CONFIGURATION GUIDE

Brocade Flow Vision Configuration Guide, 8.1.x

Supporting Fabric OS 8.1.0
Supporting Fabric OS 8.1.1

53-1004395-04
7 June 2017

Contents

Preface

Document conventions

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

Notes, cautions, and warnings

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

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

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

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

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

Text formatting conventions

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

Format	Description
bold text	Identifies command names.
	Identifies keywords and operands.
	Identifies the names of GUI elements.
	Identifies text to enter in the GUI.
italic text	Identifies emphasis.
	Identifies variables.
	Identifies document titles.
`Courier font`	Identifies CLI output.

Format	Description
	Identifies command syntax examples.

Command syntax conventions

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

Convention	Description
bold text	Identifies command names, keywords, and command options.
italic text	Identifies a variable.
value	In Fibre Channel products, a fixed value provided as input to a command option is printed in plain text, for example, **--show** WWN.
[ ]	Syntax components displayed within square brackets are optional. Default responses to system prompts are enclosed in square brackets.
{ **x** | **y** | **z** }	A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options. In Fibre Channel products, square brackets may be used instead for this purpose.
x | **y**	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, for example, passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, *member*[*member*...].
\	Indicates a “soft” line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Brocade resources

Visit the Brocade website to locate related documentation for your product and additional Brocade resources.

White papers, data sheets, and the most recent versions of Brocade software and hardware manuals are available at www.brocade.com. Product documentation for all supported releases is available to registered users at MyBrocade.

Click the **Support** tab and select **Document Library** to access product documentation on MyBrocade or www.brocade.com. You can locate documentation by product or by operating system.

Release notes are bundled with software downloads on MyBrocade. Links to software downloads are available on the MyBrocade landing page and in the Document Library.

Document feedback

Quality is our first concern at Brocade, and we have made every effort to ensure the accuracy and completeness of this document. However, if you find an error or an omission, or you think that a topic needs further development, we want to hear from you. You can provide feedback in two ways:

- Through the online feedback form in the HTML documents posted on www.brocade.com
- By sending your feedback to documentation@brocade.com

Provide the publication title, part number, and as much detail as possible, including the topic heading and page number if applicable, as well as your suggestions for improvement.

Contacting Brocade Technical Support

As a Brocade customer, you can contact Brocade Technical Support 24x7 online or by telephone. Brocade OEM customers should contact their OEM/solution provider.

Brocade customers

For product support information and the latest information on contacting the Technical Assistance Center, go to www.brocade.com and select **Support**.

If you have purchased Brocade product support directly from Brocade, use one of the following methods to contact the Brocade Technical Assistance Center 24x7.

Online	Telephone
Preferred method of contact for non-urgent issues: • Case management through the MyBrocade portal. • Quick Access links to Knowledge Base, Community, Document Library, Software Downloads and Licensing tools	Required for Sev 1-Critical and Sev 2-High issues: • Continental US: 1-800-752-8061 • Europe, Middle East, Africa, and Asia Pacific: +800-AT FIBREE (+800 28 34 27 33) • Toll-free numbers are available in many countries. • For areas unable to access a toll-free number: +1-408-333-6061

Brocade OEM customers

If you have purchased Brocade product support from a Brocade OEM/solution provider, contact your OEM/solution provider for all of your product support needs.

- OEM/solution providers are trained and certified by Brocade to support Brocade® products.
- Brocade provides backline support for issues that cannot be resolved by the OEM/solution provider.
- Brocade Supplemental Support augments your existing OEM support contract, providing direct access to Brocade expertise. For more information, contact Brocade or your OEM.
- For questions regarding service levels and response times, contact your OEM/solution provider.

About This Document

What's new in this document

This document includes new and modified information for the Fabric OS 8.1.0 release of Flow Vision.

Changes made for version 53-1004395-01

The following content is new or has been significantly revised for this release of this document:

- Supported software and hardware on page 12 was updated to include new hardware.
- New terms were added to the Glossary on page 13.
- Overview of Flow Vision on page 15 was updated to include a description of VM Insight.
- Firmware upgrade and downgrade considerations for Flow Vision on page 18 were updated for this release.
- Flow definition parameters and rules on page 23 were expanded to include a definition rules for VM Flow.
- Supported port configurations for each application on page 24 were changed to indicate that Flow Mirror is supported in Access Gateway mode.
- Numbers of flows supported on page 25 were updated.
- Limitations when monitoring SIM traffic on page 56 were updated.
- SCSI I/O latency configurations on page 81 were updated and modified for accuracy and completeness.
- Flow latency metrics restrictions and limitations on page 84 were updated.
- Flow monitoring with VM Insight capability on page 88 was added.
- VM Insight requirements on page 89 was added.
- VM Insight flow monitoring on page 89 was added.
- Creating a VM Insight flow monitoring on page 91 was added.
- Flow definition metrics for VM Insight on page 91 was added.
- VM Insight limitations and considerations on page 93 were added.
- Flow Generator limitations and considerations on page 117 were updated.
- Overview of Flow Mirror on page 119 was modified for accuracy and completeness.
- Flow Mirror limitations and restrictions on page 121 were updated.
- Using Flow Mirror with Access Gateway on page 134 was added.

Changes made for version 53-1004395-02

The following content is new or has been significantly revised for the second release of this document:

- Firmware upgrade considerations for Flow Vision on page 18 were upgraded to include additional considerations about upgrading when in-flight encryption and compression are enabled.

- The illustrations in SCSI I/O latency configurations on page 81 were updated for accuracy.
- SIM port criteria on page 114 was updated to indicate that SIM ports can be configured on ports with QoS and/or CSCTL enabled.
- Flow Generator limitations and considerations on page 117 were updated to indicate that all switches from edge to edge must be running when using Flow Generator across FCR.
- Flow Mirror limitations and restrictions on page 121 were updated to include additional information about enabling in-flight encryption and compression on the same Gen 5 ASIC that has RFM enabled.

Changes made for version 53-1004395-03

The following content is new or has been significantly revised for the third release of this document:

- Flow Generator limitations and restrictions on page 117 was modified to remove some restrictions.

Changes made for version 53-1004395-04

The following content is new or has been significantly revised for the fourth release of this document:

- Support added for Fabric OS 8.1.1.
- Updated Firmware downgrade considerations for Flow Vision on page 19 with restrictions for Brocade 6520 and DCX 8510 running Fabric OS 8.1.1.
- Updated Configuration file upload and download considerations on page 20 with restrictions for Brocade 6520 and DCX 8510 running Fabric OS 8.1.1.
- Flow Mirror limitations and restrictions on page 121 was updated with the new "vTap and Encryption/Compression Coexistence Mode" feature and restrictions.

Supported hardware and software

The following hardware platforms are supported by Fabric OS 8.1.0.

Although many different software and hardware configurations are tested and supported by Brocade for Fabric OS 8.1.0, documenting all possible configurations and scenarios is beyond the scope of this document.

Fabric OS support for the Brocade Analytics Monitoring Platform (AMP) device depends on the specific version of the software running on that platform. For more information, refer to the Brocade Analytics Monitoring Platform documentation and release notes.

Brocade Gen 5 (16-Gbps) fixed-port switches

- Brocade 6505 Switch
- Brocade 6510 Switch
- Brocade 6520 Switch
- Brocade M6505 blade server SAN I/O module
- Brocade 6542 blade server SAN I/O module
- Brocade 6543 blade server SAN I/O module
- Brocade 6545 blade server SAN I/O module
- Brocade 6546 blade server SAN I/O module

- Brocade 6547 blade server SAN I/O module
- Brocade 6548 blade server SAN I/O module
- Brocade 6558 blade server SAN I/O module
- Brocade 7840 Extension Switch

Brocade Gen 5 (16-Gbps) Directors

For ease of reference, Brocade chassis-based storage systems are standardizing on the term “Director.” The legacy term “Backbone” can be used interchangeably with the term “Director.”

- Brocade DCX 8510-4 Director
- Brocade DCX 8510-8 Director

Brocade Gen 6 (32-Gbps) fixed-port switches

- Brocade G610 Switch
- Brocade G620 Switch

Brocade Gen 6 (32-Gbps) Directors

- Brocade X6-4 Director
- Brocade X6-8 Director

Glossary

The following terminology, acronyms, and abbreviations are used in this document.

TABLE 1 Flow Vision-related terminology

Term	Description
Application ID	Identifier associated with an Entity ID that is used to "tag" frames and flows of an entity (VM).
Application Server	Component that manages VM registrations with a fabric.
Backbone E_Port	The E_Port on a Fibre Channel Routing (FCR)-enabled switch.
Defined flow	User-created flow; it can be active or inactive.
DISL	Dedicated ISL (DISL); a physically-connected link between two logical switches that belong to the same Fabric ID (FID). A DISL is dedicated to carry frames only related to the FIDs of connected logical switches.
Entity	A virtual machine.
Entity ID	An identifier associated with an entity (VM). For example, in VMWare, the VM Instance UUID.
Flow	A set of Fibre Channel frames that share similar traits.
ISL	An inter-switch link (ISL) is a link between two Fibre Channel switches, connecting the E_Port of one switch to the E_Port of the other switch.
Learned flow	Flow created by using an asterisk (*) as part of the flow definition.
LISL	Logical ISL; a logical link between two logical switches that is used for control frames. Depending on the fabric topology, a LISL may or may not map directly to a single physical ISL.
Local flow	Flow defined on the switch on which the **flow** command is being run.
Local switch	Switch on which the **flow** command is being run.

TABLE 1 Flow Vision-related terminology (continued)

Term	Description
Remote flow	Flow defined on a different switch from the one on which you are viewing it.
Remote switch	Switch other than the switch on which the **flow** command is being run.
Root flow	Instance of a static flow used to create learned flows.
SCSI I/O latency	The time it takes for an I/O operation to complete between an initiator and a target.
Static flow	Flow created when learning is not used.
VF	Virtual Fabric: an architecture that virtualizes hardware boundaries.
VM	Virtual machine. Also, referred to as an Entity.
XISL	eXtended ISL: a logical link connecting base switches together to form the base fabric. It carries frames from the base fabric and other logical fabrics using the encapsulation and inter-fabric link (IFL) header as identifiers.

Flow Vision

Overview of Flow Vision

Flow Vision is a Fibre Channel SAN network diagnostic tool supported on all platforms supported by Fabric OS 7.2.0 and later versions.

