CHAPTER 59

Troubleshooting Your Fabric

This chapter describes basic troubleshooting methods used to resolve issues with switches. This chapter includes the following sections:

fctrace

The fctrace feature allows you to:

- Trace the route followed by data traffic.
- Compute inter-switch (hop-to-hop) latency.

You can invoke fctrace by providing the FC ID, the N port, or the NL port WWN, or the device alias of the destination. The frames are routed normally as long as they are forwarded through TE ports.

Once the frame reaches the edge of the fabric (the F port or FL port connected to the end node with the given port WWN or the FC ID), the frame is looped back (swapping the source ID and the destination ID) to the originator.

If the destination cannot be reached, the path discovery starts, which traces the path up to the point of failure.

Note The fctrace feature works only on TE ports. Make sure that only TE ports exist in the path to the destination. In case there is an E port in the path, the fctrace frame is dropped by that switch. Also, fctrace times out in the originator, and path discovery does not start.

Tip You cannot use the fctrace feature in a locally configured VSAN interface (IPFC interface), but you can trace the route to a VSAN interface configured in other switches.

To perform a fctrace operation, follow this step:

	Command	Purpose

Step 1	`switch# fctrace fcid 0xd70000 vsan 1`
	`Route present for :  0xd70000`
	`20:00:00:0b:46:00:02:82(0xfffcd5)`
	`Timestamp Invalid.`
	`20:00:00:05:30:00:18:db(0xfffcd7)`
	`Timestamp Invalid.`
	`20:00:00:05:30:00:18:db(0xfffcd7)`
		Invokes fctrace for the specified FC ID of the destination N port.

	`switch# fctrace pwwn 21:00:00:e0:8b:06:d9:1d vsan 1 timeout 5`
	`Route present for : 21:00:00:e0:8b:06:d9:1d`
	`20:00:00:0b:46:00:02:82(0xfffcd5)`
	`Timestamp Invalid.`
	`20:00:00:05:30:00:18:db(0xfffcd7)`
	`Timestamp Invalid.`
	`20:00:00:05:30:00:18:db(0xfffcd7)`
		Invokes fctrace using the pWWN of the destination N port.
		By default the period to wait before timing out is 5 seconds, The range is from one through 10 seconds.

	`switch# fctrace device-alias disk1 v 1`
	`Route present for : 22:00:00:0c:50:02:ce:f8`
	`20:00:00:05:30:00:31:1e(0xfffca9)`
		Invokes fctrace using the device alias of the destination N port.

fcping

The fcping feature verifies reachability of a node by checking its end-to-end connectivity. You can invoke the fcping feature by providing the FC ID, the destination port WWN, or the device alias information.

To perform a fcping operation, follow these steps:

	Command	Purpose

Step 1	`switch# fcping fcid 0xd70000 vsan 1`
	`28 bytes from 0xd70000 time = 730 usec`
	`28 bytes from 0xd70000 time = 165 usec`
	`28 bytes from 0xd70000 time = 262 usec`
	`28 bytes from 0xd70000 time = 219 usec`
	`28 bytes from 0xd70000 time = 228 usec`
	
	`5 frames sent, 5 frames received, 0 timeouts`
	`Round-trip min/avg/max = 165/270/730 usec`
		Invokes fcping for the specified pWWN or the FC ID of the destination. By default, five frames are sent.

	`switch# fcping fcid 0xd70000 vsan 1 count 10`
	`28 bytes from 0xd70000 time = 730 usec`
	`28 bytes from 0xd70000 time = 165 usec`
	`28 bytes from 0xd70000 time = 262 usec`
	`28 bytes from 0xd70000 time = 219 usec`
	`28 bytes from 0xd70000 time = 228 usec`
	`28 bytes from 0xd70000 time = 230 usec`
	`28 bytes from 0xd70000 time = 230 usec`
	`28 bytes from 0xd70000 time = 225 usec`
	`28 bytes from 0xd70000 time = 229 usec`
	`28 bytes from 0xd70000 time = 183 usec`
	
	`10 frames sent, 10 frames received, 0 timeouts`
	`Round-trip min/avg/max = 165/270/730 usec`
		Sets the number of frames to be sent using the count option. The range is from 0 through 2147483647. A value of 0 pings forever.

	`switch# fcping fcid 0xd500b4 vsan 1 timeout 10`
	`28 bytes from 0xd500b4 time = 1345 usec`
	`...`
	`5 frames sent, 5 frames received, 0 timeouts`
	`Round-trip min/avg/max = 340/581/1345 usec`
		Sets the timeout value. The default period to wait is 5 seconds. The range is from 1 through 10 seconds.

	`switch# fcping device-alias disk1 vsan 1`
	`28 bytes from 22:00:00:0c:50:02:ce:f8 time = 1883 usec`
	`28 bytes from 22:00:00:0c:50:02:ce:f8 time = 493 usec`
	`28 bytes from 22:00:00:0c:50:02:ce:f8 time = 277 usec`
	`28 bytes from 22:00:00:0c:50:02:ce:f8 time = 391 usec`
	`28 bytes from 22:00:00:0c:50:02:ce:f8 time = 319 usec`
	
	`5 frames sent, 5 frames received, 0 timeouts`
	`Round-trip min/avg/max = 277/672/1883 usec`
		Invokes fcping for the specified device alias of the destination.

Step 2	`switch# fcping fcid 0x010203 vsan 1`
	`No response from the N port.`
	
	`switch# fcping pwwn 21:00:00:20:37:6f:db:dd vsan 1`
	`28 bytes from 21:00:00:20:37:6f:db:dd time = 1454 usec`
	`...`
	`5 frames sent, 5 frames received, 0 timeouts`
	`Round-trip min/avg/max = 364/784/1454 usec`
		Issues a `No response from the N port` message even when the N port or NL port is active. This is due to resource exhaustion at the N port or NL port.
		
		Retry the command a few seconds later.

Verifying Switch Connectivity

You can verify connectivity to a destination switch.

Note The FC ID variable used in this procedure is the domain controller address; it is not a duplication of the domain ID.

To verify connectivity to a destination switch, follow these steps:

	Command	Purpose

Step 1	`switch# show fcdomain domain-list vsan 200`
	`Number of domains: 7`
	`Domain ID WWN`
	`--------- -----------------------`
	`0x01(1) 20:c8:00:05:30:00:59:df [Principal]`
	`0x02(2) 20:c8:00:0b:5f:d5:9f:c1`
	`0x6f(111) 20:c8:00:05:30:00:60:df`
	`0xda(218) 20:c8:00:05:30:00:87:9f [Local]`
	`0x06(6) 20:c8:00:0b:46:79:f2:41`
	`0x04(4) 20:c8:00:05:30:00:86:5f`
	`0x6a(106) 20:c8:00:05:30:00:f8:e3`
		Displays the destination switch's domain ID.
		To obtain the domain controller address, concatenate the domain ID with FFFC. For example, if the domain ID is 0xda(218), the concatenated ID is 0xfffcda.

Step 2	`switch# fcping fcid 0xFFFCDA vsan 200`
	`28 bytes from 0xFFFCDA time = 298 usec`
	`28 bytes from 0xFFFCDA time = 260 usec`
	`28 bytes from 0xFFFCDA time = 298 usec`
	`28 bytes from 0xFFFCDA time = 294 usec`
	`28 bytes from 0xFFFCDA time = 292 usec`
	
	`5 frames sent, 5 frames received, 0 timeouts`
	`Round-trip min/avg/max = 260/288/298 usec`
		Verifies reachability of the destination switch by checking its end-to-end connectivity.

Cisco Fabric Analyzer

Fibre Channel protocol analyzers capture, decode, and analyze frames and ordered sets on a link. Existing Fibre Channel analyzers can capture traffic at wire rate speed. They are expensive and support limited frame decoding. Also, to snoop traffic, the existing analyzers disrupt the traffic on the link while the analyzer is inserted into the link.

The Cisco MDS 9000 Family switches support protocol analysis within a storage network with the Cisco Fabric Analyzer. You can capture Fibre Channel control traffic from a switch and decode it without having to disrupt any connectivity, and without having to be local to the point of analysis.

The Cisco Fabric Analyzer is based on two popular public-domain software applications:

- libpcap—See http://www.tcpdump.org.
- Ethereal—See http://www.ethereal.com.

Note The Cisco Fabric Analyzer is useful in capturing and decoding control traffic, not data traffic. It is suitable for control path captures, and is not intended for high-speed data path captures.

