0  125.125.125.1:0  800003   0x2939141c(-1)
```

When a VLL or VPLS peer is up, only one FEC_CB is shown.

The MTU enforcement field together with local and remote MTU will indicate whether MTU mismatch is detected.

When the local and remote VC types for a particular VC ID do not match, two FEC_CBs will be shown.

## MPLS LDP debug commands

The following sections describe the **debug mpls ldp** commands, including examples.

### debug mpls ldp

**Syntax: [no] debug mpls ldp [all | adjacency | error | event | fec| packets | socket | state | tcpdump | tunnel]**

This command displays MPLS LDP-related information.

- **all** - Displays all messages related to a MPLS LDP module.
- **adjacency** - Displays debug messages related to MPLS LDP adjacency messages.
- **error** - Displays debug messages related to MPLS LDP errors.
- **event** - Displays debug messages related to MPLS LDP events.
- **fec** - Displays MPLS LDP FEC information.
- **packets** - Displays debug messages related to MPLS LDP packets.
- **state** - Displays debug messages related to MPLS LDP states.
- **socket** - Displays debug messages related to MPLS LDP sockets.
- **tcpdump** - Displays MPLS LDP packets as raw data.
- **tunnel** - Displays debug messages related to MPLS LDP LSP interaction with other modules as virtual Interfaces.

### debug mpls ldp packets

**Syntax: [no] debug mpls ldp packets [all | detail | direction | lsr_id | pkt_type]**

This command displays MPLS LDP packets which is further filtered by direction, packet types, and LSR ID.

- **all** - Displays all MPLS LDP packets.
- **detail** - Displays messages in detailed version.
- **direction** - Limits the display of MPLS LDP packets to specific directions (send/receive).
- **pkt_type** - Limits the display of MPLS LDP packets to specific packet types.
- **lsr_id** - Limits the display of MPLS LDP packets to a specific LSR ID.

Sample output:

```
PowerConnect#debug mpls ldp packets
Msg: Keepalive(0x0201) len 4, id 0x00000131
LDP_PKT: send link Hello to e1/1
LDP_PKT: receive link Hello from 1.1.1.2 on e1/1
LDP_PKT: send link Hello to e1/1
LDP_PKT: Rcvd PDU <- 20.20.20.20, ver 1, pdu len 14, id 100.100.100.100:0 (tcb
0dfa0328)
Msg: Keepalive(0x0201) len 4, id 0x00000132
LDP_PKT: receive link Hello from 1.1.1.2 on e1/1
LDP_PKT: send link Hello to e1/1
LDP_PKT: receive link Hello from 1.1.1.2 on e1/1
LDP_PKT: Rcvd PDU <- 20.20.20.20, ver 1, pdu len 14, id 100.100.100.100:0 (tcb
0dfa0328)
Msg: Keepalive(0x0201) len 4, id 0x00000133
LDP_PKT: receive UDP packet for invalid destination address 1.1.1.1
```

### debug mpls ldp packets pkt_type

**Syntax: [no] debug mpls ldp packets pkt_type [all | address | initialization | label | notification | hello | keepalive]**

This command displays MPLS LDP packets with specific LDP packet types.

- **all** - Displays MPLS LDP packets of all the types.
- **address** - Displays information about MPLS LDP addresses, including address withdraw messages.
- **initialization** - Displays LDP Initialization messages.
- **label** - Displays LDP Label Mapping, Withdraw, Request, Release, and Abort messages.
- **notification** - Displays LDP notification information.
- **hello** - Displays information about the periodic link Hello messages sent and received.
- **keepalive** - Displays LDP KeepAlive messages after the session comes up.

### debug mpls ldp packets pkt_type address

**Syntax: [no] debug mpls ldp packets pkt_type address**

This command displays information about MPLS LDP addresses, including address withdraw messages.

```
PowerConnect#debug mpls ldp pkt_type address
LDP um 0d9098c8 fec 11.11.11.0 ses 22.22.22.1: inp 0, st 0 -> 1, act A, gen 0
LDP um 0d9098c8 fec 11.11.11.0 ses 22.22.22.1: inp 3, st 1 -> 2, act W, gen 0
LDP um 0d9098c8 fec 11.11.11.0 ses 22.22.22.1: inp 1, st 2 -> 2, act B, gen 0
LDP um 0d9098c8 fec 11.11.11.0 ses 22.22.22.1: inp 13, st 2 -> 5, act G, gen 0
LDP um 0d9098c8 fec 11.11.11.0 ses 22.22.22.1: inp 1, st 5 -> 6, act I, gen 0
LDP: Send PDU -> 22.22.22.1, ver 1, pdu len 32, id 11.11.11.1:0 (tcb 0000002f)
Msg: Address(0x0300) len 22, id 0x80000000
Tlv: Addr_lst(0x0101) len 14, fam 256
Addrs: 10.1.1.1 10.5.1.1 11.11.11.1
```

### debug mpls ldp packets pkt_type initialization

**Syntax: [no] debug mpls ldp packets pkt_type initialization**

This command currently map to the same set of traces, as shown here:

```
PowerConnect#debug mpls ldp packets pkt_type initialization
Msg: Initialize(0x0200) len 22, id 0x00000001
LDP: Send PDU -> 10.5.1.1, ver 1, pdu len 32, id 10.4.1.1:0 (tcb f0020000)
Msg: Keepalive(0x0201) len 4, id 0x00000002
LDP: Send PDU -> 10.5.1.1, ver 1, pdu len 32, id 10.4.1.1.0 (tcbf00s0000)
Msg: Address(0x0300)len 22, id 0x80000000
LDP: Send PDU -> 10.5.1.1, ver 1, pdu len 33, id 10.4.1.1:0 (tcbf0020000)
Msg: LabelMap(0x0400) len 23, id 0x80000001
LDP: Send PDU -> 10.5.1.1, ver 1, pdu len 33, id 10.4.1.1:0 (tcbf0020000)
Msg: LabelMap(0x0400) len 23, id 0x80000004
```

### debug mpls ldp packets pkt_type notification

**Syntax: [no] debug mpls ldp packets pkt_type notification**

This command displays LDP notification information, which is generated when an unexpected event occurs. In the following example, and MPLS interface has gone down and come back up.

```
PowerConnect#debug mpls ldp packets pkt_type notification
Msg: Notification(0x0001) len 18, id 0x00000056
LDP: Send PDU -> 10.5.1.1, ver 1, pdu len 28, id 10.4.1.1:0 (tcb f0030000)
Msg: Notification(0x0001) len 18, id 0x00000001
```

### debug mpls ldp packets pkt_type hello

**Syntax: [no] debug mpls ldp packets pkt_type hello**

This command displays information about the periodic link Hello messages sent and received.

```
PowerConnect#debug mpls ldp packets pkt_type hello
LDP: Send link Hello to e1/1
LDP: Send link Hello to e1/2
LDP: Rcvd link Hello from 10.1.1.2 on e1/1
LDP: Rcvd link Hello from 10.5.1.2 on e1/2
```

### debug mpls ldp packets direction

**Syntax: [no] debug mpls ldp packets direction [send | receive]**

This command limits the display of MPLS LDP packets to specified direction.

### debug mpls ldp packets lsr_id

**Syntax: [no] debug mpls ldp packets lsr-id** <*ip_address*> <*label_space*>

This command displays MPLS LDP packets for the specified LDP LSR ID.

### debug mpls ldp adjacency

**Syntax: [no] debug mpls ldp adjacency**

This command displays information about MPLS LDP adjacencies.This example shows that adjacent interfaces e1/1 and e1/2 are in the up state.

```
PowerConnect#debug mpls ldp adjacency
LDP_ADJ: Link adjacency to 140.140.140.4:0 on interface e1/2 is deleted, reason 4
LDP_ADJ: Link adjacency to 140.140.140.4:0 on interface e1/2 is added
```

### debug mpls ldp error

**Syntax: [no] debug mpls ldp error**

This command displays the errors detected by LDP components such as LDP session keepalive timer expiration, hello adjacency timeout, etc.

Sample message:

```
PowerConnect#debug mpls ldp error
LDP_ERR: Session KeepAlive timer to peer 120.120.120.2 has expired
```

### debug mpls ldp event

**Syntax: [no] debug mpls ldp event**

This command displays session up and down events, similar to the following:

```
PowerConnect#debug mpls ldp event
LDP_EVT: LDP session to 140.140.140.4, entity idx 1, type non-targeted goes DOWN,
use TCB 0XA0000010:0X12BF2B70
LDP_EVT: remove session to 140.140.140.4. Session is deleted
LDP_EVT: initiate LDP session to peer 140.140.140.4. Session is not found
LDP_EVT: initiate session block 0X344BF140, entity idx 1 and insert to tree
LDP_EVT: LDP session to 140.140.140.4 is initiated, targeted adjacency No
LDP_EVT: LDP session to 140.140.140.4, entity idx 1, type non-targeted goes UP,
use TCB 0XA0000012:0X12BF244A
```

### debug mpls ldp socket

**Syntax: [no] debug mpls ldp socket**

This command displays information about LDP sockets. Output resembles the following:

```
PowerConnect#debug mpls ldp socket
LDP_SCK: close UDP link tx socket on interface e1/2
LDP_SCK: close UDP link rx socket on e1/2, ucb 0XFF010000#1, appl_sock 0X34B8A280
LDP_SCK: start closing TCP session<130.130.130.3,646-140.140.140.4,9006>, TCB
0X12BF244A
LDP_SCK: start closing TCP listen socket TCB 0X230015D5
LDP_SCK: TCP start listen for <130.130.130.3, 646>, TCB 0X12BF2726
LDP_SCK: open link hello tx socket on interface e1/2
LDP_SCK: complete link hello tx socket open ucb 0XFF010000, count1, appl_sock
0X34B8A380
LDP_SCK: open link hello rx socket on interface e1/2
LDP_SCK: complete link hello tx socket open ucb 0XFF010000, count1, appl_sock
0X34B8A380
LDP_SCK: open link hello rx socket on interface e1/2
LDP_SCK: receive incoming connection from <140.140.140.4, 9007>, listen TCB
0X12BF2726, TCB 0X12BF2B70
LDP_SCK: accept incoming connection from <140.140.140.4, 9007>, TCP handle
0X12BF2B70, pending_data No
LDP_SCK: connection established with <140.140.140.4, 9007>, TCB 0X12BF2B70
```

### debug mpls ldp state

**Syntax: [no] debug mpls ldp state**

This command displays the MPLS LDP states. The following states are possible:

- 0 - Unknown
- 1 - No route
- 2 - Route found
- 3 - Path sent
- 4 - Path error

Output resembles the following:

```
PowerConnect#debug mpls ldp state
LDP SC 1 Initialization FSM, Session CB 207/33, input 3, old state 0, new state 2,
action 3.
LDP SC 1 Initialization FSM, Session CB 207/33, input 4, old state 2, new state 3,
action 4.
LDP SC 1 Initialization FSM, Session CB 207/33, input 5, old state 3, new state 4,
action 5.
LDP SC 1 Initialization FSM, Session CB 207/33, input 8, old state 4, new state 8,
action 7.
LDP SC 1 Initialization FSM, Session CB 207/33, input 11, old state 8, new state
9, action 11.
LDP SC 1 Initialization FSM, Session CB 207/33, input 16, old state 9, new state
10, action 24.
```

### debug mpls ldp tcpdump

**Syntax: [no] debug mpls ldp tcpdump**

This command displays information about LDP dumps. Output resembles the following:

```
PowerConnect#debug mpls ldp tcpdump
LDP_TCPDUMP: tx to 140.140.140.4, TCP length 18 0001000e 82828203 00000201
00040000 000a
LDP_TCPDUMP: rx from 140.140.140.4, TCP length 18 0001000e 8c8c8c04 00000201
00040000 000b
```

### debug mpls ldp tunnel

**Syntax: [no] debug mpls ldp tunnel [all | prefix** <*ip-address*> <*prefix-length*>**]**

This command displays MPLS LDP tunnel-related information.

- **all** - Displays all MPLS LDP tunnel up and down events.
- **prefix** <*ip-address*> <*prefix-length*> - Limits the display of information to specific prefixes.

Sample output:

```
PowerConnect#debug mpls ldp tunnel
LDP_TUNNEL: tunnel(5) to 100.100.100.100 is up
```

### debug mpls ldp fec

**Syntax: [no] debug mpls ldp fec [all | lsr-id** <*ip_address*> <*label_space*> | **key [prefix** <*ip-address*> <*prefix-length*> **| vc** <*vc_id*>**]]**

This command displays MPLS LDP FEC information.

- **all** - Displays all MPLS LDP FEC-related information.
- **lsr_id** <*ip_address*> <*label_space*> - Limits the MPLS LDP FEC information displayed to a specific LSR ID.
- **key** - Limits the information displayed to a specific FEC key value.
- **prefix** *ip-address*> <*prefix-length*> -Limits the display of MPLS LDP FEC-related information to specific prefixes.
- **vc** <*vc_id*> - Displays MPLS LDP FEC-related information to a specific VC ID.

The output resembles the following:

```
PowerConnect#debug mpls ldp fec
LDP PM 1 Ingress FSM, CB 171/33, input 0, old state 0, new state 0, action 1, FEC
CB 169/33,
FEC Prefix: 100.100.100.100, len 32.
LDP PM 1 Egress FSM, CB 205/45, input 0, old state 0, new state
0, action 1, FEC CB 184/45,
FEC Prefix: 20.20.20.20, len 32.
LDP PM 1 UM FSM, CB 164/29, input 0, old state 0, new state 1, a
ction 1, UT CB 101/5, session CB 117/25, FEC CB 184/45.
FEC Prefix: 20.20.20.20, len 32.
LDP PM 1 UM FSM, CB 164/29, input 3, old state 1, new state 2, a
ction 23, UT CB 101/5, session CB 117/25, FEC CB 184/45.
FEC Prefix: 20.20.20.20, len 32.
LDP PM 1 UM FSM, CB 164/29, input 1, old state 2, new state 2, a
ction 2, UT CB 101/5, session CB 117/25, FEC CB 184/45.
FEC Prefix: 20.20.20.20, len 32.
LDP PM 1 UM FSM, CB 164/29, input 14, old state 2, new state 5,
action 7, UT CB 101/5, session CB 117/25, FEC CB 184/45.
FEC Prefix: 20.20.20.20, len 32.
```

# MPLS VPLS

VPLS is a method for carrying Layer 2 frames between Customer Edge (CE) devices across an MPLS domain. The Dell implementation supports VPLS as described in the IETF Internet Draft, “draft-ietf-l2vpn-vpls-ldp-05.txt”.

VPLS CPU protection shields the management module CPU from being overwhelmed by VPLS traffic that requires CPU processing, such as source-MAC learning, or forwarding of unknown unicast or broadcast traffic.

## MPLS VPLS show commands

The following sections show how to display various types of information about VPLS activity and configurations, under these headings:

### show mpls vpls

**Syntax: show mpls vpls**

To display information about VPLS instances configured on the device, enter the following command.

```
PowerConnect#show mpls vpls
                      Num      Num      Ports      Num      Peers      Num
Name         Id       Vlans    Ports    Up         Peers    Up         VC-label
test         100      2        3        0          2        0          32
```

### show mpls vpls summary

**Syntax: show mpls vpls summary**

You can display a summary of VPLS statistics, including the number of VPLS instances, number of VPLS peers, label range size, and maximum size of the VPLS MAC database, by entering the following command.

```
PowerConnect#show mpls vpls summary
Total VPLS configured: 3, maximum number of VPLS allowed: 4096
Total VPLS peers configured: 1, total peers operational: 1
Maximum VPLS macentries allowed: 8192, currently installed: 3
VPLS global raw mode VC-Type is Ethernet (0x5)
VPLS global MTU is 1500, MTU enforcement is OFF
Global CPU protection: OFF
MVIDs in use: 1 of 1 total allocated
```

### show mpls vpls detail

**Syntax: show mpls vpls detail**

To display more detailed information about each VPLS instance, enter the following command.

```
PowerConnect#show mpls vpls detail
VPLS test1, Id 1, Max mac entries: 2048
Total vlans: 1, Tagged ports: 1 (1 Up), Untagged ports 0 (0 Up)
Vlan 2
Tagged: ethe 5/3
Total VPLS peers: 1 (1 Operational)
Peer address: 3.3.3.3, State: Operational, Uptime: 28 min
Tnnl: tnl0(1025), LDP session: Up, Local VC lbl: N/A, Remote VC lbl: N/A0
VPLS test2, Id 2, Max mac entries: 2048
```

```
Total vlans: 1, Tagged ports: 2 (1 Up), Untagged ports 0 (0 Up)
Vlan 3
Tagged: ethe 1/4 ethe 5/3
Total VPLS peers: 1 (1 Operational)
Peer address: 3.3.3.3, State: Operational, Uptime: 23 min
Tnnl: tnl0(1025), LDP session: Up, Local VC lbl: N/A, Remote VC lbl: N/A
```

### show mpls vpls id

**Syntax: show mpls vpls id** <*vpls id*>

This command shows the tunnel LSPs being used to forward VPLS traffic from the device to the peer. If VPLS traffic to a peer is being load-balanced across multiple tunnel LSPs, then the command lists the tunnel LSPs used for load balancing, as shown in the following example:

```
PowerConnect(config)#show mpls vpls id 1
VPLS test1, Id 1, Max mac entries: 2048
Total vlans: 1, Tagged ports: 1 (0 Up), Untagged ports 1 (1 Up)
Vlan 2
Tagged: ethe 5/4
Untagged: ethe 2/2
Total VC labels allocated: 32 (983040-983071)
Total VPLS peers: 1 (0 Operational)
Peer address: 1.1.1.1, State: Wait for remote VC label from Peer
Tnnl: tnl0(3), LDP session: Up, Local VC lbl: N/A, Remote VC lbl: N/A
CPU-Protection: ON, MVID: 0x000, VPLS FID: 0x00000205
```

### show mpls vpls down

**Syntax: show mpls vpls down**

This command displays information about VPLS instances that are not fully operational, as shown here:

```
PowerConnect#show mpls vpls down
The following VPLS'es are not completely operational:
                     Num   Num    Ports  Num    Peers   CPU
Name          Id     Vlans Ports  Up     Peers  Up      Prot
vpls1         1003   1     1      1      1      0       ON
vpls2         1004   0     0      0      1      0       ON
```

### show mac vpls

**Syntax: show mac vpls** <*vpls id*>

The VPLS MAC database stores entries associating remote MAC addresses with VC LSPs, and local MAC addresses with CE devices. Each VPLS instance has a separate VPLS MAC database.Output resembles the following:

```
PowerConnect#show mac vpls
Total VPLS mac entries in the table: 10 (Local: 5, Remote: 5)
VPLS  MAC Address    L/R  Port Vlan/Peer       Age
====  ===========    ===  ==== =========       ===
1     0016.0100.1601 R    5/1  3.3.3.3         0
1     0010.0100.1003 L    5/3  2               0
1     0016.0100.1603 R    5/1  3.3.3.3         0
1     0010.0100.1005 L    5/3  2               0
1     0010.0100.1002 L    5/3  2               0
1     0016.0100.1605 R    5/1  3.3.3.3         0
1     0016.0100.1602 R    5/1  3.3.3.3         0
1     0010.0100.1004 L    5/3  2               0
1     0010.0100.1001 L    5/3  2               0
1     0016.0100.1604 R    5/1  3.3.3.3         0
```

To display the VPLS MAC database on the management processor for a VPLS instance specified by its VPLS ID, enter the following command.

```
PowerConnect#show mac vpls 1
Total MAC entries for VPLS 1: 10 (Local: 5, Remote: 5)
VPLS  MAC Address    L/R Port   Vlan/Peer      Age
====  ===========    === ====   =========      ===
1     0016.0100.1601 R   5/1    3.3.3.3        0
1     0010.0100.1003 L   5/3    2              0
1     0016.0100.1603 R   5/1    3.3.3.3        0
1     0010.0100.1005 L   5/3    2              0
1     0010.0100.1002 L   5/3    2              0
1     0016.0100.1605 R   5/1    3.3.3.3        0
1     0016.0100.1602 R   5/1    3.3.3.3        0
1     0010.0100.1004 L   5/3    2              0
1     0010.0100.1001 L   5/3    2              0
1     0016.0100.1604 R   5/1    3.3.3.3        0
```

To display a specific entry in the MAC database on the management processor, enter the following command.

```
PowerConnect#show mac vpls 1 0016.0100.1601
VPLS: 1            MAC: 0016.0100.1601        Age: 0
Remote MAC         Port: ethe 5/1             Peer: 3.3.3.3
Trunk slot mask: 00000000
```

### show mpls statistics vpls

**Syntax: show mpls statistics vpls** [<*vpls-name*> **|** <*vpls-id*>]

- <*vpls-name*> - Indicates the configured name for a VPLS instance.
- <*vpls-id*> - Indicates the ID of a VPLS instance.

This command displays traffic statistics for a all VPLS instances, or a specific instance. The following example shows statistics for all VPLS traffic:

```
PowerConnect#show mpls statistics vpls
VPLS-Name       In-Port(s)    Endpt-Out-Pkts      Tnl-Out-Pkts
---------       ----------    --------------      ------------
test2           e1/1          0                   0
                e1/2          0                   0
                e1/3          0                   0
                e1/4          0                   0
test2           e2/1 - e2/10  0                   0
                e2/11 - e2/20 0                   0
                e2/21 - e2/30 0                   0
                e2/31 - e2/40 0                   0
test3           e1/1          0                   0
                e1/2          0                   0
                e1/3          0                   0
                e1/4          0                   0
test3           e2/1 - e2/10  0                   0
                e2/11 - e2/20 0                   0
                e2/21 - e2/30 0                   0
                e2/31 - e2/40 0                   0
test4           e1/1          0                   0
                e1/2          0                   0
                e1/3          0                   0
                e1/4          0                   0
test4           e2/1 - e2/10  0                   0
                e2/11 - e2/20 0                   0
                e2/21 - e2/30 0                   0
                e2/31 - e2/40 0                   0
test4           e5/1          10354120822         0
                e5/2          0                   0
                e5/3          0                   2992416134
                e5/4          0                   0
```

**NOTE**
The VPLS name is repeated for each module from which the statistics are collected, to be displayed on the Management console.

### show mpls debug vpls

**Syntax: show mpls debug vpls** <*vpls id*>

This command displays generic VPLS debug information. Enter a VPLS ID to display information about a specific VPLS instance, as shown in the following example:

```
PowerConnect#show mpls debug vpls 1
ID:       1      Name:     test 1
CPU-Prot: OFF    MVID:     INVD     FID:    0x00002002
MAC Info:
     Total MACs: 2  Local: 2  Remote: 0
     Max Exceed: 0 Table Full: 0
```

### show mpls debug vpls local

**Syntax: show mpls debug vpls local** <*num*>

This command displays the state of a VPLS end-point. To dump local entries for a specific VPLS instance, enter the ID number of the instance. Output resembles the following (specified for VPLS ID 2):

```
PowerConnect#show mpls debug vpls local 2
VPLS 2:
    VLAN  Port   Valid  Pending
    ====  ====   =====  =======
    4     2/19   1      0
    Local Broadcast Fids:
    =====================
    VLAN 4        --  Fid: 00008fa6, Ports: 1
      Port 2/19   --  Fid: 0000003a

Local Broadcast Fids – Internal information
```

For VPLS FID Sharing, the output is shown as follows:

```
PowerConnect#show mpls debug vpls local
VPLS 10:
   VLAN   In-Tag  Port   Valid  Blocked Pending
   ====   ======  ====   =====  ======= =======
   10     n/a     3/8    1      No        0
   10     n/a     3/1    1      No        0
   Local Broadcast Fids:
   =====================
   Vlan 10   --Fid: 00008023, Ports: 2
    Port 3/1 --Fid: 00000018
    Port 3/8 --Fid: 0000001f
VPLS 20:
   VLAN   In-Tag  Port   Valid  Blocked Pending
   ====   ======  ====   =====  ======= =======
   20     n/a     3/8    1      No        0
   20     n/a     3/1    1      No        0
   Local Broadcast Fids:
   =====================
   Vlan 20   --Fid: 00008023, Ports: 2
    Port 3/1 --Fid: 00000018
    Port 3/8 --Fid: 0000001f
```

### show mpls debug vpls remote

**Syntax: show mpls debug vpls remote <*vpls id* >**

This command displays the state of all VPLS peers configured in the system. Specify a VPLS-ID to restrict the peer listing to a specific VPLS instance. Output resembles the following (specified for remote VPLS ID 1):

```
PowerConnect#show mpls debug vpls remote 1
VPLS 1:
    Peer:  5.5.5.5             Valid:  Yes     Pending Delete: 0
ya, the output showsLabel: 983040             Tagged: No      Load
Balance:   No
                                                Num LSP Tnnls:  1
          VC        Tunnel    NHT             Use
    Port  Label     Label     Index     COS   COS
    ====  =====     ======    =====     ===   ===
    2/9   983040    3         0         0     0
    2/9   983040    3         0         0     0
    2/9   983040    3         0         0     0
    2/9   983040    3         0         0     0
    Active Trunk Index: 0
```

Internally, a maximum of four LSP tunnels are maintained to reach the peer. If Load Balancing is disabled, information for only one tunnel is displayed in the output.

**show mpls debug vpls fsm-trace**

**Syntax: show mpls debug vpls fsm-trace** *<vpls id>*

This command displays the VPLS peer FSM history trace that includes FSM state, the events that it received and also the time stamp as to when the transition happened.

```
PowerConnect#show mpls debug vpls fsm-trace 1
Time              FSM State      Rcvd Event       Action
Oct 16 02:07:34   WAIT_PT        PORT_UP           A
Oct 16 02:17:34   WAIT_TNNL      TNNL_UP           B
Oct 16 02:17:34   WAIT_PW_UP     PW_UP             E
Oct 16 03:04:24   OPER           PW_DN             I
Oct 16 03:05:34   WAIT_PW_UP     PW_UP             E
```

## *Clearing VPLS traffic statistics*

To clear the entries stored for all VPLS statistics, enter the following command.

**clear mpls statistics vpls**

**Syntax: clear mpls statistics vpls [<***vpls-name***> | <***vpls-id***>]**

To clear entries for a specific VPLS instance, enter the VPLS name or ID number.

- *<vpls-name>* - The configured name for a VPLS instance.
- *<vpls-id>* - The ID of a VPLS instance.

# MPLS VPLS debug commands

**debug vpls**

**Syntax: [no] debug vpls [cam |dy-sync | events | filter |forwarding| generic | mac | statistics | topology]**

- **cam** - Displays information about VPLS CAM or PRAM programming.
- **count** - Displays information about VPLS debug print counter.
- **dy-sync** - Displays information about VPLS table synchronization between management and CPU cards.
- **events** - Displays information about VPLS control-plane events.
- **filter** - Displays VPLS filtering options.
- **forwarding** - Displays information about VPLS CPU packet forwarding.
- **generic** - Enables generic VPLS debugging.
- **mac** - Displays VPLS MAC learning, aging, deletion, and movement.
- **statistics** - Displays information about VPLS statistics.
- **topology** - Displays information about VPLS topology group events.

### debug vpls cam

**Syntax: [no] debug vpls cam [additions |deletions |updates]**

**NOTE**
This command is available only on LP.

- **additions** - Displays information about VPLS CAM or PRAM additions.
- **deletions** - Displays information about VPLS CAM or PRAM deletions.
- **updates** - Displays information about VPLS CAM or PRAM updates.

This command displays information about VPLS CAM or PRAM programming.

```
PowerConnect#debug vpls cam
        VPLS CAM:  all debugging is on
VPLS CAM-DEL: lp_cam_del_vpls_mac_cam_all() - Delete all CAMs for MAC
0000.0404.0000.
VPLS CAM-DEL: lp_cam_del_vpls_mac_cam_single() - MAC 0000.0404.0000
(VPLS_SA_VC_ENTRY): deleted single CAM 0001800a and PRAM.
deleting cam-index 98314 for PPCR 1:1, CAM_TYPE:13
        cam-index chain:
VPLS CAM-ADD: lp_cam_add_vpls_mac_egress_one_ppcr() - Add CAM entry for MAC
0000.0404.0000, port 1/1, vc-label 000f0000, type VPLS_SA_VC_ENTRY.
VPLS CAM-ADD: lp_cam_add_vpls_mac_egress_one_ppcr() - SA-VC CAM add success. CAM
0001800a, new PRAM 000000c7.
adding cam-index 98314 for PPCR 1:1 CAM_TYPE:13
```

### debug vpls cam additions

**Syntax: [no] debug vpls cam additions**

This command generates debug output when a VPLS CAM or PRAM addition operation is performed by VPLS due to the addition of MAC CAM or PRAM entry to the hardware.

### debug vpls cam deletions

**Syntax: [no] debug vpls cam deletions**

This command generates debug output when a VPLS CAM or PRAM deletion operation is performed by VPLS due to the deletion of MAC CAM or PRAM entry from the hardware.

### debug vpls cam updates

**Syntax: [no] debug vpls cam updates**

This command generates debug output when a VPLS CAM or PRAM entry is being updated due to a change in the PRAM info in regard to forwarding decision.

### debug vpls count

**Syntax: [no] debug vpls count**

This command specifies the number of debug prints generated by debug vpls commands. Use this command to limit high-volume displays such as MAC learning activity and certain dy-sync activity.

### debug vpls forwarding

**Syntax: [no] debug vpls forwarding**

This command is used to generate various CPU packet forwarding in regard to VPLS traffic. On LP it generates how the packet was received and how it was software forwarded. On the MP it generates the packet handling of those applications such as Multicast or DOT1AG to send out control packets through VPLS to a remote peer.