Flow Vision provides a comprehensive vision of and deep insight into fabric traffic flows, along with the ability to non-disruptively create and capture copies of traffic flows for analysis of traffic flows, bottlenecks, bandwidth utilization, and similar fabric connectivity functionality. Flow Vision also provides a test flow generation capability that you can use to pre-test a SAN infrastructure for robustness. This test flow generation capability is also useful for testing the internal connections on a switch before deploying the switch into a production environment. In addition, Flow Vision allows you to test for fabric connectivity issues, such as slow drain, bandwidth utilization, and similar issues.

Brocade's **IO Insight** capability on Gen 6 platforms adds built-in device input/output (I/O) latency and performance instrumentation to Flow Vision for Fabric OS 8.0.1 and later. With IO Insight capabilities, Flow Vision provides proactive, non-intrusive, real-time monitoring and alerting of storage I/O health and performance. The additional visibility delivers deep insights into problems and concerns affecting the ability to maintain service levels.

Brocade's **VM Insight** capability for Fabric OS 8.1.0 and later on Gen 6 platforms allows you to gather Fibre Channel I/O and SCSI I/O performance and I/O latency statistics about individual traffic flows for virtual machine (VM) instances originating from a single N_Port ID (PID) to destination targets. Data about VM flows can help isolate VMs being impacted by performance issues, isolate the source of performance issues, and help plan for VM migration for load balancing or other purposes.

Flow Vision features

Flow Vision has three features: Flow Monitor, Flow Generator, and Flow Mirror.

Flow Monitor

Flow Monitor provides flow monitoring and the gathering of frame statistics and IO statistics for fabric application flows, including the ability to learn (discover) flows automatically that are flowing through a specified port. Refer to Flow Monitor on page 47 for a complete description and sample use cases.

Flow Generator

Flow Generator simulates and generates test-load traffic in specific flows, which you to validate hardware components, connectivity, and verify performance. Refer to Flow Generator on page 101 for a complete description and sample use cases.

Flow Mirror

Flow Mirror provides the ability to non-disruptively create copies of application flow frames that can be captured for deeper analysis of their contents. Refer to Flow Mirror on page 119 for a complete description and sample use cases.

Flow Vision and Advanced Performance Monitor

In Fabric OS 7.4.0 and later, Advanced Performance Monitor (APM) functionality has been replaced by Flow Vision, which has been enhanced to support monitoring features equivalent to APM monitors, including port mirroring.

Flow Vision equivalents for APM monitors

In Fabric OS prior to version 7.4.0, APM monitored statistics using multiple licensed APM performance monitors. Although Flow Vision does not use the APM configurations, it does provide the same functionality. You will have to create new monitors manually using the Flow Vision commands in the following table.

TABLE 2 Flow Vision equivalents for APM monitors

Monitor	APM command (unsupported)	Flow Vision command
Fabric mode top talker	**perfttmon --add fabricmode**	**flow --create** *flow_name* **-feature mon -srcdev** "*" **-dstdev** "*" **-ing** *e_port_number* Although you can configure learning flows on E_Port, not all E_Port learning is supported. E_Port learning flows can be activated and enforced only when the E_Port is the trunk master.

Port mode top talker	**perfttmon --add ingress** *port_ID*
	perfttmon --add egress *port_ID*
		flow --create *flow_name* **-feature mon -srcdev** "*" **-dstdev** "*" followed by either **-ing** *port_number* or **-egr** *port_number*
		
		NOTE
		You can also discover the top talking flows using the all F_Port learning flow: sys_mon_all_fports. To do this, you will have to first deactivate any user-created flows and then activate the flow. Refer to Learning for all F_Ports using the sys_mon_all_fports flow on page 55 for additional information on this flow.
		• The following command displays the top 5 talkers based on the throughput in decreasing order starting from the highest talker.
		`flow --show sys_mon_all_fports -feature mon -decrease 5`
		• The following command displays the Bottom or Least 5 talkers based on the throughput in increasing order starting from the lowest talker.
		`flow --show sys_mon_all_fports -feature mon -increase 5`

End-to-end monitor	**perfaddeemonitor** *port_number SourceID DestinationID*	**flow --create** *flow_name* **-feature mon -srcdev** *Source_ID* **-dstdev** *Destination_ID* **-ing** *port_number* **-bidir**
Frame monitor (predefined)	**fmmonitor** *port*--**addmonitor** *frame_type* **-port** *port_ID*	**flow --create** *flow_name* **-feature mon -frametype** *frame_type* followed by either **-ing** *port_number* or **-egr** *port_number*
Filter monitor (user defined)	**fmmonitor --create** *frame_type* **-pat** *bit_pattern* **--port** *port_ID*	Not supported

Unsupported Advanced Performance Monitor commands

Because Advanced Performance Monitor (APM) has been removed from Fabric OS 7.4 and later versions, the following commands have either been removed from Fabric OS entirely or have been modified. Refer to the *Brocade Fabric OS Command Reference* for information on the specific commands that have been removed or changed.

• fmmonitor	• perfAddReadMonitor
• perfcfgclear	• perfAddWriteMonitor
• perfcfgrestore	• perfAddRWMonitor
• perfcfgsave	• perfAddSCSIMonitor
• perfClearAlpaCrc	• perfAddIPMonitor
• perfShowAlpaCrc	• perfClearFilterMonitor
• perfAddEEMonitor	• perfDelFilterMonitor
• perfClearEEMonitor	• perfShowFilterMonitor
• perfDelEEMonitor	• perfMonitorClear
• perfShowEEMonitor	• perfMonitorShow
• perfSetPortEEMask	• perfttmon
• perfShowPortEEMask	• perfResourceShow
• perfAddUserMonitor	• perfHelp

Flow Vision limitations and considerations

Beyond the individual feature-specific restrictions, the following restrictions and limitations apply to Flow Vision as a whole:

- Port swap functionality is not supported. Therefore, you cannot create flows on ports that have been swapped, and you cannot swap ports on which flows are currently defined.
- For Flow Monitor learned flows, if the domain or PID is changed in a remote switch, the following should be considered:
  - The learned flows for the previously defined PID pair remain idle in the flows, they will not be actively monitored, and statistics will not be generated.
  - However, the learned flows for the newly defined PID pair will get updated in the flow, they will be monitored, and statistics will be generated.
  - To remove out-dated entries, deactivate and then activate the flows.
- For Flow Generator learned flows, if the domain or PID is changed in a remote switch, the following should be considered:
  - Flow Generator will indicate that the device is offline.
  - The flow will not be enforced.
- For EX_Ports, only flows with both **-srcdev** and **-dstdev** are supported on an EX_Port. None of the advanced options are supported on an EX_Port, such as **-bidir**, **-frametype**, and **-lun**.

Flow Vision licensing

To run Flow Vision, you need either the Fabric Vision (FV) license, or *both* the Fabric Watch (FW) and the Advanced Performance Monitor (APM) licenses. If you have both of these licenses, you do not need a separate Flow Vision license.

Refer to the *Brocade Fabric OS Software Licensing Guide* for more specific information about licensing and how to obtain the needed license keys.

Firmware upgrade and downgrade considerations for Flow Vision

Flow Vision has specific considerations that apply to upgrading or downgrading the firmware on a switch with Flow Vision installed. Both user-defined and system-defined flows support the configuration upload and download procedures.

Flow Vision in Fabric OS 8.1.0 supports only those devices listed in Supported hardware and software on page 12.

Flow Vision functionality is not affected if a switch running Fabric OS 8.1.0 is connected to a switch running Fabric OS 7.2.0 or later. Connecting to a switch running any version of Fabric OS earlier than 7.2.0 will disable Flow Vision.

Firmware upgrade considerations for Flow Vision

The following items should be taken into consideration when upgrading to Fabric OS 7.4.0 or later:

- Advanced Performance Monitoring (APM) is not supported in Fabric OS 7.4.0 or later; its functions have been incorporated into Flow Vision.
- If APM monitors are present on the switch, firmware upgrade from versions of Fabric OS earlier than 7.4.0 will not be blocked, but a warning message similar to the following comes up:

  ```
  WARNING: Advanced Performance Monitoring (APM) is obsoleted in FOS v7.4. After firmware
  upgrade, all APM monitors including End-to-End monitors, Frame monitors, and Top Talker
  monitors that have been installed on all the logical switches will be disabled. The
  configuration for these monitors will be permanently removed. Flow Vision provides the same
  functionality in FOS v7.4. Please refer to Flow Vision Administrator's Guide for further
  information.
  ```
- When moving a configuration from a switch running Fabric OS 7.2.x to one running 7.4.0 or later, predefined flows will be included automatically when the older configuration is downloaded to a switch running Fabric OS 7.4.0 or later, but they will remain deactivated after download.
- If in-flight encryption and compression functionality is enabled, then any Flow Mirror flow active before the upgrade will be deactivated after an upgrade to Fabric OS 8.1.0. RFM and inflight encryption and compression will co-exist on Gen 5 ASIC with limitations. Refer to Flow Mirror limitations and restrictions on page 121.
- If you are going to upgrade the firmware to Fabric OS 7.4.0 or later, and one or more ports have been enabled as a SIM Port through Flow Generator, the firmware upgrade will be blocked. To perform the upgrade, the SIM port configurations must be disabled. Alternatively, you can use the following procedure:
  1. Upload the np_config file before starting the upgrade operation.
  2. Disable the SIM Ports configuration.
  3. Upgrade the software.
  4. Download the np_config file to the switch.

Removing Advanced Performance Monitor monitors

All Advanced Performance Monitor (APM) monitors must be removed from a switch before upgrading to Fabric OS 8.1.0.

To remove all APM monitors, complete the following steps for each logical switch on the chassis.

1. Before upgrading the switch, connect to the switch and log in using an account with admin permissions.
2. Enter the **perfdeleemonitor** command to remove all end-to-end monitors from a port:
   perfdeleemonitor *slot /port monitorID*

3. Enter **perfcfgsave** to save this change.
4. Enter **fmmonitor --delmonitor** to remove all filter monitors.
5. Enter **fmmonitor --delete** *frametype* to remove the specified user-defined frametype. You will need to repeat this step for each user-defined frametype to remove all user-defined frametypes. If there are any user-defined frametypes, the upgrade will be blocked.
6. Enter **perfttmon --delete** to remove all switch level Top Talker monitors.
7. Enter **perfttmon --delete fabricmode** to remove fabric mode Top Talker monitors.