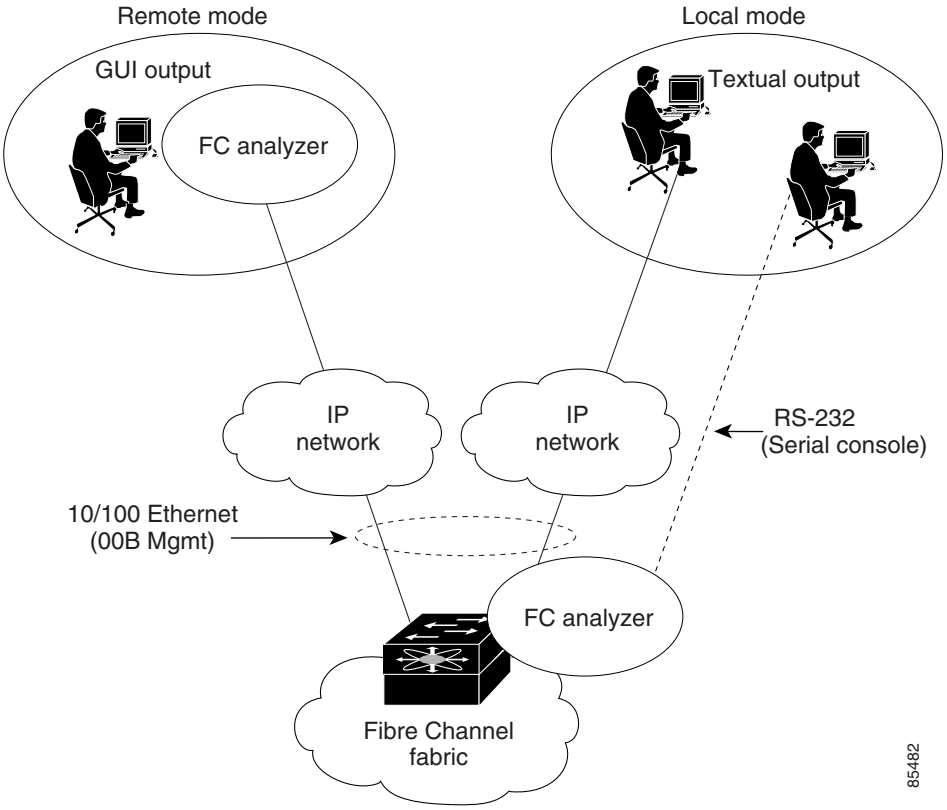
This section includes the following topics:

About the Cisco Fabric Analyzer

The Cisco Fabric Analyzer consists of two separate components (see Figure 59-1):

- Software that runs on the Cisco MDS 9000 Family switch and supports two modes of capture:
  - A text-based analyzer that supports local capture and decodes captured frames
  - A daemon that supports remote capture
- GUI-based client that runs on a host that supports libpcap such as Windows or Linux and communicates with the remote capture daemon in a Cisco MDS 9000 Family switch.

Figure 59-1 Cisco Fabric Analyzer Usage



Local Text-Based Capture

This component is a command-line driven text-based interface that captures traffic to and from the supervisor module in a Cisco MDS 9000 Family switch. It is a fully functional decoder that is useful for quick debug purposes or for use when the remote capture daemon is not enabled. Additionally, because this tool is accessed from within the Cisco MDS 9000 Family switch, it is protected by the roles-based policy that limits access in each switch.

See the "Capturing Frames Locally" section on page 59-7.

Remote Capture Daemon

This daemon is the server end of the remote capture component. The Ethereal analyzer running on a host is the client end. They communicate with each other using the Remote Capture Protocol (RPCAP). RPCAP uses two endpoints, a TCP-based control connection and a TCP or UDP-based data connection based on TCP (default) or UDP. The control connection is used to remotely control the captures (start or stop the capture, or specify capture filters). Remote capture can only be performed to explicitly configured hosts. This technique prevents an unauthorized machine in the network from snooping on the control traffic in the network.

RPCAP supports two setup connection modes based on firewall restrictions.

- Passive mode (default)—The configured host initiates connection to the switch. Multiple hosts can be configured to be in passive mode and multiple hosts can be connected and receive remote captures at the same time.
- Active mode—The switch initiates the connection to a configured host—one host at a time.

Using capture filters, you can limit the amount of traffic that is actually sent to the client. Capture filters are specified at the client end—on Ethereal, not on the switch.

See the "Sending Captures to Remote IP Addresses" section on page 59-8.

GUI-Based Client

The Ethereal software runs on a host, such as a PC or workstation, and communicates with the remote capture daemon. This software is available in the public domain from http://www.ethereal.com. The Ethereal GUI front-end supports a rich interface such as a colorized display, graphical assists in defining filters, and specific frame searches. These features are documented on Ethereal's web site.

While remote capture through Ethereal supports capturing and decoding Fibre Channel frames from a Cisco MDS 9000 Family switch, the host running Ethereal does not require a Fibre Channel connection to the switch. The remote capture daemon running on the switch sends the captured frames over the out-of-band Ethernet management port. This capability allows you to capture and decode Fibre Channel frames from your desktop or laptop.

See the "Displaying Captured Frames" section on page 59-9.

Configuring the Cisco Fabric Analyzer

You can configure the Cisco Fabric Analyzer to perform one of two captures.

- Local capture—The command setting to enable a local capture cannot be saved to persistent storage or synchronized to standby. Launches the textual version on the fabric analyzer directly on the console screen. The capture can also be saved on the local file system.

- Remote capture—The command setting to enable a remote capture can be saved to persistent storage. It can be synchronized to the standby supervisor module and a stateless restart can be issued, if required.

To use the Cisco Fabric Analyzer feature, traffic should be flowing to or from the supervisor module.

Capturing Frames Locally

To capture frames locally, follow these steps:

	Command	Purpose
Step 1	`switch# config t` `switch(config)#`	Enters configuration mode.
	Note The options within Step 2 may be performed in any order.	

Step 2	`switch(config)# fcanalyzer local`
	`Capturing on eth2`
	`switch(config)#`
		Begins capturing the frames locally (supervisor module).

	`switch(config)# fcanalyzer local brief`
	`Capturing on eth2`
	`switch(config)#`
		Displays the protocol summary in a brief format.

	`switch(config)# fcanalyzer local display-filter SampleF` `Capturing on eth2`	Displays the filtered frames.

	`switch(config)# fcanalyzer local limit-frame-size 64`
	`Capturing on eth2`
	`switch(config)#`
		Limits the size of the frame capture to the first 64 bytes. The allowed range is 64 to 65536 bytes.

	`switch(config)# fcanalyzer local limit-captured-frames 10`
	`Capturing on eth2`
	`switch(config)#`
		Limits the number of frames captured to 10. The allowed range is 0 to 2147483647 frames and the default is 100 frames. Use 0 if you do not want to limit the number of captured frames.

	Note Press **Ctrl-c** to stop a capture. Otherwise, the capture stops automatically after capturing 100 frames. You can change this default using the **fcanalyzer local limit-captured-frames** ***number*** command.	

Step 3	`switch(config)# fcanalyzer local write volatile:sample`
	`Capturing on eth2`
	`switch(config)#`
		Saves the captured frames to a specified file (sample) in the volatile: directory.
		Note Optionally, you can save the specified file to the slot0: directory.

	Note The final file name is the capture file called either SampleFile_00000_*dateandtime* or SampleFile_00001_*dateandtime*. For example, “SampleFile_00000_20021110223833” or “SampleFile_00001_20021110243833”. The maximum size of a file that can be written to is 10 MB.	

Sending Captures to Remote IP Addresses

Caution You must use the eth2 interface to capture control traffic on a supervisor module.

To capture frames remotely using IPv4, follow these steps:

	Command	Purpose
Step 1	`switch# config t` `switch(config)#`	Enters configuration mode.
Step 2	`switch(config)# fcanalyzer remote 10.21.0.3`	Configures the remote IPv4 address (10.21.0.3) to which the captured frames are sent.
	`switch(config)# fcanalyzer remote 10.21.0.3 active`	Enables active mode (passive is the default) with the remote host. Ethereal is assumed to be running when the capture is performed. The switch tries to connect forever unless a capture stop instruction is sent from Ethereal.
	`switch(config)# fcanalyzer remote 10.21.0.3 active 1`	Enables the active mode for a specified port. The valid port range is 1 to 65535.

To capture frames remotely using IPv6, follow these steps:

	Command	Purpose
Step 1	`switch# config t` `switch(config)#`	Enters configuration mode.
Step 2	`switch(config)# fcanalyzer remote 2001:0DB8:800:200C::417A`	Configures the remote IPv6 address to which the captured frames are sent.
	`switch(config)# fcanalyzer remote 2001:0DB8:800:200C::417A active`	Enables active mode (passive is the default) with the remote host. Ethereal is assumed to be running when the capture is performed. The switch tries to connect forever unless a capture stop instruction is sent from Ethereal.
	`switch(config)# fcanalyzer remote 2001:0DB8:800:200C::417A active 1`	Enables the active mode for a specified port. The valid port range is 1 to 65535.