```
PowerConnect#debug vpls forwarding
        VPLS Forwarding:  debugging is on
VPLS EGRESS FWD: lp_l2_vpls_outbound_process_packet() - RX pkt from MPLS uplink
src-port:1/1. VC label:983040
VPLS EGRESS FWD: lp_l2_vpls_outbound_process_packet() - DA missed in CAM.
VPLS EGRESS FWD: lp_l2_vpls_outbound_forward_packet() - Payload Tag 0x81000008
exisits.
VPLS EGRESS FWD: lp_l2_vpls_outbound_forward_packet() - Unknown VPLS MAC
0000.0203.0000, VPLS 1.
VPLS EGRESS FWD: lp_l2_vpls_outbound_forward_packet() - UNKNOWN: send pkt to all
end-points.
VPLS EGRESS FWD: lp_l2_vpls_broadcast_local() - Bcast pkt to VPLS VLAN 300,
inner_vid_valid:0 inner_vlan_id 0x00000000, FID 0x800b, Egress:1,
priority:0x00000000, inner_vlan_priority:0x00000000
VPLS EGRESS FWD: lp_l2_vpls_broadcast_local() - Bcast pkt to VPLS VLAN 3000,
inner_vid_valid:0 inner_vlan_id 0x00000000, FID 0x800c, Egress:1,
priority:0x00000000, inner_vlan_priority:0x00000000
```

### debug vpls dy-sync

**Syntax: [no] debug vpls dy-sync [local | mac | remote | tlv]**

- **local** - Displays information about VPLS local entry dy-sync.
- **mac** - Displays information about VPLS MAC table dy-sync.
- **remote** - Displays information about VPLS remote entry dy-sync.
- **tlv** - Displays information about VPLS TLV dy-sync.

This command monitors all VPLS dy-sync activity. Dy-sync is a method of synchronizing internal VPLS data between management and CPU cards. This includes VPLS configuration (TLV), local-entry or end-point status (local), remote-entry or VPLS peer status (remote), and VPLS MAC table activity. This data is generally used for internal debugging.

### debug vpls dy-sync local

**Syntax: [no] debug vpls dy-sync local**

This command generates information about local VPLS dy-sync activity. Output resembles the following for a specified VLAN (VLAN ID 2):

```
PowerConnect#debug vpls dy-sync local
PowerConnect(config-mpls-vpls-test1)#vlan 2
PowerConnect(config-mpls-vpls-test1-vlan-2)#no tag e 2/19
VPLS MAC-GROUP: vpls_mac_delete_local_entry() - VPLS 1, port 2/19, vlan 2: HW CAMs
flushed 620.
VPLS DY-SYNC-LOC: mpls_vpls_pack_one_vpls_local_entry() - VPLS 1, action 2, vlan
2, port 2/19,
replace-vlan 0.
VPLS DY-SYNC-TLV: mpls_vpls_pack_one_ipc_tlv_vpls_vlan_port() - VPLS 1, vlan 2,
port 2/19.
VPLS DY-SYNC-TLV: mpls_vpls_pack_one_ipc_tlv_port_config() - VPLS 1, port 2/19,
vlan 2, mode 1.
VPLS DY-SYNC-TLV: mpls_vpls_sync_timer() - Flush TLV packet
VPLS DY-SYNC-LOC: mpls_vpls_sync_timer() - Flush LOCAL packet
R4(config-mpls-vpls-test1-vlan-2)#
R4(config-mpls-vpls-test1-vlan-2)#tag e 2/19
VPLS DY-SYNC-TLV: mpls_vpls_pack_one_ipc_tlv_port_config() - VPLS 1, port 2/19,
vlan 2, mode 1.
VPLS DY-SYNC-LOC: vpls_mac_add_local_entry() - Add: VPLS 1, vlan 2, port 2/19.
```

```
VPLS DY-SYNC-LOC: mpls_vpls_pack_one_vpls_local_entry() - VPLS 1, action 1, vlan
2, port 2/19,
replace-vlan 1.
VPLS DY-SYNC-TLV: mpls_vpls_pack_one_ipc_tlv_vpls_vlan_port() - VPLS 1, vlan 2,
port 2/19.
VPLS DY-SYNC-TLV: mpls_vpls_sync_timer() - Flush TLV packet
VPLS DY-SYNC-LOC: mpls_vpls_sync_timer() - Flush LOCAL packet
```

### debug vpls dy-sync mac

**Syntax: [no] debug vpls dy-sync mac**

The following output was generated with **debug vpls dy-sync mac** enabled and indicates that an existing VPLS MAC has been deleted and then reinstalled.

```
PowerConnect#debug vpls dy-sync mac
PowerConnect#clear mac vpls e 2/19
VPLS MAC: mpls_vpls_delete_mac_entry_from_table() - VPLS 1, MAC 0006.0100.0601.
1 mac entries flushed
VPLS MAC-LOCAL: mpls_vpls_mac_sync_itc_callback() - VPLS_MAC_SYNC_LOCAL_ENTRY:
MAC
0006.0100.0601, port 2/19, vlan 2
VPLS MAC-LOCAL: vpls_local_sa_learning() - MAC 0006.0100.0601, port 2/19, vlan 2.
VPLS MAC: mpls_vpls_insert_mac_entry_in_table() - VPLS 1, MAC 0006.0100.0601.
VPLS DY-SYNC-MAC: mpls_vpls_mp_send_create_delete_one_mac_entry() - VPLS 1,
action 1, mac 0006.0100.0601.
VPLS DY-SYNC-MAC: mpls_vpls_pack_mac_table_change() - VPLS 1, action 1, mac
0006.0100.0601, local 1.
VPLS MAC-LOCAL: mpls_vpls_mac_sync_itc_callback() - VPLS_MAC_SYNC_LOCAL_ENTRY:
MAC
0006.0100.0601, port 2/19, vlan 2
VPLS MAC-LOCAL: vpls_local_sa_learning() - MAC 0006.0100.0601, port 2/19, vlan 2.
VPLS MAC: mpls_vpls_insert_mac_entry_in_table() - VPLS 1, MAC 0006.0100.0601.
VPLS MAC-LOCAL: vpls_local_sa_learning() - Existing entry.
VPLS DY-SYNC-MAC: mpls_vpls_mp_send_create_delete_one_mac_entry() - VPLS 1,
action 1, mac 0006.0100.0601.
VPLS DY-SYNC-MAC: mpls_vpls_pack_mac_table_change() - VPLS 1, action 1, mac
0006.0100.0601, local 1.
VPLS DY-SYNC-MAC: mpls_vpls_mp_send_create_delete_one_mac_entry() - VPLS 1,
action 3, mac 0006.0100.0601.
VPLS DY-SYNC-MAC: mpls_vpls_pack_mac_table_change() - VPLS 1, action 3, mac
0006.0100.0601, local 1.
VPLS DY-SYNC-MAC: mpls_vpls_mp_send_create_delete_one_mac_entry() - VPLS 1,
action 3, mac 0006.0100.0601.
VPLS DY-SYNC-MAC: mpls_vpls_pack_mac_table_change() - VPLS 1, action 3, mac
0006.0100.0601, local 1.
VPLS DY-SYNC-MAC: mpls_vpls_mp_send_create_delete_one_mac_entry() - VPLS 1,
action 3, mac 0006.0100.0601.
VPLS DY-SYNC-MAC: mpls_vpls_pack_mac_table_change() - VPLS 1, action 3, mac
```

### debug vpls dy-sync remote

**Syntax: [no] debug vpls dy-sync remote**

This command generates information about VPLS remote entry dy-sync activity. The following output was generated with this command enabled, and indicates that an existing peer was deleted and then reinstalled.

```
PowerConnect#debug vpls dy-sync remote
PowerConnect(config-mpls-vpls-test1-vlan2)#no vpls-peer 5.5.5.5
VPLS MAC-GROUP: vpls_mac_delete_remote_entry () - VPLS 1, VC-label 000f0000: HW
CAMs flushed 4.
VPLS DY-SYNC-REM: mpls_vpls_send_remote_entry_info () - called
VPLS DY-SYNC-REM: mpls_vpls_sync_timer () - Flush REMOTE packet
(config-mpls-vpls-test1-vlan-2)# vpls-peer 5.5.5.5
VPLS DY-SYNC-REM: mpls_vpls_pack_one_vpls_remote_entry_tnns () - called
VPLS DY-SYNC-REM: mpls_vpls_pack_one_vpls_remote_entry_tnnl () - called
VPLS DY-SYNC-REM: mpls_vpls_pack_one_vpls_remote_entry () - called
VPLS DY-SYNC-REM: mpls_vpls_send_remote_entry_info () - called
VPLS DY-SYNC-REM: mpls_vpls_sync_timer () - Flush REMOTE packet
```

### debug vpls dy-sync tlv

**Syntax: [no] debug vpls dy-sync tlv**

This command generates information about the VPLS configuration that is being synchronized (dy-sync) between management and CPU cards. The following output results when a VPLS instance is deleted and then reinstalled:

```
PowerConnect#debug vpls dy-sync tlv
PowerConnect(config-mpls)#no vpls 1 1
VPLS DY-SYNC-TLV: mpls_vpls_pack_one_ipc_tlv_vpls_vlan_port() - VPLS 1, vlan 2,
port 1/2.
VPLS DY-SYNC-TLV: mpls_vpls_pack_one_ipc_tlv_port_config() - VPLS 1, port 1/2,
vlan 2, mode 0.
VPLS DY-SYNC-TLV: mpls_vpls_pack_one_ipc_tlv_vpls_vlan() - VPLS 1, vlan 2,
vlan-fid 65535.
VPLS DY-SYNC-TLV: mpls_vpls_send_ipc_delete_vpls_table() - VPLS 1.

PowerConnect(config-mpls)# vpls 1 1
PowerConnect(config-mpls-vpls-1)# vpls-peer 200.200.200.1
PowerConnect(config-mpls-vpls-1)# vlan 2
PowerConnect(config-mpls-vpls-1-vlan-2)# untagged ethe 1/2
VPLS DY-SYNC-TLV: mpls_vpls_pack_one_ipc_tlv_vpls_name() - VPLS 1, action 1, name
1.
VPLS 1, action 1, value 0.
VPLS DY-SYNC-TLV: mpls_vpls_pack_one_ipc_tlv_vpls_fid() - VPLS 1, action 1, Fid
0x0000a002.
VPLS DY-SYNC-TLV: mpls_vpls_pack_one_ipc_tlv_vpls_mvid() - VPLS 1, action 1,
value 2048.
VPLS DY-SYNC-TLV: mpls_vpls_pack_one_ipc_tlv_vlan_id() - VPLS 1, vlan 2.
VPLS DY-SYNC-TLV: mpls_vpls_pack_one_ipc_tlv_vpls_vlan() - VPLS 1, vlan 2,
vlan-fid 32772.
```

### debug vpls events

**Syntax: [no] debug vpls events**

This command displays information about VPLS label-range allocation, LDP session status (up or down), LSP tunnel switchovers, VPLS end-point port transitions, VPLS peer state machine transitions, label exchanges, VC bindings, and VC withdrawals. The following example shows an MPLS uplink being disabled and then re-enabled:

```
PowerConnect#debug vpls events
PowerConnect(config-if-e1000-2/9)#disable
VPLS EVENT-LSP: mpls_vpls_process_tnnl_down()- VPLS test1: Peer 5.5.5.5, tunnel
1 is down.
VPLS EVENT-LSP: mpls_vpls_del_vpls_peer_in_tnnl_list()- VPLS test1: Delete peer
5.5.5.5 from tunnel-list of 1.
VPLS EVENT-FSM: mpls_vpls_peer_fsm_step()- VPLS fsm: vpls test1 peer 5.5.5.5
event
TNNL_DN st OPER->WAIT_TNNL act F.
VPLS EVENT-FSM: mpls_vpls_peer_fsm_step()- VPLS: vpls test1 peer 5.5.5.5 is now
DOWN, num up peer->0.
VPLS MAC-GROUP: vpls_mac_delete_remote_entry()- VPLS 1, VC-label 000f0000: HW
CAMs flush 4.
VPLS EVENT-PEER: mpls_vpls_send_vc_withdrawal()- VPLS 1, group 0, peer 5.5.5.5,
Send label withdraw for 983040.
(config-if-e1000-2/9)#enable
VPLS EVENT-LDP: mpls_ldp_peer_session_ind()- VPLS test1, peer 5.5.5.5 LDP
session is down.
VPLS EVENT-FSM: mpls_vpls_peer_fsm_ste ()- VPLS fsm: vpls test1 peer 5.5.5.5
event LDP_DN
st WAIT_TNNL->WAIT_TNNL act -.
VPLS EVENT-LSP: mpls_vpl_process_tnnl_up()- VPLS test1: Peer 5.5.5.5, tunnel 1
is up.
VPLS EVENT-LSP: mplc_vpls_add_vpls_peer_in_tnnl_list() = VPLS test1: Add peer
5.5.5.5 to tunnel-list of 1.
VPLS EVENT-FSM: mpls_vpls_peer_fsm_step()- VPLS fsm: vpls test1 peer 5.5.5.5,
send local-label 983040.
VPLS EVENT-FSM: mpls_vpls_peer_fsm_step()- VPLSfsm: vpls test 1 peer 5.5.5.5
event
RCV_LBL st WAIT_VC->OPER act D.
VPLS EVENT-FSM: mpls_vpls_peer_fsm_step()- VPLS: vpls test1 peer 5.5.5.5 is now
UP, num up peer->1.
```

### debug vpls filter

Syntax: [no] debug vpls filter [dst-mac-address |id |inner-tag | outer-vlan | src-mac-address | src-port | topo-id | topo-hw-idx | vc-label | vpls-peer-ip-address]

- **dst-mac-address -** Filter to include destination MAC address (LP only).
- **id** - Filter to include VPLS ID.
- **inner-tag** - Filter to include inner-tag ID (VLAN or ISID).
- **outer-vlan** - Filter to include VPLS outer VLAN.
- **src-mac-address** - Filter to include source MAC address.
- **src-port** - Filter to include source port.
- **topo-id** - Filter to include topology group ID (MP only).
- **topo-hw-idx** - Filter to include topology group hardware index (LP only).
- **vc-label** - Filter to include local VC Label.
- **vpls-peer-ip-address** - Filter to include this VPLS peer IP address.

This command displays VPLS filtering options.

**NOTE**
To clear a specific filter, specify the filter ID you want to clear. Enter the following command from the MP or LP console: **no debug vpls filter id** *<num>*

To clear all VPLS filters, enter the following command from the MP or LP console:
**no debug vpls filter**

### debug vpls generic

**Syntax: [no] debug vpls generic**

This command generates generic information about VPLS events such as VPLS MAC table allocation failures and VPLS MAC table deletions.

### debug vpls mac

**Syntax: [no] debug vpls mac [errors | group | local | remote]**

- **errors** - Displays information about VPLS MAC errors.
- **group** - Displays information about a VPLS MAC group.
- **local** - Displays information about VPLS local MAC learning.
- **remote** - Displays information about VPLS remote MAC learning.

This command generates information about VPLS MAC learning, aging, deletions, and topology changes. This command can be enabled for either local or remote MAC monitoring (or both).

**NOTE**
Use of this command can generate a large number of debug prints. To limit debug prints, use the **debug vpls count** command.

### debug vpls mac local

**Syntax: [no] debug vpls mac local**

This command generates information about local MAC activity. The following example indicates that an existing local VPLS MAC entry (mac vpls e 2/19) was deleted.

```
PowerConnect#debug vpls mac local
PowerConnect#clear mac vpls e 2/19
VPLS MAC: mpls_vpls_delete_mac_entry_from_table() - VPLS 1, MAC 0006.0100.0601.
VPLS MAC-GROUP: mac_group_delete_mac_entry_from_list()- 26d9f025: pt/lbl 58, vpls
1,
update_head 0, p 00000000, n 00000000.
VPLS MAC-GROUP: mp_vpls_mac_send_group_flush_msg()- Send IPC to LPs: Flush MACs for
VPLS
4294967295, port/label 0000003a,  is-vlan 0, hw-only 0.
VPLS MAC-GROUP: mac_flush_by_group() - VPLS 4294967295, port/label 0000003a,
hw-only 0:
Entries processed 1.
1 mac entries flushed
```

### debug vpls statistics

**Syntax: [no] debug vpls statistics**

This command can be used to monitor statistics collection activity for VPLS instances. The following output was generated with **debug vpls statistics** enabled, and as a result of the **show mpls statistics vpls** command.

```
PowerConnect#debug vpls statistics
debug vpls statistics is enabled
PowerConnect#show mpls statistics vpls
VPLS STATS: get_mpls_vpls_stat_from_lp () - VPLS 1.
VPLS STATS: get_mpls_vpls_stat_from_lp () - VPLS 1
VPLS-Name   In-Port(s)   Endpt-Out-Pkts   Tnl-Out-Pkts
---------   ----------   --------------   --------------
test1       e2/1 - e2/20 1252              0
test1       e4/1 - e4/2  1260              0
test1       e4/3 - e4/4  0                 0
```

### debug vpls topology

**Syntax: [no] debug vpls topology**

This command is used to debug the topology group related handling in the VPLS area. If there are any VPLS VLAN configured in a topology group, this command allows you to see what is taking place when topology group events occurred such as topology group master VLAN changes, forwarding state changes, etc.

**NOTE**
This command is available only on MP.

```
PowerConnect#debug vpls topology
VPLS TOPO: mpls_vpls_topo_itc_membership_update() - Send ITC update: Topo 2 VPLS
VLANs exist=1.
VPLS TOPO: mpls_vpls_topo_inherit_control_state() - Inherit control state for
Topo 2.
VPLS TOPO: mpls_vpls_topo_add_vpls_vlan() - Add VPLS 1 VLAN 300:0xffffffff to topo
(2) member list. Member count 1
VPLS TOPO: mpls_vpls_topo_update_end_points() - Topo 1 VLAN 300:0xffffffff old
topo_hw_index 0x0000ffff new topo_hw_index 0x00000000
VPLS TOPO: mpls_vpls_topo_update_end_points() - Update VPLS end-point state: VPLS
1 VLAN 300:0xffffffff Port 1/5 Block 0. topo_hw_index:0x00000000
VPLS TOPO: mpls_vpls_topo_add_member_vlan() - Set VPLS 1 VLAN 300:0xffffffff as
member of topo ID 2. with topo_hw_index 0x00000000
```

### debug vpls fsm-trace

**Syntax: [no] debug vpls fsm-trace** <*vpls_id*> <*Vpls Peer IP Address*>

This command shows VPLS Peer State Transition History. The output resembles the following:

```
PowerConnect#debug vpls fsm-trace 1 11.11.11.11
Time                FSM State             Rcvd Event            Action
==============      ==================== ==================== ======
Nov 16 22:52:51     0 WAIT_PT             PARAM_UPDT            -
Nov 16 22:53:24     0 WAIT_PT             PORT_UP               A
Nov 16 22:53:38     1 WAIT_TNNL           TNNL_UP               B
Nov 16 22:56:41     2 WAIT_PW_UP          PW_UP                 E
Nov 16 22:57:10     3 OPER                PW_DN                 I
Nov 16 22:57:22     2 WAIT_PW_UP          PW_UP                 E
Nov 16 22:58:02     3 OPER                PORT_DN               G
Nov 16 22:58:02     0 WAIT_PT             WITHD_SENT            -
Nov 16 22:58:02     7 WWD(WAIT_PT)        WITHD_DONE            -
Nov 16 22:58:43     0 WAIT_PT             PORT_UP               A
Nov 16 22:58:43     1 WAIT_TNNL           TNNL_UP               B
Nov 16 22:58:43     2 WAIT_PW_UP          PW_UP                 E
Nov 16 22:59:18     3 OPER                TNNL_DN               G
```

```
Nov 16 22:59:18      1 WAIT_TNNL          WITHD_SENT           -
Nov 16 22:59:18      5 WWD(TNNL_DWN)      WITHD_DONE           -
```

## Configuration notes

- **Monitoring** – Active monitoring of MAC addresses learned at each site can help determine when the VPLS is in a non-working state. Use the **show mac vpls** command at each site and verify that the local and remote MACs match for each VPLS instance. If you see any network event on the path you should check the state of the VPLS instances that traverse that path.
- **Troubleshooting** – If you discover a problem, you should gather the following information before doing a restoration of the services. Please alert Dell Technical Support, so engineering can be involved in the data capture.
- End nodes (where the VPLS instance terminates) at both ends.

  MP commands:

  ```
  Show mac vpls
  Show mpls summary
  Show mpls vpls detail
  Show mpls lsp detail
  Show mpls route
  Show mpls rsvp session detail
  Show mpls stat vpls (few times)
  Show mpls debug vpls remote
  Show mpls debug vpls local
  Show mpls debug next
  ```

  LP commands (slots 1, 2, and 4 on fr3.sjc, all others 1-3):

  ```
  Rconsole <slot>
  Show mpls vpls
  Show mac vpls
  Show mpls vpls counters
  Clear mpls vpls counters
  Show mpls vpls counters
  Show mpls next-hop
  Show mpls tunnel
  Show mpls vpls local
  Show mpls vpls remote
  ```

- Transit nodes, which identify the path each VPLS instance uses to traverse the network. Perform a traceroute between the end nodes in each direction to verify which routers are providing transit for the affected VPLS instances.

  MP commands:

  ```
  Show mpls summary
  Show mpls rsvp session detail
  Show mpls stat label (few times)
  ```

  LP commands (slots 1 and 2 on transit routers):

  ```
  Rconsole <slot>
  Show mpls lsp_xc
  Show mpls next-hop
  ```

### Recovery

The fastest way to recover is to remove the MPLS configuration from the end nodes and rebuild. This causes the end nodes to re-signal and build the tunnel, which updates the transit nodes with the correct information.

If you are still unable to correct the problem, try these additional steps:

- Delete LSPs and then reconfigure them.
- Delete MPLS interface and then reconfigure it.
- Delete VPLS peer configuration and then reconfigure it.
- Delete VPLS instance and then reconfigure it.

## Common diagnostic scenarios

- Cannot configure MPLS on a port that is part of a dynamic link aggregation (LACP - 802.1ad - dynamic trunk).

  Where MPLS is enabled globally on the device, a port that is configured in a trunk can be enabled as an MPLS interface port to create an MPLS trunk. You can either include a primary trunk port that has already been MPLS-enabled in a new trunk or MPLS-enable a primary trunk port of an already configured trunk. This feature was introduced in version 03.5.00 of the Multi-Service IronWare software. The following considerations must be considered when configuring MPLS on a trunk:

  - Only static server trunks and per-packet server trunks are supported.
  - Switch and LACP trunks are not supported.
  - MPLS is enabled on the primary port of the trunk and this enables MPLS on the entire trunk. Secondary ports of the trunk cannot be individually configured for MPLS.
- MPLS interfaces are not forwarding traffic.

  This may be caused because the customer is running software codes that do not match. It is recommended that customers always update software so that all interface modules are running the same code. If you have questions about your software version, contact Dell Technical Support for assistance.
- For a VPLS VLAN that is configured as multicast active, VPLS does not accept a receiver report.

  This issue was corrected in a later patch. The customer had not yet upgraded to the fixed patch. Once the upgrade was made, the problem disappeared.
- MPLS traffic is lost.

  This issue was corrected in a later patch. The customer had not yet upgraded to the fixed patch. Once the upgrade was made, the problem disappeared.
- Old software versions.

  Feature issues are often caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

Chapter

# 7

# ACL and QoS Diagnostics

This chapter provides diagnostic information for Access Control List (ACL) and Quality of Service (QoS) environments, including traffic management:

## ACLs

This chapter discusses the IP Access Control List (ACL) feature, which enables you to filter traffic based on the information in the IP packet header. For details on Layer 2 ACLs, refer to the Layer 2 ACL chapter in the *NetIron Configuration Guide*.

### ACL show commands

**show access-list**

**Syntax:** show access-list

To display the ACLs configured on a PowerConnect B-MLXe device, use the **show access-list** command.

**For numbered ACLs**

**Syntax: show access-list** <*number*> | **all**

For a numbered ACL, enter a command similar to the following:

```
PowerConnect(config)#show access-list 99
ACL configuration:
!
Standard IP access list 10
access-list 99 deny host 10.10.10.1
access-list 99 permit any
```

Enter the ACL number for the <*number*> parameter:

- 1 through 99 for standard ACLs
- 100 through 199 for extended ACLs

Enter **all** to display information for all ACLs configured on the device.

**For named ACLs**

**Syntax: show access-list name** <*acl-name*>

For a named ACL, enter a command similar to the following:

```
PowerConnect(config)#show access-list name entry

Standard IP access list entry
deny host 5.6.7.8
deny host 192.168.12.3|
permit any
```

Enter the ACL name for the <*acl-name*> parameter.

### show access-list accounting brief

**Syntax: show access-list accounting brief** [**l2** | **policy-based-routing** | **rate-limit**]

- **l2** - Limits the display to Layer 2 ACL accounting information.
- **policy-based-routing** - Limits the display to policy-based routing accounting information.
- **rate-limit** - Limits the display to rate limiting ACL accounting information.

IPv4 ACL accounting statistics are displayed if no option is specified.

To display a brief summary of the number of hits in all ACLs on a device, enter the following command.

```
PowerConnect(config)#show access-list accounting brief
Collecting ACL accounting summary for VE 1 ...  Completed successfully.
ACL Accounting Summary: (ac = accumulated since accounting started)
    Int    In ACL              Total In Hit   Out ACL            Total Out Hit
    VE 1    111                473963(1s)
                               25540391(1m)
                               87014178(5m)
                               112554569(ac)
```

### show access-list accounting

**Syntax: show access-list accounting** [**ethernet** <*slotnum*>/<*portnum*> | **ve** <*ve-number*> | **pos** <*slotnum*>/<*portnum*>] **in** | **out** [**l2** | **policy-based-routing** | **rate-limit**]

- **ethernet** <*slotnum*>/<*portnum*> - Displays a report for a physical interface.
- **pos** <*slotnum*>/<*portnum*> - Displays a report for a POS port.
- **ve** <*ve-number*> - Displays a report for the ports that are included in a virtual routing interface. For example, if ports 1/2, 1/4, and 1/6 are all members of ve 2, the report includes information for all three ports.
- **in** - Displays statistics for incoming traffic.
- **out** - Displays statistics for outgoing traffic.
- **l2** - Limits the display to Layer 2 ACL accounting information.
- **policy-based-routing** - Limits the display to policy-based routing accounting information. This option is only available for incoming traffic.
- **rate-limit** - Limits the display to rate limiting ACL accounting information.

This command displays statistics for an interface, as shown in the following example:

```
PowerConnect(config)#show access-list accounting ve 1 in
Collecting ACL accounting for VE 1  ...  Completed successfully.
ACL Accounting Information:
Inbound: ACL 111
    1: deny tcp any any
       Hit count: (1 sec)              237000   (1 min)12502822
                  (5 min)            87014178   (accum) 99517000
    3: permit ip any any
       Hit count: (1 sec)              236961   (1 min) 13037569
                  (5 min)                   0   (accum) 13037569
    0: deny tcp 1.1.1.0 0.0.0.255 2.2.2.0 0.0.0.255
       Hit count: (1 sec)                   0   (1 min) 0
                  (5 min)                   0   (accum) 0
    2: deny udp any any
       Hit count: (1 sec)                   0   (1 min) 0
                  (5 min)                   0   (accum) 0
```

### *Clearing ACL statistics*

Statistics on the ACL account report can be cleared:

- When a software reload occurs
- When the ACL is bound to or unbound from an interface
- When you enter the **clear access-list** command

**clear access-list**

**Syntax:** **clear access-list** [**all** | **ethernet** <*slotnum*>/<*portnum*> | **pos** <*slotnum*>/<*portnum*> | **ve** <*ve-num*>]

- **all** - Clears all statistics for all ACLs.
- **ethernet** <*slotnum*>/<*portnum*> - Clears statistics for ACLs on a physical port.
- **pos** <*slotnum*>/<*portnum*> - Clears statistics for ACLs on a POS port.
- **ve** <*ve-num*> - Clears statistics for all ACLs bound to ports that are members of a virtual routing interface.

## ACL debug commands

This section describes how to use Dell diagnostic debug commands to monitor Layer 2 and Layer 4 access lists (ACLs) for PowerConnect B-MLXe routers.

**NOTE**
To save space, date and time stamps have been removed from all output examples.

In most Layer 4 instances, a debug command must be accompanied by a user activity, such as binding or unbinding an ACL. In a few instances, background activity for Layer 4 access lists is displayed simply by enabling the debug function. Many of the debug acl commands work in conjunction with related show commands, as described in the examples in this chapter.

### debug access-list

Syntax: [no] debug access-list [accounting | ipv4 | ipv6 | l2 | mirror | policy-based-routing | rate-limit | receive]

- **accounting** - Displays ACL accounting statistics.
- **ipv4** - Displays information about IPv4 ACLs.
- **ipv6** - Displays information about IPv6 ACLs.
- **l2** - Displays information about Layer 2 ACLs.
- **mirror** - Displays ACL based mirroring.
- **policy-based-routing** - Displays information about policy-based routing activity.
- **rate-limit** - Displays information about ACL-based rate limiting.
- **receive** - Displays information about IP receive ACLs.

### debug access-list accounting

Syntax: [no] debug access-list accounting

This command displays information about inbound or outbound ACL accounting activity. For example, the following example shows inbound rate limit accounting for port 4/2:

```
PowerConnect#debug access-list accounting
PowerConnect#show access-list accounting ethernet 4/2 in
rate-limit
RL ACL accounting: retrieve for one interface
RL ACL accounting: retrieve for port 4/2, acl 101, outbound 0
RL ACL accounting: retrieve for port 4/2, acl 102, outbound 0
RL ACL Accounting Information:
Inbound: ACL 101
0: permit tcp any any
Hit count: (1 sec) 70 (1 min) 0 (5 min) 0 (accum) 0
1: deny ip any any
Hit count: (1 sec) 0 (1 min) 0(5 min) 0 (accum) 0
ACL 102
0: permit ip 192.168.2.0 0.0.0.255 40.1.1.0 0.0.0.255
Hit count: (1 sec) 0 (1 min) 0 (5 min) 0 (accum) 0
1: deny ip 192.168.2.0 0.0.0.255 50.1.1.0 0.0.0.255
Hit count: (1 sec) 0 (1 min) 0 (5 min) 0 (accum) 0
2: deny ip any any
Hit count: (1 sec) 0 (1 min) 0 (5 min) 0 (accum) 0
```

A summarized version of the inbound activity, which is easier to read, is shown in the following example:

```
PowerConnect#show access-list accounting 4/2 rate-limit brief
RL ACL accounting: retrieve brief information: MAX PORT 642
Collecting RL ACL accounting summary for 4/2, ACL 101, ACL policy 0 ... Completed
successfully
RL ACL Accounting Summary: (ac = accumulated since accounting started)
Int    In ACL       Total In Hit     Out ACL      Total Out Hit
4/2       101            69(1s)
                       2407(1m)
                          0(5m)
                       2407(ac)
4/2       102             0(1s)
                          0(1m)
                          0(ac)
```

### debug access-list ipv4

**Syntax:** [no] **debug access-list ipv4**

This command generates information about IPv4 access list activity. The following example displays information about inbound activity for access group 101:

```
PowerConnect#debug access-list ipv4
PowerConnect#ip access-gr 101 in
Bind/Unbind ACL: ACL 101, port 4/2, add 1, outbound 0
Send ITC ACL bind/unbind message: ACL_ID=101, name=, port=121, add=1, dir=in,
mask=
Received ITC ACL bind/unbind message: ACL type 0, ACL 101, add 1, outbound 0
COMMAND << ip access-group 101 in>>
Generated ACL binding command for LP: ip access-group 101 in
```

### debug access-list l2

**Syntax:** [no] **debug access-list l2**

This command generates information about Layer 2 ACL activity as shown in the following example, which shows Layer 2 inbound accounting information for interface 4/2:

```
PowerConnect#debug access-list l2
PowerConnect#show access-list accounting ethernet 4/2 in l2
L2 ACL accounting: retrieve for one interface
L2 ACL accounting: retrieve for port 4/2, acl 410, outbound 0
Collecting L2 ACL accounting for 410 on port 4/2 ... Completed successfully.
L2 ACL Accounting Information:
Inbound: ACL 410
0: permit 0000.0000.0001 ffff.ffff.ffff any any
Hit count: (1 sec) 70 (1 min) 0 (5 min) 0 (accum) 0
1: deny any any any
Hit count: (1 sec) 0 (1 min) 0 (5 min) 0 (accum) 0
```

### debug access-list policy-based-routing

**Syntax:** [no] **debug access-list policy-based-routing**

This command generates information about access-list policy-based routing, as shown in the following example:

```
PowerConnect#debug access-list policy-based-routing
PowerConnect#ip policy route-map pbrmap10
PBR: sending IPC message IPC_MSGTYPE_PBR_ROUTEMAP_AND_BINDING_UPDATE to IPC FID
53251
```

### debug access-list rate-limit

**Syntax:** [no] **debug access-list rate-limit**

This command generates information about rate limiting for IPv4 access lists. To establish a rate limit for the access group you want to observe, enter the following commands.