The following example removes all the APM monitors from the switch named switch123. There was only one user-defined frametype (D_Port22).

```
switch123:admin> perfdeleemonitor 1/23 monitor3887
switch123:admin> perfcfgsave
switch123:admin> fmmonitor --delmonitor
switch123:admin> fmmonitor --delete D_Port22
switch123:admin> perfttmon --delete
switch123:admin> perfttmon --delete fabricmode
```

Firmware downgrade considerations for Flow Vision

Flow Vision functionality will not be affected if a switch running Fabric OS 7.4.0 or later is connected to a switch running Fabric OS 8.1.1. However, the following items should be taken into consideration when downgrading.

- Remote Flow Mirroring is not supported on any version of Fabric OS earlier than 7.4.1a, except for Fabric OS 7.3.2 running on Gen 5 devices.
- Downgrading will fail if any Flow Vision-related configurations are present on the switch being downgraded.
- During downgrading from 8.1.0 to Fabric OS versions 7.4.x or 7.3.2, perform the following before proceeding with the downgrade:
  - Deactivate all active predefined sys_analytics_vtap flows.
  - Turn off the vTAP and QoS High Priority Zone Compatibility mode.
- Flows defined using **fabinfo** or **-allzoned** are not supported on Fabric OS versions earlier than 7.4.0. Functionality for **fabinfo** is limited for versions of Fabric OS earlier than 8.0.1.
- All Flow Vision-related flows or simulation ports must be deleted prior to performing a downgrade to any version of Fabric OS prior to 7.2.0; if they are not, the downgrade will be blocked and a warning message displayed.
- Flow counts that exceed the supported scalability limits will not be replayed when downgraded or failed over to a version of Fabric OS earlier than 8.0.1.
- Predefined flows will not be replayed, and flow definitions for newly introduced features will not be replayed from the flow configuration.
- Downgrading from Fabric OS 8.1.0 to a version earlier than Fabric OS 7.3.0 is not allowed if a Flow Mirror flow is active for local flow mirroring (LFM). You must first deactivate any active flow that is using a local mirror port.
- Flows that were created with the following keyword combinations are automatically deactivated.
  - **-bidir** and **-frametype**
  - **-ingrport** and **-frametype**
- Flows created using learning on E_Ports and EX_Ports will not be replayed after the downgrade.
- Flows created using monitor support on XISL ports will not be replayed after the downgrade.
- Flows will be deleted from memory configuration when the firmware is downgraded.

- If one or more ports have been enabled as a SIM Port, downgrading the firmware has the following restrictions:
  - A firmware downgrade to Fabric OS 8.0.x is allowed for only Brocade G620 devices.
  - A firmware downgrade to Fabric OS 7.4.x is allowed.
  - A firmware downgrade to Fabric OS 7.3.x is allowed.
  - To downgrade to Fabric OS 7.2.x or earlier, you should first downgrade to Fabric OS 7.3.x or 7.4.x, otherwise an error message is displayed. Then, before downgrading to Fabric OS 7.2.x or earlier, use the following command to disable SIM-port configuration on each switch:

    flow --control -simport *[SlotNumber/]PortNumber* -disable
- After downgrading to a version of Fabric OS earlier than 7.4.0, you will have to reconfigure any APM monitors that you want to have running.
- Releases previous to Fabric OS 8.0.1 do not support flows created on Virtual E_Ports (VE_Ports). The switches running previous releases cannot check if the configurations being downloaded contain flows created on VE_Ports, so to avoid errors, you should not download configurations containing flows created on VE_Ports.
- Firmware downgrade is blocked if the vTap and Encryption/Compression Coexistence Mode is enabled on Brocade 6520 and DCX 8510. On these two platforms, you must disable the coexistence mode before downgrading firmware from Fabric OS 8.1.1 to any previous version.

Flow definitions for features introduced in Fabric OS 8.1.0 are not replayed from the flow configuration if the switch is downgraded to an version earlier than Fabric OS 8.0.1.

> NOTE
> There might be other factors affecting firmware version downgrades; the ones listed here are only those that Flow Vision directly affects or is affected by.

Configuration file upload and download considerations

The following points should be considered when uploading or downloading configuration files.

Uploading from previous versions of Fabric OS

No issues will occur with flow configuration when you upload from Fabric OS 7.4.x or later.

Downloading to previous versions of Fabric OS.

- You must avoid downloading a configuration for Fabric OS 8.1.0 onto a switch that is running Fabric OS 7.4.x or a previous version.
  - Fabric OS 7.4.x and previous releases do not support flows created on VE_Ports. Therefore, if configurations are downloaded from Fabric OS 8.1.0 to a switch running Fabric OS 7.4.x or previous releases, the files would contain unsupported configurations.
  - Flows on VE_Ports must be removed before downgrading to previous releases that do not support VE_Ports.Downloading will be blocked if any flows exist on VE_Ports across all partitions of a switch.
- Only flows that are supported will be replayed during downgrade. Unsupported flows will **not** be replayed or handled internally.
- All other flow definitions for newly introduced features will not be replayed from the flow configuration when the configuration is downloaded from Fabric OS 8.1.0 to Fabric OS 8.0.x or Fabric OS 7.4.x.

Roles and access in Flow Vision

Flow Vision can be accessed by users with the following roles: Admin, Switch Admin, or Fabric Admin.

Flow Vision configuration setup

When a switch goes offline or comes online, Flow Vision reads the configuration files and then deletes flows, creates flows, and activates flows.

After a switch goes offline, any flows that were active at the time it went offline will be reactivated when it comes back online and new statistics will be generated as soon as the source and destination devices defined in the flow are online. This includes predefined flows.

Use the following commands to upload, download, and delete configurations:

- To download a Flow Vision configuration to the switch, use the **configDownload** command.
- To save the Flow Vision configuration , use the **configUpload** command.
- To delete all flows and simulation ports (SIM ports) from a switch, use the **configDefault** command.

NOTE
Statistical data for flows is not saved in the configuration database.

High availability considerations

Consider the following high availability characteristics when working with configurations:

- Flow Vision configuration data is persistent across failover.
- Data that has been captured by Flow Vision applications is not restored after failover.
- User-defined or system-defined flows that were created for newly introduced Flow Vision features will not be restored when failed over to a previous version of Fabric OS.

Flow Vision flows

A flow is a set of Fibre Channel (FC) frames or packets that share similar traits, such as an ingress port or egress port identifier or any other data that can be used to differentiate one set of related frames or packets from a different set.

These parameters are specified as part of the **flow** command, and include:

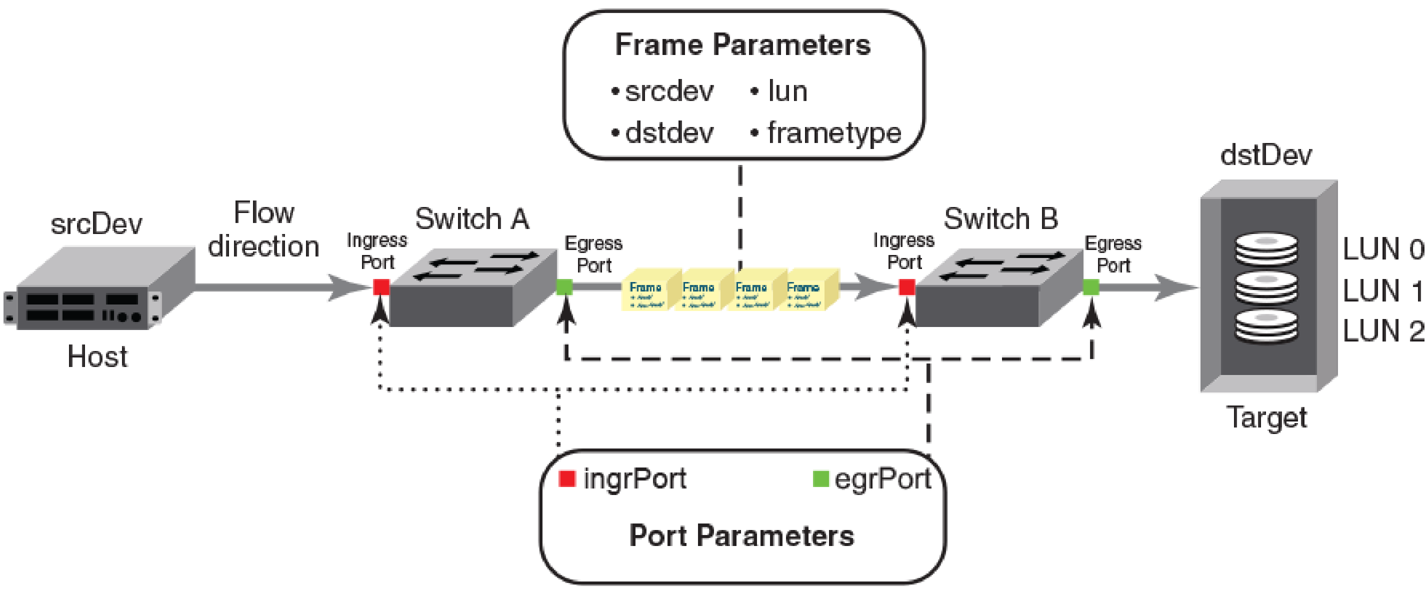
- **Port parameters:** (Also called the “Point of Interest” or the location from where the data that you want to examine originates.) This consists of an ingress port (ingrport) or an egress port (egrport). Only one can be specified when defining a flow.
- **Frame parameters:** These are the following parameters: Source Device Identification (SID or WWN), Destination Device Identification (DID or WWN), LUN, or frame type. At least one frame parameter must be present to define a flow. Refer to Flow frametype parameters on page 24 for details on frame types.

  NOTE
  Feature-specific rules apply for both lun and frametype parameters.

- **VM-flow parameters:** These include the VM entity ID (srceid), destination device ID, and the ingress port ID. Flows are monitored from the HBA N_Port, through the fabric, to the target.
- **Direction:** A direction is implicitly defined from an ingress port to an egress port, or a source device (srcdev) to a destination device (dstdev). For example, srcdev=x, dstdev=y indicates traffic flowing from x to y. The **-bidir** option causes the flow definition to be monitored in both directions. This makes the following true:
  - Entering **srcdev=x dstdev=y** specifies that only traffic flowing from x to y is the desired flow.
  - Entering **srcdev=x dstdev=y -bidir** specifies that traffic traveling from x to y and traffic traveling from y to x are both desired flows.

The following figure illustrates how the frame and port parameters apply to a flow.

FIGURE 1 Frame and port parameters



Flow definitions

To define a flow and configure Flow Vision to monitor that flow, you must provide a unique flow name and specify the flow parameters. These parameters identify the sets of related frames that compose the flow; these can either be explicitly defined or Flow Vision can learn them through observation.