To capture remote traffic, use one of the following options:

- The capture interface can be specified in Ethereal as the remote device:

  ```
  rpcap://<ipaddress or switch hostname>/eth2
  ```

  For example:

  ```
  rpcap://cp-16/eth2
  rpcap://17.2.1.1/eth2
  ```

- The capture interface can be specified either in the capture dialog box or by using the -i option at the command line when invoking Ethereal.

```
ethereal -i rpcap://<ipaddress|hostname>[:<port>]/<interface>
```

For example:

```
ethereal -i rpcap://172.22.1.1/eth2
```

or

```
ethereal -i rpcap://customer-switch.customer.com/eth2
```

Note For example, in a Windows 2000 setup, click **Start** on your desktop and select **Run**. In the resulting Run window, type the required command line option in the Open field.

Clearing Configured fcanalyzer Information

Use the **clear fcanalyzer** command to clear the entire list of configured hosts. Note that the existing connections are not terminated.

Displaying Configured Hosts

Use the **show fcanalyzer** command to display the list of hosts configured for a remote capture. See Example 59-1.

Example 59-1 Displays Configured Hosts

```
switch# show fcanalyzer
PassiveClient = 10.21.0.3
PassiveClient = 10.21.0.3
ActiveClient = 10.21.0.3, DEFAULT
```

Note The DEFAULT in the ActiveClient line indicates that the default port is used.

Displaying Captured Frames

You can selectively view captured frames by using the display filters feature. For example, instead of viewing all the frames from a capture, you may only want to view Exchange Link Protocol (ELP) request frames. This feature only limits the captured view—it does not affect the captured or the saved frames. Procedures to specify, use, and save display filters are already documented in the Ethereal web site (http://www.ethereal.com). Some examples of how you can use this feature are as follows:

- To view all packets in a specified VSAN, use this expression:

  ```
  mdshdr.vsan == 2
  ```

- To view all SW_ILS frames, use this expression:

  ```
  fcswils
  ```

- To view class F frames, use this expression:

  ```
  mdshdr.sof == SOFf
  ```

- To view all FSPF frames, use this expression:

  ```
  swils.opcode == HLO || swils.opcode == LSU || swils.opcode == LSA
  ```

- To view all FLOGI frames, use this expression:

  ```
  fcels.opcode == FLOGI
  ```

- To view all FLOGI frames in VSAN 1, use this expression:

  ```
  fcels.opcode == FLOGI && mdshdr.vsan == 2
  ```

- To view all name server frames, use this expression:

  ```
  dNS
  ```

Defining Display Filters

Display filters limit the frames that can be displayed, but not what is captured (similar to any view command). The filters to be displayed can be defined in multiple ways in the GUI application:

- Auto-definition
- Manual definition
- Assisted manual definition
- Only manual definition in local capture
- No assists

Regardless of the definition, each filter must be saved and identified with a name.

Note This GUI-assisted feature is part of Ethereal and you can obtain more information from http://www.ethereal.com.

Examples of Display Filters

Some examples of using display filters with the Fabric Analyzer local are provided in this section. The **brief** option is used in all examples to restrict the size of the output. See Example 59-2.

Example 59-2 Displays Only Fabric Login Server Traffic on VSAN 1

```
switch(config)# fcanalyzer local brief display-filter
(mdshdr.vsan==0x01)&&((fc.d_id==\"ff.ff.fe\"\|\|fc.s_id==\"ff.ff.fe\"))
Capturing on eth2
8.904145   00.00.00 -> ff.ff.fe   FC ELS 1    0x28f8 0xffff  0x3 ->  0xf FLOGI
8.918164   ff.ff.fe -> 79.03.00   FC ELS 1    0x28f8 0x12c6 0xff ->  0x0 ACC (FLOGI)
```

You can trace all frames to and from a particular N port device. For example, you can observe RSCNs from the Fabric Controller and registration, and/or you can query requests to the name server. See Example 59-3.

Note The filter requires prior knowledge of the FC ID that is assigned to the N port. Issue the **show flogi database interfac**e command before running fcanalyzer to obtain the FC ID. In this example, the N port FC ID is 79.03.00.

Example 59-3 Displays All Traffic for a Particular N Port on VSAN 1

```
switch(config)# fcanalyzer local brief
display-filter(mdshdr.vsan==0x01)&&((fc.d_id==\"79.03.00\"\|\|fc.s_id==\"79.03.00\"))
Capturing on eth2
8.699162   ff.ff.fe -> 79.03.00   FC ELS 1    0x35b8 0x148e 0xff ->  0x0 ACC (FLOGI)
8.699397   79.03.00 -> ff.ff.fc   FC ELS 1    0x35d0 0xffff  0x3 ->  0xf PLOGI
8.699538   ff.ff.fc -> 79.03.00   FC ELS 1    0x35d0 0x148f 0xff ->  0x0 ACC (PLOGI)
8.699406   79.03.00 -> ff.ff.fd   FC ELS 1    0x35e8 0xffff  0x3 ->  0xf SCR
8.700179   79.03.00 -> ff.ff.fc   dNS    1    0x3600 0xffff  0x3 ->  0xf GNN_FT
8.702446   ff.ff.fd -> 79.03.00   FC ELS 1    0x35e8 0x1490 0xff ->  0x0 ACC (SCR)
8.704210   ff.ff.fc -> 79.03.00   dNS    1    0x3600 0x1491 0xff ->  0x0 ACC (GNN_FT)
8.704383   79.03.00 -> ff.ff.fc   dNS    1    0x3618 0xffff  0x3 ->  0xf GPN_ID
8.707857   ff.ff.fc -> 79.03.00   dNS    1    0x3618 0x1496 0xff ->  0x0 ACC (GPN_ID)
```

The VSAN ID is specified in hex. See Example 59-4.

Example 59-4 Displays All Traffic for a Specified VSAN

```
switch(config)# fcanalyzer local brief display-filter mdshdr.vsan==0x03e7
Capturing on eth2
12.762577   ff.ff.fd -> ff.ff.fd   SW_ILS 999   0xb2c 0xffff  0x1 ->  0xf HLO
12.762639   ff.ff.fd -> ff.ff.fd   FC     999   0xb2c  0xd32 0xff ->  0x0 Link Ctl, ACK1
13.509979   ff.ff.fd -> ff.ff.fd   SW_ILS 999   0xd33 0xffff 0xff ->  0x0 HLO
13.510918   ff.ff.fd -> ff.ff.fd   FC     999   0xd33  0xb2d  0x1 ->  0xf Link Ctl, ACK1
14.502391   ff.fc.64 -> ff.fc.70   SW_ILS 999   0xd34 0xffff 0xff ->  0x0 SW_RSCN
14.502545   ff.ff.fd -> 64.01.01   FC ELS 999   0xd35 0xffff 0xff ->  0x0 RSCN
14.502804   64.01.01 -> ff.ff.fd   FC ELS 999   0xd35  0x215  0x0 ->  0xf ACC (RSCN)
14.503387   ff.fc.70 -> ff.fc.64   FC     999   0xd34  0xb2e  0x1 ->  0xf Link Ctl, ACK1
14.503976   ff.fc.70 -> ff.fc.64   SW_ILS 999   0xd34  0xb2e  0x1 ->  0xf SW_ACC (SW_RSCN)
14.504025   ff.fc.64 -> ff.fc.70   FC     999   0xd34  0xb2e 0xff ->  0x0 Link Ctl, ACK1
```

By excluding FSPF `hellos` and `ACK1`, you can focus on the frames of interest. See Example 59-5.