```
PowerConnect#debug access-list rate-limit
PowerConnect#rate-limit in access-group 101 500000000 750000000
PowerConnect#ip access-gr 101 in
Bind/Unbind ACL: ACL 101, port 4/2, add 1, outbound 0
Send ITC ACL bind/unbind message: ACL_ID=101, name=, port=121, add=1, dir=in, mask=
Received ITC ACL bind/unbind message: ACL type 0, ACL 101, add 1, outbound 0
COMMAND << ip access-group 101 in>>
Generated ACL binding command for LP: ip access-group 101 in
ACL-based Rate-Limiting: debugging is ON
```

### debug access-list receive generic

**Syntax:** [no] **debug access-list receive generic**

This command generates information about generic access list receive activity, as shown in the following example for access-list 101, sequence 10, which shows an ACL bind/unbind message being sent and received:

```
PowerConnect#debug access-list receive generic
PowerConnect#ip receive access-list 101 sequence 10
Send ITC Receive ACL bind/unbind message:
    ACL ID 101
    Sequence num 10
    Policy name
    strict acl enabled FALSE
    add 1
    Received ITC Receive-ACL bind/unbind message:
    ACL ID 101
    Sequence num 10
    Policy name
    strict acl enabled FALSE
    add 1
IP Receive ACL: Set global ACL 101 sequence 10 add 1
IP Receive ACL: Create/update ACL 101
Generated IP Receive ACL binding command for LP: ip receive access-list
```

### debug ipv6 access-list ipv6

**Syntax:** [no] **debug ipv6 access-list ipv6**

This command generates information about IPv6 access list activity. The following example assumes an IPv6 traffic filter on incoming traffic to virtual interface abc10.

Output similar to the following is displayed:

```
PowerConnect#debug ipv6 access-list ipv6
PowerConnect#ipv6 traffic-filter abc10 in
Send ITC ACL bind/unbind message: ACL_ID=0, name=abc10, port=651, add=1, dir=in,
mask=
Received ITC IPv6 ACL bind/unbind message: ACL type 2, ACL abc10, add 1, outbound
0
Set IPv6 ACL abc10 in internal: port number 651, enable 1
COMMAND(b) <<ipv6 traffic-filter abc10 in>>
```

Remove the inbound traffic filter for virtual interface abc10 by entering **no ipv6 traffic-filter abc10 in**. With debugging enabled, output similar to the following is displayed:

```
Send ITC ACL bind/unbind message: ACL_ID=0, name=abc10, port=651, add=0, dir=in,
mask=
Received ITC IPv6 ACL bind/unbind message: ACL type 2, ACL abc10, add 0, outbound
0
Set IPv6 ACL abc10 in internal: port number 651, enable 0
```

### debug ipv6 access-list stats

**Syntax:** [no] **debug ipv6 access-list stats**

This command generates statistical information about IPv6 access lists. In the following example, with this command enabled, the **show ipv6 access-list accounting brief** generates a brief accounting summary:

```
PowerConnect#debug ipv6 access-list stats
PowerConnect#show ipv6 access-list accouting ethernet 4/2
IPv6 ACL accounting: interface 122, port id 121
IPv6 ACL accounting: retrieve for port 4/2, acl abc10, outbound 0
IPv6 inbound ACL accounting: abc10 filter from 0, num 1
Collecting IPv6 ACL accounting for 4/2  ...  Completed successfully.
IPv6 ACL Accounting Information:
Inbound: IPv6 ACL abc10
   10:  permit tcp any any
       Hit count: (1 sec)                  99   (1 min)                1515
                  (5 min)                   0   (accum)                1515
PowerConnect#show ipv6 access-list accounting brief
IPv6 in/out ACL accounting: retrieve brief information: Max IPv6 interfaces 1193
Collecting IPv6 ACL accounting summary for 4/2 ...  Completed successfully.
IPv6 ACL Accounting Summary: (ac = accumulated since accounting started)
    Int    In ACL             Total In Hit   Out ACL             Total Out Hit
    4/2    abc10                   99(1s)
                                 1515(1m)
                                    0(5m)
                                 1515(ac)
```

## Configuration notes

- Do not apply an empty ACL (an ACL ID without any corresponding entries) to an interface. If you accidentally do this, the software applies the default ACL action, deny all, to the interface and denies all traffic.

### *Considerations when implementing ACL CAM sharing*

The following considerations apply when implementing this feature:

- If you enable ACL CAM sharing, ACL statistics will be generated per-PPCR instead of per-port. If you require the statistics  per-port granularity for your application, you cannot use this feature.
- This feature is only applicable for inbound IPv4 ACLs, IPv6 ACLs, VPNv4 ACLs, Layer-2 ACLs, and Global PBR policies.
- This feature is not applicable for ACL-based rate-limiting and interface-level PBR policies.
- This feature cannot be applied to a virtual interface.
- CAM entry matching within this feature is based on the ACL group ID.

### *ACL deny logging*

Carefully consider the following statements before configuring the ACL deny logging feature on your router.

- The ACL deny logging feature cannot be used in conjunction with the deny traffic redirection feature (command: **ip access-group redirect-deny-to-interf**). If you configure both features on the same interface, the ACL deny logging feature will take precedence and the deny traffic redirection will be disabled. Although disabled, deny traffic redirection will still be shown in the running configuration.
- ACL deny logging is a CPU-based feature. Consequently, to maintain maximum performance we recommend that you selectively enable the logging option only on the deny filters where you are interested in seeing the logs.
- ACL deny logging generates Syslog entries only. No SNMP traps are issued.
- The ACL deny logging feature is supported for inbound ACLs only.
- You can configure the maximum number of ACL session entries using the **system-max session-limit** command as described in the *NetIron Series Configuration Guide*.
- ACL logging is applicable only for traffic matching ACL deny clauses. It is not applicable for traffic matching ACL permit clauses.
- Do not apply an empty ACL (an ACL ID without any corresponding entries) to an interface. If you accidentally do this, the software applies the default ACL action, deny all, to the interface and thus denies all traffic.

## Common diagnostic scenarios

- Interrupted policy routing causing severe network problems.

  This happened because the customer was running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

- Error message: ACL: List 101 denied tcp.

  This problem was resolved by reapplying ACL list 101.

- The **acl replace** command failed.

  This occurred because the **show access-list accounting ethernet 5/3 in | include ntp** command that was used incorrectly counted the ACL statements. The **show access-list 101** command displays line #70 as the HTTP statement and line # 71 as the NTP statement, so the **acl replace** command replaced the wrong line. This issue was fixed in a patch. If you encounter these types of issues, contact Dell Technical Support for assistance.

- Need to enable ACL filtering based on VLAN membership or VE port.

  This was a customer request for help with an ACL configuration. The following information helped complete this configuration:

  Before you can bind an ACL to specific VLAN members on a port, you must first enable support for this feature. If this feature is not already enabled on your device, enable it as instructed here:

  ```
  PowerConnect(config)#enable acl-per-port-per-vlan
  PowerConnect(config)#write memory
  PowerConnect(config)#exit
  ```

- When an ACL was removed from a port with port mapping (ACL-based rate-limiting) configured, the MLXe stopped all traffic on this port.

  If you make an ACL configuration change, you must reapply the ACLs to their interfaces for he change to take effect. An ACL configuration change includes any of the following:

  - Adding, changing, or removing an ACL or an entry in an ACL
  - Changing a PBR policy
  - Changing ToS-based QoS mappings

  To reapply ACLs following an ACL configuration change, enter the following command at the global CONFIG level of the CLI:

  ```
  PowerConnect(config)# ip rebind-acl all
  ```

  **NOTE**
  Do not apply an empty ACL (an ACL ID without any corresponding entries) to an interface. If you accidentally do this, the software applies the default ACL action, deny all, to the interface and thus denies all traffic.

- Old software versions.

  Feature issues are often caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# QoS

Quality of Service (QoS) features prioritize the use of bandwidth in a router. When QoS features are enabled, traffic is classified as it arrives at the router, and processed on the basis of configured priorities. Traffic can be dropped, prioritized for guaranteed delivery, or subject to limited delivery options, depending on how you configure QoS features.

# QoS show commands

### show qos-tos

**Syntax: show qos-tos**

This command displays QoS and ToS configuration information, as shown in the following example:

```
PowerConnect(config)#show qos-tos
Interface QoS , Marking and Trust Level:

i/f    | QoS       | Mark     | Trust-Level
-------+-----------+----------+---------------
1      | No        | No       | L2 CoS
2      | No        | No       | L2 CoS
3      | No        | No       | L2 CoS
4      | No        | No       | L2 CoS
... <lines omitted for brevity>
49     | No        | No       | L2 CoS
50     | No        | No       | L2 CoS
ve5    | No        | No       | L2 CoS
... <lines omitted for brevity>

COS-DSCP map:
     COS: 0 1 2 3 4 5 6 7
  ---------------------------------------------------------
     dscp: 0 8 16 24 32 40 48 56
IP Precedence-DSCP map:
  ip-prec: 0 1 2 3 4 5 6 7
  ---------------------------------------------------------
     dscp: 0 8 16 24 32 40 48 56
DSCP-Priority map: (dscp = d1d2)
     d2| 0 1 2 3 4 5 6 7 8 9
  d1   |
  -----+----------------------------------------
     0 | 0 0 0 0 0 0 0 0 1 1
     1 | 1 1 1 1 1 1 2 2 2 2
     2 | 2 2 2 2 3 3 3 3 3 3
    ... <lines omitted for brevity>
DSCP-DSCP map: (dscp = d1d2)
     d2| 0 1 2 3 4 5 6 7 8 9
  d1   |
  -----+----------------------------------------
     0 | 0 1 2 3 4 5 6 7 8 9
     1 | 10 11 12 13 14 15 16 17 18 19
     2 | 20 21 22 23 24 25 26 27 28 29
     3 | 30 31 32 33 34 35 36 37 38 39
     4 | 40 41 42 43 44 45 46 47 48 49
     5 | 50 51 52 53 54 55 56 57 58 59
     6 | 60 61 62 63
```

### show qos wred

**Syntax:** show qos wred

On the PowerConnect B-MLXe device router, traffic levels that exceed the bandwidth of individual ports are buffered by queues. For each output port, a set of eight priority queues is allocated on each inbound traffic manager. When traffic exceeds the bandwidth of a port, packets are dropped randomly as long as the congestion persists. Under these conditions, traffic of greater priority can be dropped instead of traffic with a lesser priority.

Instead of being subject to this random process, you can configure a PowerConnect B-MLXe device router to monitor traffic congestion and drop packets according to a Weighted Random Early Discard (WRED) algorithm. This algorithm enables the system to detect the onset of congestion and take corrective action. In practice, WRED causes a router to start dropping packets as traffic in the router starts to back up. WRED provides various control points that can be configured to change a system's reaction to congestion.

The **show qos wred** command displays output similar to that shown in this example:

```
PowerConnect#show qos wred
QType Enable AverWt    MaxQSz DropPrec MinAvgQSz MaxAvgQSz MaxDropProb MaxPktSz
0     Yes    9(0.19%)  16384  0           5696     16384          2%      16384
                               1           4864     16384          4%      16384
                               2           4096     16384          9%      16384
                               3           3264     16384         10%      16384
1     No
2     No
3     Yes    9(0.19%)  16384  0           6528     16384          2%      16384
                               1           5696     16384          4%      16384
                               2           4864     16384          9%      16384
                               3           4096     16384          9%      16384
4     No
5     No
6     No
7     No
```

## QoS debug commands

There are no debug commands specific to QoS.

## Configuration notes

- You cannot use advanced TOS based QoS and other Layer 4 features such as:
  - IPv4 ACLs and IPv4 ACL-based rate-limiting
  - Layer 2 ACLs and Layer 2 ACL-based rate-limiting
  - PBR
  - VLAN ID and Inner VLAN ID translation on the same interface
- QoS mappings are globally configurable and apply to all interfaces.
- To place a QoS mapping change into effect, you must enter the **ip rebind-acl all** command at the global CONFIG level of the CLI after making the mapping change. This applies to mappings that are configured using the **qos-tos map** command.

- Because excess traffic is buffered, rate shaping must be used with caution. In general, it is not advisable to rate shape delay-sensitive traffic.

## Common diagnostic scenarios

- Having trouble configuring QoS.

  To configure QoS, you must first enable **port-priority** globally on the device, using the commands shown:

  ```
  PowerConnect(config)# port-priority
  PowerConnect(config)# write memory
  PowerConnect(config)# end
  PowerConnect# reload
  ```

- Old software versions.

  Feature issues are often caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# Traffic management

## Traffic management show commands

The following sections describe the show commands you can use to display traffic management information.

### show tm statistics

**Syntax:** show tm statistics

This command displays all traffic manager statistics for the port groups that belong to each traffic manager, as shown in the following example:

```
PowerConnect#show tm statistics
--------- Ports 2/1 - 2/20 ---------
Ingress Counters:
   Total Ingress Pkt Count:                 464418
   EnQue Pkt Count:                         464418
   EnQue Byte Count:                        51904240
   DeQue Pkt Count:                         464418
   DeQue Byte Count:                        51904240
   TotalQue Discard Pkt Count:              0
   TotalQue Discard Byte Count:             0
   Oldest Discard Pkt Count:                0
   Oldest Discard Byte Count:               0
Egress Counters:
   EnQue Pkt Count:                         701812
   EnQue Byte Count:                        78785888
   Discard Pkt Count:                       0
   Discard Byte Count:                      0
--------- Ports 4/1 - 4/20 ---------
Ingress Counters:
   Total Ingress Pkt Count:                 0
   EnQue Pkt Count:                         0
   EnQue Byte Count:                        0
   DeQue Pkt Count:                         0
   DeQue Byte Count:                        0
   TotalQue Discard Pkt Count:              0
   TotalQue Discard Byte Count:             0
   Oldest Discard Pkt Count:                0
   Oldest Discard Byte Count:               0
Egress Counters:
   EnQue Pkt Count:                         0
   EnQue Byte Count:                        0
   Discard Pkt Count:                       0
   Discard Byte Count:                      0
```

### show tm statistics ethernet

**Syntax: show tm statistics ethernet** <*slotnum*>/<*portnum*>

This command displays traffic manager statistics for a specified port group (identified by a slot and port within the group), as shown in the following example:

```
PowerConnect#show tm statistics ethernet 2/1
--------- Ports 2/1 - 2/20 ---------
Ingress Counters:
   Total Ingress Pkt Count:                 464454
   EnQue Pkt Count:                         464454
   EnQue Byte Count:                        51907696
   DeQue Pkt Count:                         464454
   DeQue Byte Count:                        51907696
   TotalQue Discard Pkt Count:              0
   TotalQue Discard Byte Count:             0
   Oldest Discard Pkt Count:                0
   Oldest Discard Byte Count:               0
Egress Counters:
   EnQue Pkt Count:                         701866
   EnQue Byte Count:                        78791072
   Discard Pkt Count:                       0
   Discard Byte Count:                      0
```

### show tm statistics all-counters ethernet

**Syntax:** show tm statistics all-counters ethernet <*slotnum*>/<*portnum*>

This command displays traffic manager statistics with all the counters.

```
PowerConnect#show tm statistics all-counters ethernet 1/1
--------- Ports 1/1 ---------
Ingress Counters:
  Total Ingress Pkt Count:         14119009
  EnQue Pkt Count:                 12741416
  DeQue Pkt Count:                 7944950
  TotalQue Discard Pkt Count:      1377495
  Oldest Discard Pkt Count:        4801282
Programmable Ingress Counters:
[Queue Select: 8000, Queue Mask 0x0007]
  Prg IQM EnQue Pkt Count:         0
  Prg IQM DeQue Pkt Count:         0
  Prg IQM Tail Delete Pkt Count:   0
  Prg IQM Head Delete Pkt Count:   0
  Prg IQM Max Occupancy Q size:    0
Egress Counters:
  EnQue Pkt Count:                 14875149
  DeQue Pkt Count:                 14875257
  Discard Pkt Count:               0
  Reassembly error Discard Count:  0
  Pruning Discard Count:           0
```

### show tm statistics slot

**Syntax: show tm statistics slot** <*num*>

This command displays all traffic manager statistics for an interface module identified by slot number, as shown in the following example:

```
PowerConnect#show tm statistics slot 4
--------- Ports 4/1 - 4/20 ---------
Ingress Counters:
   Total Ingress Pkt Count:                 0
   EnQue Pkt Count:                         0
   EnQue Byte Count:                        0
   DeQue Pkt Count:                         0
   DeQue Byte Count:                        0
   TotalQue Discard Pkt Count:              0
   TotalQue Discard Byte Count:             0
   Oldest Discard Pkt Count:                0
   Oldest Discard Byte Count:               0
Egress Counters:
   EnQue Pkt Count:                         0
   EnQue Byte Count:                        0
   Discard Pkt Count:                       0
   Discard Byte Count:                      0
```

**NOTE**
The byte counts displayed from the **show tm statistics** command incorporate proprietary internal headers of various lengths.

## Clearing traffic management statistics

You can clear traffic management statistics selectively for a specified port group, an interface module, or for an entire PowerConnect B-MLXe router using the following commands.

### clear tm statistics

**Syntax: clear tm statistics** [**ethernet** <*slotnum*>/<*portnum*> | **slot** <*slot-number*>]

- **ethernet** <*slotnum*>/<*portnum*> - Clears traffic manager statistics for a specific port group.
- **slot** <*slot-number*> - Clears traffic manager statistics for a specific interface module.

## Configuration notes

- The byte counts displayed from the **show tm statistics** command incorporate proprietary internal headers of various lengths.
- A traffic manager contains a specific number of ports depending on the Interface module. Specifying a particular port and slot gathers statistics for all ports that belong to the same port group.
- The PowerConnect B-MLXe device routers classify packets into one of eight internal priorities. Traffic scheduling allows you to selectively forward traffic according to the forwarding queue that is mapped to according to one of the following schemes:

- **Strict priority-based scheduling** – This scheme guarantees that higher-priority traffic is always serviced before lower priority traffic. The disadvantage of strict priority-based scheduling is that lower-priority traffic can be starved of any access.
- **WFQ weight-based traffic scheduling** – With WFQ destination-based scheduling enabled, some weight-based bandwidth is allocated to all queues. With this scheme, the configured weight distribution is guaranteed across all traffic leaving an egress port and an input port is guaranteed allocation in relationship to the configured weight distribution.
- **Mixed strict priority and weight-based scheduling** – This scheme provides a mixture of strict priority for the three highest priority queues and WFQ for the remaining priority queues.

Chapter

# Multicast Diagnostics 8

This chapter provides diagnostic information about IP multicast environments on PowerConnect B-MLXe routers. The following protocols are documented:

## IP multicasting

Multicast protocols allow a group or channel to be accessed over different networks by multiple stations (clients) for the receipt and transmission of multicast data.

PowerConnect B-MLXe device routers support two multicast routing protocols—Distance Vector Multicast Routing Protocol (DVMRP) and Protocol-Independent Multicast (PIM) protocol along with the Internet Group Membership Protocol (IGMP).

PIM and DVMRP are broadcast and pruning multicast protocols that deliver IP multicast datagrams. DVMRP and PIM build a different multicast tree for each source and destination host group.

## DVMRP

PowerConnect B-MLXe device routers provide multicast routing with the Distance Vector Multicast Routing Protocol (DVMRP) routing protocol. DVMRP uses IGMP to manage the IP multicast groups.

DVMRP is a broadcast and pruning multicast protocol that delivers IP multicast datagrams to intended receivers. The receiver registers the interested groups using IGMP. DVMRP builds a multicast delivery tree with the sender forming the root. Initially, multicast datagrams are delivered to all nodes on the tree. Those leaves that do not have any group members send prune messages to the upstream router, noting the absence of a group. The upstream router maintains a prune state for this group for the given sender. A prune state is aged out after a given configurable interval, allowing multicasts to resume.

### DVMRP show commands

You can use show commands to display the following DVMRP information:

- DVMRP group information
- DVMRP interface information
- DVMRP multicast cache information
- DVMRP neighbor information
- DVMRP active prune information
- Available multicast resources
- IP multicast route information
- Active multicast traffic information

### show ip mcache

**Syntax:** show ip mcache

This command displays information about the DVMRP multicast cache, as shown in the following example:

```
PowerConnect#show ip mcache
Total 2 entries
1    (192.2.1.2, 226.1.1.1) in v20 (e2/2)
     L3 (HW) 1: e2/4(VL40)
     fast=1 slow=0 leaf=0 prun=0 frag=0 tag=0 tnnl=0 swL2=0 hwL2=0 swRepl=0
     age=0 fid: 8012,
2    (192.1.1.2, 225.1.1.1) in v10 (e2/1)
     L3 (HW) 1: e2/3(VL30)
     fast=1 slow=0 leaf=0 prun=0 frag=0 tag=0 tnnl=0 swL2=0 hwL2=0 swRepl=0
     age=0 fid: 8011,
Total number of mcache entries 2
```

### show ip dvmrp group

**Syntax:** show ip dvmrp group

This command displays information about DVMRP groups, as shown in this example:

```
PowerConnect#show ip dvmrp group
Total number of groups: 2
1    Group 225.1.1.1           Ports
     Group member at  e2/3: v30
2    Group 226.1.1.1           Ports
     Group member at  e2/4: v40
```

### show ip dvmrp interface

**Syntax: show ip dvmrp interface**

This command displays DVMRP Interface information, as shown in the following example.

```
PowerConnect#show ip dvmrp interface
Interface e5/2
TTL Threshold: 1, Enabled, Querier
Local Address: 172.5.1.1
DR: itself
Neighbor:
  172.5.1.2
Interface e8/1
TTL Threshold: 1, Enabled, Querier
Local Address: 172.8.1.1
DR: itself
Interface v10
TTL Threshold: 1, Enabled, Querier
Local Address: 192.1.1.1 logical Vid=1
DR: itself
Interface v20
TTL Threshold: 1, Enabled, Querier
Local Address: 192.2.1.1 logical Vid=2
DR: itself
Interface v30
TTL Threshold: 1, Enabled, Querier
Local Address: 192.3.1.1 logical Vid=3
DR: itself
Interface v40
TTL Threshold: 1, Enabled, Querier
Local Address: 192.4.1.1 logical Vid=4
DR: itself
```

### show ip dvmrp nbr

**Syntax:** show ip dvmrp nbr

This command displays information about DVMRP neighbors, as shown in this example:

```
PowerConnect#show ip dvmrp nbr
Port   Phy_p  Neighbor        GenId    Age   UpTime
e5/2   e5/2   172.5.1.2       00000000 170   440
```

### show ip dvmrp prune

**Syntax:** show ip dvmrp prune

This command displays DVMRP prune information, as shown in this example:

```
PowerConnect#show ip dvmrp prune
Port SourceNet       Group           Nbr             Age
                                                     sec
e5/2  192.2.1.2       226.1.1.1      172.5.1.1       60
e5/2  192.1.1.2       225.1.1.1      172.5.1.1       60
```

### show ip dvmrp resource

**Syntax: show ip dvmrp resource**

This command displays information about available multicast resources, as shown in the following example:

```
PowerConnect#show ip dvmrp resource
                     allocated    in-use available allo-fail  up-limita
DVMRP route               2048        13      2035         0      2048
route interface           2048        13      2035         0      8192
NBR list                   128         1       127         0      1874
prune list                  64         2        62         0       256
graft list                  64         0        64         0       256
mcache                     128         2       126         0      4096
mcache hash link           547         2       545         0  no-limit
graft if no mcache         197         0       197         0  no-limit
IGMP group                 256         2       254         0      2048
pim/dvm intf. group        256         2       254         0  no-limit
pim/dvm global group       256         2       254         0      4096
HW replic vlan            2000         4      1996         0  no-limit
HW replic port            1024         2      1022         0  no-limit
```

### show ip dvmrp route

**Syntax:** show ip dvmrp route

This command displays IP multicast route information, as shown in this example:

```
PowerConnect#show ip dvmrp route
                     allocated    in-use available allo-fail  up-limita
DVMRP route               2048        13      2035         0      2048
route interface           2048        13      2035         0      8192
NBR list                   128         1       127         0      1874
prune list                  64         2        62         0       256
graft list                  64         0        64         0       256
mcache                     128         2       126         0      4096
mcache hash link           547         2       545         0  no-limit
graft if no mcache         197         0       197         0  no-limit
IGMP group                 256         2       254         0      2048
pim/dvm intf. group        256         2       254         0  no-limit
pim/dvm global group       256         2       254         0      4096
HW replic vlan            2000         4      1996         0  no-limit
HW replic port            1024         2      1022         0  no-limit
```

### show ip dvmrp traffic

**Syntax: show ip dvmrp traffic**

This command displays active multicast traffic information, as shown in this example:

```
PowerConnect#show ip dvmrp traffic
Port          Probe                   Graft                   Prune
     [Rx      Tx      Dscrd] [Rx      Tx      Dscrd] [Rx      Tx      Dscrd]
e5/2   111     112     0       0       0       0       9       0       1
e8/1   0       220     0       0       0       0       0       0       0
v10    0       211     0       0       0       0       0       0       0
v20    0       210     0       0       0       0       0       0       0
Total 111     1718     0       0       0       0       9       0       1
IGMP Statistics:
  Total Discard/chksum  0/0
```

## DVMRP debug commands

### debug ip pim-dvmrp

**Syntax:** [no] **debug ip pim-dvmrp** [**add-del-oif** | **bootstrap** | **clear** | **event** | **group** | **ipc** | **join-prune** | **level** | **nbr-change** | **route-change** | **show** | **source** | **vlan-id** | **vpls-id**]

- **add-del-oif** - Displays mcache additions or deletions.
- **bootstrap** - Displays bootstrap messages in detail.
- **clear** - Clears PIM-DVMRP debug settings.
- **event** - Displays information about infrastructure events and callback handling.
- **group** - Displays activity for a specific group.
- **ipc** - Displays information about ipc messages between management processor and line processor.
- **join-prune** - Displays information about join/prune messages.
- **level** - Sets level of debug information from 1 through 3 (3 generates the most detailed information).
- **nbr-change** - Displays information about neighbor port changes.
- **route-change** - Displays information about route change events.
- **show** - Shows PIM-DVMRP debug settings.
- **source** - Displays information about multicast traffic from a specific source.
- **vlan-id** - Displays information a specified VLAN.
- **vpls-id** - Displays information about a specific VPLS ID.

### debug ip pim-dvmrp add-del-oif

**Syntax:** [no] **debug ip pim-dvmrp add-del-oif** [**stack**]

This command monitors and displays instances of mcache activity, such as OIF (outbound interface) additions or deletions. When the stack option is selected, this command also generates a stack trace of the add or delete event. Output is similar to the following:

```
PowerConnect#debug ip pim-dvmrp add-del-oif
Added oif v10, e2/1 to (10.10.10.2, 224.225.0.1) entry
```

This indicates that vlan 10 on port 2/1 has been added to the OIF table for multicast stream (10.10.10.2, 224.225.0.1).

### debug ip pim-dvmrp clear

**Syntax:** [no] **debug ip pim-dvmrp clear**

This command clears all PIM-DVMRP debug settings.

### debug ip pim-dvmrp ipc

**Syntax:** [no] debug ip pim-dvmrp ipc

This command displays IPC messages between the management processor and a line processor. Output is similar to the following, which Indicates a line processor notification has been enabled for stream (10.10.10.1, 224.225.0.1). This stream originates from port e2/1 on vlan 10.

```
PowerConnect#debug ip pim-dvmrp ipc
receive slave messages S_G_CREAT_NOTIF, entry (10.10.10.1, 224.255.0.1)intf v10,
e2/1
```

### debug ip pim-dvmrp join-prune

**Syntax:** [no] **debug ip pim-dvmrp join-prune**

This command displays information about join/prune activity. Output resembles the following:

```
PowerConnect#debug ip pim-dvmrp join-prune
PIMDM:Rx Join/Prune from 30.30.30.1, on intf v10, e2/1. RPF Addr 20.20.20.1.ToME 1
```

This indicates that a join or prune message has been received from 30.30.30.1, on Ethernet port 2/1, vlan 10 for RPF Addr 20.20.20.1. ToME means this message should be processed.

### debug ip pim-dvmrp level

**Syntax:** [no] **debug ip pim-dvmrp level** <*num*>

This command sets the level of detail for debug output. Levels range from 0 through 3, with 3 being the most detailed. Levels currently supported are:

- **Level 0** - Receive input or send output messages
- **Level 1** - Process control message

### debug ip pim-dvmrp nbr-change

**Syntax:** [no] **debug ip pim-dvmrp nbr-change**

This command displays information about neighbor port changes. Output resembles the following:

```
PowerConnect#debug ip pim-dvmrp nbr-change
nbr 30.30.30.1 phy change from e2/1 to e3/1
```

This output indicates that neighbor 30.30.30.1 has changed from port e2/1 to port e3/1.

### debug ip pim-dvmrp show

**Syntax:** [no]**debug ip pim-dvmrp show**

This command displays current debug settings for PIM-DVMRP. For example:

```
PowerConnect#debug ip pim-dvmrp show
debug ip pim is enabled
```

## Common diagnostic scenarios

- Old software versions.

  Feature issues are often caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# IGMP V2 and V3

The Internet Group Management Protocol (IGMP) allows an IPV4 system to communicate IP Multicast group membership information to its neighboring routers. The routers in turn limit the multicast of IP packets with multicast destination addresses to only those interfaces on the router that are identified as IP Multicast group members. This release introduces the support of IGMP version 3 (IGMP V3) on PowerConnect B-MLXe device Series routers.

In IGMP V2, when a router sends a query to the interfaces, the clients on the interfaces respond with a membership report of multicast groups to the router. The router can then send traffic to these groups, regardless of the traffic source. When an interface no longer needs to receive traffic from a group, it sends a leave message to the router which in turn sends a group-specific query to that interface to see if any other clients on the same interface is still active.

IGMP V3 provides selective filtering of traffic based on traffic source. A router running IGMP V3 sends queries to every multicast enabled interface at the specified interval. These general queries determine if any interface wants to receive traffic from the router.

## IGMP show commands

### show ip igmp group

**Syntax: show ip igmp** [**vrf** <*vrf-name*>] **group** [<*group-address*> [**detail**] [**tracking**]]

- <*group-address*> - Displays a report for a specific multicast group. Omit the <*group-address*> for a report for all multicast groups.
- **vrf** <*vrf-name*> - Specifies that you want to display IGMP group information for the VRF specified.
- **detail** - Displays the source list of the multicast group.
- **tracking** - Displays information on interfaces that have tracking enabled.

This command displays the status of all IGMP multicast groups on a device, as shown in the following example.

```
PowerConnect#show ip igmp group
Total 2 entries
------------------------------------------------------
Idx Group Address    Port   Intf   Mode    Timer Srcs
---+----------------+------+------+-------+-----+----
  1 232.0.0.1        e6/2   v30    include     0   7
  2 226.0.0.1        e6/2   v30    exclude   240   2
                     e6/3   e6/3   include     0   3
Total number of groups 2
```

### show ip igmp group detail

**Syntax: show ip igmp group** <*group-address*> **detail**

This command displays the status of a specific IGMP multicast group, as shown in this example.