> NOTE
> These flow definitions are stored on the switch on which the flow is created, and are not distributed across the fabric. This means that each switch (logical or physical) knows only its own unique flows and does not know what flows exist on other switches.

When creating or viewing a flow, you can specify any combination of the three features (monitor, mirror, generator) in the **flow** command.

It does not matter what order the operands are in, but once you choose to use an option, then the next argument has to be correct. In the case of **flow --activate sys_gen_all_simports --feature generator**, as **sys_gen_all_simports** is the flowname, it must immediately follow **--activate**.

Flow definition parameters and rules

The rules listed in the following table identify the parameters that can be used to define a flow.

TABLE 3 Flow definition rules

Parameters	Field names	Rules
Port	ingrport egrport	• One field only must be specified. • Values must be explicit. • If the ingrport or the egrport is an E_Port, then the port must be the trunk master for the associated trunk group; if it is not, then the flow will not be enforced.
Frame	srcdev dstdev lun frametype	• At least one field must be specified. • Values for srcdev and dstdev can be explicit or `"*"`. (`"*"` indicates learned flows, and must include the quotation marks.) • Values for lun and frametype must be explicit. • On XISL monitors, the SFID and DFID values are mandatory but srcdev and dstdev are not.
VM flow	srceid	• Must be an hexadecimal or ASCII entity ID registered with the application server. • Can be explicit or `"*"`. (`"*"` indicates learned flows and must include the quotation marks.)

The following restrictions apply to the flow definition parameters and rules:

- Feature-specific rules apply for both lun and frametype parameters.
- On 8 Gbps-capable Fibre Channel platforms, possible frame monitoring flow classifiers include: egrport, ingrport, srcdev, dstdev, and lun.
- On Gen 5 and Gen 6 Fibre Channel platforms and Brocade FC8-32E and FC8-48E blades, the possible frame monitoring flow classifiers include: ingrport, egrport, dstdev, srcdev, and lun.

Refer to Flow frametype parameters on page 24 for more information about frame types.

Duplicate flow definition support

Flow Vision allows duplicate flow definitions to be created as long as the duplicate flows are not active. Duplicate flow definitions are detected during flow activation. If a flow is a duplicate of an active flow, the duplicate will not be activated.

Flows that are defined with the same basic flow definition for the same port are considered duplicates, even if one of the definitions includes additional advanced options. For example, the following flow definitions are considered duplicates, because both use the same basic definition:

- `flow --create myflow1 -feature mon -srcdev 014800 -egrport 2/8`
- `flow --create myflow2 -feature mon -srcdev 014800 -egrport 2/8 -frametype scsiwrite`

Any flow that is considered to be a duplicate will remain deactivated as long as there is an existing matching flow definition active irrespective of the defined application. A warning message is displayed when you try to create (which implicitly activates) or activate a flow if there is an existing matching flow definition active. You must manually deactivate the active flow to activate the new flow definition.

A flow definition must be active to be considered a duplicate. For example, the following user-defined flow definitions are considered to be duplicates, because the `"*"` value for dstdev in the second example would include the 0xa20c81 value specified in the first.

- `flow --create flow1 -feature monitor -ingrport 9/46 -srcdev 0xb2c680 -dstdev 0xa20c81`
- `flow --create flow2 -feature monitor -ingrport 9/46 -srcdev 0xb2c680 -dstdev "*"`

However, these user-defined flow definitions will not be considered to be duplicates:

- `flow --create flow1 -feature monitor -ingrport 9/46 -srcdev 0xb2c680 -dstdev 0xa20c81`

- `flow --create flow2 -feature monitor -ingrport 9/46 -srcdev 0xb2c680 -dstdev 0xa20c81 -noactivate`
- `flow --create flow3 -feature generator -ingrport 9/46 -srcdev 0xb2c680 -dstdev 0xa20c81 -noactivate`

Predefined flows are considered when checking for duplicate flows. When a predefined flow is active for any feature (for example Flow Generator), all user-defined flows for that feature are checked to see if they are duplicate flows; however, user-defined flows for other features (in this case Flow Monitor or Flow Mirror) are not checked to see if they are duplicate flows. Duplicate predefined flow definitions can be active for different applications. Duplicate predefined flow definitions cannot be active simultaneously for the *same* application.

Supported port configurations for each application

The following table lists the supported configurations for each Flow Vision feature that can be made using only the basic flow identification parameters (ingrport and srcdev; egrport and dstdev).

TABLE 4 Port configurations supported in Flow Vision

Feature	Platforms			Switch Configuration Mode	
	32 Gbps-capable Fibre Channel (Gen 6)	16 Gbps-capable Fibre Channel (Gen 5)	8 Gbps-capable Fibre Channel	Access Gateway	Virtual Fabric
Flow Generator	Supported (SIM ports only)	Supported (SIM ports only)	Supported (Destination SIM ports only)	Not Supported	Supported
Flow Mirror	Supported (F_Ports and F_Port trunks)	Supported (F_Ports and F_Port trunks)	Not Supported	Supported	Supported
Flow Monitor	Supported (E_Ports, EX_Ports, F_Ports, SIM ports, VE_Ports, and XISL_Ports)	Supported (E_Ports, EX_Ports, F_Ports, SIM ports, and XISL_Ports)	Supported (E_Ports, EX_Ports, F_Ports, and XISL_Ports)	Supported (F_Ports only)	Supported

Notes on supported configurations

- Neither ranges nor lists are supported for any parameter.
- If you are using at least one advanced parameter (lun, frametype, or bidir), then feature-specific rules apply. Refer to the individual Flow Vision features for specific details.
- Disabling a SIM port that is receiving traffic might produce Class 3 discards for the simulated traffic; however, this will have no effect on other traffic flows.
- Refer to Supported hardware and software on page 12 for a list of Gen 5 and Gen 6 devices supporting F_Ports and F_Port trunks:

Flow frametype parameters

Frame monitoring can be done for a variety of frames using predefined frametype parameters.

The following table lists these parameters and the type of frames counted for each.

TABLE 5 Supported frametype parameters

Frametype parameter	Frames counted
abts	Abort Sequence
baacc	All frames accepted
barjt	All frames rejected

TABLE 5 Supported frametype parameters (continued)

Frametype parameter	Frames counted
scsi	All SCSI frames (including both command and data frames)
scsiread	Only SCSI read command frames
scsiwrite	Only SCSI write command frames
scsirw	Both SCSI read and write command frames
scsi2reserve	Only SCSI 2 reserve command frames
scsi3reserve	Only SCSI 3 reserve command frames
scsi2release	Only SCSI 2 release command frames
scsi3release	Only SCSI 3 release command frames
scsi2reserverelease	Only SCSI 2 reserve-release command frames
scsi3reserverelease	Only SCSI 3 reserve-release command frames
scsitur	Only SCSI test unit ready frames
scsistatus	Only SCSI status frames
scsicmdsts	Only SCSI command status frames **NOTE** This parameter is valid only for Flow Mirror. It implicitly assumes "-bidir" and looks for both SCSI command and status frames.
scsigoodstatus	Only SCSI status frames with status marked as good (all 0s [zeros] in the status byte)
scsicheckstatus	Only SCSI status frames with check status (Check Condition, Busy, Reservation Conflict, Task Full Set)
scsiinquiry	Only SCSI inquiry frames
scsiresvconflict	Only SCSI status frames with reservation conflict set
scsixferrdy	Only SCSI FCP XFER_RDY (transfer ready) frames
srr	Sequence Retransmission Request

Numbers of flows supported

Flow Vision has both overall limits on the number of flows supported on a switch as well as separate limits for Flow Generator, Flow Mirror, and Flow Monitor.

TABLE 6 Total flow count limits

	User-defined static and learning flows (combined)	Learned flows	Predefined flows
Chassis-based platforms	512	2048	4
Fixed-port platforms	128	512	4 (3 flows for Brocade 7840 Extension Switch)

TABLE 7 Per-port flow count limits

	User-defined static and learning flows (combined)	Learned flows
Chassis-based platforms	64	512
Fixed-port platforms	64	128

The limits for the user-defined static flows and learning flows include both active and inactive flows, and the combined limit applies to all features on a per-port basis. This means (as an example) that two different learning flows on a given port can each have 64 learned flows.

Beyond these limits, there are limits for each individual feature, as described in the following table. In addition, refer to the individual features for other feature-specific restrictions.

NOTE
A verification is done for each flow when it is created or activated to ensure that there is no identical flow active. Duplicate flows will not be created or activated when there is a identical flow already active. To create a new flow that duplicates an active flow, you must use the **-noactivate** keyword as part of the **flow --create** command. Refer to the "Creating an inactive flow" section of each feature for instructions on creating an inactive flow for that feature.

TABLE 8 Feature-specific flow count restrictions in Flow Vision

Feature	Limit to number of flows
Flow Monitor	Up to 64 active flows per port, including static flows and learning flows. However, there is a restriction of only one Flow Monitor flow per ASIC if the flow uses `"*"` as the srcdev or dstdev definition. (See note.)
Flow Generator	Up to 39 active flows per port for ingress ports and 64 active flows per port for egress ports. (See note.)
Flow Mirror	Multiple instances of the predefined flow sys_analytics_vtap can be active across a chassis or the Brocade 6520 switch; however, only one Flow Mirror flow can be active per Virtual Fabric (VF). No other fixed-port switches support multiple instances of sys_analytics_vtap. (See note.)

NOTE
For predefined flows, the learned flow limits are reduced by 1, because one sub-flow is used internally for managing the flows.

Flow learning

Flow Vision creates a learned flow when you specify an asterisk (*) instead of the source device, the destination device, or both devices. This allows you to discover which flows are active on a port without having to explicitly identify all the devices.

The following items should be kept in mind when constructing learning flows:

- Learning is enabled on a port if the flow definition has an asterisk as the value for any of the flow parameters. The learning flow is expanded to include learned flows based on the parameters indicated by the asterisk. Cumulative data is presented for parameters for which learning is not requested.
- When you enter an asterisk as part of the command to indicate a learned flow, you must enclose it in double quotes, like this: `"*"`.
- Learning source device (srcdev) or destination device (dstdev) values are only supported on Gen 5 and Gen 6 Fibre Channel ports.
- Each Flow Vision feature uses learning as follows:
  - Flow Monitor can learn all the source device and destination device pairs passing through the ingress or egress port defined in a flow. Learning is **not** supported for Flow Monitor flows defined using the lun, frametype, or bidir parameters. Refer to Learning in Flow Monitor flows on page 51 for additional information.
  - Flow Generator can generate traffic to or from every source or destination device that shares the zone with the ingress or egress port defined in a flow. Refer to Learning in Flow Generator flows on page 107 for additional information.
  - Flow Mirror can capture all the source device and destination device pairs passing through the ingress or egress port defined in a flow. Learning is **not** supported for Flow Mirror flows defined using the lun, frametype, or bidir parameters. Refer to Learning in Flow Mirror flows on page 131 for additional information.
- Hardware resources for a learned flow are limited. Consequently, the number of actual sub-flows for a flow definition might exceed the number of sub-flows permitted by limitations. Flow Monitor refreshes the sub-flows when the number of sub-flows reaches or exceeds the limit. Refer to Learning in Flow Monitor flows on page 51 for more information.