Example 59-5 Displays All VSAN 1 Traffic Excluding FSPF Hellos and ACK1 Frames.

```
switch(config)# fcan lo bri dis
(mdshdr.vsan==0x01)&&not((swils.opcode==0x14)or(fc.r_ctl==0xc0))
Capturing on eth2
10.589934   ff.fc.79 -> ff.fc.7a   FC-FCS 1    0x1b23 0xffff 0xff ->  0x0 GCAP
10.591253   ff.fc.7a -> ff.fc.79   FC-FCS 1    0x1b23 0x2f70  0x4 ->  0xf MSG_RJT (GCAP)
25.277981   ff.fc.79 -> ff.fc.7a   SW_ILS 1    0x1b27 0xffff 0xff ->  0x0 SW_RSCN
25.278050   ff.fc.79 -> ff.fc.89   SW_ILS 1    0x1b28 0xffff 0xff ->  0x0 SW_RSCN
25.279232   ff.fc.89 -> ff.fc.79   SW_ILS 1    0x1b28 0xadd7  0x5 ->  0xf SW_ACC (SW_RSCN)
25.280023   ff.fc.7a -> ff.fc.79   Unzoned NS 1    0x3b2b 0xffff  0x5 ->  0xf GE_PT
25.280029   ff.fc.7a -> ff.fc.79   SW_ILS 1    0x1b27 0x2f71  0x4 ->  0xf SW_ACC (SW_RSCN)
25.282439   ff.fc.79 -> ff.fc.7a   dNS    1    0x3b2b 0x1b29 0xff ->  0x0 RJT (GE_PT)
38.249966   00.00.00 -> ff.ff.fe   FC ELS 1    0x36f0 0xffff  0x3 ->  0xf FLOGI
38.262622   ff.ff.fe -> 79.03.00   FC ELS 1    0x36f0 0x1b2b 0xff ->  0x0 ACC (FLOGI)
38.262844   79.03.00 -> ff.ff.fc   FC ELS 1    0x3708 0xffff  0x3 ->  0xf PLOGI
38.262984   ff.ff.fc -> 79.03.00   FC ELS 1    0x3708 0x1b2c 0xff ->  0x0 ACC (PLOGI)
38.262851   79.03.00 -> ff.ff.fd   FC ELS 1    0x3720 0xffff  0x3 ->  0xf SCR
38.263514   ff.fc.79 -> ff.fc.7a   SW_ILS 1    0x1b2e 0xffff 0xff ->  0x0 SW_RSCN
38.263570   ff.fc.79 -> ff.fc.89   SW_ILS 1    0x1b2f 0xffff 0xff ->  0x0 SW_RSCN
38.263630   79.03.00 -> ff.ff.fc   dNS    1    0x3738 0xffff  0x3 ->  0xf GNN_FT
38.263884   ff.ff.fd -> 79.03.00   FC ELS 1    0x3720 0x1b2d 0xff ->  0x0 ACC (SCR)
38.264066   ff.fc.89 -> ff.fc.79   SW_ILS 1    0x1b2f 0xaddf  0x5 ->  0xf SW_ACC (SW_RSCN)
38.264417   ff.fc.89 -> ff.fc.79   dNS    1    0xade0 0xffff  0x5 ->  0xf GE_ID
38.264585   ff.fc.79 -> ff.fc.89   dNS    1    0xade0 0x1b31 0xff ->  0x0 ACC (GE_ID)
38.265132   ff.ff.fc -> 79.03.00   dNS    1    0x3738 0x1b30 0xff ->  0x0 ACC (GNN_FT)
38.265210   ff.fc.7a -> ff.fc.79   Unzoned NS 1    0x3b2f 0xffff  0x5 ->  0xf GE_PT
38.265414   79.03.00 -> ff.ff.fc   dNS    1    0x3750 0xffff  0x3 ->  0xf GPN_ID
```

```
38.265502   ff.fc.7a -> ff.fc.79   SW_ILS 1    0x1b2e 0x2f73  0x4 ->  0xf SW_ACC (SW_RSCN)
38.267196   ff.fc.79 -> ff.fc.7a   dNS    1    0x3b2f 0x1b32 0xff ->  0x0 ACC (GE_PT)
```

Use this command to focus on TE port initialization. This example allows two VSANs on the TE port and the port VSAN is 666. Hence the ELP, ESC, and EPP (0x71) go out on VSAN 666. Once the EPP negotiation is complete, we see EFP, DIA, RDI, MR, FSPF, and other updates flow for each allowed VSAN. See Example 59-6.

Example 59-6 Displays SW_ILS Traffic Between Fabric Controllers for all VSANs and Exclude FSPF Hellos and ACK1 Frames.

```
switch(config)# fcan lo bri dis
fc.type==0x22&&((fc.d_id==\"ff.fc.ef\"\|\|fc.s_id==\"ff.fc.ef\"))
Warning:Couldn't obtain netmask info (eth2:no IPv4 address assigned).
Capturing on eth2
9.472181 ff.fc.ef -> ff.fc.61 0x5e0a 0xffff SW_ILS ACA
9.472777 ff.fc.61 -> ff.fc.ef 0x5e0a 0x5e09 SW_ILS SW_ACC (ACA)
9.474551 ff.fc.ef -> ff.fc.61 0x5e0b 0xffff SW_ILS SFC
9.475706 ff.fc.61 -> ff.fc.ef 0x5e0b 0x5e0a SW_ILS SW_ACC (SFC)
9.476694 ff.fc.ef -> ff.fc.61 0x5e0c 0xffff SW_ILS UFC
9.483612 ff.fc.61 -> ff.fc.ef 0x5e0c 0x5e0b SW_ILS SW_ACC (UFC)
9.488187 ff.fc.ef -> ff.fc.61 0x5e0d 0xffff SW_ILS RCA
9.493703 ff.fc.61 -> ff.fc.ef 0x5e0d 0x5e0c SW_ILS SW_ACC (RCA)
```

This example focuses on zone server changes. Prior knowledge of the domain controller ID is required. The switch domain ID where the fcanalyzer is run is x79, the domain controller is FF.FC.79. See Example 59-7.

Example 59-7 Display Switch Internal Link Services (SW_ILS) Traffic to and from Fabric Domain Controller ff.fc.79

```
switch(config)# fcan lo bri dis fc.type==0x22&&((fc.d_id==\ ff.fc.79\
\|\|fc.s_id==\"ff.fc.79\"))
Capturing on eth2
64.053927 ff.fc.79 -> ff.fc.7a SW_ILS 0x1e15 0xffff 0xff -> 0x0 ACA
64.053995 ff.fc.79 -> ff.fc.89 SW_ILS 0x1e16 0xffff 0xff -> 0x0 ACA
64.054599 ff.fc.89 -> ff.fc.79 SW_ILS 0x1e16 0xb1e2 0x5 -> 0xf SW_ACC (ACA)
64.054747 ff.fc.7a -> ff.fc.79 SW_ILS 0x1e15 0x3037 0x4 -> 0xf SW_ACC (ACA)
64.057643 ff.fc.79 -> ff.fc.7a SW_ILS 0x1e17 0xffff 0xff -> 0x0 SFC
64.057696 ff.fc.79 -> ff.fc.89 SW_ILS 0x1e18 0xffff 0xff -> 0x0 SFC
64.058788 ff.fc.7a -> ff.fc.79 SW_ILS 0x1e17 0x3038 0x5 -> 0xf SW_ACC (SFC)
64.059288 ff.fc.89 -> ff.fc.79 SW_ILS 0x1e18 0xb1e3 0x5 -> 0xf SW_ACC (SFC)
64.062011 ff.fc.79 -> ff.fc.7a SW_ILS 0x1e19 0xffff 0xff -> 0x0 UFC
64.062060 ff.fc.79 -> ff.fc.89 SW_ILS 0x1e1a 0xffff 0xff -> 0x0 UFC
64.073513 ff.fc.7a -> ff.fc.79 SW_ILS 0x1e19 0x3039 0x5 -> 0xf SW_ACC (UFC)
64.765306 ff.fc.89 -> ff.fc.79 SW_ILS 0x1e1a 0xb1e4 0x5 -> 0xf SW_ACC (UFC)
64.765572 ff.fc.79 -> ff.fc.7a SW_ILS 0x1e1b 0xffff 0xff -> 0x0 RCA
64.765626 ff.fc.79 -> ff.fc.89 SW_ILS 0x1e1c 0xffff 0xff -> 0x0 RCA
64.766386 ff.fc.7a -> ff.fc.79 SW_ILS 0x1e1b 0x303a 0x4 -> 0xf SW_ACC (RCA)
64.766392 ff.fc.89 -> ff.fc.79 SW_ILS 0x1e1c 0xb1e5 0x5 -> 0xf SW_ACC (RCA)
```

Note You can find the fabric domain controller address in the Mgmt-Id field in the **show fcs ie vsan** command output.

```
switch# show fcs ie vsan 999
IE List for VSAN:999
------------------------------------------------------------------------
IE-WWN                       IE-Type             Mgmt-Id         Mgmt-Addr
------------------------------------------------------------------------
```

```
23:e7:00:05:30:00:91:5f     Switch (Remote)     0xfffc04         10.66.78.51
23:e7:00:05:30:00:9b:9f     Switch (Adjacent)   0xfffc01         10.66.78.52
23:e7:00:0d:ec:00:93:81     Switch (Local)      0xfffc79         10.66.78.54
[Total 3 IEs in Fabric]
```

Capture Filters

You can limit what frames are captured by using the capture filters feature in a remote capture. This feature limits the frames that are captured and sent from the remote switch to the host. For example, you can capture only class F frames. Capture filters are useful in restricting the amount of bandwidth consumed by the remote capture.

Unlike display filters, capture filters restrict a capture to the specified frames. No other frames are visible until you specify a completely new capture.

The syntax for capture filters is different from the syntax for display filters. Capture filters use the Berkeley Packet Filter (BPF) library that is used in conjunction with the libpcap freeware. The list of all valid Fibre Channel capture filter fields are provided later in this section.

Procedures to configure capture filters are already documented in the Ethereal web site (http://www.ethereal.com). Some examples of how you can use this feature follows:

- To capture frames only on a specified VSAN, use this expression:

  ```
  vsan = 1
  ```

- To capture only class F frames, use this expression:

  ```
  class_f
  ```

- To capture only class Fibre Channel ELS frames, use this expression:

  ```
  els
  ```

- To capture only name server frames, use this expression:

  ```
  dns
  ```

- To capture only SCSI command frames, use this expression:

  ```
  fcp_cmd
  ```

Note This feature is part of libpcap and you can obtain more information from http://www.tcpdump.org.