```
PowerConnect#show ip igmp group 239.0.0.1 detail
Total 2 entries
------------------------------------------------------
Idx Group Address    Port   Intf   Mode    Timer Srcs
---+---------------+------+------+-------+-----+----
  1 226.0.0.1        e6/2   v30    exclude   218   2
     S: 40.40.40.12
     S: 40.40.40.11
     S: 40.40.40.10
     S: 40.40.40.2          (Age: 218)
     S: 40.40.40.3          (Age: 218)
    226.0.0.1        e6/3   e6/3   include     0   3
     S: 30.30.30.3          (Age: 165)
     S: 30.30.30.2          (Age: 165)
     S: 30.30.30.1          (Age: 165)
```

### show ip igmp group tracking

**Syntax: show ip igmp group** <*group-address*> **tracking**

If tracking and fast leave are enabled, you can display the list of clients that belong to a particular group by entering commands similar to those in the following example:

```
PowerConnect#show ip igmp group 224.1.10.1 tracking
Total 2 entries
------------------------------------------------------
Idx Group Address    Port   Intf   Mode    Timer Srcs
---+---------------+------+------+-------+-----+----
  1 226.0.0.1        e6/2   v30    exclude   253   3
     S: 40.40.40.12
     S: 40.40.40.11
     S: 40.40.40.10
     S: 40.40.40.2          (Age: 253)
                    C: 10.10.10.1       (Age: 253)
    S: 40.40.40.3          (Age: 253)
                    C: 10.10.10.1       (Age: 253)
    226.0.0.1        e6/3   e6/3   include     0   3
     S: 30.30.30.3          (Age: 196)
                    C: 10.2.0.1         (Age: 196)
     S: 30.30.30.2          (Age: 196)
                    C: 10.2.0.1         (Age: 196)
     S: 30.30.30.1          (Age: 196)
                    C: 10.2.0.1         (Age: 196)
```

### show ip igmp vrf eng static

**Syntax: show ip igmp vrf eng** [<*vrf-name*>] **static**

The **vrf** parameter with the <*vrf-name*> variable displays static IGMP group information for a specified VRF, as shown in the following example:

```
PowerConnect#show ip igmp vrf eng
static
Group Address    Interface Port List
---------------+---------+---------
      229.1.0.12       4/1 ethe 4/1
      229.1.0.13       4/1 ethe 4/1
      229.1.0.14       4/1 ethe 4/1
```

### show ip igmp interface

**Syntax: show ip igmp** [**vrf** <*vrf-name*>] **interface** [**ve** <*num*> | **ethernet** <*slotnum*>/<*portnum*> | **pos** <*slotnum*>/<*portnum*>]

- **vrf** <*vrf-name*> - Displays IGMP interface information for the VRF specified by the <*vrf-name*> variable.
- **ve** <*num*> - Displays information for a specific virtual routing interface.
- **pos** <*slotnum*>/<*portnum*> - Displays information for a specific pos interface.
- **ethernet** <*slotnum*>/<*portnum*> - Displays information for a specific ethernet interface.

This command displays the status of a multicast enabled port, as shown in the following example:

```
PowerConnect#show ip igmp interface
---------+------+---------+---------------+---------+-----+-----+---------
Intf/Port|Groups| Version |Querier        |  Timer  |V1Rtr|V2Rtr|Tracking
         |      |Oper  Cfg|               |OQrr GenQ|     |     |
---------+------+----+----+---------------+----+----+-----+-----+---------
e6/3           1    3    3 Self               0   94 No    No     Disabled
e6/4           0    2    - Self               0   94 No    No     Disabled
v30            1    3    3                                        Disabled
     e6/2           3    - Self               0   20 No    No
v40            0    3    3                                        Disabled
     e6/2           3    - Self               0   20 No    No
v50            0    2    -                                        Disabled
    e12/1           2    - Self               0   29 No    No
     e6/8           2    - 50.1.1.10         46    0 No    Yes
     e6/1           2    - Self               0  115 No    Yes
```

### show ip igmp traffic

**Syntax: show ip igmp** [**vrf** <*vrf-name*>] **traffic**

The **vrf** parameter with <*vrf-name*> variable displays IGMP traffic information for a specific VRF.

This command displays the traffic status on each virtual routing interface, as shown in the following example:

```
PowerConnect#show ip igmp traffic
Recv  QryV2 QryV3 G-Qry GSQry MbrV2 MbrV3 Leave  IsIN  IsEX ToIN ToEX ALLOW BLK
v5       29     0     0     0     0     0     0     0     0    0    0    0    0
v18      15     0     0     0     0    30     0    60     0    0    0    0    0
v110      0     0     0     0     0    97     0   142    37    2    2    3    2
Send  QryV1 QryV2 QryV3 G-Qry GSQry
v5        0     2     0     0     0
v18       0     0    30    30     0
v110      0     0    30    44    11
```

**show ip igmp settings**

**Syntax: show ip igmp** [**vrf** <*vrf-name*>] **settings**

The **vrf** parameter with <*vrf-name*> variable displays IGMP setting information for a specified VRF.

This command displays global IGMP settings or IGMP settings for a specified VRF. Global IGMP settings are shown in the following example:

```
PowerConnect#show ip igmp  settings
IGMP Global Configuration
  Query Interval            : 125s
  Configured Query Interval : 125s
  Max Response Time         : 10s
  Group Membership Time     : 260s
  Configured Version        : 2
  Operating Version         : 2
```

## Clearing the IGMP group membership table

To clear the IGMP group membership table, enter the following command.

**clear ip igmp cache**

**Syntax: clear ip igmp** [**vrf** <*vrf-name*>] **cache**

This command clears the IGMP membership for the default router instance or for a specified VRF. Use the **vrf** option with the <*vrf-name*> variable to clear the traffic information for a specific VRF instance.

## Clearing IGMP traffic statistics

To clear statistics for IGMP traffic, enter the following command.

**clear ip igmp traffic**

**Syntax: clear ip igmp** [**vrf** <*vrf-name*>] **traffic**

This command clears all the multicast traffic information on all interfaces on the device.

Use the **vrf** option to clear the traffic information for a VRF instance specified by the <*vrf-name*> variable. This option became available in version 03.5.00 of the Multi-Service IronWare software.

## Clearing IGMP group flows

To clear all the IGMP flows, enter the following command at the Privileged EXEC level of the CLI:

**clear ip multicast all**

**Syntax:** clear ip multicast all

## IGMP debug commands

**debug ip igmp**

**Syntax:** [no] **debug ip igmp**

This command generates information about IGMP activity, including IGMP membership queries, membership responses, and the conversion of IGMPv2 to IGMPv3 through DNS lookup. Output resembles the following:

```
PowerConnect#debug ip igmp
IGMP.RCV: Type Query Port 0/0 PktLen 8. GrpAddr 0.0.0.0 Src: 69.28.172.53
IGMP.RCV: Type Query Port 1/1 PktLen 8. GrpAddr 0.0.0.0 Src: 69.28.172.110
```

**debug ip vrf**

**Syntax:** [no] **debug ip vrf**

This command generates information about synchronization of VRF routing information to line cards, similar to the following, which shows a download request from the line card to start the tree download for VFR-1:

```
PowerConnect#debug ip vrf
RTM (vrf): Processing tree download for vrf 1
```

## Configuration notes

- Each of the multicast protocols uses IGMP. IGMP is automatically enabled on an interface when you configure PIM or DVMRP on an interface and is disabled on the interface if you disable PIM or DVMRP on the interface.
- IGMPv3 does not support static IGMP group members.
- Static IGMP groups are supported only in Layer 3 mode.
- Since VLANs are not VRF-aware, any changes to default-vlan or tagged port moves is counted by all VRFs in existence at the time, including the default VRF.

## Common diagnostic scenarios

- With flow-control and IGMP enabled, performance is considerably slower than expected.

  This was caused in this case because the device was running old code. The problem disappeared when the code version was updated.

- UDP multicasts sent to the IGMP client are dropped.

  In this instance, Dell Technical Support worked with the customer to make some configuration changes to help consistently force the traffic through the RP. Some VRRP-E priorities were also changed to make the vrrp-e router the master router for the server aggregation area. This cleared the problem.

- Sporadic difficulty in accessing web management console.

  This issue disappeared once the customer upgraded to more recent code, and disabled the following IGMP settings:

  ```
  no ip igmp snooping
  no ip igmp snooping querier
  ```

- Even though multicast function is not configured, LP and MP CPU usage is high when the NetIron MLXe receives a large number of invalid IGMP packets.

  The following solutions were proposed to avoid this issue:

  Enter the following commands to deny IGMP packets:

  ```
  PowerConnect#ip receive access-list 100 sequence 5
  PowerConnect#access-list 100 deny igmp any any
  PowerConnect#access-list 100 permit ip any any
  ```

  Control packets can be flooded to the router and cause high CPU because the CPU must process these packets.PowerConnect B-MLXe devices use IP receive ACL (firewall) to filter these packets.

  The following commands to deny any IP fragment packets and prevent excessive LP and MP CPU usage:

  ```
  ip receive access-list 100 sequence 5
  access-list 100 deny ip any any fragment
  access-list 100 permit ip any any
  ```

  The router will bind this rACL to all interfaces.

- Old software versions.

  Feature issues are often be caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# Multicast traffic reduction

When you enable IP Multicast Traffic Reduction, you also can configure the following features:

- **IGMP mode** – When you enable IP Multicast Traffic Reduction, the device passively listens for IGMP Group Membership reports by default. If the multicast domain does not have a router to send IGMP queries to elicit these Group Membership reports, you can enable the device to actively send the IGMP queries.
- **Query interval** – The query interval specifies how often the device sends Group Membership queries. This query interval applies only to the active IGMP mode. The default is 60 seconds. You can change the interval to a value from 10 through 600 seconds.
- **Age interval** – The age interval specifies how long an IGMP group can remain in the IGMP group table without the device receiving a Group Membership report for the group. If the age interval expires before the device receives another Group Membership report for the group, the device removes the entry from the table. The default is 140 seconds. You can change the interval to a value from 10 through 1220 seconds.

Furthermore, when you enable IP Multicast Traffic Reduction, the device forwards all IP multicast traffic by default but you can enable the device to do the following:

- Forward IP multicast traffic only for groups for which the device has received a Group Membership report.
- Drop traffic for all other groups.

**NOTE**
If the “route-only” feature is enabled on the PowerConnect B-MLXe device, then IP Multicast Traffic Reduction will not be supported. This feature is also not supported on the default VLAN of the PowerConnect B-MLXe device.

To verify that IP Multicast Traffic Reduction is enabled, enter the **show ip multicast** command at any level of the CLI:

## Multicast show commands

### show ip multicast

**Syntax: show ip multicast**

This command displays IP multicast traffic reduction information as shown in this example:

```
PowerConnect#show ip multicast
IP multicast is enabled - Passive
IP pimsm snooping is enabled

VLAN ID 23
Active 10.10.10.10 Report ports: 1/1 7/1
Report FID 0X0400
Number of Multicast Groups: 2

1     Group: 225.1.0.291
      IGMP report ports  :
      Mapped mac address : 0100.5e01.001d Fid:0x041b
      PIMv2*G join ports : 1/1

2     Group: 225.1.0.24
      IGMP report ports : 4/48
      Mapped mac address : 0100.5e01.0018 Fid:0x041a
      PIMv2*G join ports : 1/1
```

### show ip multicast pimsm-snooping

**Syntax: show ip multicast pimsm-snooping**

This command displays PIM SM information, as shown in the following example:

```
PowerConnect(config)#show ip multicast pimsm-snooping
PIMSM snooping is enabled
VLAN ID 100
  PIMSM neighbor list:
          31.31.31.4 : 12/2 expires 142 s
         31.31.31.13 : 10/7 expires 136 s
          31.31.31.2 : 3/1 expires 172 s
Number of Multicast Groups: 2
1    Group: 239.255.162.4 Num SG 4
     Forwarding ports : 3/1 12/2
     PIMv2 *G join ports : 3/1 12/2
     1     Source: (165.165.165.165, 10/7) FID 0x0bb3
           SG join ports: 12/2 10/7
     2     Source: (161.161.161.161, 10/7) FID 0x0bb2
           SG join ports: 12/2 3/1
     3     Source: (158
.158.158.158, 10/7) FID 0x0bb1
           SG join ports: 12/2 3/1
     4     Source: (170.170.170.170, 10/7) FID 0x0baf
           SG join ports: 3/1 10/7
           (S, G) age 0 s
2    Group: 239.255.163.2 Num SG 1
     Forwarding ports : 10/7 12/2
     PIMv2 *G join ports : 10/7 12/2
     1     Source: (165.165.165.165, 3/1) FID 0x0bb5
            SG join ports: 12/2 10/7
```

### show ip multicast statistics

**Syntax: show ip multicast statistics**

This command displays IP multicast statistics, as shown in the following example, which shows statistics for two port-based VLANs:

```
PowerConnect#show ip multicast statistics
IP multicast is enabled - Passive

VLAN ID 1
Reports Received:               34
Leaves Received:                21
General Queries Received:       60
Group Specific Queries Received: 2
Others Received:                 0
General Queries Sent:            0
Group Specific Queries Sent:     0

VLAN ID 2
Reports Received:                0
Leaves Received:                 0
General Queries Received:       60
Group Specific Queries Received: 2
Others Received:                 0
General Queries Sent:            0
Group Specific Queries Sent:     0
```

## Clearing IP multicast statistics

To clear IP multicast statistics on a device, enter the following command at the Privileged EXEC level of the CLI:

**clear ip multicast statistics**

**Syntax: clear ip multicast** [**all** | **group** <*group-id*>]

- **all** - Clears the learned flows for all groups.
- **group** <*group-id*> - Clears the flows for the specified group but does not clear the flows for other groups.

This command resets statistics counters for all the statistics displayed by the **show ip multicast statistics** command to zero.

The following example shows IGMP flows information listed by the **show ip multicast** command, followed by removal of the information by the **clear ip multicast all** command.

```
PowerConnect#show ip multicast
IP multicast is enabled - Active
VLAN ID 1
Active 192.168.2.30 Router Ports  4/13
Multicast Group: 239.255.162.5, Port: 4/4 4/13
Multicast Group: 239.255.162.4, Port: 4/10 4/13

PowerConnect#clear ip multicast all

PowerConnect#show ip multicast
IP multicast is enabled - Active
VLAN ID 1
Active 192.168.2.30 Router Ports  4/13
```

To clear the learned IGMP flows for a specific IP multicast group, enter a command similar to the following:

**clear ip multicast group**

**Syntax: clear ip multicast group** <*group address*>

The following example shows how to clear the IGMP flows for a specific group and retain reports for other groups:

```
PowerConnect#show ip multicast
IP multicast is enabled - Active
VLAN ID 1
Active 192.168.2.30 Router Ports  4/13
Multicast Group: 239.255.162.5, Port: 4/4 4/13
Multicast Group: 239.255.162.4, Port: 4/10 4/13

PowerConnect#clear ip multicast group 239.255.162.5
PowerConnect#show ip multicast
IP multicast is enabled - Active
VLAN ID 1
Active 192.168.2.30 Router Ports  4/13
Multicast Group: 239.255.162.4, Port: 4/10 4/13
```

## Configuration notes

- You must enter the **ip multicast-routing** command before changing the global IP Multicast parameters. Otherwise, the changes do not take effect and the software uses the default values. Also, entering **no ip multicast-routing** will reset all parameters to their default values.

## Common diagnostic scenarios

- Multicast is not working.

  In this incident, route-only was configured on the device. VLANs and route-only will not work together. Removing the route-only setting eliminated the problem.

- High CPU usage with Multicast.

  In this incident, multicast packets were hitting the CPU with code 149, which indicates an RPF check failure. The customer was asked to confirm the route to the source (route incorrectly configured).

- Multicast packets are not being transferred between two PowerConnect B-MLXe devices.

  In this incidence, the two PowerConnect B-MLXe devices had network address overlap. The problem resolved with the network address on one of the devices was changed.

- Old software versions.

  Feature issues are often be caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# MSDP

The Multicast Source Discovery Protocol (MSDP) is used by Protocol Independent Multicast (PIM) Sparse routers to exchange routing information for PIM Sparse multicast groups across PIM Sparse domains. Routers running MSDP can discover PIM Sparse sources that are in other PIM Sparse domains.

## MSDP show commands

You can display the following MSDP information using show commands:

- Summary information – The IP addresses of the peers, the state of the PowerConnect B-MLXe device router's MSDP session with each peer, and statistics for Keepalive, Source Active, and Notification messages sent to and received from each of the peers.
- Peer information – The IP address of the peer, along with detailed MSDP and TCP statistics.
- Source Active cache entries – The Source Active messages cached by the router.

### show ip msdp summary

**Syntax: show ip msdp summary**

This command displays summary MSDP information, as shown in the following example:

```
PowerConnect#show ip msdp summary

MSDP Peer Status Summary
 KA: Keepalive SA:Source-Active NOT: Notification
 Peer Address     State         KA          SA          NOT
                              In     Out   In    Out   In   Out
 206.251.17.30    ESTABLISH   3     3     0     640   0     0
 206.251.17.41    ESTABLISH   0     3     651   0     0     0
```

### show ip msdp peer

**Syntax:** show ip msdp peer

This command displays MSDP peer information, as shown in this example:

```
PowerConnect#show ip msdp peer

        Total number of MSDP Peers: 2

        IP Address           State
1       206.251.17.30        ESTABLISHED
        Keep Alive Time  Hold Time
        60               90

                             Message Sent         Message Received
        Keep Alive           2                    3
        Notifications        0                    0
        Source-Active        0                    640
        Last Connection Reset Reason:Reason Unknown
        Notification Message Error Code Received:Unspecified
        Notification Message Error SubCode Received:Not Applicable
        Notification Message Error Code Transmitted:Unspecified
        Notification Message Error SubCode Transmitted:Not Applicable
        TCP Connection state: ESTABLISHED
          Local host:  206.251.17.29, Local  Port: 8270
          Remote host: 206.251.17.30, Remote Port: 639
          ISentSeq:      16927  SendNext:     685654  TotUnAck:          0
          SendWnd:       16384  TotSent:      668727  ReTrans:           1
          IRcvSeq:    45252428  RcvNext:    45252438  RcvWnd:        16384
          TotalRcv:         10  RcvQue:            0  SendQue:           0
```

### show ip msdp sa-cache

**Syntax:** show ip msdp sa-cache

This command displays the Source Actives in the MSDP cache, as shown in this example:

```
PowerConnect#show ip msdp sa-cache
Total of 10 SA cache entries
Index  RP address       (Source, Group)                    Orig Peer        Age
1      2.2.2.2          (192.6.1.10, 227.1.1.1)            192.1.1.2        0
2      2.2.2.2          (192.6.1.10, 227.1.1.2)            192.1.1.2        0
3      2.2.2.2          (192.6.1.10, 227.1.1.3)            192.1.1.2        0
4      2.2.2.2          (192.6.1.10, 227.1.1.4)            192.1.1.2        0
5      2.2.2.2          (192.6.1.10, 227.1.1.5)            192.1.1.2        0
6      2.2.2.2          (192.6.1.10, 227.1.1.6)            192.1.1.2        0
7      2.2.2.2          (192.6.1.10, 227.1.1.7)            192.1.1.2        0
8      2.2.2.2          (192.6.1.10, 227.1.1.8)            192.1.1.2        0
9      2.2.2.2          (192.6.1.10, 227.1.1.9)            192.1.1.2        0
10     2.2.2.2          (192.6.1.10, 227.1.1.10)           192.1.1.2        0
```

### show ip msdp debug

**Syntax:** show ip msdp debug

This command displays information about internal MSDP activity, such as number of peers, timer settings, internal clock ticks, and source-active (SA) cache memory pool data. Output resembles the following:

```
PowerConnect#show ip msdp debug
[BEGIN] MSDP Debug Info
Oper is On
Max # of peers 1, # of peers 1
Srvr IP 0.0.0.0, sockInit Yes
Orig-id 0/1/0/0.0.0.0, SA filter-orig No, orig-rmap ""/00000000
entry-per-ticks 3200, adv-int 60, adv-entry-per-tick 35, state-ticks 533, start-
offset 10
SA agetime 6, Holddown 75, KA 60, hold-timer 90, conn-rety 30
SA Cache memory Pool Information:
pool: 25da9600, unit_size: 24, initial_number:256, upper_limit:32000
  total_number:256, allocated_number:10, alloc_failure 0
  flag: 0, pool_index:1, avail_data:25ef60f0
[END] MSDP Debug Info
```

## MSDP debug commands

### debug ip msdp

**Syntax:** [**no**] **debug ip msdp** [**alarms** | **events** | **message**]

- **alarms** - Displays information about MSDP alarms.
- **events** - Displays information about MSDP events.
- **message** - Displays information about MSDP messages.

The **debug ip msdp** command generates information about Multicast Source Discovery Protocol (MSDP) alarms, events, and messages.

### debug ip msdp alarms

**Syntax:** [no] **debug ip msdp alarms**

This command generates information about MSDP RX processing errors, such as invalid headers or incomplete or truncated information, errors during transmission of SA advertisement transmission, (for example buffer unavailability), and peer connection socket errors and notification messages. Output resembles the following:

```
PowerConnect#debug ip msdp alarms
         MSDP:  alarms debugging is on
MSDP: S=xxxxxxx P=0 Initiate Transport Connection to MSDP peer
```

### debug ip msdp events

**Syntax:** [no] **debug ip msdp events**

This command tracks originating SA advertisements, major peer events, and peer keepalive timer events. Output resembles the following:

```
PowerConnect#debug ip msdp events
MSDP: events debugging is on
MSDP: 192.1.1.2: Process START event, local = 192.1.1.2
MSDP: 192.1.1.2: TCP Connection to Remote Peer is Open
MSDP: 192.1.1.2: MSDP-TCP Connection opened
MSDP: 192.1.1.2: TCP_OPEN DONE, State 4
MSDP: 192.1.1.2: Originating SA
MSDP: 192.1.1.2: TX Keep Alive timer expired, send keep alive to peer
MSDP: 192.1.1.2: Originating SA
MSDP: 192.1.1.2: TX Keep Alive timer expired, send keep alive to peer
```

### debug ip msdp message

**Syntax:** [no] **debug ip msdp message**

This command generates information (including message contents) about MSDP messages received, transmitted and forwarded, and flags errors in MSDP messages. Output resembles the following:

```
PowerConnect#debug ip msdp message
MSDP: 192.1.1.2: Xmt SA
RP 1.1.1.1, SA count 10
(172.1.2.10,226.1.1.1) (172.1.2.10,226.1.1.2)
(172.1.2.10,226.1.1.3) (172.1.2.10,226.1.1.4)
(172.1.2.10,226.1.1.5) (172.1.2.10,226.1.1.6)
(172.1.2.10,226.1.1.7) (172.1.2.10,226.1.1.8)
MSDP: 192.1.1.2: State=4, Rcv SA
RP 2.2.2.2, SA count 10
(192.2.2.10,225.1.1.1) (192.2.2.10,225.1.1.2)
(192.2.2.10,225.1.1.3) (192.2.2.10,225.1.1.4)
(192.2.2.10,225.1.1.5) (192.2.2.10,225.1.1.6)
(192.2.2.10,225.1.1.7) (192.2.2.10,225.1.1.8)
MSDP: 192.1.1.2: State=4, Rcv KA
```

## Clearing MSDP information

You can clear the following MSDP information:

- Peer information
- Source Active cache
- MSDP statistics

To clear MSDP peer information, enter the following command at the Privileged EXEC level of the CLI:

**clear ip msdp peer**

**Syntax: clear ip msdp peer** <*ip-addr*>

The command displays a message to indicate when the connection has been successfully closed.To clear all the peers, omit the <*ip-addr*> variable from the command.

To clear the Source Active cache, enter the following command at the Privileged EXEC level of the CLI:

**clear ip msdp sa-cache**

**Syntax: clear ip msdp sa-cache** <*ip-addr*>

The command shown above clears all of the cache entries. Use the <*ip-addr*> variable to clear only the entries matching either a source or a group.

To clear MSDP statistics, enter the following command at the Privileged EXEC level of the CLI:

**clear ip msdp statistics**

**Syntax: clear ip msdp statistics** [<*ip-addr*>]

The command shown above clears statistics for all the peers. To clear statistics for only a specific peer, enter the peer's IP address using the variable <*ip-addr*>.

## Configuration notes

- MSDP depends on BGP and MBGP for inter-domain operations.
- Routers that run MSDP usually also run BGP. The source address used by the MSDP router is normally configured to be the same source address used by BGP.
- For MSDP mesh groups, on each device that will be part of the mesh group, there must be a mesh group definition for all the peers in the mesh-group.
- It is recommended that you use the connect-source loopback <num> parameter when issuing the **msdp-peer** command. If you do not use this parameter, the PowerConnect B-MLXe device router uses the outgoing interface's IP address. You should also make sure the IP address of the connect-source loopback is the source IP address used by the PIM-RP, and the BGP router.

## Common diagnostic scenarios

- High CPU usage with MSDP traffic between two peers.

  This issue may be resolved by resetting both peers.
- PIM/SM Multicast packets are not being successfully transmitted.

This was caused by a device downstream that had multicast passive and pim snooping enabled. Once PIM snooping was disabled on the downstream device, and added on MBGP links, the problem was resolved.

- Unable to remove an MSDP peer.

  This problem was resolved when the customer upgraded the software.

- Unable to remove the ttl threshold command or an MSDP peer that was somehow mistakenly entered.

  An outage at the customer data center probably triggered this issue. After the customer reloaded the system, they were able to remove the ttl threshold.

- Old software versions.

  Feature issues are often be caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# PIM DM and PIM SM

Protocol-independent multicast (PIM) helps simplify the complexity of the routing protocol. PIM is similar to DVMRP in that PIM builds source-routed multicast delivery trees and employs reverse path check when forwarding multicast packets.

There are two PIM modes: Dense and Sparse. The Dense Mode is suitable for densely populated multicast groups, primarily in the LAN environment. The Sparse Mode is suitable for sparsely populated multicast groups with the focus on WAN.

The PowerConnect B-MLXe device supports PIM DM V1 and V2. The default is V2. You can specify the version on an individual interface basis.

The primary difference between PIM DM V1 and V2 is the methods the protocols use for messaging:

- PIM DM V1 – Uses the IGMP to send messages.
- PIM DM V2 – Sends messages to the multicast address 224.0.0.13 (ALL-PIM-ROUTERS) with protocol number 103.

The CLI commands for configuring and managing PIM DM are the same for V1 and V2. The only difference is the command you use to enable the protocol on an interface.

**NOTE**

If you want to continue to use PIM DM V1 on an interface, you must change the version, then save the configuration. However, this does not mean you can run different PIM versions on devices that are connected to each other. The devices must run the same version of PIM. To connect a PowerConnect B-MLXe device running PIM to a device that is running PIM V1, you must change the PIM version on the PowerConnect B-MLXe device to V1 (or change the version on the device to V2, if supported).

PIM was introduced to simplify some of the complexity of the routing protocol at the cost of additional overhead tied with a greater replication of forwarded multicast packets. PIM is similar to DVMRP in that PIM builds source-routed multicast delivery trees and employs reverse path check when forwarding multicast packets.

There are two modes in which PIM operates: Dense and Sparse. The Dense Mode is suitable for densely populated multicast groups, primarily in the LAN environment. The Sparse Mode is suitable for sparsely populated multicast groups with the focus on WAN.

## PIM DM and PIM SM show commands

The following sections describe show commands you can use to display PIM DM and PIM SM information.

### show ip aim counter nsr

**Syntax:** show ip pim counter nsr

This command displays Multicast NSR-related status counter from the management processor. The output resembles the following:

```
PowerConnect#show ip pim counter nsr
Mcache sync (entity id: 203)
  pack: 1078
  unpack: 0
  ack: 1078
RPset sync (entity id: 201)
  pack: 3
  unpack: 0
  ack: 3
BSR status (entity id: 202)
  pack: 3
  unpack: 0
  ack: 3
```

### show ip pim counter

**Syntax:** show ip pim counter

This command displays the PIM counters from the line processor as shown in the following example:

```
PowerConnect#show ip pim counter
Forward:
  Packets : 41649 Regsters: 0
Drops:
  RPF-Fail: 0 No-RP   : 0    IfMsmtch: 0
  OIFEmpty: 0 InvldIf : 0    TTLXpire: 0
  NoFwEntr: 0 TrkMove : 0    PortMove: 0
  NoCause : 0 FwEntrFl: 0    ResFail : 0
  SSMNoEnt: 0
IPC:
  MCCreate: 0 MCFirDta: 8000 SGAgeOut: 4000
  Register: 0 WGFirDta: 0    SGAbvThr: 4000
  WrongIf : 0
  NoSGFDta: 0
ISSU:
MVID Save: BufferFull(0); Num Travrse(4000);Num Savd(4000);Num MVID(1)
MVID Restore: Num Travrse(4000);Num Rest(4000);Num MVID(1)
MVID Restore: Invalid MVID(0);InvalidNumOif(0);MVID Resrv Fail(0)
CAM Restore: Total CAMs(4000);Unique CAMs(4000);Inv.VRF(0);
Inv.fwd.entry(0);Dup.CAM(0)
```

### show ip pim neighbor

**Syntax: show ip pim neighbor**

This command displays PIM neighbor states, as shown in the following example:

```
PowerConnect#show ip pim neighbor
--------+--------+---------------+--------+-----+----------------+-----------+---
Port     PhyPort  Neighbor        Holdtime Age   UpTime            VRF        Prio
                                  sec      sec
--------+--------+---------------+--------+-----+----------------+-----------+---
e1/22    e1/22    43.5.1.1        105      0     00:04:40         default-vrf 1
v46      e3/4     56.5.1.1        105      10    00:04:30         default-vrf 1
v59      e1/1     111.10.10.1     105      0     00:04:40         default-vrf 1
v67      e1/1     61.3.1.1        105      0     00:04:40         default-vrf 1
v420     e1/1     121.100.100.1   105      0     00:04:40         default-vrf 1
```

### show ip pim sparse

**Syntax: show ip pim sparse**

This command displays global PIM SM configuration information, as shown in the following example:

```
PowerConnect#show ip pim sparse
Global PIM Sparse Mode Settings
  Hello interval        : 30          Neighbor timeout                   : 105
  Bootstrap Msg interval: 60          Candidate-RP Advertisement interval: 60
  Join/Prune interval   : 60          SPT Threshold                      : 1
  Inactivity interval   : 180
  SSM Enabled: Yes
  SSM Group Range: 226.0.0.0/8
---------+----------------+----+---+----------------------+------
Interface|Local           |Mode|Ver|  Designated Router   |TTL
         |Address         |    |   |Address          Port |Thresh
---------+----------------+----+---+----------------------+------
     e6/3 10.2.0.10         SM  V2  Itself                     1
     e6/4 10.3.0.10         SM  V2  Itself                     1
      v30 10.10.10.10       SM  V2  Itself                     1
```

### show ip pim group

**Syntax: show ip pim group**

This command displays PIM Sparse configuration information, as shown in this example:

```
PowerConnect#show ip pim group

Total number of Groups: 2
Index 1         Group 239.255.162.1       Ports e3/11
```

### show ip pim bsr

**Syntax: show ip pim bsr**

This command displays PIM bootstrap router (BSR) information, as shown in this example:

```
PowerConnect#show ip pim bsr
PIMv2 Bootstrap information
This system is the elected Bootstrap Router (BSR)
  BSR address: 207.95.7.1
  Uptime: 00:33:52, BSR priority: 5, Hash mask length: 32
  Next bootstrap message in 00:00:20

Next Candidate-RP-advertisement in 00:00:10
  RP: 207.95.7.1
    group prefixes:
    224.0.0.0 / 4
Candidate-RP-advertisement period: 60
```

This example shows information for a router that has been elected as the BSR. The following example shows information for a router that is not the BSR. Notice that some fields shown in the first example do not appear in the second example.