Viewing flows

Flow Vision allows you to view the configuration parameters for each flow on a switch.

- To display all Flow Vision flows, enter **flow --show**.
- To display all flows for a specific feature, enter **flow --show all -feature** *feature_name*.
- To display the definition for a specific flow, enter **flow --show** *flow_name* . In this case, only the flow definition for the specified flow is displayed; this includes feature-specific information. The following example shows a typical result of this command.

```
switch:admin> flow --show flow1
=======================================================================
Name      : flow1    Features: gen(Activated)    mir(Not Active) noConfig: Off
Definition: IngrPort(2),SrcDev(0x050240),DstDev(0x05c200)

Flow Generator (Activated):
-----------------------
|  SrcDev  |  DstDev  |
-----------------------
| 0x050240 | 0x05c200 |
-----------------------
Number of frames generated from IngrPort :   1.02G
Note: More than 1 flow active on this port.
=======================================================================
```

- In flows containing multiple features, if you enter **flow --show** *flow_name* **-feature** *feature_name*, only the information for that feature will display. For root and static flows, this command shows both the Source ID–Destination ID pair and the cumulative frame count for the ingress or egress port specified in the flow definition. The following example shows a typical result of this command.

```
switch:admin> flow --show gen1 -fea gen
=======================================================================
Name      : gen1    Features: gen(Activated)    mir(NotActive) noConfig: Off
Definition: IngrPort(2),SrcDev(0x050240),DstDev(0x05c200)

Flow Generator (Activated):
-----------------------
|  SrcDev  |  DstDev  |
-----------------------
| 0x050240 | 0x05c200 |
-----------------------
Number of frames generated from IngrPort :   1.21G
Note: More than 1 flow active on this port.
=======================================================================
```

- To display all the zoned devices for a flow, enter **flow --show -allzoned** *port_options frame_options*. Other flow parameters should not be specified. Refer to Identifying all zoned devices in a flow on page 36 for more details.
- Specifying **fabinfo** as the feature for the **flow --show** command displays a summary topology of all the paths taken by the flows in the flow definition, including switch level information and MAPS rule violations for each of the switches the flow passes through. Refer to Viewing fabric information with the Fabric Flow Dashboard on page 40 for additional information on this feature.
- For Flow Monitor flows only, including either the **-increase** or **-decrease** option affects the order in which the flows are displayed. This (in combination with **-count** option) allows you to define how many flow definition monitors to display in increasing or decreasing order (based on the throughput) for a learning flow monitor. For example, when you specify **-increase 3 -count 2**, the output of three flows is displayed and repeated twice.

  NOTE
  The **-increase** and **-decrease** options can be used only for monitor flows. These options are not used for Flow Generator or Flow Mirror flows.

The following example displays all the existing flows on the switch. Flow names with a "sys_" prefix are predefined flows, and "+" in the Feature column indicates that the named feature is currently active for the flow.

```
switch:admin> flow --show
--------------------------------------------------------------------------------------
      Flow Name   |Feat.|SrcDev|DstDev|IngrPt|EgrPt|BiDir|LUN|FrameType|SFID|DFID|MirPt
--------------------------------------------------------------------------------------
sys_mon_analytics|mon+ |*     |*     |*     |-    |no   |*  |-        |-   |-   |-
--------------------------------------------------------------------------------------
  + Denotes feature is currently activated for the flow
  The flow name with prefix sys_ denotes predefined flow
```

Refer to the individual features to see feature-specific output, and to the *Brocade Fabric OS Command Reference* for a complete description of the **flow** command options.

Repeating flow output

You can configure the Flow Vision Flow Monitor, Flow Mirror, and Flow Generator features to repeat their flow output. The purpose of repeating a flow is so that you can view sample frames or output over time to look for differences, allowing you to continuously monitor a changing situation. You can specify that a flow output is to be repeated from one to ten times.

To specify the number of times a flow output should be repeated, complete the following steps.

1. Connect to the switch and log in using an account with admin permissions.
2. Use the **flow --show** *flow_name* **-feature** *feature_name* **-count** *num* command.

   The *num* value can range from 1 through 10. The default value is 1.

Example of repeating a Flow Monitor flow

The following example creates a Flow Monitor flow named "ag159_flow_2", and then repeats the output two times.

```
switch:admin> flow --create ag159_flow_2 -feature monitor
                    -srcdev 10:05:00:11:0d:78:45:02 -dstdev 10:00:8c:7c:ff:43:c0:01
                    -ingrport 3/2 -bidir

switch:admin> flow --show ag159_flow_2 -feature monitor -count 2
===========================================================================================
Name      : ag159_flow_2        Features: mon(Activated)        noConfig: Off
Definition: IngrPort(3/2),SrcDev(10:05:00:11:0d:78:45:02),DstDev(10:00:8c:7c:ff:43:c0:01),BiDir

Flow Monitor (Activated):
Monitor time:   | Mon Mar 14 19:04:42 UTC 2016 |
---------------------------------------------------------
--------------------------------------------------------------------------------------------------------
|      Frame Count       |  Frames Per Sec.    |    Byte Count       |    Throughput(Bps)    |FrmSize(B)  |
|   Tx  /    Rx   / Total | Tx     / Rx   /Total | Tx     /Rx    /Total  | Tx    / Rx    /Total   | Tx / Rx    |
--------------------------------------------------------------------------------------------------------
|382.91M/394.17M/777.08M|46.93k/48.16k/95.10k|703.75G/701.23G/1.37T|88.34M/87.72M/176.07M|1976/1912   |
--------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------
|      I/O Count         |   I/O Per Sec.(IOPS)   | I/O bytes Transferred |    I/O bytes Per Sec.  |
| Reads / Writes/ Total | Reads / Writes/ Total | Reads / Writes/ Total | Reads / Writes/ Total |
--------------------------------------------------------------------------------------------------
| 11.26M/ 11.26M/ 22.52M|  1.40k/  1.39k/  2.80k|687.43G/687.43G/  1.34T| 87.85M/ 87.46M/175.31M|
--------------------------------------------------------------------------------------------------
==========================================================================================================
==========================================================================================================
Name      : ag159_flow_2        Features: mon(Activated)        noConfig: Off
Definition: IngrPort(3/2),SrcDev(10:05:00:11:0d:78:45:02),DstDev(10:00:8c:7c:ff:43:c0:01),BiDir

Flow Monitor (Activated):
Monitor time:   | Mon Mar 14 19:04:48 UTC 2016 |
---------------------------------------------------------
----------------------------------------------------------------------------------------------------------
```

```
|      Frame Count       |  Frames Per Sec.    |     Byte Count        |    Throughput(Bps)      |FrmSize(B)
|   Tx  /    Rx  / Total | Tx    / Rx    /Total  | Tx     /Rx    /Total  | Tx     / Rx     /Total  |Tx  / Rx |
-----------------------------------------------------------------------------------------------------------
|383.23M/394.50M/777.74M|53.69k/55.40k/109.09k|704.34G/701.82G/1.37T|101.04M/100.94M/201.98M|1976/1912|
-----------------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------
|      I/O Count        |   I/O Per Sec.(IOPS)   | I/O bytes Transferred |   I/O bytes Per Sec.  |
| Reads / Writes/ Total | Reads / Writes/ Total | Reads / Writes/ Total | Reads / Writes/ Total |
-------------------------------------------------------------------------------------------------
| 11.27M/ 11.27M/ 22.54M|  1.53k/  1.54k/  3.08k|   688G/688.00G/  1.34T| 96.20M/ 96.69M/192.89M|
-------------------------------------------------------------------------------------------------
===========================================================================================================
```

Example of repeating a Flow Generator flow

The following example creates a Flow Generator flow named "simflow_1", and then repeats the output three times.

```
switch:admin> flow --create simflow_1 -feature generator
                            -srcdev 07f000 -dstdev 371400 -ingrport 12/16

switch:admin> flow --show simflow_1 -feature generator -count 3
================================================================================
Name      : simflow_1   Features: gen(Activated),mon(Activated) noConfig: Off
Definition: IngrPort(12/16),SrcDev(0x07f000),DstDev(0x371400)
Flow Generator (Activated):
-----------------------
|  SrcDev   |  DstDev  |
-----------------------
| 0x07f000 | 0x371400 |
-----------------------
Number of frames generated from IngrPort :   2.57G
================================================================================
================================================================================
Name      : simflow_1   Features: gen(Activated),mon(Activated) noConfig: Off
Definition: IngrPort(12/16),SrcDev(0x07f000),DstDev(0x371400)
Flow Generator (Activated):
-----------------------
|  SrcDev   |  DstDev  |
-----------------------
| 0x07f000 | 0x371400 |
-----------------------
Number of frames generated from IngrPort :   2.57G
================================================================================
================================================================================
Name      : simflow_1   Features: gen(Activated),mon(Activated) noConfig: Off
Definition: IngrPort(12/16),SrcDev(0x07f000),DstDev(0x371400)
Flow Generator (Activated):
-----------------------
|  SrcDev   |  DstDev  |
-----------------------
| 0x07f000 | 0x371400 |
-----------------------
Number of frames generated from IngrPort :   2.58G
================================================================================
```