Permitted Capture Filters

This section lists the permitted capture filters.

```
o vsan
o src_port_idx
o dst_port_idx
o sof
o r_ctl
o d_id
o s_id
o type
o seq_id
o seq_cnt
o ox_id
```

```
o rx_id
o els
o swils
o fcp_cmd   (FCP Command frames only)
o fcp_data (FCP data frames only)
o fcp_rsp   (FCP response frames only)
o class_f
o bad_fc
o els_cmd
o swils_cmd
o fcp_lun
o fcp_task_mgmt
o fcp_scsi_cmd
o fcp_status
o gs_type      (Generic Services type)
o gs_subtype   (Generic Services subtype)
o gs_cmd
o gs_reason
o gs_reason_expl
o dns   (name server)
o udns (unzoned name server)
o fcs   (fabric configuration server)
o zs    (zone server)
o fc    (use as fc[x:y] where x is offset and y is length to compare)
o els   (use as els[x:y] similar to fc)
o swils (use as swils[x:y] similar to fc)
o fcp   (use as fcp[x:y] similar to fc)
o fcct (use as fcct[x:y] similar to fc)
```

Loop Monitoring

This section includes the following topics:

About Loop Monitoring

By default, loop monitoring is disabled in all switches in the Cisco MDS 9000 Family. When a disk is removed from a loop port, the loop stays active based on the bypass circuit. Thus the disk removal is not known until you try to communicate with the disk. To detect such removals, the disks can be polled periodically (every 20 seconds).

Caution Changes to the loop monitoring feature should be made by an administrator or individual who is completely familiar with switch operations.

Enabling Loop Monitoring

To enable the loop monitoring feature, follow these steps:

	Command	Purpose
Step 1	`switch# config t` `switch(config)#`	Enters configuration mode.
Step 2	`switch(config)# fcinterop loop-monitor`	Enables the loop polling for FL ports.
	`switch(config)# no fcinterop loop-monitor`	Disables (default) the loop monitoring feature and reverts the switch to the factory defaults.

Verifying Loop Monitoring Configuration

Use the show running-config command to verify the loop monitoring configuration.

```
switch# show running-config | include loop-monitor
fcinterop loop-monitor
```

The show tech-support Command

The **show tech-support** command is useful when collecting a large amount of information about your switch for troubleshooting purposes. The output of this command can be provided to technical support representatives when reporting a problem.

The **show tech-support** command displays the output of several **show** commands at once. The output from this command varies depending on your configuration. Use the **show tech-support** command in EXEC mode to display general information about the switch when reporting a problem.

You can choose to have detailed information for each command or even specify the output for a particular interface, module, or VSAN. Each command output is separated by line and the command precedes the output.

Note Explicitly set the **terminal length** command to 0 (zero) to disable auto-scrolling and enable manual scrolling. Use the **show terminal** command to view the configured the terminal size. After obtaining the output of this command, remember to reset your terminal length as required (see the "Setting the Terminal Screen Length" section on page 2-19).

Tip You can save the output of this command to a file by appending **>** (left arrow) and the filename to the **show tech-support** command (see the "Saving Command Output to a File" section on page 2-32). If you save this file, verify you have sufficient space to do so—each of these files may take about 1.8 MB. However, you can zip this file using the **gzip** *filename* command (see the "Compressing and Uncompressing Files" section on page 2-32). Copy the zipped file to the required location using the **copy** command and unzip the file using the **gunzip** command (see the "Copying Files" section on page 2-30).

The default output of the **show tech-support** command includes the output of the following commands:

- **show version**
- **show environment**

- **show module**
- **show hardware**
- **show running-config**
- **show interface**
- **show accounting log**
- **show process**
- **show process log**
- **show processes log details**
- **show flash**

Each command is discussed in both the *Cisco MDS 9000 Family CLI Configuration Guide* and the *Cisco MDS 9000 Family Command Reference*. Refer to the *Cisco MDS 9000 Family Troubleshooting Guide* to obtain debug processes, procedures, and examples.

The show tech-support brief Command

Use the **show tech-support brief** command to obtain a quick, condensed review of your switch configurations. This command provides a summary of the current running state of the switch (see Example 59-8).

The **show tech-support brief** command is useful when collecting information about your switch for troubleshooting purposes. The output of this command can be provided to technical support representatives when reporting a problem.

Tip You can save the output of this command to a file by appending **>** (left arrow) and the filename to the **show tech-support brief** command (see the "Saving Command Output to a File" section on page 2-32).

Example 59-8 Displays the Condensed View of Switch Configurations

```
vegas01# show tech-support brief
Switch Name           : vegas01
Switch Type           : DS-X9216-K9-SUP
Kickstart Image       : 1.3(2) bootflash:///m9200-ek9-kickstart-mz.1.3.1.10.bin
System Image          : 1.3(2) bootflash:///m9200-ek9-mz.1.3.1.10.bin
IP Address/Mask       : 10.76.100.164/24
Switch WWN            : 20:00:00:05:30:00:84:9e
No of VSANs           : 9
Configured VSANs      : 1-6,4091-4093

VSAN    1:    name:VSAN0001, state:active, interop mode:default
              domain id:0x6d(109), WWN:20:01:00:05:30:00:84:9f [Principal]
              active-zone:VR, default-zone:deny

VSAN    2:    name:VSAN0002, state:active, interop mode:default
              domain id:0x7d(125), WWN:20:02:00:05:30:00:84:9f [Principal]
              active-zone:<NONE>, default-zone:deny

VSAN    3:    name:VSAN0003, state:active, interop mode:default
              domain id:0xbe(190), WWN:20:03:00:05:30:00:84:9f [Principal]
              active-zone:<NONE>, default-zone:deny

VSAN    4:    name:VSAN0004, state:active, interop mode:default
```

```
                domain id:0x5a(90), WWN:20:04:00:05:30:00:84:9f [Principal]
                active-zone:<NONE>, default-zone:deny

VSAN    5:      name:VSAN0005, state:active, interop mode:default
                domain id:0x13(19), WWN:20:05:00:05:30:00:84:9f [Principal]
                active-zone:<NONE>, default-zone:deny

VSAN    6:      name:VSAN0006, state:active, interop mode:default
                domain id:0x1f(31), WWN:20:06:00:05:30:00:84:9f [Principal]
                active-zone:<NONE>, default-zone:deny

VSAN 4091:      name:VSAN4091, state:active, interop mode:default
                domain id:0x08(8), WWN:2f:fb:00:05:30:00:84:9f [Principal]
                active-zone:<NONE>, default-zone:deny

VSAN 4092:      name:VSAN4092, state:active, interop mode:default
                domain id:0x78(120), WWN:2f:fc:00:05:30:00:84:9f [Principal]
                active-zone:<NONE>, default-zone:deny

VSAN 4093:      name:VSAN4093, state:active, interop mode:default
                domain id:0x77(119), WWN:2f:fd:00:05:30:00:84:9f [Principal]
                active-zone:<NONE>, default-zone:deny


--------------------------------------------------------------------------------
Interface  Vsan   Admin  Admin   Status          FCOT   Oper  Oper    Port
                  Mode   Trunk                         Mode  Speed   Channel
                         Mode                                 (Gbps)
--------------------------------------------------------------------------------
fc1/1      1      auto   on      fcotAbsent       --     --           --
fc1/2      1      auto   on      fcotAbsent       --     --           --
fc1/3      1      auto   on      fcotAbsent       --     --           --
fc1/4      1      auto   on      fcotAbsent       --     --           --
fc1/5      1      auto   on      notConnected     swl    --           --
fc1/6      1      auto   on      fcotAbsent       --     --           --
fc1/7      1      auto   on      fcotAbsent       --     --           --
fc1/8      1      auto   on      fcotAbsent       --     --           --
fc1/9      1      auto   on      fcotAbsent       --     --           --
fc1/10     1      auto   on      fcotAbsent       --     --           --
fc1/11     1      auto   on      fcotAbsent       --     --           --
fc1/12     1      auto   on      fcotAbsent       --     --           --
fc1/13     1      auto   on      fcotAbsent       --     --           --
fc1/14     1      auto   on      fcotAbsent       --     --           --
fc1/15     1      auto   on      fcotAbsent       --     --           --
fc1/16     1      auto   on      fcotAbsent       --     --           --

--------------------------------------------------------------------------------
Interface          Status                            Speed
                                                     (Gbps)
--------------------------------------------------------------------------------
sup-fc0            up                                1

--------------------------------------------------------------------------------
Interface               Status     IP Address        Speed        MTU
--------------------------------------------------------------------------------
mgmt0                   up         10.76.100.164/24  100 Mbps     1500
```

The show tech-support zone Command

Use the **show tech-support zone** command to obtain information about the zoning configuration on your switch (see Example 59-9).