```
PowerConnect#show ip pim bsr

PIMv2 Bootstrap information
 BSR address  = 207.95.7.1
 BSR priority = 5
```

### show ip pim rp-candidate

**Syntax: show ip pim rp-candidate**

This command displays candidate RP information, as shown in the following example:

```
PowerConnect#show ip pim rp-candidate

Next Candidate-RP-advertisement in 00:00:10
  RP: 207.95.7.1
    group prefixes:
    224.0.0.0 / 4

  Candidate-RP-advertisement period: 60
```

This example shows information displayed on a candidate RP router. The following example shows the message displayed for a non-candidate RP router.

```
PowerConnect#show ip pim rp-candidate
This system is not a Candidate-RP.
```

### show ip pim rp-map

**Syntax: show ip pim rp-map**

This command displays RP-to-group-mappings, as shown in the following example:

```
PowerConnect#show ip pim rp-map
Number of group-to-RP mappings: 6

Group address        RP address
------------------------------
1 239.255.163.1  99.99.99.5
2 239.255.163.2  99.99.99.5
3 239.255.163.3  99.99.99.5
4 239.255.162.1  99.99.99.5
5 239.255.162.2  43.43.43.1
6 239.255.162.3  99.99.99.5
```

### show ip pim rp-hash

**Syntax: show ip pim rp-hash** <*group-address*>

The <*group-address*> parameter is the address of a PIM Sparse IP multicast group.

This command displays RP hash information for a PIM Sparse group (identified by the <*group-address*> variable), as shown in this example:

```
PowerConnect#show ip pim rp-hash 239.255.162.1

  RP: 207.95.7.1, v2
    Info source: 207.95.7.1, via bootstrap
```

### show ip pim rp-set

**Syntax: show ip pim rp-set**

To display the RP set list, enter this command at any CLI level:

```
PowerConnect#show ip pim rp-set
Group address Static-RP-address Override
--------------------------------------------------
Access-List 44 99.99.99.5 On
Number of group prefixes Learnt from BSR: 1
Group prefix = 239.255.162.0/24 # RPs expected: 1
# RPs received: 1
RP 1: 43.43.43.1 priority=0 age=0
```

### show ip pim nbr

**Syntax: show ip pim nbr**

This command displays information about PIM neighbors, enter the following command at any CLI level:

```
PowerConnect#show ip pim nbr

Port   Neighbor        Holdtime   Age     UpTime
                       sec        sec     sec
e3/8   207.95.8.10     180        60      900
Port   Neighbor        Holdtime   Age     UpTime
                       sec        sec     sec
v1     207.95.6.2      180        60      900
```

### show ip pim mcache

**Syntax:** show ip pim mcache

This command displays information about the PIM multicast cache, as shown in the following example:

```
PowerConnect#show ip pim mcache
Total 3 entries
1 (10.161.32.200, 237.0.0.1) in v87 (tag e3/1), cnt=0
   Sparse Mode, RPT=0 SPT=1 Reg=0
   upstream neighbor=10.10.8.45
   num_oifs = 1 v2
   L3 (HW) 1: e4/24(VL2702)
   fast=1 slow=0 leaf=0 prun=0 frag=0 tag=0 tnnl=0 swL2=0 hwL2=0 msdp_adv=0
   age=0 fid: 0416 l2vidx: none
2 (*, 237.0.0.1) RP10.161.2.1 in v93, cnt=0
   Sparse Mode, RPT=1 SPT=0 Reg=0
   upstream neighbor=10.10.8.33
   num_oifs = 1 v2
   L3 (SW) 1: e4/24(VL2702)
   fast=1 slow=0 leaf=0 prun=0 frag=0 tag=0 tnnl=0 swL2=0 hwL2=0 msdp_adv=0
   age=0 fid: none l2vidx: none
3 (*, 239.255.255.250) RP10.159.2.2 in v87, cnt=0
   Sparse Mode, RPT=1 SPT=0 Reg=0
   upstream neighbor=10.10.8.45
   num_oifs = 1 v2
   L3 (SW) 1: e4/23(VL2702)
   fast=1 slow=0 leaf=0 prun=0 frag=0 tag=0 tnnl=0 swL2=0 hwL2=0 msdp_adv=0
   age=0 fid: none l2vidx: none
```

### show ip pim traffic

**Syntax:** show ip pim traffic

This command displays PIM traffic statistics, as shown in the following example:

```
PowerConnect#show ip pim traffic

Port    Hello           J/P          Register       RegStop          Assert
     [Rx     Tx]    [Rx     Tx]    [Rx     Tx]    [Rx     Tx]    [Rx     Tx]
e3/8   19      19      32      0       0       0       37      0       0       0
v1     18      19      0       20      0       0       0       0       0       0
v2     0       19      0       0       0       16      0       0       0       0

Total 37       57      32      0       0       0       0       0       0       0
IGMP Statistics:
  Total Recv/Xmit 85/110
  Total Discard/chksum  0/0
```

**NOTE**

If you have configured interfaces for standard PIM (dense mode) on your router, statistics for these interfaces are listed first in the display.

## Clearing the PIM forwarding cache

You can clear the PIM forwarding cache using the following command.

**clear pim-cache**

**Syntax: clear pim-cache**

## PIM SM debug commands

There are no debug commands specific to PIM SM, however, there are debug commands for PIM-DVMRP, as described in the following sections.

**debug ip pim-dvmrp**

**Syntax:** [**no**] **debug ip pim-dvmrp** [**add-del-oif** | **bootstrap** | **clear** | **event** | **group** | **ipc** | **join-prune** | **level** | **nbr-change** | **route-change** | **show** | **source** | **vlan-id** | **vpls-id**]

- **add-del-oif** - Displays mcache additions or deletions.
- **bootstrap** - Displays bootstrap messages in detail.
- **clear** - Clears PIM-DVMRP debug settings.
- **event** - Displays information about infrastructure events and callback handling.
- **group** - Displays activity for a specific group.
- **ipc** - Displays information about IPC messages between management processor and line processor.
- **join-prune** - Displays information about join/prune messages.
- **level** - Sets level of debug information from 1 through 3 (3 generates the most detailed information).
- **nbr-change** - Displays information about neighbor port changes.
- **route-change** - Displays information about route change events.
- **show** - Shows ip PIM-DVMRP debug settings.
- **source** - Displays information about multicast traffic from a specific source.
- **vlan-id** - Displays information a specified VLAN.
- **vpls-id** - Displays information about a specific VPLS ID.

**debug ip pim-dvmrp add-del-oif**

**Syntax:** [**no**] **debug ip pim-dvmrp add-del-oif** [**stack**]

This command monitors and displays instances of mcache activity, such as OIF (outbound interface) additions or deletions. When the stack option is selected, this command also generates a stack trace of the add or delete event. Output is similar to the following:

```
PowerConnect#debug ip pim-dvmrp add-del-oif
Added oif v10, e2/1 to (10.10.10.2, 224.225.0.1) entry
```

This example indicates that vlan 10 on port 2/1 has been added to the OIF table for multicast stream (10.10.10.2, 224.225.0.1).

**debug ip pim-dvmrp clear**

**Syntax:** [**no**] **debug ip pim-dvmrp clear**

This command clears all PIM-DVMRP debug settings.

### debug ip pim-dvmrp ipc

**Syntax:** [no] **debug ip pim-dvmrp ipc**

This command displays IPC messages between the management processor and a line processor. Output is similar to the following, which Indicates a line processor notification has been enabled for stream (10.10.10.1, 224.225.0.1). This stream originates from port e2/1 on vlan 10.

```
PowerConnect#debug ip pim-dvmrp ipc
receive slave messages S_G_CREAT_NOTIF,entry(10.10.10.1, 224.255.0.1)intf v10,
e2/1
```

### debug ip pim-dvmrp join-prune

**Syntax:** [no] **debug ip pim-dvmrp join-prune**

This command displays information about join/prune activity. Output resembles the following:

```
PowerConnect#debug ip pim-dvmrp join-prune
PIMDM: Rx Join/Prune from 30.30.30.1, on intf v10, e2/1. RPF Addr 20.20.20.1. ToME
1
```

This indicates that a join or prune message has been received from 30.30.30.1, on Ethernet port 2/1, vlan 10 for RPF Addr 20.20.20.1. ToME means this message should be processed.

### debug ip pim-dvmrp level

**Syntax:** [no] **debug ip pim-dvmrp level**

This command sets the level of detail for debug output. Levels range from 0 through 3, with 3 being the most detailed. Levels currently supported are:

- **Level 0** - Receive input or send output messages
- **Level 1** - Process control message

### debug ip pim-dvmrp nbr-change

**Syntax:** [no] **debug ip pim-dvmrp nbr-change**

This command displays information about neighbor port changes. Output resembles the following:

```
PowerConnect#debug ip pim-dvmrp nbr-change
nbr 30.30.30.1 phy change from e2/1 to e3/1
```

This output indicates that neighbor 30.30.30.1 has changed from port e2/1 to port e3/1.

### debug ip pim-dvmrp show

**Syntax:** [no] **debug ip pim-dvmrp show**

This command displays current debug settings for PIM-DVMRP. For example:

```
PowerConnect#debug ip pim-dvmrp show
debug ip pim is enabled
```

## Configuration notes

The following limitations apply to implementation of PIM Sparse:

- PIM Border Routers (PMBRs) are not supported. Thus, you cannot configure a routing interface as a PMBR interface for PIM Sparse.
- PIM Sparse and regular PIM (dense mode) cannot be used on the same interface.
- You cannot configure or display PIM Sparse information using the Web management interface. (You can display some general PIM information, but not specific PIM Sparse information.)

- If you want to continue to use PIM DM V1 on an interface, you must change the version, then save the configuration. This doesn't mean you can run different PIM versions on devices that are connected to each other. The devices must run the same version of PIM. If you want to connect a device running PIM to a device that is running PIM V1, you must change the PIM version on the device to V1 (or change the version on the device to V2, if supported).
- When PIM routing is enabled, the line rate for receive traffic is reduced by about 5%. The reduction occurs due to overhead from the VLAN multicasting feature, which PIM routing uses. This behavior is normal and does not indicate a problem with the device.
- You do not need to globally enable IP multicast routing when configuring PIM Sparse.
- It is recommended that you configure the same device as both the BSR and the RP.
- It is possible to configure the device as only a candidate BSR or RP, but it is recommended that you configure the same interface on the same device as both a BSR and an RP.
- Specify the same IP address as the RP on all PIM Sparse routers within the PIM Sparse domain. Make sure the router is on the backbone or is otherwise well connected to the rest of the network.
- Use the same Join/Prune message interval on all the PIM Sparse routers in the PIM Sparse domain. If the routers do not all use the same timer interval, the performance of PIM Sparse can be adversely affected.

## Common diagnostic scenarios

- The following PIM Sparse error appears: ERROR - cannot enable PIM sparse or DVMRP on an interface.

  This occurred because the customer had not updated their software to reflect the most current patch version. Once the software was updated, the problem resolved.

- Specific multicast sources are not being forwarded properly.

  In this instance, the customer's dmzo router was directed to an incorrect rpf. After modifying the configuration, the problem resolved.

- Enabling IP PIM-SPARSE caused high CPU usage.

  This can occur when the device runs out of CAM due to large amounts of multicast traffic flooding the CAM. Make sure the device is using the most current patch of the software, and contact Dell Technical Support for assistance.

- Old software versions.

  Feature issues are often be caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

Chapter

# Security Diagnostics 9

This chapter describes diagnostic information for security environments on PowerConnect B-MLXe routers.

## 802.1x

802.1x port security allows you to configure a PowerConnect B-MLXe device to grant access to a port based on information supplied by a client to an authentication server.

This section describes how to use Dell show commands and debug commands to monitor 802.1x configurations and activity on PowerConnect B-MLXe routers.

**CAUTION**

**Enabling diagnostic commands may degrade system performance. These commands are best used to troubleshoot specific problems while working with qualified Dell service technicians. Whenever possible, troubleshoot your system during periods of low network traffic and user activity to preserve system performance.**

**NOTE**

To save space, date and time stamps have been removed from all output examples.

### 802.1x show commands

You can display the following 802.1x-related information:

- Information about the 802.1x configuration on the device and on individual ports
- Statistics about the EAPOL frames passing through the device
- Information about 802.1x-enabled ports dynamically assigned to a VLAN
- Information about the user-defined and dynamically applied Mac address and IP ACLs currently active on the device
- Information about the 802.1x multiple client configuration

### show dot1x

**Syntax: show dot1x**

This command displays information about the 802.1x configuration, as shown in the following example:

```
PowerConnect#show dot1x
PAE Capability               : Authenticator Only
system-auth-control          : Enable
Number of ports enabled      : 25
re-authentication            : Disable
global-filter-strict-security: Enable
quiet-period                 : 60 Seconds
tx-period                    : 30 Seconds
supptimeout                  : 30 Seconds
servertimeout                : 30 Seconds
maxreq                       : 3
re-authperiod                : 3600 Seconds
Protocol Version             : 1
auth-fail-action             : Block Traffic
MAC Session Aging            : All
MAC Session Max Age          : 120 Seconds
Maximum Failed Attempts      : 3
```

### show dot1x config ethernet

**Syntax: show dot1x ethernet** <*slotnum*>/<*portnum*>

To display information about the 802.1x configuration on an individual port, enter a command similar to the following:

```
PowerConnect#show dot1x config ethernet 1/3

Port 1/3 Configuration:
AuthControlledPortControl    : Auto
max-clients                  : 32
multiple-clients             : Enable
filter-strict-security       : Enable
```

### show dot1x statistics

**Syntax: show dot1x statistics** [**all** | **ethernet** <*slotnum*>/<*portnum*>]

This command displays 802.1x statistics for all ports, or for a specified port. The following example shows the output from this command when issued for a specific port:

```
PowerConnect#show dot1x statistics ethernet 3/3

Port 1/3 Statistics:
RX EAPOL Start:                 0
RX EAPOL Logoff:                0
RX EAPOL Invalid:               0
RX EAPOL Total:                 2
RX EAP Resp/Id:                 1
RX EAP Resp other than Resp/Id: 1
RX EAP Length Error:            0
Last EAPOL Version:             1
Last EAPOL Source:              0050.da0b.8bef
TX EAPOL Total:                 3
TX EAP Req/Id:                  1
TX EAP Req other than Req/Id:   1
Num Sessions:                   1
Num Restricted Sessions:        0
Num Authorized Sessions:        1
```

### show interface

#### Syntax: show interface

The **show interface** command displays the VLAN to which an 802.1x-enabled port has been dynamically assigned, as well as the port from which it was moved (that is, the port's default VLAN).

The following example indicates the port's dynamically assigned VLAN. Information about the dynamically assigned VLAN is shown in bold type. In this example, the 802.1x-enabled port has been moved from VLAN 1 to VLAN 4094. When the client disconnects, the port will be moved back to VLAN 1.

```
PowerConnect#show interface ethernet 12/2
GigabitEthernet1/3 is up, line protocol is up
  Hardware is GigabitEthernet, address is 000c.dbe2.5800 (bia 000c.dbe2.5800)
  Configured speed auto, actual 100Mbit, configured duplex fdx, actual fdx
  Configured mdi mode AUTO, actual MDIX
  Member of L2 VLAN ID 4094 (dot1x-RADIUS assigned), original L2 VLAN ID is 1,
  port is untagged, port state is Forwarding
  STP configured to ON, Priority is level0, flow control enabled
  Force-DSCP disabled
  mirror disabled, monitor disabled
  Not member of any active trunks
  Not member of any configured trunks
  No port name
  Internet address is 12.12.12.250/24, MTU 1522 bytes, encapsulation ethernet
  300 second input rate: 810 bits/sec, 0 packets/sec, 0.00% utilization
  300 second output rate: 1253 bits/sec, 1 packets/sec, 0.00% utilization
  70178 packets input, 7148796 bytes, 0 no buffer
  Received 0 broadcasts, 0 multicasts, 70178 unicasts
  0 input errors, 0 CRC, 0 frame, 0 ignored
  0 runts, 0 giants, DMA received 70178 packets
  91892 packets output, 10081165 bytes, 0 underruns
  Transmitted 9853 broadcasts, 13330 multicasts, 68709 unicasts
  0 output errors, 0 collisions, DMA transmitted 91892 packets
```

### show dot1x mac-address

**Syntax:** **show dot1x mac-address-filter** [**all** | **ethernet** <*slotnum*>/<*portnum*>][**begin** <*expression*> | **exclude** <*expression*> | **include** <*expression*>]

The **all** keyword displays all dynamically applied MAC address filters active on the device.

Use the **ethernet** <*slotnum*>/<*portnum*> parameter to display information for one port.

This command displays information about MAC filters. If you specify a specific port, and If the MAC address filter is dynamically assigned by 802.1x, the output resembles the following example:

```
PowerConnect#show dot1x mac-address-filter ethernet 1/1
Port 1/1 MAC Address Filter information:
  802.1x dynamic MAC Filter (user defined) :
    mac access-list 401 in
  Port default MAC Filter :
    mac access-list 400 in
```

"Port default MAC Filter" appears if a default MAC filter has been configured on the port. This is the filter that will be applied to the port once the dynamically assigned MAC filter is removed. If a default MAC filter has not been configured, the message "No Port default MAC" is displayed.

When the dynamically assigned MAC address filter is removed, output resembles the following example:

```
PowerConnect#show dot1x mac-address ethernet 1/1
Port 1/1 MAC Address Filter information:
  Port default MAC Filter :
    mac access-list 400 in
```

### show dot1x ip-acl

**Syntax:** **show dot1x ip-acl** [**all** | **ethernet** <*slotnum*>/<*portnum*>] [**begin** <*expression*> | **exclude** <*expression*> | **include** <*expression*>]

- **all** - Displays all dynamically applied IP ACLs active on the device.
- **ethernet** <*slotnum*>/<*portnum*> - Displays information for one port.

This command displays information about what IP ACLs have been applied to an 802.1x-enabled port. If an IP ACL was dynamically applied by 802.1x, output from this command resembles this example:

```
PowerConnect#show dot1x ip-acl ethernet 1/1
Port 1/1 IP ACL information:
  802.1x dynamic IP ACL (user defined) in:
    ip access-list extended Port_1/1_E_IN in
  Port default IP ACL in:
    ip access-list 100 in
  No outbound ip access-list is set
```

"Port default IP ACL" appears if a default IP ACL has been configured on the port. This is the IP ACL that will be applied to the port once the dynamically assigned IP ACL is removed. If a default IP ACL has not been configured, the message "No Port default IP ACL" is displayed.

When the dynamically assigned IP ACL is removed from the port, the display shows the following information:

```
PowerConnect#show dot1x ip-acl ethernet 1/1
Port 1/1 IP ACL information:
  Port default IP ACL in:
    ip access-list 100 in
  No outbound ip access-list is set
```

### show dot1x mac-session

**Syntax: show dot1x mac-session** [**brief** | [**begin** <*expression*> | **exclude** <*expression*> | **include** <*expression*>]]

This command displays information about the dot1x MAC sessions on all ports, as shown in the following example:

```
PowerConnect#show dot1x mac-session
Port  MAC             Username                      VLAN Auth State ACL|MAC  Age
                                                                    i|o|f
-------------------------------------------------------------------------------
1/1   0050.da0b.8cd7  Mary M                           1    DENIED  n|n|n   0
1/2   0050.da0b.8cb3  adminmorn                     4094 PERMITTED  y|n|n   0
1/3   0050.da0b.8bef  reports                       4094 PERMITTED  y|n|n   0
1/4   0010.5a1f.6a63  testgroup                     4094 PERMITTED  y|n|n   0
1/5   0050.da1a.ff7e  admineve                      4094 PERMITTED  y|n|n   0
```

### show dot1x mac-session brief

**Syntax: show dot1x mac-session brief**

This command displays information about the ports in an 802.1x multiple client configuration, as shown in the following example:

```
PowerConnect#show dot1x mac-session brief
Port             Number of users      Dynamic Dynamic     Dynamic
          Restricted Authorized Total VLAN    ACL (In/Out)MAC-Filt
---------+----------+----------+-----+-------+-----------+--------
1/1               0          0     1 no          no/no   no
1/2               0          1     1 yes        yes/no   no
1/3               0          1     1 yes        yes/no   no
1/4               0          1     1 yes        yes/no   no
1/5               0          1     1 yes        yes/no   no
```

## Clearing 802.1x statistics

You can clear the 802.1x statistics counters on all interfaces at once, on individual interfaces, or on a range of interfaces.

For example, to clear the 802.1x statistics counters on all interfaces on the device, enter the following command.

### clear dot1x statistics all

**Syntax: clear dot1x statistics all**

Use the following command to clear dot1x statistics for a specified MAC address or a specified port.

**clear dot1x statistics**

**Syntax: clear dot1x statistics** [<*mac-address*> | **ethernet** <*slotnum*>/<*portnum*>]

## 802.1x debug commands

**debug dot1x**

**Syntax:** [**no**] **debug dot1x** [**all** | **dumpclass** | **events** | **fault** | **packets** | **port** | **state** | **timers**]

This command displays information about 802.1x authentication events, activity, and settings.

- **all** - Displays general information about dot1x activity for all ports.
- **dumpclass** - Displays internal data structure.
- **events** - Displays significant events for all ports.
- **fault** - Displays any internal errors for all ports.
- **packets** - Displays information about 802.1x packets.
- **port** - Displays information about 802.1 events and timers for specified ports.
- **state** - Displays 802.1x port state information.
- **timers** - Displays 802.1x timer information.

**debug dot1x all**

**Syntax:** [**no**] **debug dot1x all**

This is an extremely useful command that provides a complete profile of the authentication process, including events, faults, timers and packets. This command works globally across all ports. Output resembles the following:

```
PowerConnect#debug dot1x all
802.1X All debugging ON
Dec 21 17:11:48 : 802.1X: Timer tick expired
: 802.1X 3/16 Tx EAPOL on vlan_id 1 for dst 0180.c200.0003, len 8:
EAP PACKET - EAPCode: FAILURE
dump TX 802.1X: 8 bytes|
01000004 04000004                       ........
: 802.1X: port 3/16 Tx (OK) EAPOL-FAIL Pkt (EAPId: 0)
: 802.1X: Port 3/16 txEAP timer expired. Transmitting an EAP ReqId
: 802.1X 3/16 Tx EAPOL on vlan_id 1 for dst 0180.c200.0003, len 9:
EAP PACKET - EAPCode: REQUEST  EAPType: IDENTITY
dump TX 802.1X: 9 bytes
01000005 01000005 010000fc              .........
: 802.1X: port 3/16 Tx (OK) EAPOL-EAP-REQUEST-ID Pkt (EAPId: 0)
: 802.1X: Timer tick expired
: 802.1X: Timer tick expired
: 802.1X: Timer tick expired
: 802.1X: Timer tick expired
: 802.1X: port 4/1 Rx EAPOL Pkt SA=0000.00aa.0001, DA=0180.c200.0003, len=0
EAP START   Dec 21 17:12:02   dump RX 802.1X: 18 bytes
0180c200 00030000 00aa0001 888e0101     ................
000071ae                                ..
: 802.1X: port 4/1 Rx EAPOL_START
: 802.1X: port 4/1 BkEnd BKEND_INVALID --> BKEND_INIT
: 802.1X: port 4/1 BkEnd BKEND_INIT --> BKEND_IDLE
: 802.1X: port 4/1 AuthPAE AUTH_INVALID --> AUTH_INIT
: 802.1X: port 4/1 AuthPAE AUTH_INIT --> AUTH_DISCONCTED
: 802.1X: port 4/1 AuthPAE AUTH_DISCONCTED --> AUTH_CONNTING
: 802.1X 4/1 Tx EAPOL on vlan_id 1 for dst 0000.00aa.0001, len 9:
EAP PACKET - EAPCode: REQUEST  EAPType: IDENTITY
dump TX 802.1X: 9 bytes
01000005 01010005 01011607              .........
: 802.1X: port 4/1 Tx (OK) EAPOL-EAP-REQUEST-ID Pkt (EAPId: 1)
: 802.1X 4/1 Tx EAPOL on vlan_id 1 for dst 0000.00aa.0001, len 9:
EAP PACKET - EAPCode: REQUEST  EAPType: IDENTITY
dump TX 802.1X: 9 bytes
01000005 01010005 01011607              .........
: 802.1X: port 4/1 Tx (OK) EAPOL-EAP-REQUEST-ID Pkt (EAPId: 1)
: 802.1X: port 4/1 Rx EAPOL Pkt SA=0000.00aa.0001, DA=0180.c200.0003, len=6
EAP PACKET - EAPCode: RESPONSE  EAPType: IDENTITY
dump RX 802.1X: 24 bytes
0180c200 00030000 00aa0001 888e0100     ................
00060201 00060161                       .......a
: 802.1X: port 4/1 Rx EAPOL-EAP-RESPONSE-ID Pkt (EAPId: 1)
: 802.1X: port 4/1 AuthPAE AUTH_CONNTING --> AUTH_ENTCATING
: 802.1X: port 4/1 BkEnd BKEND_IDLE --> BKEND_RESPONSE
: 802.1X: port 4/1 aWhile timer (AuthServer) started for 30 secs
: 802.1X: port 4/1 Tx EAP PDU (EAPId: 1) to AuthServer
: 802.1X: port 4/1 Rx EAPOL Pkt SA=0000.00aa.0001, DA=0180.c200.0003, len=6
EAP PACKET - EAPCode: RESPONSE  EAPType: IDENTITY
```

```
dump RX 802.1X: 24 bytes
0180c200 00030000 00aa0001 888e0100    ................
00060201 00060161                      .......a
: 802.1X: port 4/1 Rx EAPOL-EAP-RESPONSE-ID Pkt (EAPId: 1)
: 802.1X: port 4/1 Rx AAA_INTERACTIVE from AuthServer
: 802.1X: port 4/1 Rx from Server: EAP-REQUEST-MD5 (EAPId: 2) Len 22
: 802.1X 4/1 Tx EAPOL on vlan_id 1 for dst 0000.00aa.0001, len 26:
EAP PACKET - EAPCode: REQUEST  EAPType: MD5
dump TX 802.1X: 26 bytes
01000016 01020016 04102fb9 32b79134    ........../.2..4
17b5ea71 89b9eb79 b42b0b04             ...q...y.+
: 802.1X: port 4/1 BkEnd BKEND_RESPONSE --> BKEND_REQUEST
: 802.1X: port 4/1 Rx AAA_INTERACTIVE from AuthServer
: 802.1X: port 4/1 Rx from Server: EAP-REQUEST-MD5 (EAPId: 2) Len 22
: 802.1X 4/1 Tx EAPOL on vlan_id 1 for dst 0000.00aa.0001, len 26:
EAP PACKET - EAPCode: REQUEST  EAPType: MD5
dump TX 802.1X: 26 bytes
01000016 01020016 04102fb9 32b79134    ........../.2..4
17b5ea71 89b9eb79 b42b0b04             ...q...y.+
: 802.1X: port 4/1 BkEnd BKEND_RESPONSE --> BKEND_REQUEST
: 802.1X: port 4/1 Fwd (OK) EAPOL-EAP Pkt (EAPId: 2) from AuthServer to
Supplicant
: 802.1X: port 4/1 aWhile timer (Supplicant) started for 30 secs
: 802.1X: port 4/1 Rx EAPOL Pkt SA=0000.00aa.0001, DA=0180.c200.0003, len=24
EAP PACKET - EAPCode: RESPONSE  EAPType: MD5
dump RX 802.1X: 42 bytes
0180c200 00030000 00aa0001 888e0100    ................
00180202 00180410 0462366a 6cb18e4a    .........b6jl..J
e739af16 80a64756 61000000             .9....GVa.
: 802.1X: port 4/1 Rx EAP-RESPONSE-MD5 Pkt (EAPId: 2) Len 24
: 802.1X: port 4/1 BkEnd BKEND_REQUEST --> BKEND_RESPONSE
: 802.1X: port 4/1 aWhile timer (AuthServer) started for 30 secs
: 802.1X: port 4/1 Tx EAP PDU (EAPId: 2) to AuthServer
: 802.1X: port 4/1 Rx AAA_ACCEPT from AuthServer
: 802.1X: Port 4/1 Created user-defined mac filter 400.
: 802.1X: Port 4/1 Binding with the RADIUS assigned MAC ACL ID: 400
: 802.1X: port 4/1 Rx Tunnel Data (Type=0, Medium_Type=0, PvtGrpId=NULL)
: 802.1X 4/1 Tx EAPOL on vlan_id 1 for dst 0000.00aa.0001, len 8:
EAP PACKET - EAPCode: SUCCESS
dump TX 802.1X: 8 bytes
01000004 03020004                      ........
: 802.1X: port 4/1 Tx (OK) EAPOL-SUCCESS Pkt (EAPId: 2)
: 802.1X: port 4/1 BkEnd BKEND_RESPONSE -->
BKEND_SUCCESS
: 802.1X: port 4/1 BkEnd BKEND_SUCCESS --> BKEND_IDLE
: 802.1X: port 4/1 AuthPAE AUTH_ENTCATING --> AUTH_ENTCATED
: 802.1X: Port 4/1 Programming Permitted MAC 0000.00aa.0001 on VLAN 1
: 802.1X: Timer tick expired
: 802.1X: Timer tick expired
```

### debug dot1x dumpclass

**Syntax:** [no] **debug dot1x dumpclass**

This command dumps internal data structure. Output from this command is primarily useful to Dell technicians.