Example of repeating a Flow Mirror flow

The following example creates a bidirectional Flow Mirror flow named "count_cfm" that is mirrored to the CPU, and repeats the output three times.

```
switch:admin> flow --create count_cfm -feature mir -ingrport 10
                            -srcdev 010403 -dstdev 020504 -bidir

switch:admin> flow --show count_cfm -fea mir -count 3
==============================================================================================
Name      : count_cfm    Features: mir(Activated)    noConfig: Off
Definition: IngrPort(14),SrcDev(0x010e00),DstDev(0x010f00),BiDir
```

```
Flow Mirror (Activated):
------------------------------------------------------------------------------------
| OXID | RXID | SOF    | EOF      | Frame_type  | LUN(*) | Dir | Time-Stamp         |
------------------------------------------------------------------------------------
| 0001 | ffff | SOFn3  | EOFn     | Data        | ----   | Rx  | Jun 05 07:54:27:100 |
| 0044 | ffff | SOFn3  | EOFn     | Data        | ----   | Tx  | Jun 05 07:54:27:100 |
(output truncated)
| 0001 | ffff | SOFn3  | EOFn     | Data        | ----   | Rx  | Jun 05 07:54:31:109 |
| 0044 | ffff | SOFn3  | EOFn     | Data        | ----   | Tx  | Jun 05 07:54:31:109 |
-------------------------------------------------------------------------------------------
 No of Mirrored Frames : 1280, No of RX Mirrored Frames : 640, No of TX Mirrored Frames : 640
-------------------------------------------------------------------------------------------
===========================================================================================
===========================================================================================
Name      : count_cfm    Features: mir(Activated)    noConfig: Off
Definition: IngrPort(14),SrcDev(0x010e00),DstDev(0x010f00),BiDir

Flow Mirror (Activated):
------------------------------------------------------------------------------------
| OXID | RXID | SOF    | EOF      | Frame_type  | LUN(*) | Dir | Time-Stamp         |
------------------------------------------------------------------------------------
| 0001 | ffff | SOFn3  | EOFn     | Data        | ----   | Rx  | Jun 05 07:54:34:100 |
| 0044 | ffff | SOFn3  | EOFn     | Data        | ----   | Tx  | Jun 05 07:54:34:100 |
(output truncated)
| 0001 | ffff | SOFn3  | EOFn     | Data        | ----   | Rx  | Jun 05 07:54:38:109 |
| 0044 | ffff | SOFn3  | EOFn     | Data        | ----   | Tx  | Jun 05 07:54:38:109 |
-------------------------------------------------------------------------------------------
 No of Mirrored Frames : 1280, No of RX Mirrored Frames : 640, No of TX Mirrored Frames : 640
-------------------------------------------------------------------------------------------
===========================================================================================
===========================================================================================
Name      : count_cfm    Features: mir(Activated)    noConfig: Off
Definition: IngrPort(14),SrcDev(0x010e00),DstDev(0x010f00),BiDir

Flow Mirror (Activated):
------------------------------------------------------------------------------------
| OXID | RXID | SOF    | EOF      | Frame_type  | LUN(*) | Dir | Time-Stamp         |
------------------------------------------------------------------------------------
| 0001 | ffff | SOFn3  | EOFn     | Data        | ----   | Rx  | Jun 05 07:54:40:100 |
| 0044 | ffff | SOFn3  | EOFn     | Data        | ----   | Tx  | Jun 05 07:54:40:100 |
(output truncated)
| 0001 | ffff | SOFn3  | EOFn     | Data        | ----   | Rx  | Jun 05 07:54:44:109 |
| 0044 | ffff | SOFn3  | EOFn     | Data        | ----   | Tx  | Jun 05 07:54:44:109 |
-------------------------------------------------------------------------------------------
 No of Mirrored Frames : 1280, No of RX Mirrored Frames : 640, No of TX Mirrored Frames : 640
-------------------------------------------------------------------------------------------
===========================================================================================
```

The following example creates a Flow Mirror flow named “count16_lfm” that is mirrored to local port 16, and repeats the output five times.

```
switch:admin> flow --create count16_lfm -feature mir -ingrport 14
                            -srcdev 010403 -dstdev 020504 -mirrorport 16

switch:admin> flow --show count16_lfm -feature mir -count 5
====================================================================================================
Name      : count16_lfm      Features: mir(Activated)        noConfig: Off
Definition: IngrPort(14),SrcDev(0x010e00),DstDev(0x010f00),MirPort(16)

Flow Mirror (Activated):
----------------------------------------------------------------------------------------------------
 No of Mirrored Frames : 1032316, No of RX Mirrored Frames : 1032316, No of TX Mirrored Frames : 0
----------------------------------------------------------------------------------------------------
====================================================================================================
====================================================================================================
Name      : count16_lfm      Features: mir(Activated)        noConfig: Off
Definition: IngrPort(14),SrcDev(0x010e00),DstDev(0x010f00),MirPort(16)

Flow Mirror (Activated):
----------------------------------------------------------------------------------------------------
 No of Mirrored Frames : 1267119, No of RX Mirrored Frames : 1267119, No of TX Mirrored Frames : 0
```

```
-----------------------------------------------------------------------------------------------
===============================================================================================
===============================================================================================
Name      : count16_lfm      Features: mir(Activated)        noConfig: Off
Definition: IngrPort(14),SrcDev(0x010e00),DstDev(0x010f00),MirPort(16)

Flow Mirror (Activated):
-----------------------------------------------------------------------------------------------
 No of Mirrored Frames : 1501921, No of RX Mirrored Frames : 1501921, No of TX Mirrored Frames : 0
-----------------------------------------------------------------------------------------------
===============================================================================================
===============================================================================================
Name      : count16_lfm      Features: mir(Activated)        noConfig: Off
Definition: IngrPort(14),SrcDev(0x010e00),DstDev(0x010f00),MirPort(16)

Flow Mirror (Activated):
-----------------------------------------------------------------------------------------------
 No of Mirrored Frames : 1736723, No of RX Mirrored Frames : 1736723, No of TX Mirrored Frames : 0
-----------------------------------------------------------------------------------------------
===============================================================================================
===============================================================================================
Name      : count16_lfm      Features: mir(Activated)        noConfig: Off
Definition: IngrPort(14),SrcDev(0x010e00),DstDev(0x010f00),MirPort(16)

Flow Mirror (Activated):
-----------------------------------------------------------------------------------------------
 No of Mirrored Frames : 1971525, No of RX Mirrored Frames : 1971525, No of TX Mirrored Frames : 0
-----------------------------------------------------------------------------------------------
===============================================================================================
```

Sorting flow output

In Flow Vision, frames can be sorted whether or not sub-flows are present. Sorting the output allows you to highlight a selected aspect of the flow data.

To sort the flow output, complete the following steps.

1. Connect to the switch and log in using an account with admin permissions.
2. Use the **flow --show** *flow_name* **-feature** *feature_name* **-sortby column***column_num* command. The **column***column_num* value is the number of the output column on which the data is to be sorted. There is no space between “column” and the column number.

   NOTE
   The **-sortby** parameter can only be applied when there is only one feature (monitor, mirror, or generator) specified in the **flow --show** *flow_name* command.

Example of sorting a Flow Monitor flow

Flow Monitor flows only need sorting if they are learned flows. Sorting with the dstdev and srcdev explicitly defined does not make sense, because there will be only one line of data in the output.

The following example creates the Flow Monitor flow called “neutrons,” and then shows the output sorted by column 4, the Destination ID. The table headings have been edited so that they will display more clearly in this document.

```
switch:admin> flow --create neutrons -feature monitor
                    -egrport 1212 -dstdev "*" -srcdev "*"

switch:admin> flow --show neutrons -feature monitor -sortby column4

Monitor time:  | Mon Mar 14 21:46:52 UTC 2016 |
---------------------------------------------------------
=========================================================================================
```

```
Name      : neutrons    Features: mon(Activated)        noConfig: Off
Definition: EgrPort(1212),SrcDev(*),DstDev(*),SFID(*),DFID(*)

Flow Monitor (Activated):
-------------------------------------------------------------------------------------------
|SFID(*)|DFID(*)|SID(*)|DID(*)|Tx Frm Cnt|Tx Frm/Sec.|Tx Bytes   |Tx Through- |Avg Tx     |
|       |       |      |      |          |           |  Cnt      |  put(Bps)  | Frm Sz(B) |
-------------------------------------------------------------------------------------------
|     25|     25|0ffe80|01cd40|  50.59k  |   8.38k   |   1.28G   |   16.97M   |    2124   |
|     25|     25|0fffc0|01cec0| 752.38k  |   9.65k   |   1.48G   |   19.55M   |    2124   |
|     25|     25|0fff40|01e800| 634.85k  |   8.17k   |   1.25G   |   16.55M   |    2124   |
|     25|     25|0ffe00|01efc0| 742.98k  |   9.53k   |   1.46G   |   19.30M   |    2124   |
|     45|     45|3c8340|5ac9c0| 303.27k  |   3.92k   | 614.30M   |    7.94M   |    2120   |
|     45|     45|3cf140|5ac9c0| 174.55k  |   2.22k   | 353.58M   |    4.49M   |    2124   |
|     45|     45|3cfc00|5ac9c0| 562.38k  |   7.27k   |   1.11G   |   14.73M   |    2120   |
|     45|     45|3cfd00|5aca00| 981.76k  |  12.61k   |   1.94G   |   25.54M   |    2120   |
|     45|     45|3cb000|5aca00|   1.02M  |  12.99k   |   2.03G   |   26.31M   |    2120   |
|     45|     45|3c8340|5aca00| 301.74k  |   3.90k   | 611.20M   |    7.90M   |    2124   |
|     45|     45|3cdcc0|5aca00| 653.07k  |   8.48k   |   1.29G   |   17.17M   |    2120   |
|     45|     45|3cfd00|5aca40| 960.02k  |  12.33k   |   1.89G   |   24.97M   |    2120   |
|     45|     45|3cbe00|5aca40| 418.35k  |   5.33k   | 847.41M   |   10.81M   |    2120   |
|     45|     45|3c8340|5aca80| 262.76k  |   3.41k   | 532.25M   |    6.91M   |    2120   |
|     45|     45|3cdcc0|5aca80| 630.51k  |   8.16k   |   1.24G   |   16.54M   |    2124   |
|     45|     45|3cfd00|5aca80| 946.89k  |  12.17k   |   1.87G   |   24.65M   |    2120   |
|     45|     45|3cf140|5aca80| 175.48k  |   2.22k   | 355.45M   |    4.51M   |    2124   |
|     45|     45|3cbd00|5aca80|  52.19k  |   661     | 105.72M   |    1.33M   |    2124   |
|     45|     45|3cbd00|5acac0|  64.22k  |   807     | 130.09M   |    1.63M   |    2124   |
|     45|     45|3c82c0|5acac0| 209.24k  |   2.75k   | 423.85M   |    5.58M   |    2124   |
|     45|     45|3cfc00|5acac0| 570.87k  |   7.36k   |   1.12G   |   14.92M   |    2124   |
|     45|     45|3c8340|5acac0| 404.76k  |   5.23k   | 819.88M   |   10.61M   |    2124   |
|     45|     45|3cf140|5acac0| 168.94k  |   2.14k   | 342.21M   |    4.34M   |    2124   |
|     45|     45|3cf040|5acac0| 108.99k  |   1.40k   | 220.78M   |    2.84M   |    2124   |
|     45|     45|3cdd40|5acac0| 412.87k  |   5.34k   | 836.30M   |   10.81M   |    2120   |
|     45|     45|3cbe00|5acac0| 396.50k  |   5.05k   | 803.16M   |   10.23M   |    2124   |
|     45|     45|3c8300|5ace00| 970.02k  |  12.43k   |   1.91G   |   25.19M   |    2124   |
|     45|     45|3cbcc0|5ace00|  42.39k  |   538     |  85.87M   |    1.09M   |    2124   |
|     45|     45|3cdd00|5ace00| 952.45k  |  12.37k   |   1.88G   |   25.07M   |    2120   |
|     45|     45|3cbf40|5ace00| 448.57k  |   5.67k   | 908.62M   |   11.49M   |    2124   |
```

Sorting flow output based on device performance

You can view the performance of devices being monitored by a flow. You can also sort the output based on the values in a column of the output or the throughput of the devices.