The output of the **show tech-support zone** command includes the output of the following commands:

- **show zone status vsan**
- **show zone active vsan**
- **show zoneset vsan**
- **show zone vsan**
- **show zone-attribute-group vsan**
- **show zone policy vsan**
- **show zoneset pending active vsan**
- **show zoneset pending vsan**
- **show zone active vsan**
- **show zone pending active vsan**
- show fcalias pending vsan
- **show zone-attribute-group pending vsan**
- **show zone policy pending vsan**
- **show zone pending-diff vsan**
- **show zone analysis active vsan**
- **show zone analysis vsan**
- **show zone ess vsan**
- **show zone statistics vsan**
- **show zone statistics lun-zoning vsan**
- **show zone statistics read-only-zoning vsan**

Tip You can save the output of this command to a file by appending **>** (left arrow) and the filename to the **show tech-support zone** command (see the "Saving Command Output to a File" section on page 2-32).

Example 59-9 Displays the Zoning Configurations

```
switch# show tech-support zone vsan 1

`show zone status vsan 1`
VSAN: 1 default-zone: permit distribute: active only Interop: default
mode: basic merge-control: allow session: none
hard-zoning: enabled
Default zone:
qos: disabled broadcast: disabled ronly: disabled
Full Zoning Database :
Zonesets:0 Zones:0 Aliases: 0
Active Zoning Database :
Name: vhost-zone Zonesets:1 Zones:9
Status: Activation failed [Error: Unknown error Dom 21]:
at 23:36:44 UTC Dec 19 2005
```

The show tech-support port-channel Command

Use the **show tech-support port-channel** command to obtain information about the PortChannel configuration on your switch (see Example 59-10).

The output of the **show tech-support port-channel** command includes the output of the following commands:

- **show port-channel internal event-history all**
- **show port-channel internal event-history errors**
- **show port-channel internal event-history lock**
- **show port-channel internal mem-stats detail**
- **show port-channel usage**
- **show port-channel summary**
- **show port-channel internal database**
- **show port-channel consistency detail**

Tip You can save the output of this command to a file by appending **>** (left arrow) and the filename to the **show tech-support port-channel** command (see the "Saving Command Output to a File" section on page 2-32).

Example 59-10 Displays the PortChannel Configurations

```
switch# show tech-support port-channel
cp: missing destination file
Try `cp --help' for more information.

`show port-channel internal event-history all`
Low Priority Pending queue: len(0), max len(1) [Wed Jan  4 18:29:18 2006]
High Priority Pending queue: len(0), max len(14) [Wed Jan  4 18:29:18 2006]
PCM Control Block info:
pcm_max_channels        : 128
pcm_max_channel_in_use  : 1
has Vegas Line Card
Total of 1 Vegas Line cards
PCM total_vlans info: 0x0
================================================
PORT CHANNELS:
================================================

ALL PORTS:
GigabitEthernet3/1
peer          : 00:00:00:00:00:00:00:00
my wwn        : 00:00:00:00:00:00:00:00
state         : down
update        : none
intent        : unknown
status        : unknown
mode          : on
fcip timeout  : 0 ms
sigloss       : FALSE
flags         :
```

```
cfg flags      :
up_time        : 0 usecs after Thu Jan  1 00:00:00 1970
auto pc        : none
auto retry     : 0
last pcp err   : 0 at 0 usecs after Thu Jan  1 00:00:00 1970
No auto create compat failure
...
```

The show tech-support vsan Command

Use the **show tech-support vsan** command to obtain information about the VSAN configuration on your switch (see Example 59-11).

The output of the **show tech-support vsan** command includes the output of the following commands:

- **show vsan**
- **show vsan membership**
- **show interface brief**
- **show port-channel database**
- **show port-channel consistency**
- **show flogi database vsan**
- **show fcdomain vsan**
- **show fcdomain domain-list vsan**
- **show fcdomain address-allocation vsan**
- **show fcns database vsan**
- show fcs ie vsan
- **show rscn statistics vsan**
- **show fspf vsan**
- **show fspf database vsan**
- **show span session**
- **show snmp**
- **show zone tech-support vsan**

Tip You can save the output of this command to a file by appending **>** (left arrow) and the filename to the **show tech-support vsan** command (see the "Saving Command Output to a File" section on page 2-32).

Example 59-11 Displays the VSAN Configurations

```
switch# show tech-support vsan 1

`show vsan 1`
vsan 1 information
         name:VSAN0001  state:active
         interoperability mode:default
         loadbalancing:src-id/dst-id/oxid
         operational state:up
```

```
`show vsan 1 membership`
vsan 1 interfaces:
        fc3/1   fc3/2   fc3/3   fc3/4   fc3/5   fc3/6   fc3/7   fc3/8
        fc3/9   fc3/10  fc3/11  fc3/12  fc3/13  fc3/14  port-channel 1  iscsi3/1 iscsi3/2
...
```

The show tech-support fcdomain Command

Use the **show tech-support fcdomain** command to obtain information about the fcdomain configuration on your switch (see Example 59-9).

The output of the **show tech-support fcdomain** command includes the output of the following commands:

- **show fcdomain**
- **show fcdomain domain-list**
- **show fcdomain allowed**
- **show fcdomain pending-diff**
- **show fcdomain address-allocation**
- **show fcdomain address-allocation cache**
- **show fcdomain fcid persistent**
- **show fcdomain internal event-history**
- **show fcdomain internal event-history fcid**
- **show fcdomain internal mem-stats detail**
- show fcdomain statistics
- **show fcdomain internal info mts**
- **show fcdomain internal info fcidp-tbl range**

Tip You can save the output of this command to a file by appending **>** (left arrow) and the filename to the **show tech-support fcdomain** command (see the "Saving Command Output to a File" section on page 2-32).

Example 59-12 Displays the fcdomain Configurations

```
switch# show tech-support fcdomain

`show fcdomain status`
fcdomain distribution is disabled

`show fcdomain session-status`

Session parameters for VSAN 1
-----------------------------------
Last Action: none yet
Result: not available

`show fcdomain`
```

```
VSAN 1
The local switch is the Principal Switch.

Local switch run time information:
        State: Stable
        Local switch WWN:    20:01:00:0c:85:90:3e:81
        Running fabric name: 20:01:00:0c:85:90:3e:81
        Running priority: 128
        Current domain ID: 0x72(114)

Local switch configuration information:
        State: Enabled
        FCID persistence: Enabled
        Auto-reconfiguration: Disabled
        Contiguous-allocation: Disabled
        Configured fabric name: 20:01:00:05:30:00:28:df
        Configured priority: 128
        Configured domain ID: 0x00(0) (preferred)

Principal switch run time information:
        Running priority: 128

No interfaces available.
...
```

IP Network Simulator

The IP Network Simulator tool is supported on the 8-port IP Storage Services (IPS-8) module and 4-port IP Storage Services (IPS-4) module only. You must also have either the SAN extension over IP package for IPS-8 modules (SAN_EXTN_OVER_IP) or SAN extension over IP package for IPS-4 modules (SAN_EXTN_OVER_IP_IPS4) so that you can enable the SAN Extension Tuner, which is a prerequisite for enabling and using the network simulator.

Note As of Cisco MDS SAN-OS Release 3.3(1a), IP Network Simulator is supported on the Multiservice Module (MSM) and the Multiservice Modular Switch.

Note You must have a pair of Gigabit Ethernet ports dedicated for each Ethernet path requiring simulation; these ports cannot provide FCIP or iSCSI functionality while simulation occurs. Of course, the remaining ports that are not performing network simulations can run FCIP or iSCSI.

Ports dedicated to network simulation must be adjacent, and always begin with an odd-numbered port. For example, GE 1/1 and GE 1/2 would be a valid pair, while GE 2/2 and GE 2/3 would not.

Network simulator enables you to simulate a variety of IP data network conditions, including the ability to test the impact of network latency. Network simulator is a generic tool that can provide simulation features for all Ethernet traffic; it is not limited to FCIP and iSCSI traffic to or from the Cisco MDS 9000 Family.

The simulation handles full duplex Gigabit Ethernet traffic at full line rate. Figure 59-2 depicts the physical topology using a Cisco MDS 9506 director with an IPS-8 module. GE ports 1 and 2 serve as the network simulator. The FCIP tunnel runs between the Cisco MDS 9506 director port GE 2/1 and the Cisco 9216 module port GE 2/2.