```
PowerConnect# debug dot1x dumpclass
DOT1X Class: 0x2085e280
 Flags:  EnabAll: 0 ReAuthEnab/ReAuthMax: 0/3 MaxReq: 3 SysAuthEnab: 0
         AuthFailAction/Vlanid: Restricted/99 Aging/Age: All/120
 Timers: SecHold: 60 Quiet: 60 TxWhen: 30 ReAuth: 3600 SuppTmO: 30 SrvrTmO: 30
```

### debug dot1x events

**Syntax:** [no] **debug dot1x events**

This command displays authentications that have failed or succeeded, the application of VLAN/ACLs requested by RADIUS, etc. This command works globally across all ports. Output resembles the following:

```
PowerConnect#debug dot1x event
        events:  debugging is on
: 802.1X: port 4/1 Rx EAPOL_START
: 802.1X: port 4/1 BkEnd BKEND_INVALID --> BKEND_INIT
: 802.1X: port 4/1 BkEnd BKEND_INIT --> BKEND_IDLE
: 802.1X: port 4/1 AuthPAE AUTH_INVALID --> AUTH_INIT
: 802.1X: port 4/1 AuthPAE AUTH_INIT --> AUTH_DISCONCTED
: 802.1X: port 4/1 AuthPAE AUTH_DISCONCTED --> AUTH_CONNTING
: 802.1X: port 4/1 Tx (OK) EAPOL-EAP-REQUEST-ID Pkt (EAPId: 1)
: 802.1X: port 4/1 Tx (OK) EAPOL-EAP-REQUEST-ID Pkt (EAPId: 1)
: 802.1X: port 4/1 Rx EAPOL-EAP-RESPONSE-ID Pkt (EAPId: 1)
: 802.1X: port 4/1 AuthPAE AUTH_CONNTING --> AUTH_ENTCATING
: 802.1X: port 4/1 BkEnd BKEND_IDLE --> BKEND_RESPONSE
: 802.1X: port 4/1 Tx EAP PDU (EAPId: 1) to AuthServer
: 802.1X: port 4/1 Rx EAPOL-EAP-RESPONSE-ID Pkt (EAPId: 1)
: 802.1X: port 4/1 Rx AAA_INTERACTIVE from AuthServer
: 802.1X: port 4/1 Rx from Server: EAP-REQUEST-MD5 (EAPId: 2) Len 22
: 802.1X: port 4/1 BkEnd BKEND_RESPONSE --> BKEND_REQUEST
: 802.1X: port 4/1 Fwd (OK) EAPOL-EAP Pkt (EAPId: 2) from AuthServer to
Supplicant
: 802.1X: port 4/1 Rx EAP-RESPONSE-MD5 Pkt (EAPId: 2) Len 24
: 802.1X: port 4/1 BkEnd BKEND_REQUEST --> BKEND_RESPONSE
: 802.1X: port 4/1 Tx EAP PDU (EAPId: 2) to AuthServer
: 802.1X: port 4/1 Rx AAA_ACCEPT from AuthServer
: 802.1X: Port 4/1 Created user-defined mac filter 400.
: 802.1X: Port 4/1 Binding with the RADIUS assigned MAC ACL ID: 400
: 802.1X: port 4/1 Rx Tunnel Data (Type=0, Medium_Type=0, PvtGrpId=NULL)
: 802.1X: port 4/1 Tx (OK) EAPOL-SUCCESS Pkt (EAPId: 2)
: 802.1X: port 4/1 BkEnd BKEND_RESPONSE --> BKEND_SUCCESS
: 802.1X: port 4/1 BkEnd BKEND_SUCCESS --> BKEND_IDLE
: 802.1X: port 4/1 AuthPAE AUTH_ENTCATING --> AUTH_ENTCATED
: 802.1X: Port 4/1 Programming Permitted MAC 0000.00aa.0001 on VLAN 1
: 802.1X: port 4/1 Rx EAPOL_LOGOFF
: 802.1X: port 4/1 AuthPAE AUTH_ENTCATED --> AUTH_DISCONCTED
: 802.1X: Port 4/1 Deleting Permitted MAC 0000.00aa.0001 on VLAN 1
: 802.1X: Port 4/1 Unbinding with the dynamic assigned L2 ACL: 400
: 802.1X: Port 4/1 Deleting the user defined L2 ACL: 400
: 802.1X: port 4/1 Tx (OK) EAPOL-FAIL Pkt (EAPId: 3)
: 802.1X: port 4/1 Tx (OK) EAPOL-EAP-REQUEST-ID Pkt (EAPId: 0)
```

### debug dot1x fault

**Syntax:** [no] **debug dot1x fault**

This command reports any kind of internal errors, such as out-of-memory; invalid RADIUS-response, etc. This command works globally across all ports. Output resembles the following:

```
PowerConnect#debug dot1x fault
fault:  debugging is on
: 802.1X: Port 4/1 Preprocess of user-defined L2-ACL
failed due to invalid ACL
: 802.1X: Port 4/1 Preprocess of user-defined L2-ACL failed due to invalid ACL
: 802.1X: Port 4/1 Preprocess of user-defined L2-ACL failed due to invalid ACL
: 802.1X: Port 4/1 Preprocess of user-defined L2-ACL failed due to invalid ACL
: 802.1X: Port 4/1 Preprocess of user-defined L2-ACL failed due to invalid ACL
```

### debug dot1x packets

**Syntax:** [no] **debug dot1x packets**

This command displays information about 802.1x packets. Output resembles the following:

```
PowerConnect#debug dot1x packets
    packets:  debugging is on
: 802.1X: port 4/1 Rx EAPOL Pkt SA=0000.00aa.0001, DA=0180.c200.0003, len=0
EAP START   Dec 21 17:26:04   dump RX 802.1X: 18 bytes
0180c200 00030000 00aa0001 888e0101    ................
00004ef5                               ..
: 802.1X 4/1 Tx EAPOL on vlan_id 1 for dst 0000.00aa.0001, len 9:
EAP PACKET - EAPCode: REQUEST  EAPType: IDENTITY
dump TX 802.1X: 9 bytes
01000005 01010005 01000000             .........
802.1X 4/1 Tx EAPOL on vlan_id 1 for dst 0000.00aa.0001, len 9:
EAP PACKET - EAPCode: REQUEST  EAPType: IDENTITY
dump TX 802.1X: 9 bytes
01000005 01010005 01000000             .........
: 802.1X: port 4/1 Rx EAPOL Pkt SA=0000.00aa.0001, DA=0180.c200.0003, len=6
EAP PACKET - EAPCode: RESPONSE  EAPType: IDENTITY
dump RX 802.1X: 24 bytes
0180c200 00030000 00aa0001 888e0100    ................
: 802.1X: port 4/1 Rx EAPOL Pkt SA=0000.00aa.0001, DA=0180.c200.0003, len=6
```

### debug dot1x port

**Syntax:** [no] **debug dot1x port** [**all** | **event** | **timer**] <*slotnum*>/<*portnum*>

- **all** - Displays 802.1x event and timer information for a specified port.
- **event** - Displays 802.1x event information for a specified port.
- **timer** - Displays 802.1x timer settings for a specified port.

This command displays events and timer information for a specified port, as shown in the following example:

```
PowerConnect#debug dot1x port all 3/15
802.1X Events debugging on port 94 ON
802.1X Timers debugging on port 94 ON

SYSLOG: Nov 7 17:39:22:<12>MLX_4K, DOT1X: Port 3/15, MAC Address 0002.b3cd.5f4e
Access: unauthenticated
: 802.1X: port 3/15 Rx EAPOL-EAP-RESPONSE-ID Pkt (EAPId: 1)
```

```
: 802.1X: port 3/15 AuthPAE AUTH_CONNTING --> AUTH_ENTCATING
: 802.1X: port 3/15 BkEnd BKEND_IDLE --> BKEND_RESPONSE
: 802.1X: port 3/15 aWhile timer (AuthServer) started for 35 secs
: 802.1X: port 3/15 Tx EAP PDU (EAPId: 1) to AuthServer
: 802.1X: port 3/15 Rx AAA_INTERACTIVE from AuthServer
: 802.1X: port 3/15 Rx from Server: EAP-REQUEST-PEAP(EAPId: 237)Len 6
: 802.1X: port 3/15 BkEnd BKEND_RESPONSE --> BKEND_REQUEST
: 802.1X: port 3/15 Fwd(OK) EAPOL-EAP Pkt (EAPId:237) from AuthServer to
Supplicant
: 802.1X: port 3/15 aWhile timer (Supplicant) started for 30 secs
: 802.1X: port 3/15 Rx EAP-RESPONSE-NAK Pkt (EAPId: 237) Len 6
: 802.1X: port 3/15 BkEnd BKEND_REQUEST --> BKEND_RESPONSE
: 802.1X: port 3/15 aWhile timer (AuthServer) started for 35 secs
: 802.1X: port 3/15 Tx EAP PDU (EAPId: 237) to AuthServer
: 802.1X: port 3/15 Rx AAA_INTERACTIVE from AuthServer
: 802.1X: port 3/15 Rx from Server: EAP-REQUEST-MD5 (EAPId: 238) Len 30
: 802.1X: port 3/15 BkEnd BKEND_RESPONSE --> BKEND_REQUEST
: 802.1X: port 3/15 Fwd(OK) EAPOL-EAP Pkt (EAPId: 238) from AuthServer to
Supplicant
: 802.1X: port 3/15 aWhile timer (Supplicant) started for 30 secs
: 802.1X: port 3/15 Rx EAP-RESPONSE-MD5 Pkt (EAPId: 238) Len 23
: 802.1X: port 3/15 BkEnd BKEND_REQUEST --> BKEND_RESPONSE
: 802.1X: port 3/15 aWhile timer (AuthServer) started for 35 secs
: 802.1X: port 3/15 Tx EAP PDU (EAPId: 238) to AuthServer
: 802.1X: port 3/15 Rx AAA_ACCEPT from AuthServer
: 802.1X: port 3/15 Tx (OK) EAPOL-SUCCESS Pkt (EAPId: 238)
: 802.1X: port 3/15 BkEnd BKEND_RESPONSE --> BKEND_SUCCESS
: 802.1X: port 3/15 BkEnd BKEND_SUCCESS --> BKEND_IDLE
: 802.1X: port 3/15 AuthPAE AUTH_ENTCATING --> AUTH_ENTCATED
: SYSLOG: Nov 7 17:39:22:<14>MLX_4K, DOT1X: Port 3/15, MAC Address 0002.b3cd.5f4e
802.1X: Port 3/15 Programming Permitted MAC 0002.b3cd.5f4e on VLAN 1
Access: authorized
```

### debug dot1x state

**Syntax:** [no] **debug dot1x state** <*slotnum*>/<*portnum*>

This command displays information about the 802.1x state of a specified port. Output resembles the following:

```
PowerConnect#debug dot1x state 4/1
Port 4/1. #Sess 0 #RestSess 0 #AuthSess 0. #DeniedSess 0
DfACLIn 0 DfACLOut 0 DfL2ACL 0. DfVLAN 4096 InVLAN 4096(UserCfg)
Flags pEnab pCtrl reAut ACLSS MClnt |X| Aging RvtVl DynVl OvRst DoSPr
1    AUTO   0     1     1      None     0     1     1     0
Timers TxEAP 23. EAPTries: 0
```

### debug dot1x timers

**Syntax:** [no] **debug dot1x timers**

This command enables debugging, and displays information about 802.1x timers. Output resembles the following:

```
PowerConnect#debug dot1x timers
timers:  debugging is on
: 802.1X: Timer tick expired
: 802.1X: Timer tick expired
: 802.1X: Timer tick expired
: 802.1X: Timer tick expired
: 802.1X: Port 4/1 txEAP timer expired. Transmitting an EAP ReqId
: 802.1X: Timer tick expired
```

```
: 802.1X: Timer tick expired
: 802.1X: Timer tick expired
: 802.1X: port 4/1 aWhile timer (Supplicant) started for 30 secs
: 802.1X: port 4/1 aWhile timer (AuthServer) started for 30 secs
: 802.1X: port 4/1 aWhile timer (AuthServer) started for 30 secs
: 802.1X: Timer tick expired
: 802.1X: port 4/1 aWhile timer (Supplicant) started for 30 secs
: 802.1X: port 4/1 aWhile timer (AuthServer) started for 30 secs
: 802.1X: Timer tick expired
: 802.1X: Port 3/15 txEAP timer expired. Transmitting an EAP ReqId
: 802.1X: Port 3/17 txEAP timer expired. Transmitting an EAP ReqId
: 802.1X: Port 3/18 txEAP timer expired. Transmitting an EAP ReqId
: 802.1X: Port 3/19 txEAP timer expired. Transmitting an EAP ReqId
: 802.1X: Port 3/20 txEAP timer expired. Transmitting an EAP ReqId
: 802.1X: port 4/1 aWhile timer (AuthServer) started for 30 secs
: 802.1X: port 4/1 aWhile timer (Supplicant) started for 30 secs
: 802.1X: port 4/1 aWhile timer (AuthServer) started for 30 secs
: 802.1X: Timer tick expired
: 802.1X: port 4/1 aWhile timer (AuthServer) started for 30 secs
: 802.1X: port 4/1 aWhile timer (Supplicant) started for 30 secs
: 802.1X: port 4/1 aWhile timer (AuthServer) started for 30 secs
: 802.1X: Timer tick expired
```

## Configuration notes

- The client's dot1x-mac-session establishes a relationship between the username and MAC address used for authentication. If a user attempts to gain access from different clients (with different MAC addresses), he or she would need to be authenticated from each client.
- If a client has been denied access to the network (that is, the client's dot1x-mac-session is set to "access-denied"), then you can cause the client to be re-authenticated by manually disconnecting the client from the network, or by using the **clear dot1x mac-session** command.
- When a client has been denied access to the network, its dot1x-mac-session is aged out if no traffic is received from the client's MAC address over a fixed hardware aging period (70 seconds), plus a configurable software aging period. You can optionally change the software aging period for dot1x-mac-sessions or disable aging altogether. After the denied client's dot1x-mac-session is aged out, traffic from that client is no longer blocked, and the client can be re-authenticated.
- To implement 802.1x port security, at least one of the RADIUS servers identified to the PowerConnect B-MLXe device must support the 802.1x standard.

## Common diagnostic scenarios

- Old software versions.

  Feature issues are often caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# Denial of Service attacks

In a Denial of Service (DoS) attack, a router is flooded with useless packets, hindering normal operation. Dell devices include measures for defending against two types of DoS attacks: Smurf attacks and TCP SYN attacks.

A Smurf attack is a kind of DoS attack where an attacker causes a victim to be flooded with ICMP echo (Ping) replies sent from another (intermediary) network. For detailed information about how to prevent Smurf attacks, see the *NetIron Series Configuration Guide.*

## DoS show commands

### show statistics dos-attack

**Syntax:** **show statistics dos-attack** [**begin** <*expression*> | **exclude** <*expression*> | **include** <*expression*>]

This command displays information about ICMP and TCP SYN packets dropped, passed, and blocked because burst thresholds were exceeded. Output resembles the following example:

```
PowerConnect# show statistics dos-attack
------ Local Attack Statistics ------
ICMP Drop Count     Port Block Count    SYN Drop Count    SYN Block Count
---------------     ----------------    --------------    ---------------
              0                   0                 0                  0
```

## Clearing DoS attack statistics

To clear statistics about ICMP and TCP SYN packets dropped because burst thresholds were exceeded, enter the following command.

### clear statistics dos-attack

**Syntax:** clear statistics dos-attack

## DoS debug commands

There are no debug commands specific to DoS attacks.

## Configuration notes

For detailed information about configuring to prevent DoS attacks, see the *NetIron Series Configuration Guide.*

## Common diagnostic scenarios

The following sections describe how to avoid various types of DoS attacks. For more information, see the *NetIron Series Configuration Guide*.

### *Avoiding being an intermediary in a Smurf attack*

To avoid being an ***intermediary*** in a Smurf attack, make sure that the forwarding of directed broadcasts is disabled on your PowerConnect B-MLXe device. Directed broadcast forwarding is disabled by default. To disable directed broadcast forwarding, do one of the following:

#### Using the CLI

```
PowerConnect(config)#no ip directed-broadcast
```

#### Using the Web Management Interface

1. Log on to the device with a valid user name and password for read-write access. The System configuration panel is displayed.
2. Click the plus sign (+) next to Configure in the tree view to see the list of configuration options.
3. Click the plus sign (+) next to IP to see the list of IP configuration options.
4. Select the General link to display the IP configuration panel.
5. Select Disable next to Directed Broadcast Forward.
6. Click the Apply button to save the change to the running-config file.
7. Select the Save link at the bottom of the dialog box. Select Yes when prompted to save the configuration change to the startup-config file on the device flash memory.

### *Avoiding being a victim in a Smurf attack*

You can configure the PowerConnect B-MLXe device to drop ICMP packets when it encounters excessive numbers, as is the case when the device is the *victim* of a Smurf attack. You can set threshold values for ICMP packets that are targeted at the router itself or passing through an interface, and drop them when the thresholds are exceeded.

#### ip icmp burst-normal

**Syntax: ip icmp burst-normal** <*value*> **burst-max** <*value*> **lockup** <*seconds*>

- The **burst-normal** <*value*> can be from 1 through 100000.
- The **burst-max** <*value*> can be from 1 through 100000.
- The**lockup** <*seconds*> can be from 1 through 10000.

For example, to set threshold values for ICMP packets targeted at the router, enter the following command in CONFIG mode:

```
PowerConnect(config)#ip icmp burst-normal 5000 burst-max 10000 lockup 300
```

To set threshold values for ICMP packets received on interface 3/11, enter:

```
PowerConnect(config)#interface ethernet 3/11
PowerConnect(config-if-e100-3/11)#ip icmp burst-normal 5000 burst-max 10000
lockup 300
```

This command is supported on Ethernet, POS, and Layer 3 ATM interfaces.

The number of incoming ICMP packets per second is measured and compared to the threshold values as follows:

- If the number of ICMP packets exceeds the **burst-normal** value, the excess ICMP packets are dropped.
- If the number of ICMP packets exceeds the **burst-normal** value, all ICMP packets are dropped for the number of seconds specified by the **lockup** value. When the lockup period expires, the packet counter is reset and measurement is restarted.

In this example, if the number of ICMP packets received per second exceeds 5,000, the excess packets are dropped. If the number of ICMP packets received per second exceeds 10,000, the device drops all ICMP packets for the next 300 seconds (5 minutes).

## *Protecting against TCP SYN attacks*

In a TCP SYN attack, an attacker floods a host with TCP SYN packets that have random source IP addresses. For each of these TCP SYN packets, the destination host responds with a SYN ACK packet and adds information to the connection queue. However, since the source host does not exist, no ACK packet is sent back to the destination host, and an entry remains in the connection queue until it ages out (after around a minute). If the attacker sends enough TCP SYN packets, the connection queue can fill up, and service can be denied to legitimate TCP connections.

To protect against TCP SYN attacks, you can configure your device to drop TCP SYN packets when excessive numbers are encountered. You can set threshold values for TCP SYN packets that are targeted at the router itself or passing through an interface, and drop them when the thresholds are exceeded.

### ip tcp burst-normal

**Syntax: ip tcp burst-normal** <*value*> **burst-max** <*value*> **lockup** <*seconds*>

- The **burst-normal** <*value*> can be from 1 through 100000.
- The **burst-max** <*value*> can be from 1 through 100000.
- The**lockup** <*seconds*> value can be from 1 through 10000.

The number of incoming TCP SYN packets per second is measured and compared to the threshold values as follows:

- If the number of TCP SYN packets exceeds the **burst-normal** value, the excess TCP SYN packets are dropped.
- If the number of TCP SYN packets exceeds the **burst-normal** value, all TCP SYN packets are dropped for the number of seconds specified by the **lockup** value. When the lockup period expires, the packet counter is reset and measurement is restarted.

  For example, to set threshold values for TCP SYN packets targeted at the router, enter the following command in CONFIG mode:

  ```
  PowerConnect(config)#ip tcp burst-normal 10 burst-max 100 lockup 300
  ```

To set threshold values for TCP SYN packets received on interface 3/11, enter:

```
PowerConnect(config)#interface ethernet 3/11
PowerConnect(config-if-e100-3/11)#ip tcp burst-normal 10 burst-max 100 lockup 300
```

In this example, if the number of TCP SYN packets received per second exceeds 10, the excess packets are dropped. If the number of TCP SYN packets received per second exceeds 100, the device drops all TCP SYN packets for the next 300 seconds (5 minutes).

# HTTPS Web management access

Web management is disabled by default. You can enable it through HTTP or HTTPS using the following command. For details, see the *NetIron Series Configuration Guide.*

**web-management hp-top-tools**

**Syntax:** [no] **web-management hp-top-tools**

The **hp-top-tools** parameter disables TCP port 280.

## HTTPS show commands

There are no show commands directly related to HTTPS.

## HTTPS debug commands

There are no debug commands specific to HTTPS configurations.

## Configuration notes

- To enable Web management through HTTPS, enter the **web-management https** command. In addition, you must generate a crypto SSL certificate or import digital certificates issued by a third-party Certificate Authority (CA).
- By default, TCP port 80 is enabled on the Dell device. TCP port 80 (HTTP) allows access to the Web management interface on the device.
- By default, TCP port 280 for HP Top tools is disabled. This port allows access to the device by HP ProCurve Manager.
- The **no web-management** command disables both TCP ports. However, if you want to disable only port 280 and leave port 80 enabled, use the **hp-top-tools** option with the command.

## Common diagnostic scenarios

- Old software versions.

  Feature issues are often caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# Port loop detection

The Port Loop Detection protocol allows the Dell device to detect loop and disable a port that is on the receiving end of a loop. Loop is detected by sending test packets BPDU.

## Port loop detection show command

### show loop-detection

**Syntax:** show loop-detection

This command to display the loop detection configuration and current status as shown in the following:

```
PowerConnect#show loop-detection
loop detection packets interval: 10 (unit 100 msec)
loop detection disable duration: 10 (In minutes, 0 means permanently disabled)
Ports mode loop detection
=========================
port-num disable-count
1/12 0
1/11 0
Vlan mode loop detection
========================
vlan-id disable-count
100 2
10 0
200 0
Ports disabled by loop detection
================================
port age(minutes) disable cause
1/11 1 Disabled by VLAN: 100 loopdetect 1/11
1/12 1 Disabled by VLAN: 100 loopdetect 1/12
```

## Port loop detection debug command

### debug loopdetect

**Syntax:** [no] **debug loopdetect** [**detail** | **error** | **info**]

This command enables the loop detect debugging.

- **detail** - Displays loop detect detail messages.
- **error** - Displays loop detect error messages.
- **info** - Displays loop detect information messages.

## Configuration notes

The following information applies to Loose Mode loop detection:

- Loop detection is configured on the VLAN. Different VLANs may disable different ports.
- Loose Mode can disable multiple ports of a loop. A disabled port affects every VLAN using it.
- Loose Mode disables the receiving port if packets originate from any port or member port of a VLAN on the same device.

- The VLAN of the receiving port must be configured for loop detection in order to disable the port.
- Loose Mode floods test packets to the entire VLAN. This can impact system performance if too many VLANs are configured for Loose Mode loop detection.

The following information applies to Strict Mode loop detection:

- A port is disabled only if a packet is looped back to that same port. Loop detection must be configured on the physical port.
- Strict Mode overcomes specific hardware issues where packets are echoed back to the input port.

# Port mirroring and monitoring

## Port mirroring show commands

### show monitor config

**Syntax: show monitor config**

This command displays the inbound and outbound traffic that is being mirrored to each mirror port, as shown in the following example:

```
PowerConnect#show monitor config
Monitored Port 3/1
    Input traffic mirrored to: 2/1
    Output traffic mirrored to: 1/1
Monitored Port 4/1
    Input traffic mirrored to: 1/2
    Output traffic mirrored to: 2/1
```

### show monitor actual

**Syntax: show monitor actual**

This command displays the actual traffic being mirrored to each mirror port, as shown in the following example:

```
PowerConnect#show monitor actual
Monitored Port 3/1
    Output traffic mirrored to: 1/1
Monitored Port 4/1
    Input traffic mirrored to: 1/2
```

This output displays the output traffic mirrored to mirror port 1/1 from port 3/1 and input traffic mirrored to mirror port 1/2 from port 4/1, which are explicitly configured.

## Port mirroring debug commands

**debug access list mirror**

**Syntax:** [**no**] **debug access list mirror**

This command displays information about generic access list mirroring activity.

## Configuration notes

The following must be considered when configuring ACL-based Inbound Mirroring:

- Configuring a common destination ACL mirror port for all ports of a PPCR.
- Support with ACL CAM sharing enabled.
- The **mirror** and **copy-sflow** keywords are mutually exclusive on a per-ACL clause basis.
- ACL-based inbound mirroring and port-based inbound mirroring are mutually exclusive on a per-port basis.

For ACL CAM sharing to function, either of the following conditions must be operative:

- All ports that belong to a PPCR must have the **acl-mirror-port** command configured to direct mirrored traffic to the same port
- None of the ports that belong to the PPCR can have the **acl-mirror-port** command configured.

### *Applying the ACL to an Interface*

You must apply the ACL to an interface using the **ip access-group** command as shown in the following example:

```
PowerConnect(config)#interface ethernet 1/1
PowerConnect(config-if-e10000-1/1)#ip access-group 101 in
```

## Common diagnostic scenarios

- Old software versions.

  Feature issues are often caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# RADIUS

You can use a Remote Authentication Dial In User Service (RADIUS) server to secure the following types of access to the PowerConnect B-MLXe device routers:

- Telnet access
- SSH access
- Web management access
- Access to the Privileged EXEC level and CONFIG levels of the CLI

**NOTE**
The PowerConnect B-MLXe device routers do not support RADIUS security for SNMP access.

## RADIUS show commands

### show aaa

**Syntax: show aaa**

This command displays information about all TACACS/TACACS+ and RADIUS servers identified on the device. For example:

```
PowerConnect#show aaa
Tacacs+ key: foundry
Tacacs+ retries: 1
Tacacs+ timeout: 15 seconds
Tacacs+ dead-time: 3 minutes
Tacacs+ Server: 207.95.6.90 Port:49:
                opens=6 closes=3 timeouts=3 errors=0
                packets in=4 packets out=4
no connection
Radius key: networks
Radius retries: 3
Radius timeout: 3 seconds
Radius dead-time: 3 minutes
Radius Server:  207.95.6.90 Auth Port=1645 Acct Port=1646:
                opens=2 closes=1 timeouts=1 errors=0
                packets in=1 packets out=4
no connection
```

### show web

**Syntax: show web**

This command displays the privilege level of Web management interface users. For example:

```
PowerConnect(config)#show web
User                              Privilege       IP address
set                                      0       192.168.1.234
```

## RADIUS debug commands

There are no RADIUS-specific debug commands.

## Configuration notes

- You must deploy at least one RADIUS server in your network.
- PowerConnect B-MLXe device routers support authentication using up to eight RADIUS servers. The device tries to use the servers in the order you add them to the device's configuration. If one RADIUS server is not responding, the PowerConnect device tries the next one in the list.

- You can select only one primary authentication method for each type of access to a device (CLI through Telnet, CLI Privileged EXEC and CONFIG levels). For example, you can select RADIUS as the primary authentication method for Telnet CLI access, but you cannot also select TACACS+ authentication as the primary method for the same type of access. However, you can configure backup authentication methods for each access type.

## Common diagnostic scenarios

- Old software versions.

  Feature issues are often caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

- Without access to RADIUS, sending an SSH session to the management module causes it to switch to the backup module.

  This problem was fixed in a patch release of the software. The customer was instructed to update his software version.

# sFlow

sFlow is a system for observing traffic flow patterns and quantities within and among a set of PowerConnect B-MLXe devices. sFlow performs the following tasks:

- Sample packet flows
- Collect the packet headers from sampled packets and collect ingress-egress information on these packets
- Compose the collected information into flow sample messages
- Relay these messages to an external device known as a collector

Participating devices also relay byte and packet counter data (counter samples) for ports to the collector.

## sFlow show commands

### show sflow

**Syntax: show sflow**

This command displays sFlow configuration information and statistics, as shown in the following example:

```
PowerConnect(config)#show sflow
sFlow services are enabled.
sFlow agent IP address: 30.30.30.2
Collector IP 10.10.10.1, UDP 6343
Polling interval is 20 seconds.
Configured default sampling rate: 1 per 2048 packets.
0 UDP packets exported
0 sFlow samples collected.
sFlow ports   Global Sample Rate   Port Sample Rate   Hardware Sample Rate
        3/1                2048               2048   2048
        3/2                2048               2048   2048
        3/3                2048               2048   2048
        3/4                2048               2048   2048
```

### *Clearing sFlow statistics*

To clear the UDP packet and sFlow sample counters in the **show sflow** display, enter the following command.

**clear statistics**

**Syntax: clear statistics**

This command clears the values in the following fields of the **show sflow** display:

- UDP packets exported
- sFlow samples collected

**NOTE**
This command also clears the statistics counters used by other features.

## sFlow debug commands

There are no debug commands specific to sFlow.

## Configuration notes

- Before you enable sFlow, make sure the device has an IP address that sFlow can use as its source address.
- It is recommended that you do not change the denominator to a value lower than the default. Sampling requires CPU resources. Using a low denominator for the sampling rate can cause high CPU utilization.
- sFlow uses CPU resources to send sFlow samples to the collector. If you set a low sampling value on a high rate interface (for example 10 GE), the Interface module CPU utilization can become high.
- If you change the router ID or other IP address value that sFlow uses for its agent_address, you need to disable and then re-enable sFlow to cause the feature to use the new source address.
- Sample data is collected from inbound traffic on ports enabled for sFlow. However, both traffic directions are counted for byte and packet counter statistics sent to the collector.
- sflow and Port Mirror cannot be enabled simultaneously on the same port.

## Common diagnostic scenarios

- Improving system performance for systems with large sFlow configurations.

  If your system is running a large sFlow configuration, is forwarding sFlow on several hundred ports, or is generating syslog messages saying the server is overloaded, you can increase the database server cache size to improve the throughput, using the following steps.

  1. Shut down IronView Network Manager.
  2. Open the startup.bat or startup.sh file in a text editor. This file is under the <install-directory>.
  3. Look for the line “start <install-directory>\bin\java” and change the “-Xmx” setting to “-XmxnnnM”. The variable nnn indicates the number of megabytes of memory the IronView Network Manager Server Java Virtual Machine should use as cache (for example, 512, 1024, etc.). The character “M” represents megabytes. However, make sure you have enough memory in the system to run other processes simultaneously without significant disk swapping by the operating system.
  4. Look for the line “start dbsrv8” and remove any “-c xxx” command line option and add the line “-cl nnnM -ch nnnM”. The variable nnnM indicates the number of megabytes of memory the database engine should use as cache (for example, 512, 1024, etc.). The character “M” represents megabytes. However, make sure you have enough memory in the system to run other processes simultaneously without significant disk swapping by the operating system.
  5. Save the file.
  6. Restart IronView Network Manager.

- Unable to see traffic reports under the accounting tab.

  Traffic reports can be generated using either snmp or sflow, and show up on the monitoring tab under service director, not accounting. Accounting is used for sflow accounting for billing information, etc.

- Old software versions.

  Feature issues are often caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# SNMP

The Simple Network Management Protocol (SNMP) forms part of the internet protocol suite as defined by the Internet Engineering Task Force (IETF). SNMP is used in network management systems to monitor network-attached devices administration and management.

SNMP is disabled by default on PowerConnect B-MLXe device routers. SNMP must be enabled in order to manage a PowerConnect B-MLXe device router using IronView Network Manager.

## SNMP show commands

### show snmp server

**Syntax: show snmp server**

This command displays both the read-only and read-write community strings in the clear.

**NOTE**
If display of the strings is encrypted, the strings are not displayed. Encryption is enabled by default.

To display the SNMP community string, enter the following commands.