To display the performance of devices being monitored by a flow, use the following syntax:

- **flow --show sys_mon_all_fports -feature mon**

To sort the flow output based on the values in a particular column, use the following syntax:

- **flow --show sys_mon_all_fports -feature mon -sortby column***column_number*

You can also sort the flow output based on the throughput of devices. This capability is useful when you want to display the devices with the most or least throughput (top or bottom talkers) being monitored by the flow.

The syntax for showing top or bottom talkers is as follows:

- **flow --show sys_mon_all_fports -feature mon -increase** *count* |**-decrease** *count*

Listing bottom talkers: The option **-increase** *count* lists the number of devices specified with *count* and orders them starting with the device with the least throughput.

Listing top talkers: The option **-decrease** *count* lists the number of devices specified with *count* and orders them starting with the device with the most throughput.

Example of viewing the performance of devices being monitored by a flow

The following example displays a snapshot of device performance.

```
switch:admin> flow --show sys_mon_all_fports -feat mon
==============================================================================
Name      : sys_mon_all_fports  Features: mon(Activated)        noConfig: Off
Definition: IngrPort(*),SrcDev(*),DstDev(*)


Flow Monitor (Activated):
Monitor time:   | Mon Mar 21 07:37:18 UTC 2016 |
----------------------------------------------------------
------------------------------------------------------------------------------
|Port|SID(*)|DID(*)| Rx Frames| Rx Frames| Rx Bytes| Rx Through-| Avg Rx Frm|
|                  |   Count  |  per Sec.|   Count |   put(Bps) |  Sz(Bytes)|
------------------------------------------------------------------------------
| 10 |630400|630200|  1.65M   |  276.42k |   1.62G |  277.52M   |  1052     |
| 10 |630400|630600|  1.65M   |  276.31k |   2.46G |  420.99M   |  1600     |
| 10 |630400|630500|  1.65M   |  276.28k |   1.91G |  326.14M   |  1240     |
| 10 |630400|630100|  1.65M   |  276.21k |   2.46G |  420.83M   |  1600     |
| 12 |630200|630600|  2.01M   |  335.81k |   2.37G |  405.09M   |  1264     |
| 12 |630200|630500|  2.01M   |  335.75k |   2.19G |  375.03M   |  1172     |
| 12 |630200|630100|  2.01M   |  335.72k |   2.05G |  350.73M   |  1096     |
| 12 |630200|630300|  2.01M   |  335.69k |   1.77G |  302.84M   |   948     |
| 14 |630600|630500|  2.18M   |  364.68k |   1.65G |  282.07M   |   812     |
| 14 |630600|630100|  2.18M   |  364.65k |   2.25G |  385.66M   |  1108     |
| 14 |630600|630300|  2.18M   |  364.59k |   2.81G |  480.47M   |  1384     |
| 14 |630600|630b00|  2.18M   |  364.56k |   1.62G |  277.06M   |   796     |
| 15 |630500|630100|  1.88M   |  313.96k |   2.41G |  411.44M   |  1376     |
| 15 |630500|630300|  1.88M   |  313.81k |   1.99G |  340.04M   |  1136     |
------------------------------------------------------------------------------
| *  |   *  |  *   | 19.63G   |   19.01M |  19.83T |   20.52G   |  1084     |
------------------------------------------------------------------------------
```

Example of sorting flow output based on values in a specified column

The following example displays flow output that has been sorted by the values in column 5, Rx Frames per Sec. (The "SID(*)" column is skipped.)

```
switch:admin> flow --show sys_mon_all_fports -feat mon –sortby column5
==============================================================================
Name      : sys_mon_all_fports  Features: mon(Activated)        noConfig: Off
Definition: IngrPort(*),SrcDev(*),DstDev(*)


Flow Monitor (Activated):
Monitor time:   | Mon Mar 21 07:29:11 UTC 2016 |
----------------------------------------------------------
----------------------------------------------------------------------------------
|Port|SID(*)|DID(*)| Rx Frames | Rx Frames|  Rx Bytes| Rx Through- | Avg Rx Frm |
|    |      |      |  Count    |  per Sec.|   Count |   put (Bps) |  Sz (Bytes)|
----------------------------------------------------------------------------------
| 14 |630600|630500| 197.86M   | 322.18k  |  192.68G |   331.82M   |      1048  |
| 14 |630600|630500| 197.86M   | 322.18k  |  192.68G |   331.82M   |      1048  |
| 14 |630600|630100| 194.75M   | 322.18k  |  206.60G |   310.81M   |      1140  |
| 14 |630600|630100| 194.75M   | 322.18k  |  206.60G |   310.81M   |      1140  |
| 11 |630100|630500| 147.39M   | 323.55k  |  157.56G |   336.92M   |      1148  |
| 17 |632f00|630300| 147.03M   | 338.76k  |  151.37G |   320.75M   |      1108  |
| 13 |630300|630500| 150.47M   | 344.95k  |  159.29G |   327.21M   |      1136  |
| 16 |630b00|632f00| 203.56M   | 378.37k  |  220.78G |   268.43M   |      1164  |
| 16 |630b00|632f00| 203.56M   | 378.37k  |  220.78G |   268.43M   |      1164  |
| 15 |630500|630100| 191.28M   | 379.46k  |  195.65G |   321.34M   |      1100  |
| 15 |630500|630100| 191.28M   | 379.46k  |  195.65G |   321.34M   |      1100  |
| 15 |630500|630300| 190.55M   | 379.46k  |  206.52G |   300.92M   |      1164  |
| 15 |630500|630300| 190.55M   | 379.46k  |  206.52G |   300.92M   |      1164  |
| 15 |630400|630200| 185.78M   | 382.43k  |  185.88G |   306.35M   |      1076  |
| 15 |630400|630200| 185.78M   | 382.43k  |  185.88G |   306.35M   |      1076  |
| 12 |630200|630400| 149.11M   | 392.42k  |  146.26G |   343.28M   |      1056  |
```

```
--------------------------------------------------------------------------------
| *  |  *   |  *   |  19.63G   |  19.01M  |   19.83T |    20.52G   |     1084   |
--------------------------------------------------------------------------------
```

Examples of sorting flow output based on device throughput

The following example displays the 16 bottom talkers (sorted by *increasing* values in the column labeled "Rx Throughput (Bps)."

```
switch:admin> flow --show sys_mon_all_fports -feat mon -increase 16
=================================================================================
Name      : sys_mon_all_fports  Features: mon(Activated)        noConfig: Off
Definition: IngrPort(*),SrcDev(*),DstDev(*)


Flow Monitor (Activated):
Monitor time:  | Mon Mar 21 07:42:21 UTC 2016 |
---------------------------------------------------------
------------------------------------------------------------------------------------
|Ingr(*)|SID(*)|DID(*)| Rx Frames| Rx Frames | Rx Bytes | Rx Through- | Avg Rx Frm |
|       |      |      |   Count  | per Sec.  |   Count  |   put (Bps) |  Sz (Bytes)|
------------------------------------------------------------------------------------
| 16    |630b00|632f00|  203.56M |  378.37k  |   220.78G | 268.43M    |  1164      |
| 16    |630b00|632f00|  203.56M |  378.37k  |   220.78G | 268.43M    |  1164      |
| 15    |630500|630300|  190.55M |  379.46k  |   206.52G | 300.92M    |  1164      |
| 15    |630500|630300|  190.55M |  379.46k  |   206.52G | 300.92M    |  1164      |
| 15    |630400|630200|  185.78M |  382.43k  |   185.88G | 306.35M    |  1076      |
| 15    |630400|630200|  185.78M |  382.43k  |   185.88G | 306.35M    |  1076      |
| 14    |630600|630100|  194.75M |  322.18k  |   206.60G | 310.81M    |  1140      |
| 14    |630600|630100|  194.75M |  322.18k  |   206.60G | 310.81M    |  1140      |
| 17    |632f00|630300|  147.03M |  338.76k  |   151.37G | 320.75M    |  1108      |
| 15    |630500|630100|  191.28M |  379.46k  |   195.65G | 321.34M    |  1100      |
| 15    |630500|630100|  191.28M |  379.46k  |   195.65G | 321.34M    |  1100      |
| 13    |630300|630500|  150.47M |  344.95k  |   159.29G | 327.21M    |  1136      |
| 14    |630600|630500|  197.86M |  322.18k  |   192.68G | 331.82M    |  1048      |
| 14    |630600|630500|  197.86M |  322.18k  |   192.68G | 331.82M    |  1048      |
| 11    |630100|630500|  147.39M |  323.55k  |   157.56G | 336.92M    |  1148      |
| 12    |630200|630400|  149.11M |  392.42k  |   146.26G | 343.28M    |  1056      |
------------------------------------------------------------------------------------
|  *    |  *   |  *   |   19.63G |   19.01M  |    19.83T |  20.52G    |  1084      |
------------------------------------------------------------------------------------
```