Figure 59-2 Network Simulator - Physical Topology Example

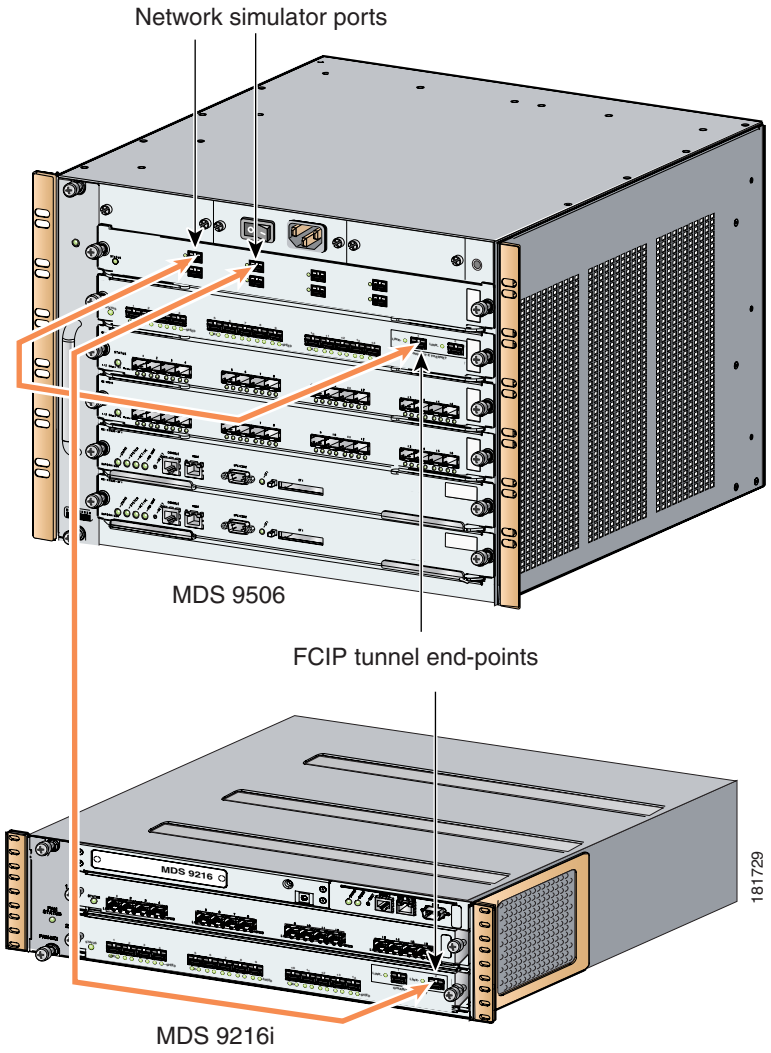
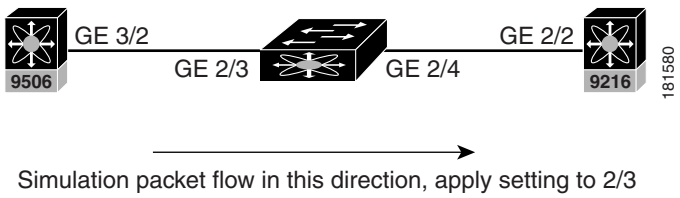


Figure 59-3 depicts the packet flow between the Cisco MDS 9506 and Cisco MDS 9216. Simulations such as delays, drops, and packet reordering are applied independently in each direction. To configure a delay simulation in both directions, you must configure the simulation on both the Cisco MDS 9506 GE 1/1 and 1/2 ports. Simulations are applied to ingress traffic only. All packets received on one Gigabit Ethernet port are sent out of the other Gigabit Ethernet port, and all network configuration simulations are made with respect to the ingress Gigabit Ethernet port.

Figure 59-3 Network Simulator Packet Flow



The network simulator tool can simulate the following network functions:

- Network delays (maximum network delays of 150 ms)

- Limiting maximum bandwidth
- Finite queue size
- Dropping packets
- Reordering packets

Enabling the IP Network Simulator

Because the network simulator commands and functionality are part of the SAN Extension Tuner, you must first enable the tuner; after doing so, you can view and use the network simulator commands in EXEC mode.

To enable the network simulator (in this case, on a Cisco MDS 9506 director), follow these steps:

	Command	Purpose
Step 1	`switch# config t` `switch(config)#`	Enters configuration mode.
Step 2	`switch(config)# san-ext-tuner enable`	Enables the SAN Extension Tuner.
Step 3	`switch(config)# exit` `switch#`	Exits to EXEC mode.
Step 4	`switch# ips netsim enable interface gigabitethernet 2/3 gigabitethernet 2/4`	Configures the pair of Gigabit Ethernet ports in network simulation mode. **Note** The two ports must be adjacent to each; the first port must be an odd-numbered port.
	`switch# ips no netsim enable interface gigabitethernet 2/3 gigabit ethernet 2/4`	Disables network simulation mode and resets the Gigabit Ethernet ports.

Simulating Network Delays

You can configure the network simulator to delay all packets entering the Gigabit Ethernet ports. After configuring the delay in one direction, you need to also enter the same command to introduce the delay in the opposite direction, if desired. You can specify the delay in either milliseconds (allowable range is 0 to150 ms) or microseconds (allowable range is 0 to 150000 µs).

To configure the network simulator to delay all packets entering the Gigabit Ethernet ports 2/3 and 2/4 by 100 ms (round-trip), follow these steps:

	Command	Purpose
Step 1	`switch# ips netsim delay-ms 50 ingress gigabitethernet 2/3`	Configures the network simulator to delay all packets entering the Gigabit Ethernet port 2/3 by 50 ms.
	`switch# ips netsim delay-us 50 ingress gigabitethernet 2/3`	Configures the network simulator to delay all packets entering the Gigabit Ethernet port 2/3 by 50 µs.
Step 2	`switch# ips netsim delay-ms 50 ingress gigabitethernet 2/4`	Configures the network simulator to delay all packets entering the Gigabit Ethernet port 2/4 by 50 ms.
	`switch# ips netsim delay-us 50 ingress gigabitethernet 2/4`	Configures the network simulator to delay all packets entering the Gigabit Ethernet port 2/4 by 50 µs
	`switch# ips netsim delay-ms 0 ingress gigabitethernet 2/3 gigabitethernet 2/4` `switch# ips netsim delay-us 0 ingress gigabitethernet 2/3 gigabitethernet 2/4`	Disables network packet delay simulation.

Simulating Maximum Bandwidth

You can configure the network simulator to restrict the maximum bandwidth in a single direction. Simulating a maximum bandwidth less than that provided by Gigabit Ethernet allows you to control the pacing of packets through the network. So simulating maximum bandwidth in this way actually gives you an idea of the actual bandwidth across a WAN link (for example, an OC3).

You can specify the allowable bandwidth range in either kilobits per second (1000 to 1000000) or megabits per second (1 to 1000).

To configure the network simulator to limit the bandwidth in a specified direction, follow these steps.

	Command	Purpose
Step 1	`switch# ips netsim max-bandwidth-kbps 4500 ingress gigabitethernet 2/3`	Configures the network simulator to limit the bandwidth rate to 4500 kbps for the Gigabit Ethernet port 2/3 in one direction only.
	`switch# ips netsim max-bandwidth-mbps 45 ingress gigabitethernet 2/3`	Configures the network simulator to limit the bandwidth rate to 45 mbps for the Gigabit Ethernet port 2/3 in one direction only.

	Command	Purpose
Step 2	`switch# ips netsim max-bandwidth-kbps 4500 ingress gigabitethernet 2/4`	Configures the network simulator to limit the bandwidth rate to 4500 kbps for the Gigabit Ethernet port 2/4 in one direction only.
	`switch# ips netsim max-bandwidth-mbps 45 ingress gigabitethernet 2/4`	Configures the network simulator to limit the bandwidth rate to 45 mbps for the Gigabit Ethernet port 2/4 in one direction only.

	`switch# ips netsim max-bandwidth-kbps 0 ingress gigabitethernet 2/3`
	
	`switch# ips netsim max-bandwidth-kbps 0 ingress gigabitethernet 2/4`
		Disables network bandwidth rate simulation.

Simulating a Finite Queue Size

You can configure network simulator to simulate a finite queue size in a network device. Data packets are dropped after the queue is full. To simulate a realistic network device, you should specify a queue size of 50 to150 KB. The maximum acceptable queue size is 1000 KB.

To configure the network simulator to simulate a finite queue size, follow these steps.

	Command	Purpose
Step 1	`switch# ips netsim qsize 75 ingress gigabitethernet 2/3`	Configures the network simulator to simulate a finite queue size of 75 KB for the Gigabit Ethernet port 2/3 in one direction only.
Step 2	`switch# ips netsim qsize 75 ingress gigabitethernet 2/4`	Configures the network simulator to simulate a finite queue size of 75 KB for the Gigabit Ethernet port 2/4 in one direction only.
	`switch# ips netsim qsize 1000 ingress gigabitethernet 2/3 gigabitethernet 2/4`	Disables finite queue size simulation.

Simulating Packet Drops

You can configure network simulator to simulate packet drops (even when the queue is not full) randomly (specified as a percentage) or every Nth packet.

Percentage is represented as the number of packets in 10000. For example, if you wish to drop one percent of packets, then you would specify it as 100 packets in 10000. To simulate a realistic scenario for IP networks using random drops, the drop percentage should be between zero and one percent of packet drops in the specified traffic direction.