```
PowerConnect(config)#enable password-display
PowerConnect(config)#show snmp server
```

The **enable password-display** command allows the community string to be displayed, but only in the output of the **show snmp server** command. Display of the string is still encrypted in the startup configuration file and running configuration. Enter the **enable password-display** command at the global CONFIG level of the CLI.

### show snmp engineid

**Syntax: show snmp engineid**

This command displays the engine ID of a management module, as shown in this example:

```
PowerConnect(config)#show snmp engineid
Local SNMP Engine ID: 800007c70300e05290ab60
Engine Boots: 3
Engine time: 5
```

### show snmp group

**Syntax: show snmp group**

This command displays the definition of an SNMP group, as shown in the following example:

```
PowerConnect(config)#show snmp group
groupname = exceptifgrp
security model = v3
security level = authNoPriv
ACL id = 2
readview = exceptif
writeview = <none>
```

## Displaying SNMP user information

**show snmp user**

**Syntax: show snmp user**

To display the definition of an SNMP user account, enter a command similar to the following:

```
PowerConnect#show snmp user
username = bob
acl id = 2
group = admin
security model = v3
group acl id = 0
authtype = md5
authkey = 3aca18d90b8d172760e2dd2e8f59b7fe
privtype = des,  privkey = 1088359afb3701730173a6332d406eec
engine ID= 800007c70300e052ab0000
```

## SNMP debug commands

There are no debug commands specific to SNMP.

## Configuration notes

- SNMP read or read-write community strings are always required for SNMP access to the device.
- SNMP access is disabled by default.
- The syntax for using ACLs for SNMP access is different from the syntax for controlling Telnet, SSH, and Web management access using ACLs.
- When **snmp-server community** is configured, all incoming SNMP packets are validated first by their community strings and then by their bound ACLs. Packets are permitted if no filters are configured for an ACL.
- If you do not enable Telnet access, you can access the CLI using a serial connection to the management module. If you do not enable SNMP access, you will not be able to use IronView Network Manager or third-party SNMP management applications.
- You cannot authenticate IronView Network Manager (SNMP) access to a PowerConnect B-MLXe device router using TACACS/TACACS+.
- PowerConnect B-MLXe device routers do not support RADIUS security for SNMP access.
- The TACACS/TACACS+, RADIUS, and Telnet login password authentication methods are not supported for SNMP access.
- For CLI access, you must configure authentication-method lists if you want the device to authenticate access using local user accounts or a RADIUS server. Otherwise, the device will authenticate using only the locally based password for the Super User privilege level.

- When no authentication-method list is configured specifically for Web management access, the device performs authentication using the SNMP community strings:
  - For read-only access, use the user name “get” and the password “public”. The default read-only community string is “public”.
  - There is no default read-write community string, which means you cannot open a read-write management session using the Web management interface. You first must configure a read-write community string using the CLI. Then you can log on using “set” as the user name and the read-write community string you configure as the password.
- If you configure an authentication-method list for Web management access and specify “local” as the primary authentication method, users who attempt to access the device using the Web management interface must supply a user name and password configured in one of the local user accounts on the device. You *cannot* access the device by entering “set” or “get” and the corresponding SNMP community string.
- For devices that can be managed using IronView Network Manager, the default authentication method (if no authentication-method list is configured for SNMP) is the CLI Super User level password. If no Super User level password is configured, then access through IronView Network Manager is not authenticated. To use local user accounts to authenticate access through IronView Network Manager, configure an authentication-method list for SNMP access and specify “local” as the primary authentication method.

## Common diagnostic scenarios

- Unable to see traffic reports.
- Traffic reports can be generated using either snmp or sflow, and the reports actually show up on the monitoring tab under service director, not accounting. Accounting is used for sflow accounting for billing information. If you use snmp, first configure an snmp collector by going into SNMP Collectors under Service Director, then click new. Name the collector, choose a Collectible for traffic, then fill out the rest of the information. Once data is collected, you can see it by going to Monitoring, locating the device you are interested in, which will have a icon under the snmp column. Click the icon to see the collected statistics.
- Old software versions.

  Feature issues are often caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# TACACS and TACACAS+

TACACS is a simple UDP-based access control protocol originally developed by BBN for MILNET. TACACS+ is an enhancement to the TACACS security protocol. TACACS+ improves on TACACS by separating the functions of authentication, authorization, and accounting (AAA) and by encrypting all traffic between the PowerConnect B-MLXe device router and the TACACS+ server.

## TACACS show commands

### show aaa

**Syntax: show aaa**

This command displays information about all TACACS+ and RADIUS servers identified on the device. For example:

```
PowerConnect#show aaa
Tacacs+ key: foundry
Tacacs+ retries: 1
Tacacs+ timeout: 15 seconds
Tacacs+ dead-time: 3 minutes
Tacacs+ Server: 207.95.6.90 Port:49:
                opens=6 closes=3 timeouts=3 errors=0
                packets in=4 packets out=4
no connection
Radius key: networks
Radius retries: 3
Radius timeout: 3 seconds
Radius dead-time: 3 minutes
Radius Server:  207.95.6.90 Auth Port=1645 Acct Port=1646:
                opens=2 closes=1 timeouts=1 errors=0
                packets in=1 packets out=4
no connection
```

### show web

**Syntax: show web**

The **show web** command displays the privilege level of Web management interface users. For example:

```
PowerConnect(config)#show web
User                              Privilege       IP address
set                                      0       192.168.1.234
```

## TACACS debug commands

### debug ip aaa

**Syntax:** [**no**] **debug ip aaa** [**event** | **ipc** | **itc** | **packet**]

- **event** - Displays information about ARP events.
- **ipc** - Displays information about ARP IPC messages.
- **itc** - Displays information about ARP ITC messages.
- **packet** - Displays information about ARP packets.

This command displays information about AAA or TACACS+ authentication, as the following example illustrates:

```
PowerConnect#debug ip aaa
COMMAND ACCOUNTING STARTS...
RADIUS accounting for context 2
Reseting RADIUS Client structure
RADIUS: Reset client 0, Total number of active clients=1

AAA: Open RADIUS UDP port

Tracing the outgoing Radius Accounting packet..
UDP packet source IP=172.20.1.2, port=1061, destination IP=172.2
0.1.1, port=1813
**Radius timer - 0 kicks in.
RADIUS retransmission for user lab, context=2, client=0
Tracing the outgoing Radius Accounting packet..
UDP packet source IP=172.20.1.2, port=1061, destination IP=172.20.1.1, port=1813
**Radius timer - 0 kicks in.
RADIUS retransmission for user lab, context=2, client=0
Tracing the outgoing Radius Accounting packet..
UDP packet source IP=172.20.1.2, port=1061, destination IP=172.20.1.1, port=1813
**Radius timer - 0 kicks in.
RADIUS timeout for user lab, context=2, client=0
RADIUS Timer cancelled for client 0.
Closing RADIUS UDP port
RADIUS: radius_authenticate_stop for client Idx 0. Actv Clients left 0
Reseting RADIUS Client structure
Accounting status - timeout.
Accounting status - reject.
aaa_send_aaa_response()..session 2, err_code=3
Unsuccessful accounting for session 2, code 3.
```

## Configuration notes

- You must deploy at least one TACACS/TACACS+ server in your network.
- The PowerConnect B-MLXe device router supports authentication using up to eight TACACS/TACACS+ servers. The device tries to use the servers in the order you add them to the device's configuration.
- You can select one primary authentication method for each type of access to a device (CLI through Telnet, CLI Privileged EXEC and CONFIG levels). For example, you can select TACACS+ as the primary authentication method for Telnet CLI access. You cannot also select RADIUS as a primary method for the same type of access, but you can configure backup authentication methods for each access application.
- You can configure the device to authenticate using a TACACS or TACACS+ server, not both.
- TACACS+ provides for authentication, authorization, and accounting, but an implementation or configuration is not required to employ all three.

## Common diagnostic scenarios

- During enable authentication with login+ password, the password checked is the login password and not the enable password configured on the TACACS+ server.

  This problem resolved when the "implicit-user" option was configured. The enable password then correctly checked against the enable password configured on the TACACS+ server.

- TACACS does not work after reboot. The key is still in config but authentication does not work till the key is removed and added back to configuration.

  This problem was resolved when the customer updated their software version.

- Old software versions.

  Feature issues are often caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# Telnet and SSH connections

The first time you log on to the Console port, you must use a serial connection in order to assign an IP address to the port. Once an IP address is assigned, you can access the CLI through a local Telnet, SSH, or SNMP connection through the management port. When accessing the CLI through Telnet, you maybe prompted for a password. By default, the password required is the one you enter for general access at initial setup.

**NOTE**
Telnet, SSH, Web, and SNMP servers are disabled by default, and can be enabled selectively.

## Telnet and SSH show commands

### show telnet

**Syntax: show telnet**

This command shows you the number of open Telnet sessions at any given time, including information about each session. Output from this command resembles the following:

```
PowerConnect#show telnet
Console connections:
       established
       3 days 17 hours 31 minutes 27 seconds in idle
Telnet server status: Enabled
Telnet connections (inbound):
1      established, client ip address 10.53.1.65, privilege super-user
       you are connecting to this session
2      closed
3      closed
4      closed
5      closed
Telnet connections (outbound):
6      established, server ip address 10.47.2.200, from Telnet session 1
       4 seconds in idle
7      closed
8      closed
9      closed
10     closed
SSH server status: Enabled
SSH connections:
1      closed
..........
```

## Telnet and SSH debug commands

### debug ip telnet

**Syntax:** [no] **debug ip telnet**

This command generates information about incoming Telnet connections, as shown in this example:

```
PowerConnect#debug ip telnet
TELNET: Data is ready to receive
TELNET: Data is ready to receive
```

### debug ip ssh

**Syntax:** [no] **debug ip ssh**

This command generates information about SSH connections,

## Configuration notes

- By default, a user logging into the device through Telnet or SSH would first enter the User EXEC level. The user can enter the **enable** command to get to the Privileged EXEC level.
- You can select only one primary authentication method for each type of access to a device (CLI through Telnet, CLI Privileged EXEC and CONFIG levels). For example, you can select TACACS+ as the primary authentication method for Telnet CLI access, but you cannot also select RADIUS authentication as a primary method for the same type of access. However, you can configure backup authentication methods for each access type.
- If you erase a **tacacs-server** command (by entering "**no**" followed by the command), make sure you also erase the **aaa** commands that specify TACACS/TACACS+ as an authentication method. Otherwise, when you exit from the CONFIG mode or from a Telnet session, the system continues to believe it is TACACS/TACACS+ enabled and you will not be able to access the system.
- TACACS+ command authorization can be performed only for commands entered from Telnet or SSH sessions, or from the console. No authorization is performed for commands entered at the Web management interface or IronView Network Manager.

## Common diagnostic scenarios

- Old software versions.

  Feature issues are often caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# SNTP

SNTP server enables the device to synchronize its clock with an upstream time server. From this release, SNTP server will also allow the PowerConnect device to function as an SNTP server to its downstream clients.

## SNTP show commands

### show sntp server-mode

**Syntax: show sntp server-mode**

This command displays the information about the SNTP Server.

```
PowerConnect(config)#show sntp server-mode
Status                      : up
Stratum                     : 6
Authentication              : md5
Clock Source                : 10.50.3.120
Last upstream sync          : 15:55:00 Pacific Sun Jul 5 2009
Last responses sent to 5 unique downstream clients:
    Client Address              Reference Time
    10.1.50.23                  16:10:32 Pacific Sun Jul 5 2009
    10.1.52.34                  15:50:40 Pacific Sun Jul 5 2009
    10.1.50.41                  10:22:08 Pacific Fri Jul 3 2009
    10.1.50.10                  06:21:03 Pacific Fri Jul 3 2009
    10.1.50.29                  21:17:39 Pacific Fri Jul 2 2009
```

### show sntp associations

**Syntax: show sntp associations**

This command displays the information about SNTP associations as shown in the following example:

```
PowerConnect(config)#show sntp associations
address              ref clock     st     when   poll   delay  disp
*~10.50.3.120        10.40.2.93    5       3      300   0.000  0.012
~10.50.7.118         0.0.0.0       16     -1      300   0.000  0.000
~216.218.254.202     0.0.0.0       16     -1      300   0.000  0.000
```

## SNTP debug command

### debug ip sntp

**Syntax: [no] debug ip sntp**

This command displays the debugging traces about the SNTP.

Output resembles the following:

```
PowerConnect#debug ip sntp
cu_get_time_utc_seconds_double: set_clock=3456685950, current ticks=936191, last
tick=0, ntp_time_counter=334, returned time=3456685950.02246856 sec

Sending SNTP Query to 64.125.78.85
SNTP Client Request Packet==========
Version =1, Mode=3, Originate TS = 3456685950.02246856
```

```
cu_get_time_utc_seconds_double: set_clock=3456685950, current ticks=1232321, last
tick=0, ntp_time_counter=334, returned time=3456685950.02957582 sec

^^^Delay calculation: dest-org=0.00710725, xmt-rec=0.00001144, Delay=0.00709581
org=3456685950.02246856, rec=3456686030.25115203, xmt=3456686030.25116348, dest=
3456685950.02957582

^^^Time offset calculation: rec-org=80.22868347, dest-xmt=-80.22
158765, Offset=0.00354790
Jul 15 15:33:10 SNTP Server Response Packet==========
Jul 15 15:33:10 Leap = 0, Stratum= 1, Precision= -29, Delay= 0, Disp= 0,
RefId=1094
Reference TS = 3456685990.3182691034
Originate TS = 3456685950.02246856
Receive TS = 3456686030.25115203
Transmit TS = 3456686030.25116348
**Dest TS = 3456685950.02957582
Roundtrip delay = 0.00709581
Time Offset = 0.00354790
```

## Configuration notes

- When using the device as an SNTP server, you can also set it to use its own internal clock as the reference source if an upstream server becomes unavailable.
- If synchronization with any of the configured upstream servers become impossible and the server was configured not to fall back to use the local-clock (RFC mode), it will respond to a client request with a “kiss-of-death” response (stratum number = 0).
- There is no support to configure SNTP server from through SNMP or Web management.

# IP security

The debug commands in this section apply to Internet Protocol security operation (IPsec) for OSPFv3. You should use them only if you suspect that something is not working correctly. For example, if you remove (or change) authentication configuration from one side of a connection, but the other side still expects OSPF packets inside ESP packets, the OSPF adjacency does not come up because the other side does not trust the OSPFv3 it receives. In this example, the **show ipv6 ospf neighbor** command would not indicate why the adjacency failed to come up. By turning on debugging, you can see that hello packets are being dropped due to no-authentication.

### debug ipsec esp

**Syntax:** [no] **debug ipsec esp**

This command turns on debugging of ESP. After key rollover finishes (if it is in process), the ESP debugging facility can show error data (if it exists). Note that the first row of information shows the SPI in hexadecimal format (0x1d97c), and this value is the equivalent of decimal 121212.

```
PowerConnect#debug ipsec sa
          IPSec:  esp debugging is on
PowerConnect#show ipsec esp
Dec 12 11:37:39 IPSEC,ESP: decrypt ok, seq=0  (SA: ESP  in spi=0x1d97c
dst=FE80::)
psec sa
             IPSEC Security Association Database(Entries:2)
SPDID Dir Encap SPI        Destination          AuthAlg  EncryptAlg
8     out ESP   121212     ::                   sha1     Null
8     in  ESP   121212     FE80::               sha1     Null
PowerConnect#Dec 12 11:37:49 IPSEC,ESP: decrypt ok, seq=0  (SA: ESP  in
spi=0x1d97c dst=FE80::)
Dec 12 11:37:59 IPSEC,ESP: decrypt ok, seq=0  (SA: ESP  in spi=0x1d97c
dst=FE80::)
Dec 12 11:38:08 IPSEC,ESP: decrypt ok, seq=0  (SA: ESP  in spi=0x1d97c
dst=FE80::)
Dec 12 11:38:18 IPSEC,ESP: decrypt ok, seq=0  (SA: ESP  in spi=0x1d97c
dst=FE80::)
Dec 12 11:38:30 IPSEC,ESP: decrypt ok, seq=0  (SA: ESP  in spi=0x1d97c
dst=FE80::)
Dec 12 11:38:40 IPSEC,ESP: decrypt ok, seq=0  (SA: ESP  in spi=0x1d97c
dst=FE80::)
```

### debug ipsec sa

**Syntax:** [no] **debug ipsec sa**

This command enables the display of debugging information related to the security associations used by IPsec for OSPFv3 packets.

```
PowerConnect#debug ipsec sa
          IPSec:  sa debugging is on
PowerConnect(config)#interface ethernet 1/8
PowerConnect(config-if-e1000-1/8)#ipv6 ospf authentication ipsec spi 121212 esp
sha1 no-encrypt 1234567890123456789012345678901234567890
PowerConnect(config-if-e1000-1/8)#Dec 12 11:45:37 IPSEC,SA:
ipipsec_pfkeyv2_input() :: receiving 'ADD' command
Dec 12 11:45:37 IPSEC,SA: Adding SA: ESP  in spi=0x1d97c dst=FE80::, replay=0
Dec 12 11:45:37 IPSEC,SA: ipipsec_pfkeyv2_input() :: succeeded
Dec 12 11:45:37 IPSEC,SA: ipipsec_pfkeyv2_input() :: receiving 'X_ADDFLOW'
command
Dec 12 11:45:37 IPSEC,SA: ipipsec_pfkeyv2_input() :: succeeded
Dec 12 11:45:47 IPSEC,SA: ipipsec_pfkeyv2_input() :: receiving 'ADD' command
Dec 12 11:45:47 IPSEC,SA: Adding SA: ESP  out spi=0x1d97c dst=::, replay=0
Dec 12 11:45:47 IPSEC,SA: ipipsec_pfkeyv2_input() :: succeeded
Dec 12 11:45:47 IPSEC,SA: ipipsec_pfkeyv2_input() :: receiving 'X_ADDFLOW'
command
Dec 12 11:45:47 IPSEC,SA: ipipsec_pfkeyv2_input() :: succeeded
```

### debug ipsec policy

**Syntax:** [no] **debug ipsec policy**

This command enables the display of debugging information for IPsec security policy.

```
PowerConnect#debug ipsec policy
          IPSec:  policy debugging is on
PowerConnect(config)#interface ethernet 1/8
PowerConnect(config-if-e1000-1/8)#ipv6 ospf authentication ipsec spi 121212 esp
sha1 no-encrypt 1234567890123456789012345678901234567890
PowerConnect(config-if-e1000-1/8)#Dec 12 11:47:45 IPSEC,Policy: Creating flow
[input use 'prot=OSPF src=FE80::/10:0 dst=::/0:0' -> SA: ESP  in spi=0x1d97c
dst=FE80::] : ok
PowerConnect(config-if-e1000-1/8)#Dec 12 11:47:55 IPSEC,Policy: Creating flow
[output use 'prot=OSPF src=FE80::/10:0 dst=::/0:0' -> SA: ESP  out spi=0x1d97c
dst=::] : ok
```

### debug ipsec in

**Syntax:** [no] **debug ipsec in**

This command enables the display of debugging information related to inbound OSPFv3 packets with IPsec.

```
PowerConnect#debug ipsec in
          IPSec:  in debugging is on
PowerConnect#Dec 12 11:39:49 IPSEC,IN:  ESP  spi=121212 (pkt 'ESP  FE80:: ->
FE80::') payloadlength =64
Dec 12 11:39:49 IPSEC,IN: Incoming packet matches Policy :  input use 'prot=OSPF
src=FE80::/10:0 dst=::/0:0' -> SA: ESP  in spi=0x1d97c dst=FE80::
PowerConnect#sh ipDec 12 11:39:59 IPSEC,IN:  ESP  spi=121212 (pkt 'ESP  FE80:: ->
FE80::') payloadlength =64
Dec 12 11:39:59 IPSEC,IN: Incoming packet matches Policy :  input use 'prot=OSPF
src=FE80::/10:0 dst=::/0:0' -> SA: ESP  in spi=0x1d97c dst=FE80::
sec po
             IPSEC Security Policy Database(Entries:2)
PType  Dir Proto Source(Prefix:TCP/UDP Port)    Destination(Prefix:TCP/UDPPort)
                   SA: SPDID Dir Encap SPI       Destination
use    in  OSPF  FE80::/10:any                  ::/0:any
                   SA: 8     in  ESP   121212     FE80::
use    out OSPF  FE80::/10:any                  ::/0:any
                   SA: 8     out ESP   121212     ::
```

### debug ipsec out

**Syntax:** [no] **debug ipsec out**

This command enables the display of debugging information related to outbound OSPFv3 packets with IPsec.

### debug ipv6 ospf ipsec

**Syntax:** [no] **debug ipv6 ospf ipsec**

Of the debugging commands for IPsec, this command is the most relevant. It can show if IPsec is actually providing its services to IPv6 OSPFv3. For example, the example output shows success in the attempts to provide various IPsec services to OSPFv3.

```
PowerConnect(config-if-e1000-1/8)#ipv6 ospf authentication ipsec spi 121212 esp
sha1 no-encrypt 1234567890123456789012345678901234567890
PowerConnect(config-if-e1000-1/8)#Dec 12 11:53:24 OSPFv3:
ITC_AUTHENTICATION_CONFIG message received
Dec 12 11:53:24 OSPF6: Sending request to IPSEC to ADD Inbound SA for SA with
SPI=121212 SPDID=8
Dec 12 11:53:24 OSPFv3: IPSEC ADD Inbound SA SUCCESS for SA with SPI=121212,
SPDID=8!
Dec 12 11:53:24 OSPF6: Sending request to IPSEC to ADD Inbound Policy with
SPI=121212
Dec 12 11:53:24 OSPFv3: IPSEC ADD Inbound Policy SUCCESS for SA with SPI=121212,
SPDID=8!
Dec 12 11:53:24 OSPF6: Auth timer started
Dec 12 11:53:24 OSPFv3: Key Rollover, for 1/8, state change NOT_ACTIVE->STARTED
Dec 12 11:53:34 OSPFv3: Key Rollover, for 1/8, state change STARTED->IN-PROGRESS
Dec 12 11:53:34 OSPF6: Sending request to IPSEC to ADD Outbound SA for SA with
SPI=121212 SPDID=8
Dec 12 11:53:34 OSPFv3: IPSEC ADD Outbound SA SUCCESS for SA with SPI=121212,
SPDID=8!
Dec 12 11:53:34 OSPF6: Sending request to IPSEC to ADD Outbound Policy with
SPI=121212
Dec 12 11:53:34 OSPFv3: IPSEC ADD Outbound Policy SUCCESS for SA with
SPI=121212, SPDID=8!
Dec 12 11:53:44 OSPFv3: Key Rollover, for 1/8, state change
IN-PROGRESS->NOT_ACTIVE
Dec 12 11:53:44 OSPF6: Auth timer stopped
```

Chapter

# Forwarding Diagnostics

10

This chapter describes diagnostics for forwarding protocols and environments on PowerConnect B-MLXe routers.

## ARP

Address Resolution Protocol (ARP) is a standard IP protocol that enables a router to obtain the MAC address of an interface on another device when the router knows the IP address of the interface. ARP is enabled by default and cannot be disabled.

### ARP show commands

#### show ip arp-inspection

**Syntax: show ip arp-inspection** [**vlan** <*vlan_id*>]

This command displays the ARP inspection status and the trusted and untrusted ports in a VLAN, as shown in the following example:

```
PowerConnect#show ip arp-inspection
ARP inspected VLANs:
1000

ARP inspection trusted ports:
ethe 2/1
```

#### show ip static-arp

**Syntax: show ip static-arp**

This command displays the ARP Inspection table, as shown in the following example:

```
PowerConnect#show ip static-arp
Total no. of entries: 6
  Index   IP Address          MAC Address         Port          Vlan
  1       10.10.10.10         0123.0000.0000
  2       110.0.0.2           0000.0002.0000      10/4
  3       110.0.0.3           0000.0003.0000      10/4
  4       110.0.0.4           0000.0004.0000      10/4
  5       110.0.0.5           0000.0005.0000      10/4
  6       110.0.0.6           0000.0006.0000      10/4
  7       110.0.0.7           0000.0007.0000      10/4
```

# ARP debug commands

### debug ip arp

**Syntax:** [no] **debug ip arp** [**event** | **ipc** | **itc** | **packet**]

- **event** - Displays information about ARP events.
- **ipc** - Displays information about ARP IPC messages.
- **itc** - Displays information about ARP ITC messages.
- **packet** - Displays information about ARP packets.

This command displays information about ARP transactions, either for all ARP variables, or specified variables only.

### debug ip arp event

**Syntax:** [no] **debug ip arp event**

This command displays information about ARP events, which indicates whether the router is sending and receiving ARP requests. Output is similar to the following, which shows send and receive activity for ARP packets:

```
PowerConnect#debug ip arp event
IP/ARP: sent request for 206.223.143.22
IP/ARP: sent packet src 206.223.143.16 000cdbe2b000: dst 206.223.143.22
000000000000: Port 1062
IP/ARP: sent request for 206.223.143.3
IP/ARP: sent packet src 206.223.143.16 000cdbe2b000: dst 206.223.143.3
000000000000: Port 1062
IP/ARP: sent request for 69.28.144.206
IP/ARP: sent packet src 69.28.144.205 000cdbe2b000: dst 69.28.144.206
000000000000: Port 843
IP/ARP: Received arp request from Lp for dest 69.28.144.206 Port: 843 Router: 1
IP/ARP: sent request for 69.28.181.171
IP/ARP: sent packet src 69.28.181.161 000cdbe2b000: dst 69.28.181.171
000000000000: Port 753
```

### debug ip arp ipc

**Syntax:** [no] **debug ip arp ipc**

This command generates information about ARP Interprocess Communication (IPC) activity. Output resembles the following:

```
PowerConnect#debug ip arp ipc
IP/ARP: Received arp request from Lp for dest 68.142.108.94 Port: 1049 Router: 1
IP/ARP: Received arp request from Lp for dest 68.142.106.82 Port: 848 Router: 1
IP/ARP: Received arp request from Lp for dest 68.142.108.94 Port: 1049 Router: 1
IP/ARP: Received arp request from Lp for dest 68.142.106.82 Port: 848 Router: 1
IP/ARP: Received arp request from Lp for dest 68.142.108.94 Port: 1049 Router: 1
IP/ARP: Received arp request from Lp for dest 69.28.144.206 Port: 843 Router: 1
IP/ARP: Received arp request from Lp for dest 68.142.78.18 Port:846 Router: 1
IP/ARP: Received arp request from Lp for dest 69.28.144.206Port: 843 Router: 1
IP/ARP: Received arp request from Lp for dest 68.142.78.18 Port: 846 Router: 1
```

### debug ip arp itc

**Syntax:** [no] **debug ip arp itc**

This command generates information about ARP inter-task communications (ITC) activity, which is communications between processing tasks. Output resembles the following:

```
PowerConnect#debug ip arp itc
ARP: itc debugging is on
PowerConnect#configure terminal
PowerConnect(config)#arp 19 6.1.1.19 1001.2001.3019 ethernet 1/2
IP/ARP: Add static arp for Addr: 6.1.1.19 Mac: 100120013019 Port: 1
Vrf_index: 0 Add: 1
```

### debug ip arp packet

**Syntax:** [no] **debug ip arp packet**

This command displays information about ARP packet activity. Output resembles the following, which indicates that the source router is polling routers 68.142.106.98, 69.28.181.122, 206.223.143.27 and 68.142.108.94 to learn their MAC addresses and add them to the source router ARP table:

```
PowerConnect#debug ip arp packet
IP/ARP: sent request for 68.142.106.98
IP/ARP: sent packet src 68.142.106.97 000cdbe2b000: dst 68.142.106.98
000000000000: Port 114
IP/ARP: sent request for 69.28.181.122
IP/ARP: sent packet src 69.28.181.121 000cdbe2b000: dst 69.28.181.122
000000000000: Port 1045
IP/ARP: sent request for 69.28.181.172
IP/ARP: sent packet src 69.28.181.161 000cdbe2b000: dst 69.28.181.172
000000000000: Port 753
IP/ARP: sent request for 206.223.143.27
IP/ARP: sent packet src 206.223.143.16 000cdbe2b000: dst 206.223.143.27
000000000000: Port 1062
IP/ARP: sent request for 68.142.106.82
IP/ARP: sent packet src 68.142.106.81 000cdbe2b000: dst 68.142.106.82
000000000000: Port 848
IP/ARP: sent request for 68.142.108.94
```

## Configuration notes

- To delete the static MAC entry, you must delete the static ARP entry first.

## Common diagnostic scenarios

- Old software versions.

  Feature issues are often caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# ECMP

Equal-Cost MultiPath (ECMP) supports traffic that is forwarded in software. ECMP applies only to traffic forwarded by software, not to traffic forwarded by hardware. Normally, traffic is forwarded in software when you configure a CPU-based feature such as ACLs, rate limiting, or NetFlow Switching. Traffic also is forwarded by software if the CAM (used for hardware forwarding) becomes full.

## ECMP show commands

### show ipv6

**Syntax: show ipv6**

This command displays the status of ECMP load sharing for IPv6, as shown in the following example:

```
PowerConnect#show ipv6
Global Settings

  unicast-routing enabled, hop-limit 64
  No Inbound Access List Set
  No Outbound Access List Set
  Prefix-based IPv6 Load-sharing is Enabled, Number of load share paths: 4
```

### show ipv6 cache

**Syntax: show ipv6 cache** [<*index-number*> | <*ipv6-prefix*>/<*prefix-length*> | <*ipv6-address*> | **ethernet** <*port*> | **ve** <*number*> | **tunnel** <*number*>]

This command displays all the entries in the IPv6 forwarding cache, or just those specified using the syntax variables. The following example shows all entries:

```
PowerConnect#show ipv6 cache
Total number of cache entries: 10
    IPv6 Address                      Next Hop                    Port
1   5000:2::2                         LOCAL                       tunnel 2
2   2000:4::106                       LOCAL                       ethe 2
3   2000:4::110                       DIRECT                      ethe 2
4   2002:c0a8:46a::1                  LOCAL                       ethe 2
5   fe80::2e0:52ff:fe99:9737          LOCAL                       ethe 2
6   fe80::ffff:ffff:feff:ffff         LOCAL                       loopback 2
7   fe80::c0a8:46a                    LOCAL                       tunnel 2
8   fe80::c0a8:46a                    LOCAL                       tunnel 6
9   2999::1                           LOCAL                       loopback 2
10  fe80::2e0:52ff:fe99:9700          LOCAL                       ethe 1
```

## ECMP debug commands

There are no debug commands specific to ECMP.

## Configuration notes

The following considerations should be taken into account when configuring RPF with ECMP routes:

- For a source IP address matching an ECMP route, RPF will permit the packet if it arrives on any of the next-hop interfaces for that route. For example, if there are two best next-hops for a network route 11.11.11.0/24, one pointing to 10.10.10.1 (Gigabit Ethernet 7/1) and the other to 10.10.30.1 (Gigabit Ethernet 7/12), then incoming packets with source address matching 11.11.11.0/24 will be permitted on either Gigabit Ethernet 7/1 or Gigabit Ethernet 7/12.
- A disadvantage of this configuration is that if some other route shares any of these next-hops, the packets with a source IP address matching that route will also be permitted from any of the interfaces associated with those next hops. For example, say 12.12.12.0/24 has the next-hop 10.10.10.1, then packets from 12.12.12.0/24 will also be permitted on either Gigabit Ethernet 7/1 or Gigabit Ethernet 7/12.

## Common diagnostic scenarios

- Old software versions.

  Feature issues are often caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# Multicast/VRF Forwarding

The provider edge (PE) router maintains a Virtual Routing and Forwarding table (VRF) for each customer that is attached to it through a customer edge (CE) router. The VRF contains routes between the PE and the CE and Label Switched Paths (LSPs) across the MPLS domain for each PE that is a member of the customer's VPN. VRFs are defined on interfaces of the PEs.

## Multicast/VRF show commands

**show ip bgp vpnv4 neighbor flap-statistics**

**Syntax: show ip bgp vpnv4** <*vrf-name*> **neighbor** <*ip-addr*> **flap-statistics**

- <*vrf-name*> - Specifies the VPNv4 neighbor for which you want to display flap-statistics.
- <*ip-addr*> - Specifies a particular route.

This command displays flap statistics for routes learned from the specified VRF neighbor, as shown in this example:

```
PowerConnect(config)#show ip bgp vpnv4 neighbor 2.2.2.2 flap-statistics
Total number of flapping routes: 0
```

This output shows the total number of routes in the Layer 3 switch BGP4 route table that have changed state and have been marked as flapping routes.

### show ip pim vrf prune

**Syntax: show ip pim** [**vrf** <*vrf-name*>] **prune**

This command displays all multicast cache entries that are currently in a pruned state and have not yet aged out, as shown in this example:

```
PowerConnect(config-if-e10000-1/1)#show ip pim vrf med prune
Index Port    PhyPort SourceNet       Group           Nbr             Age
                                                                      sec
    1 v12     1/1     130.47.2.10     228.172.0.77    0.0.0.0         40
    2 v12     1/1     130.47.2.10     228.172.0.73    0.0.0.0         40
    3 v12     1/1     130.47.2.10     228.172.0.69    0.0.0.0         40
    4 v12     1/1     130.47.2.10     228.172.0.65    0.0.0.0         40
    5 v12     1/1     130.47.2.10     228.172.0.61    0.0.0.0         40
    6 v12     1/1     130.47.2.10     228.172.0.57    0.0.0.0         40
    7 v12     1/1     130.47.2.10     228.172.0.53    0.0.0.0         40
    8 v12     1/1     130.47.2.10     228.172.0.49    0.0.0.0         40
    9 v12     1/1     130.47.2.10     228.172.0.45    0.0.0.0         40
   -------------------------------------
Total Prune entries: 9
```

### show ip pim group

**Syntax: show ip pim** [**vrf** <*vrf-name*>] **group**

This command displays PIM group information for an entire PIM group, or for the VRF instance identified by the <*vrf-name*> variable. Output resembles the following:

```
PowerConnect#show ip pim group

Total number of Groups: 2
Index 1         Group 239.255.162.1       Ports e3/11
```

### show cam ifl

**Syntax: show cam ifl** [<*slot/port*>]

This command displays CAM information for a specific port. Output resembles the following:

```
PowerConnect#show cam ifl 7/7
Slot Index   Port  Outer VLAN Inner VLAN PRAM   IFL ID IPV4/V6
(Hex)                         (Hex)         Routing
7    0081fe9 7/4   4000       0          181fe9 131071 1/1
7    0081fea 7/3   4000       0          181fea 131071 1/1
7    0081feb 7/2   4000       0          181feb 131071 1/1
7    0081fec 7/1   4000       0          181fec 131071 1/1
7    0081fed 7/8   607        0          181fed 131071 1/1
7    0081fee 7/7   607        0          181fee 131071 1/1
7    0081fef 7/8   606        0          181fef 131071 1/1
7    0081ff0 7/7   606        0          181ff0 131071 1/1
7    0081ff1 7/8   605        0          181ff1 131071 1/1
7    0081ff2 7/7   605        0          181ff2 131071 1/1
7    0081ff3 7/8   604        0          181ff3 131071 1/1
7    0081ff4 7/7   604        0          181ff4 131071 1/1
7    0081ff5 7/8   603        0          181ff5 131071 1/1
7    0081ff6 7/7   603        0          181ff6 131071 1/1
7    0081ff7 7/8   602        0          181ff7 131070 1/1
7    0081ff8 7/7   602        0          181ff8 131070 1/1
7    0081ff9 7/8   601        0          181ff9 131071 1/1
7    0081ffa 7/7   601        0          181ffa 131071 1/1
```

# Multicast/VRF debug commands

### debug ip vrf

**Syntax:** [no] **debug ip vrf**

This command generates information about synchronization of VRF routing information to line cards, as shown in the following example:

```
PowerConnect#debug ip vrf
RTM (vrf): Processing tree download for vrf 1
```

This is a download request from the line card to start the tree download for VRF 1.

### debug ip bgp all-vrfs

**Syntax:** [no] **debug ip bgp all-vrfs** [ *A.B.C.D* | **dampening** | **events** | **graceful-restart** | **keepalives** | **updates**]

- *A.B.C.D* - Displays information about a BGP neighbor address.
- **dampening** - Displays information about BGP dampening.
- **events** - Displays information abuot BGP events.
- **graceful-restart** - Displays information about graceful-restart events.
- **keepalives** - Displays information about BGP keepalives.
- **updates** - Displays information about BGP updates.

This command displays information about BGP activity, with output that is limited to VRF events.

### debug ip ospf all-vrfs

**Syntax:** [no] **debug ip ospf all-vrfs** [*A.B.C.D* | **adj** | **bfd** | **error** | **events** | **flood** | **graceful-restart** | **log-debug-message** | **log-empty-lsa** | **lsa-generation** | **max-metric** | **packet** | **retransmission** | **route** | **sham-link** | **shortcuts** | **spf**]

- *A.B.C.D* - OSPF neighbor address.
- **adj** - Displays OSPF adjacency events.
- **bfd** - Displays OSPF BFD events.
- **error** - Displays possible OSPF error in run time.
- **events** - Displays OSPF events.
- **flood** - Displays OSPF flooding.
- **graceful-restart** - Displays OSPF graceful restart events.
- **log-debug-message** - Enables OSPF debug message logging.
- **log-empty-lsa** - Enables OSPF empty LSA logging.
- **lsa-generation** - Enables OSPF LSA generation.
- **max-metric** - Displays information about OSPF Stub Router Advertisement.
- **packet** - Displays information about OSPF packets.
- **retransmission** - Displays OSPF retransmission events.
- **route** - Displays information about OSPF routes.
- **sham-link** - Displays OSPF sham-link traces.
- **shortcuts** - Displays OSPF shortcuts.
- **spf** - Displays OSPF SPF traces.

This command displays OSPF information for all VRF activity.

## Configuration notes

- You must configure a VRF on an interface before configuring a Virtual Router (VRRPE) on it. If you enable the Virtual Router before you enable the VRF, the Virtual Router configuration will be deleted.

## Common diagnostic scenarios

- Old software versions.

  Feature issues are often caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# RPF

Reverse Path Forwarding (RPF) prevents malicious users from spoofing a source IP address. It does this by checking that the source address specified for a packet is received from a network to which the router has access. Packets with invalid source addresses aren't forwarded. Packets that fail the RPF test can be logged.

## RPF show commands

### show ip interface

**Syntax: show ip interface ethernet** <*slotnum/portnum*>

This command displays information about RPF configurations and packets that have been dropped because they failed the RPF check, as shown in this example in bold:

```
PowerConnect#show ip interface ethernet 7/1
Interface Ethernet 7/1 (384)
   port enabled
   port state: UP
   ip address: 1.2.3.4/8
   Port belongs to VRF: default
   encapsulation: Ethernet, mtu: 1500
   MAC Address 000c.db24.a6c0
   directed-broadcast-forwarding: disabled
   No inbound ip access-list is set
   No outbound ip access-list is set
   No Helper Addresses are configured
   RPF mode: strict RFP Log: Disabled
   376720 unicast RPF drop 36068 unicast RPF suppressed drop
```

**NOTE**
RPF accounting information is always available through the physical interface, even if the physical port belongs to one or more VEs.

### *Clearing RPF statistics for a specified interface*

To clear RPF statistics on a specific physical interface use the following command.

**clear ip interface ethernet 7/1**

**Syntax: clear ip interface ethernet** <*slot/port*>

**show logging**

**Syntax: show logging**

If you have enabled the **log** option of the **rpf-mode** command, packet information is saved to the system log. To display the log, enter the **show logging** command, as shown in the following example:

```
PowerConnect#show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 1305 overruns)
    Buffer logging: level ACDMEINW, 50 messages logged
    level code: A=alert C=critical D=debugging M=emergency E=error
                I=informational N=notification W=warning
Dynamic Log Buffer (50 lines):
May 11 12:12:54:I:RPF: Denied 1 packets on port 7/5 tcp 4.4.4.1(0) -> 5.6.7.8(0)
```

**NOTE**
A maximum of 256 RPF messages are logged per minute.

## RPF debug commands

There are no debug commands specific to RPF.

## Configuration notes

- IP packets with source IP address of 0.0.0.0 will always fail RPF check
- If you attempt to enable the global RPF command on a system with incompatible CAM settings, the command will be rejected and you will receive a console message describing this.
- Since the RPF feature requires that the entire IP route table is available in hardware, the feature must work in conjunction with Foundry Direct Routing (FDR). FDR is the default mode of operation for the PowerConnect B-MLXe.
- You cannot configure RPF on a physical port that has VRF configured on it or if the physical port belongs to a virtual interface with a VRF configuration.
- Only RPF loose mode is supported for GRE routes.
- If a default route is present on the router, loose mode will permit all traffic.
- RPF can only be configured at the physical port level. It should not be configured on virtual interfaces.

## Common diagnostic scenarios

- RPF check fails.

  Problem is due to the fact that the OSPF best route back to the BSR is different from the interface where PIM is enabled.

- RPF packet drop occurs.

  Rate-limiting was configured on the port where the drops occurred. Problem resolved when configuration was changed.

- Old software versions.

  Feature issues are often caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# Trunking

Trunk groups are manually-configured aggregate links containing multiple ports. Trunk groups enable load sharing of traffic, and they also provide redundant, alternate paths for traffic if any of the segments fail.

The following sections describe how to display information about trunking configurations.

## Trunking show commands

Trunk group configuration information can be displayed using the same command for either a server or switch trunk. The information is displayed in two sections: configured trunks and operational trunks.

### show lag

**Syntax: show lag ethernet** <*slotnum*>/<*portnum*> **to** <*slotnum*>/<*portnum*>

This command displays information about trunk groups, and is divided into sections for configured trunks and operational trunks. A configured trunk group is one that has not yet been activated.

To display server trunk group information for a *range of ports*, enter a command similar to the following:

```
PowerConnect(config)#show lag ethernet 12/1 to 12/3
Max number of trunks: 128
available: 127
Configured number of server trunks: 1
Configured trunks:
Trunk ID: 1
Type: Server
Ports_Configured: 3
Base FID: 0x0400
FID count: 16
Ports         12/1    12/2      12/3
Port Names    none    none      none
Port_Status   enable  enable    enable
Operational trunks:
Trunk ID: 1
```

```
Type: Server
Duplex: Full
Speed: 1G
Tag: No
Priority: level0
Active Ports: 3
Ports                12/1    12/2    12/3
Link_Status          active  active  active
```

To display switch trunk group information for *specific* ports, enter a command similar to the following:

```
PowerConnect(config)#show lag ethernet 9/1 to 9/2
Max number 206 (128 server trunks, 78 switch trunks)
Number of hash buckets per server trunk: 256
Configured number of server trunks: 0
Configured trunks:
Trunk ID: 66
Type: Switch
Ports_Configured: 2
Ports         9/1     9/2
Port Names    none    none
Port_Status   enable  enable

Operational trunks:
Trunk ID: 66
Type: Switch
Duplex: Full
Speed: 10G
Tag: No
Priority: level0
Active Ports: 1
Ports                9/1    9/2
Link_Status          active down
Load Sharing
 Mac Address         0      0
 IP                  0      0
 Multicast           0      0
 PBR                 0      0
```

## Trunking debug commands

There are no debug commands specific to trunking.

## Configuration notes

There are several trunk group rules. For a full description of these trunk rules, see the *NetIron Series Configuration Guide.*

The following items should also be considered when configuring trunk groups:

- You can use both static trunk groups and 802.3ad trunking on the same device. However, you can use only one type of trunking for a given port. For example, you can configure port 1/1 as a member of a static trunk group or you can enable 802.3ad link aggregation on the port, but you cannot do both.
- The ports in a trunk group make a single logical link. Therefore, all the ports in a trunk group must be connected to the same device at the other end.

- Trunk threshold should be configured on only one end of the trunk. If threshold is set on both sides, link failures will result in race-conditions and the trunk will not function properly.
- If you connect physical cables before configuring the trunk groups and then reboot, traffic on the ports can create a spanning tree loop.

## Common diagnostic scenarios

- Trunk transaction failed: Ports overlap with other trunks

  Customer is using static trunk. With static trunk, they must first remove the existing trunk and reconfigure a new one. If they were using dynamic trunk configuration (LACP - 802.1ad), they would be able to add port dynamically in trunk.

- Old software versions.

  Feature issues are often caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# MCT

Multi-Chassis Trunking provides switch level redundancy. If one of the switches goes down, then the LAG is still up and the traffic flows without any disruption.

## MCT show command

### show cluster

**Syntax:** show cluster

This command displays the complete cluster information on ICL, peer, and clients as shown in the following example:

```
PowerConnect#show cluster
Cluster abc 1
=============
Rbridge Id: 100, Session Vlan: 4090
Cluster State: Deploy
Clients State: All client ports are administratively disabled [Optional]
Client Isolation Mode: Strict [Optional]
Configured Member Vlan Range: 20 to 30
Active Member Vlan Range: 20
ICL Info:
---------
Name Port Trunk
icl1 1/1 -
Peer Info:
----------
Peer IP: 10.10.10.2, Peer Rbridge Id: 200, ICL: icl1
KeepAlive Interval: 20 , Hold Time: 60, Fast Failover
Active Vlan Range: 20
Peer State: CCP Up (Up Time: 0 days: 0 hr:15 min:44 sec)
Client Info:
------------
```

```
Name Rbridge-id Config Port Trunk FSM-State
c1 300 Deployed 1/13 - Up
```

## MCT debug commands

### debug cluster forwarding

**Syntax:** [no] **debug cluster forwarding**

This command displays all the MCT forwarding-related events or messages in MP which can affect traffic forwarding. Some examples include remote CCEP status changes, MCT FID creation and FID updates, etc.

The output resembles the following:

```
PowerConnect#debug cluster forwarding
CLUSTER FORWARDING: MCT CCEP control FID 0xa006 for vlan 101 - REMOVE CCEP port
1/1
CLUSTER FORWARDING: Processing remote CCEP UP event for port eth 1/1
CLUSTER FORWARDING: Processed remote CCEP event for port 1/1 for 100 vlans
CLUSTER FORWARDING: MCT CCEP control FID 0xa006 for vlan 101 - ADD CCEP port 1/1
CLUSTER FORWARDING: Processing remote CCEP DOWN event for port eth 1/1
```

### debug cluster actions

**Syntax:** [no] **debug cluster actions**

This command displays the debug messages for MCT-related events like MCT fid port changes, etc.

The output resembles the following:

```
PowerConnect#debug cluster actions
CLUSTER ACTIONS: Mac learning disabled for MAC 0000.2222.0001 on ICL  4/2, vlan
101
CLUSTER ACTIONS: Received remote CCEP UP IPC
```

### debug cluster cam

**Syntax:** [no] **debug cluster cam**

This command displays all the CAM or PRAM-related activities for MCT-related events in LP.

The output resembles the following:

```
PowerConnect# debug cluster cam
CLUSTER CAM: Added peer interface cam:mac 001b.ed3b.a200, ppcr 0
CLUSTER CAM: Deleted peer interface cam:mac 001b.ed3b.a200, ppcr 0
```

### debug cluster ipc

**Syntax:** [no] **debug cluster ipc**

This command is helpful to debug IPC to LP to sync forwarding information in LP.

The output resembles the following:

```
PowerConnect#debug cluster ipc
CLUSTER IPC: MCT FID 0xa005 received for vlan 101
CLUSTER IPC: cluster 1, vlan mask change
```

## Configuration notes

- ICL ports should not be untagged member of any VLAN and ICL is preferably a LAG to provide port level redundancy and higher bandwidth for cluster communication.
- ICL ports can be part of MCT VLANs as well as regular VLANs.
- ICL VLAN mask should be superset of client VLAN mask.
- MAC learning is disabled on ICL ports for the VLANs configured in the cluster. However, MAC learning is enabled on ICL port for non-cluster VLANs.
- MUDP will synchronize all MAC entries for VLANs served by ICL link.
- Interaction with Other Features: On CCEP ports, we do not support MPLS, VLL, VPLS, 802.1ah, 802.1ad and Routing protocols.
- The following are supported:
  - Supports only 2-node MCT topology.
  - Support for Layer 2 switching, VE for VRRP or VRRP-E, IPv4 unicast only.
  - Supports static-LAG and LACP LAG for MCT ICL and CCEP ports.
  - Support for port-loop detection feature.
  - Hardware flooding features are supported on cluster VLANs.
  - STP BPDU flooding or dropping are supported using a CLI.
- The following are not supported:
  - Multi-port MAC configuration is not supported on ICL or CCEP ports. Configuration will be rejected if we try to configure multi-port MAC addresses with a port mask which contains either a CCEP port or ICL port and vice versa.
  - Similarly, multi-port ARP configuration is allowed only on ICL or CCEP ports.
  - MP hitless failover is not supported but it is compatible, for example, the system behavior will be similar to reload from the cluster point of view.
  - Hitless OS Upgrade is not supported but it is compatible, for example, the system behavior will be similar to reload from the cluster point of view.

# VPLS unicast forwarding

VPLS enhances the point-to-point connectivity defined in the Draft-Martini IETF documents by specifying a method for Virtual Circuits (VCs) to provide point-to-multipoint connectivity across the MPLS domain, allowing traffic to flow between remotely connected sites as if the sites were connected by a Layer 2 switch.

VPLS can be used to transport Ethernet frames to and from multiple, geographically dispersed sites belonging to a customer VPN. The Provider Edge (PE) devices connecting the customer sites provide functions similar to a Layer 2 switch. The PE devices learn the MAC addresses of locally connected customer devices, flood broadcast and unknown unicast frames to other PE devices in the VPN, and create associations between remote MAC addresses and the VC LSPs used to reach them.

For more information about MPLS VPLS diagnostics, see Chapter 6, “MPLS Diagnostics” .

## VPLS unicast forwarding show commands

### show mpls debug vpls

**Syntax: show mpls debug vpls** <*vpls id*>

This command displays generic VPLS debug information, as shown in the following example.

```
PowerConnect#show mpls debug vpls 1
ID:       1      Name:     test 1
CPU-Prot: OFF    MVID:     INVD     FID:    0x00002002
MAC Info:
     Total MACs: 2  Local: 2  Remote: 0
     Max Exceed: 0 Table Full: 0
```

### show mpls debug vpls local

**Syntax: show mpls debug vpls local** <*num*>

This command displays the state of a VPLS end-point. Specify the vpls-id to see the local entries for a specific VPLS. Output resembles the following (specified for VPLS 2):

```
PowerConnect#show mpls debug vpls local 2
VPLS 2:
    VLAN  Port   Valid  Pending
    ====  ====   =====  =======
    4     2/19   1      0
    Local Broadcast Fids:
    =====================
    VLAN 4        --  Fid: 00008fa6, Ports: 1
      Port 2/19   --  Fid: 0000003a
```

### show mpls debug vpls remote

**Syntax: show mpls debug vpls remote** <*num*>

This command displays the state of all VPLS peers configured in the system. To display information for a specific VPLS, enter a VPLS-ID number. Output resembles the following (specified for remote VPLS 1):

```
 PowerConnect#show mpls debug vpls remote 1
 VPLS 1:
     Peer:  5.5.5.5             Valid:  Yes     Pending Delete: 0
     Label: 983040              Tagged: No      Load Balance:   No
                                                Num LSP Tnnls:  1
           VC        Tunnel    NHT            Use
     Port  Label     Index     COS  COS
     ====  =====     ======    =====     ===  ===
     2/9   983040    3         0         0    0
     2/9   983040    3         0         0    0
     2/9   983040    3         0         0    0
     2/9   983040    3         0         0    0
     Active Trunk Index: 0
```

Internally, a maximum of fourLSP tunnels are maintained to reach the peer. If Load Balancing is disabled, information for only one tunnel is displayed in the output.

## Configuration notes

- When you enable IP multicast for a specific VLAN or VPLS instance, IGMP snooping is enabled. The device uses IGMP to maintain a table of the Group Membership reports received by the device for the specified VLAN or VPLS instance. You can use active or passive IGMP mode. There is no default mode.
- Active – When active IGMP mode is enabled, the router actively sends out IGMP queries to identify IP multicast groups within the VLAN or VPLS instance and makes entries in the IGMP table based on the Group Membership reports received from the network.
- Passive – When passive IGMP mode is enabled, the router listens for IGMP Group Membership reports on the VLAN or VPLS instance specified but does not send IGMP queries. The passive mode is called “IGMP snooping”. Use this mode when another device in the VLAN or VPLS instance is actively sending queries.
- The **multicast static-group uplink** command cannot be configured globally per VPLS basis. It can be configured under the VLAN configuration only.

## Common diagnostic scenarios

- VPLS does not accept a receiver report where VPLS VLAN 1002 is configured with multicast active.

  This issue was resolved in a software patch. The customer was instructed to upgrade their software version.

- VPLS drops multicast traffic with multicast addresses that have not been reported to the PowerConnect B-MLXe (through IGMP Report).

  This issue was resolved in a software patch. The customer was instructed to upgrade their software version.

- Old software versions.

  Feature issues are often caused because the customer is running an old version of the software. It is recommended that customers always update software to reflect the latest patches and versions. If you have questions about your software version, contact Dell Technical Support for assistance.

# GRE and IPv6 tunnel debug commands

The following commands provide information about GRE and ipv6 tunnels. Since both tunnel types are used by various applications (Multicast, PBR etc.),the following commands display specific debugging messages with detailed information. These commands help to isolate events for a specific tunnel, a range of tunnels, or all tunnels.

**debug iptunnel**

**Syntax:** [**no**] **debug iptunnel** [ **errors** | **events** | **ipc** | **keepalives** | **packets** | |**statistics** | **tunnel-type** [**gre** | **ipv6**]] [**range** <*tunnel-id low*> **-** <*tunnel-id high*>]

- **errors** - Displays IP tunnel errors.
- **events** - Dispjays IP tunnel events.
- **ipc** - Displays IP tunnel IPC messages.
- **keepalives** - Displays keepalive information (for GRE tunnels only).

- **packets** - Displays IP tunnel packets information.
- **statistics** - Displays tunnel statistics.
- **tunnel-type** - Displays information for a specific tunnel type.
- **gre** - Displays information for a specific GRE tunnel.
- **ipv6** - Displays information for a specific IPv6 tunnel.
- **range** <*tunnel-id low*> **-** <*tunnel-id high*> - Displays information for a range of tunnels (specify lower and higher tunnel ID values).

The following examples describe some of these commands and their output.

### debug iptunnel errors

**Syntax:** [**no**] **debug iptunnel errors** [**range** <*tunnel-id low*> **-** <*tunnel-id high*>]

This command displays any error messages for the specified range of tunnels, including unexpected events as part of the packet flow, IPC flow, or configuration of ip tunnels.

Output resembles the following.

```
Apr 28 11:54:24 TNNL_GRE:ERRORS: tnnl 2- IP tunnel is invalid
Apr 28 11:54:24 TNNL_GRE:ERRORS: tnnl 2- L3 encapsulate and send - Dropping as
Tnnl is Down
```

### debug iptunnel events

**Syntax:** [**no**] **debug iptunnel events** [**range** <*tunnel-id low*> **-** <*tunnel-id high*>]

This command displays any tunnel events for the specified range of tunnels, including.route changes, IP/IPv6 port state notifications, tunnel status changes because of destination route changes or source interface changes, any events that cause the tunnel status and operational information to be changed which can affect the routes.

Output resembles the following.

```
//Disable the tunnel on local side
TNNL_GRE:EVENTS:tnnl 1-IP notify -  Removing Tunnel IP - tnnl, state DOWN
TNNL_GRE:EVENTS:tnnl 1-IP notify-Tnnl state DOWN: Notifying all routing protocols
TNNL_GRE:EVENTS:tnnl 1-Deleting NHT for tnnl, nht 0
TNNL_GRE:EVENTS:tnnl 1-Active MP Sending NHT for tnl,nht_idx 0, action Delete
TNNL_GRE:EVENTS:tnnl 1-Update NHT Entry as Tnnl Dest route change for Tunnel

//Enable the tunnel on local side
TNNL_GRE:EVENTS:tnnl 1-IP notify -  Adding Tunnel IP - tnnl, state UP
TNNL_GRE:EVENTS:tnnl 1-IP notify-Tnnl state UP: Notifying all routing protocols
TNNL_GRE:EVENTS:tnnl 1-NHT entry Created for tnnl, nht 0, IP 11.11.11.21
TNNL_GRE:EVENTS:tnnl 1-Creating NHT and sending info to LP for tnnl
TNNL_GRE:EVENTS:tnnl 1-Active MP Sending NHT for tnl,nht_idx 0, action Add
TNNL_GRE:EVENTS:tnnl 1-Adding NHT index for tnnl, nht 0, outport 2/12
```

### debug iptunnel ipc

**Syntax:** [**no**] **debug iptunnel ipc** [**range** <*tunnel-id low*> **-** <*tunnel-id high*>]

This command displays IPC messages related to debugging information for the specified range of tunnels. Output resembles the following.

With tunnel disabled manually:

```
TNNL_GRE:IPC:tnnl 1-Sending GRE Tunnel Update for tnnl
TNNL_GRE:IPC:tnnl 1-One Tunnel update - src-ip 11.11.11.1, dst-ip 11.11.11.21

With tunnel enabled manually
TNNL_GRE:IPC:tnnl 1-Sending GRE Tunnel Update for tnnl
TNNL_GRE:IPC:tnnl 1-Tnnl is UP
TNNL_GRE:IPC:tnnl 1-One Tunnel update - src-ip 11.11.11.1, dst-ip 11.11.11.21
```

### debug iptunnel keepalives

**Syntax:** [**no**] **debug iptunnel keepalives** [**range** <*tunnel-id low*> **-** <*tunnel-id high*>]

This command displays keepalive events for the specified range of tunnels.This command dispjays messages about the receiving/transmitting of keepalive packets, keepalive timer actions, bringing up/down the tunnel based on keepalive actions.

**NOTE**
This command only applies to GRE tunnels and does not work for IPv6 tunnels.

Normal keepalive messages:

```
TNNL_GRE:KALIVE:tnnl 1-Keepalive Timer- For Tunnel, remaining time 10
TNNL_GRE:KALIVE:tnnl 1-TX Keepalive packet on tnnl, src 11.11.11.1, dst
11.11.11.21
TNNL_GRE:KALIVE:tnnl 1-Keepalive packet received at MP for tunnel
TNNL_GRE:KALIVE:tnnl 1-Rx Keepalive packet, src 11.11.11.21, dst 11.11.11.1
TNNL_GRE:KALIVE:tnnl 1-Reset the keepalives in the Keepalive List for tunnel
```

With tunnel disabled on the remote side so that keepalive brings down the tunnel:

```
TNNL_GRE:KALIVE:tnnl 1-Bring DOWN GRE Tunnel due to keepalive timeout
TNNL_GRE:EVENTS:tnnl 1-IP notify -  Removing Tunnel IP - tnnl, state DOWN
TNNL_GRE:EVENTS:tnnl 1-IP notify-Tnnl state DOWN: Notifying all routing protocols
TNNL_GRE:KALIVE:tnnl 1-TX Keepalive packet on tnnl, src 11.11.11.1, dst
11.11.11.21
```

With tunnel enabled on the remote side so that the tunnel comes back up on the local side:

```
TNNL_GRE:KALIVE:tnnl 1-TX Keepalive packet on tnnl, src 11.11.11.1, dst
11.11.11.21
TNNL_GRE:KALIVE:tnnl 1-Keepalive packet received at MP for tunnel
TNNL_GRE:KALIVE:tnnl 1-Rx Keepalive packet, src 11.11.11.21, dst 11.11.11.1
TNNL_GRE:KALIVE:tnnl 1-Reset the keepalives in the Keepalive List for tunnel
TNNL_GRE:KALIVE:tnnl 1-Bring UP GRE Tunnel as keepalive response is received
TNNL_GRE:EVENTS:tnnl 1-IP notify -  Adding Tunnel IP - tnnl, state UP
TNNL_GRE:EVENTS:tnnl 1-IP notify-Tnnl state UP: Notifying all routing protocols
```

### debug iptunnel tunnel-type

**Syntax:** [**no**] **debug iptunnel tunnel-type** <**gre** | **ipv6**> [**range** <*tunnel-id low*> **-** <*tunnel-id high*>]

You can specify the type of tunnel to debug. The valid options are **gre** for GRE tunnels and **ipv6** for both manual and 6to4 IPv6 tunnels. Within a specified range of tunnels, information is filtered for the tunnel type if it is configured. If a tunnel type is not configured, debug messages are printed based on range only.

The **no debug iptunnel tunnel-type** command resets the previously configured tunnel-type (if any), and.prints messages for tunnels within the specified range.

### debug iptunnel packets

**Syntax:** [**no**] **debug iptunnel packets** [**range** <*tunnel-id low*> **-** <*tunnel-id high*>]

This command displays packet processing details for the specified range of tunnels. Output shows the packets received and transmitted on any particular tunnel including control packets, keepalive packets, etc.

Output resembles the following.

```
TNNL_GRE:PKTS:tnnl 1-Sending packets to tunnel, dest 111.111.111.21
TNNL_GRE:PKTS:tnnl 1-Sending packets to tunnel, dest 111.111.111.21
```

### debug iptunnel statistics

**Syntax:** [**no**] **debug iptunnel statistics** [**range** <*tunnel-id low*> **-** <*tunnel-id high*>]

Displays tunnel statistics for the specified range of tunnels, including messages about statistics collection, statistics IPC actions, background polling of statistics information, etc.

Output resembles the following:

```
TNNL_GRE:STATS: tnnl 1-Statistics sync processed for IP tunnel
TNNL_GRE:STATS: tnnl 1-Statistics sync processed for IP tunnel
TNNL_GRE:STATS: tnnl 1-Statistics sync processed for IP tunnel
TNNL_GRE:STATS: tnnl 1-LP to MP IPC:PPCR0: recv_ucast 2, recv_mcast 0, xmit_ucast
0, xmit_mcast 0
```

Chapter

# 11

# Software Licensing Diagnostics

## Software licensing

Software licensing enables premium features in the software.Software licensing uses the same command parsing control that is used in the EEPROM version of packaging. So it is built on an already proven infrastructure.Software licenses can be pre-installed in the factory or ordered and installed on demand by customer.

### Software licensing show command

**show license**

**Syntax: show license**

This command displays the licenses in the system, as shown in the following example:

```
PowerConnect#show license
Total no. of entries: 6
Index      Package Name      Lid        Valid  Type    Period
1       NI-CES-2024-L3U   egut-cdOJ     yes    trial   2 hours
2       NI-CES-2024-L3U   egut-cdOJ     yes    trial   48 hours
3       NI-CER-2048-ADV   ucHJFOFGOH    no     trial   1 days
4       NI-CER-2048-ADV   ucHJFOFGOH    no     normal  unlimited
5       NI-CES-2024-L3U   egut-cdOJ     yes    normal  unlimited
```

For MLX-X cards, the output is as follows:

```
PowerConnect#show license
Index  Package Name         Lid           Slot   License    Status     License
                                                 Type                  Period
1      BR-MLX-10Gx4-MLUpg   doaFIGFhFGG   S4     normal     active     unlimited
2      BR-MLX-10Gx4-MLUpg   doaFIGFhFGG   S4     trial      not used   3 hours
3      BR-MLX-1Gx24-MLUpg   ONMLKJIHG     S3     normal     active     unlimited
4      BR-MLX-1Gx24-MLUpg   ONMLKJIHG     S3     trial      not used   3 hours
```

### Software licensing debug command

**debug license**

**Syntax:** [**no**] **debug license**

This command is used to display the package information on which the license has been loaded. It is encoded as a Hex value. This information can be displayed only when the **show** command is used with the license index, for example, **show license 1**:

Before enabling debugging:

```
PowerConnect#show license
Index      Package Name         Lid          License Type    Status     License
Period
1        NI-CES-2048-L3U     ucGNFOGHGO   normal        active    unlimited
PowerConnect#show license 1
License information for license <1>:
        +package name:        NI-CES-2048-L3U
        +lid:                 ucGNFOGHGO
        +license type:        normal
        +status:              active
        +license period:      unlimited
```

After enabling debugging:

```
PowerConnect#debug license
License all debugging ON
PowerConnect#show license 1
License information for license <1>:
        +package name:        NI-CES-2048-L3U
        +lid:                 ucGNFOGHGO
        +license type:        normal
        +status:              active
        +license period:      unlimited
Dell license information:
        +pkg info:     0X00000003
```

# Diagnostic Command Index

## C



## D

# H

# I



# P



# S

# T



# X