If you use the optional burst parameter, then the specified number of packets will be dropped each time a decision is made to drop a packet. If you do not specify the burst parameter, then only one packet is dropped each time a decision is made to drop packets. The burst limit for either random or Nth drops is between 1 and 100 packets. Take the burst parameter into account when specifying the percentage of packet drops. For example, if you select random drops of 100 packets in 10,000 (one percent) with a burst size of 2, then 200 packets (or two percent) are dropped every 10,000 packets.

To configure the network simulator to simulate packet drops, follow these steps:

	Command	Purpose
Step 1	`switch# ips netsim drop random 100 burst 1 ingress gigabitethernet 2/3`	Configures the network simulator to simulate random packet drops of 1% for the Gigabit Ethernet port 2/3 in one direction only. The burst is one packet.
	`switch# ips netsim drop nth 100 burst 2 ingress gigabitethernet 2/3`	Configures the network simulator to drop 2 packets after every 100 packets for the Gigabit Ethernet port 2/3 in one direction only (meaning that when the drop is made, two consecutive packages are dropped).
Step 2	`switch# ips netsim drop random 100 burst 1 ingress gigabitethernet 2/4`	Configures the network simulator to simulate a random packet drop of 1% for the Gigabit Ethernet port 2/4 in one direction only. The burst is one packet.
	`switch# ips netsim drop nth 100 burst 2 ingress gigabitethernet 2/4`	Configures the network simulator to drop 2 packets after every 100 packets for the Gigabit Ethernet port 2/4 in one direction only. The burst is two packets, meaning that when the drop is made, two consecutive packages are dropped.
	`switch# ips netsim drop random 0 burst 1 ingress gigabitethernet 2/3` `switch# ips netsim drop nth 0 burst 1 ingress gigabitethernet 2/4`	Disables packet drop simulation.

Simulating Packet Reordering

You can configure network simulator to simulate that a percentage of packets be reordered, either randomly or every Nth packet. Percentage is represented as the number of packets to be reordered in 10000 packets. The acceptable range is between 0 and 10000. So, a specified value of 100 is equal to 1 percent; a value of 1000 is equal to 10 percent.

If you specify the optional distance parameter, then the packet at the head of the queue is reordered with the packet at the distance specified. For example, if you specify a distance of 2 for every 100 packets, then packets 100 and 102 are reordered. The packet sequence would be 1...99, 101, 102, 103...199, 201, 202, 200, 203 and so on. Hence, distance determines how far back in the queue a reordered packet is placed.

To configure the network simulator to simulate packet reordering, follow these steps:

	Command	Purpose
Step 1	`switch# ips netsim reorder random 50 distance 2 ingress gigabitethernet 2/3`	Configures the network simulator to randomly simulate packet reordering at 50% for the Gigabit Ethernet port 2/3 in one direction only. The distance limit is 5.
	`switch# ips netsim reorder nth 50 distance 2 ingress gigabitethernet 2/3`	Configures the network simulator to simulate packet reordering every 50th packet the Gigabit Ethernet port 2/3 in one direction only. The distance limit is 2. So every 50th packet is reordered as the 52nd packet.
Step 2	`switch# ips netsim reorder random 50 distance 2 ingress gigabitethernet 2/4`	Configures the network simulator tool to randomly simulate packet reordering at 50% for the Gigabit Ethernet port 2/4 in one direction only. The distance limit is 2.
	`switch# ips netsim reorder nth 50 distance 2 ingress gigabitethernet 2/4`	Configures the network simulator to simulate packet reordering every 50th packet for the Gigabit Ethernet port 2/4 in one direction only. The distance limit is 2.

	`switch# ips netsim reorder random 0 ingress gigabitethernet 2/3 gigabitethernet 2/4`
	
	`switch# ips netsim reorder nth 0 ingress gigabitethernet 2/3 gigabitethernet 2/4`
		Disables packet reorder simulation.

Displaying IP Network Simulator Statistics

You can view a summary of the IP ports that are currently operating in network simulation mode using the **show ips netsim** command.

```
switch# show ips netsim
Following ports operate in network simulator mode
GigabitEthernet2/3 and GigabitEthernet2/4
```

You can view a summary of the configured parameters and statistics of network simulation using the **show ips stats netsim ingress gigabit ethernet x/y** command. The configuration parameters displayed by default are:

- Delay
- Bandwidth
- Qsize
- Qdelay

The optional configuration parameters are displayed only if they are currently configured on the specified port.

The following network statistics are also displayed:

- Number of packets dropped
- Queue size

- Number of packets reordered
- Average speed

```
switch# show ips stats netsim ingress gigabitethernet 2/3
Network Simulator Configuration for Ingress on GigabitEthernet2/3
    Delay            : 50000 microseconds
    Rate             : 1000000 kbps
    Max_q            : 100000 bytes
    Max_qdelay       : 600000 clocks
    Random Drop %    : 1.00%

Network Simulator Statistics for Ingress on GigabitEthernet2/3
    Dropped (tot)      = 28
    Dropped (netsim)   = 14
    Reordered (netsim) = 0
    Max Qlen(pkt)      = 7
    Qlen    (pkt)      = 0
    Max Qlen (byte)    = 326
    Qlen     (byte)    = 0
    Mintxdel(poll)     = 852
    Mintxdel(ethtx)    = 360
    empty              = 757
    txdel              = 8
    late               = 617
    Average speed      = 0 Kbps
switch# show ips stats netsim ingress gigabitethernet 2/4
Network Simulator Configuration for Ingress on GigabitEthernet2/4
    Delay            : 50000 microseconds
    Rate             : 1000000 kbps
    Max_q            : 100000 bytes
    Max_qdelay       : 600000 clocks
    Reorder nth pkt  : 50
      distance       : 2

Network Simulator Statistics for Ingress on GigabitEthernet2/4
    Dropped (tot)      = 0
    Dropped (netsim)   = 0
    Reordered (netsim) = 2
    Max Qlen(pkt)      = 8
    Qlen    (pkt)      = 0
    Max Qlen (byte)    = 0
    Qlen     (byte)    = 0
    Mintxdel(poll)     = 3788
    Mintxdel(ethtx)    = 360
    empty              = 595
    txdel              = 0
    late               = 335
    Average speed      = 0 Kbps
```

IP Network Simulator Configuration Example

The following example shows how to set up and use the network simulator to introduce a network delay simulation. For continuity, the procedures for creating the Gigabit Ethernet interfaces and enabling the FCIP tunnels are included.

Step 1 Before enabling the network simulator, you must configure two Gigabit Ethernet interfaces to create an FCIP tunnel link (Gigabit Ethernet interfaces 2/3 and 2/4), and then enable the tunnel.

```
switch# config t
switch(config)# interface gigabitethernet 2/3 no shut
```

```
switch(config)# interface gigabitethernet 2/4 no shut
```

Step 2 Enable the SAN Extension Tuner; this is required for the network simulator tool to work.

```
switch(config)# san-ext-tuner enable
switch(config)# exit
```

Step 3 Enable the network simulator on Gigabit Ethernet ports 2/3 and 2/4. Then check that the Gigabit Ethernet ports are operating in network simulation mode.

```
switch# ips netsim enable interface gigabitethernet 2/3 gigabitethernet 2/4
switch# show ips netsim
Following ports operate in network simulator mode
GigabitEthernet2/3 and GigabitEthernet2/4
```

Step 4 Configure a delay of 100 ms round trip (sum of both trips) for all the packets that are arriving on the specified Gigabit Ethernet port.

```
switch# ips netsim delay-ms 50 ingress gigabitethernet 2/3
switch# ips netsim delay-ms 50 ingress gigabitethernet 2/4
```

Step 5 Confirm that the delay you introduced is configured.

```
switch# show ips stats netsim ingress gigabitethernet 2/3
Network Simulator Configuration for Ingress on GigabitEthernet2/3
    Delay            : 50000 microseconds
    Rate             : 1000000 kbps
    Max_q            : 100000 bytes
    Max_qdelay       : 600000 clocks

Network Simulator Statistics for Ingress on GigabitEthernet2/3
    Dropped (tot)    = 0
    Dropped  (ne)    = 0
    Reordered (ne)   = 0
    Max Qlen(pkt)    = 5
    Qlen     (pkt)   = 0
    Max Qlen (byte)  = 0
    Qlen     (byte)  = 0
    Mintxdel(poll)   = 128322
    Mintxdel(ethtx)  = 360
    empty            = 9
    txdel            = 0
    late             = 7
    Average speed    = 0 Kbps
```

Default Settings

Table 59-1 lists the default settings for the features included in this chapter.

Table 59-1 Default Settings for Fabric Troubleshooting Features

Parameters	Default
Timeout period to invoke fctrace	5 seconds
Number of frame sent by the fcping feature	5 frames
Remote capture connection protocol	TCP
Remote capture connection mode	Passive

Table 59-1 Default Settings for Fabric Troubleshooting Features (continued)

Parameters	Default
Local capture frame limits	10 frames
FC ID allocation mode	Auto mode.
Loop monitoring	Disabled.