The following example displays the 5 top talkers (sorted by *decreasing* values in the column labeled "Rx Throughput (Bps)."

```
switch:admin> flow --show sys_mon_all_fports -fe mon -decrease 5
=====================================================================================
Name      : sys_mon_all_fports  Features: mon(Activated)        noConfig: Off
Definition: IngrPort(*),SrcDev(*),DstDev(*)


Flow Monitor (Activated):
Monitor time:  | Wed Apr 13 23:44:42 UTC 2016 |
---------------------------------------------------------

-------------------------------------------------------------------------------------
|Ingr(*)|SID(*)|DID(*)| Rx Frames| Rx Frames | Rx Bytes | Rx Through- | Avg Rx Frm |
|       |      |      |  Count   | per Sec.  |  Count   |  put (Bps)  |  Sz (Bytes)|
 ------------------------------------------------------------------------------------
|19     |011300|011000| 602.69M  |  373.06k  | 613.67G  |  515.91M    |    1096    |
|16     |011000|010e00| 612.84M  |  367.26k  | 608.71G  |  500.46M    |    1068    |
|19     |011300|011200| 598.89M  |  373.06k  | 594.41G  |  491.57M    |    1068    |
|13     |010d00|010f00| 619.69M  |  353.45k  | 653.47G  |  477.04M    |    1132    |
|18     |011200|010a00| 489.73M  |  445.34k  | 497.91G  |  475.86M    |    1092    |
-------------------------------------------------------------------------------------
|*      |  *   |   *  |  57.11G  |   16.16M  |  57.68T  |   16.89G    |    1112    |
-------------------------------------------------------------------------------------
```

Example of sorting a Flow Generator flow

The following example creates a Flow Generator flow, and then shows the output sorted by column 2.

```
switch:admin> flow --create fgSort -fea gen -ing 10 -srcdev "*" -dstdev "*"
```

Output before sorting:

```
switch:admin> flow --show fgSort
=====================================================================================
Name      : gen1    Features: gen(Activated)    noConfig: Off
Definition: IngrPort(10),SrcDev(*),DstDev(*)

Flow Generator (Activated):
-----------------------
|  SrcDev   |  DstDev  |
-----------------------
| 0x040a00 | 0x041c00 |
-----------------------
| 0x040a00 | 0x040b00 |
-----------------------
| 0x040a00 | 0x041d00 |
-----------------------
| 0x040a00 | 0x012500 |
-----------------------
Number of frames generated from IngrPort :   2.33G
```

Output after sorting:

```
switch:admin> flow --show fgSort -feature generator -sortby column2
=====================================================================================
Name      : gen1    Features: gen(Activated)    noConfig: Off
Definition: IngrPort(10),SrcDev(*),DstDev(*)

Flow Generator (Activated):
-----------------------
|  SrcDev   |  DstDev  |
-----------------------
| 0x040a00 | 0x012500 |
-----------------------
| 0x040a00 | 0x040b00 |
-----------------------
| 0x040a00 | 0x041c00 |
-----------------------
| 0x040a00 | 0x041d00 |
-----------------------
Number of frames generated from IngrPort :   2.42G
=====================================================================================
```

Example of sorting a Flow Mirror flow

The following example creates a Flow Mirror flow, and then shows the output sorted by column 3, the OXID.

```
switch:admin> flow --create sortMirror -feature mirror -egrport 15 -srcdev "*" -bidir

switch:admin> flow --show sortMirror -feature mirror -sortby column3
=======================================================================================================
Name      : sortMirror    Features: mir(Activated)    noConfig: Off
Definition: EgrPort(15),SrcDev(*),BiDir

Flow Mirror (Activated):
-------------------------------------------------------------------------------------------------------
| SID(*)   | DID(*)   | OXID | RXID | SOF    | EOF      | Frame_type | LUN(*)  | Dir | Time-Stamp          |
-------------------------------------------------------------------------------------------------------
| 010e00   | 010f00   | 0001 | ffff | SOFn3 | EOFn     | Data       | ----    | Tx  | Jun 05 10:16:50:100 |
| 010e00   | 010f00   | 0001 | ffff | SOFn3 | EOFn     | Data       | ----    | Tx  | Jun 05 10:16:50:101 |
| 010e00   | 010f00   | 0001 | ffff | SOFn3 | EOFn     | Data       | ----    | Tx  | Jun 05 10:16:50:102 |
| 010e00   | 010f00   | 0001 | ffff | SOFn3 | EOFn     | Data       | ----    | Tx  | Jun 05 10:16:50:103 |
| 010e00   | 010f00   | 0001 | ffff | SOFn3 | EOFn     | Data       | ----    | Tx  | Jun 05 10:16:50:104 |
```

```
| 010e00   | 010f00   | 0001 | ffff | SOFn3 | EOFn     | Data        | ----     | Tx   | Jun 05 10:16:50:105 |
| 010e00   | 010f00   | 0001 | ffff | SOFn3 | EOFn     | Data        | ----     | Tx   | Jun 05 10:16:50:106 |
| 010e00   | 010f00   | 0001 | ffff | SOFn3 | EOFn     | Data        | ----     | Tx   | Jun 05 10:16:50:107 |
| 010e00   | 010f00   | 0001 | ffff | SOFn3 | EOFn     | Data        | ----     | Tx   | Jun 05 10:16:50:108 |
| 010e00   | 010f00   | 0001 | ffff | SOFn3 | EOFn     | Data        | ----     | Tx   | Jun 05 10:16:50:109 |
| 010e00   | 010f00   | 0001 | ffff | SOFn3 | EOFn     | Data        | ----     | Tx   | Jun 05 10:16:50:110 |
| 010e00   | 010f00   | 0001 | ffff | SOFn3 | EOFn     | Data        | ----     | Tx   | Jun 05 10:16:51:104 |
| 010f00   | 010e00   | 0044 | ffff | SOFn3 | EOFn     | Data        | ----     | Rx   | Jun 05 10:16:51:111 |
| 010f00   | 010e00   | 0044 | ffff | SOFn3 | EOFn     | Data        | ----     | Rx   | Jun 05 10:16:52:104 |
| 010f00   | 010e00   | 0044 | ffff | SOFn3 | EOFn     | Data        | ----     | Rx   | Jun 05 10:16:52:104 |
| 010f00   | 010e00   | 0044 | ffff | SOFn3 | EOFn     | Data        | ----     | Rx   | Jun 05 10:16:53:106 |
| 010f00   | 010e00   | 0044 | ffff | SOFn3 | EOFn     | Data        | ----     | Rx   | Jun 05 10:16:54:109 |
  (output truncated)
----------------------------------------------------------------------------------------------
 No of Mirrored Frames : 1280, No of RX Mirrored Frames : 640, No of TX Mirrored Frames : 640
 ---------------------------------------------------------------------------------------------
=========================================================================================================
```

Identifying all zoned devices in a flow

Flow Vision allows you to identify all the flows (actual and possible) passing through a specified E_Port or F_Port, subject to zoning restrictions.

This feature is valid for all Flow Vision flows. If a flow is already defined, you can view all the zoned devices in the flow using **flow --show** *flow_name* **-allzoned**. Alternatively you can enter the **flow show** command and completely define in this command the parameters for the flow definition and include the **-allzoned** parameter. This allows you to see all the zoned devices matching the flow definition without creating an actual flow. The advantage to this techniques is that it does not interfere with any existing flows. One use case for this feature would be to assist you in determining a host-to-target ratio or device-to-ISL ratio and keeping such a ratio to a user-determined value for optimal performance based on standard best practice recommendations for SANs.

The following table lists the combinations of flow definition values (Source ID, Destination ID, and flow terminus) for flows that will let you identify devices in a zone.

TABLE 9 Supported flow definition values for zoned device identification

Source ID	Destination ID	Monitorable port
*	WWN or PID	Ingress or Egress (fixed)
WWN or PID	*	Ingress or Egress (fixed)
*	*	Ingress or Egress (fixed)
* or WWN or PID	Not defined	Ingress or Egress (fixed)
Not defined	* or WWN or PID	Ingress or Egress (fixed)
WWN or PID	WWN or PID	Ingress or Egress (fixed)

Examples of show command output for zoned devices

The following example displays all devices zoned with the device at port 11 (an F_Port) as the destination port.

```
switch:admin> flow --show -allzoned -srcdev "*" -dstdev 0x190b00 -egr 11
Active Flows:
---------------------------------------------------------
| SrcDev                                      | DstDev   |
---------------------------------------------------------
| 0x21400 0x21e00 0x21f00 0x31000 0x31100      | 0x190b00 |
---------------------------------------------------------
```

The following example displays all devices zoned with the device at port 11 (an F_Port) as the source port.

```
switch:admin> flow --show -allzoned -srcdev "*" -dstdev "*" -ingport 11
Active Flows:
----------------------------------------------------------
| SrcDev   | DstDev                                      |
----------------------------------------------------------
| 0x190b00 | 0x21400 0x21e00 0x21f00 0x31000 0x31100     |
----------------------------------------------------------
```

The following example displays all the zoned devices which can communicate over port 15 (an E_Port).

```
switch:admin> flow --show -allzoned -srcdev "*" -dstdev "*" -ingport 15
Active Flows:
----------------------------------------------------------
| SrcDev   | DstDev                                      |
----------------------------------------------------------
| 0x190a00 | 0x21400 0x21e00 0x21f00 0x31000 0x31100     |
----------------------------------------------------------
| 0x190b00 | 0x21400 0x21e00 0x21f00 0x31000 0x31100     |
----------------------------------------------------------
```

The following example demonstrates creating a flow called "AllHostFlow" and then displaying all the zoned devices in the flow using **flow --show** *flow_name* **-allzoned**.

```
switch:admin> flow --create AllHostFlow -fe mon -srcdev "*" -dstdev 0x190b00 -egrport 11

switch:admin> flow --show AllHostFlow -allzoned
Active Flows:
----------------------------------------------------------
| SrcDev                                      | DstDev   |
----------------------------------------------------------
| 0x21400 0x21e00 0x21f00 0x31000 0x31100     | 0x190b00 |
----------------------------------------------------------
```

Limitations on viewing zoned devices

The following limitations apply to viewing zoned devices through Flow Vision flows:

- This feature is supported only on E_Ports and F_Ports.
- This feature is not supported on EX_Ports or XISL ports.
- Only Master trunk ports are supported — Slave ports are not supported.
- The maximum number of flows that can be discovered is the maximum number of sub-flows supported for a flow. This count will not be affected by the maximum number of sub-flows supported across the chassis.
- This feature is not available on switches running in Access Gateway mode.

Flow deletion

Flow Vision allows you to delete either individual flows or all flows at one time.

When you delete a flow, the following actions occur:

- The specified flow is automatically deactivated before it is deleted.
- All instances of the specified flow are removed.
- Any sub-flows associated with the specified flow are removed.
- If the specified flow is a Flow Monitor or Flow Mirror flow, all flow statistics for it are automatically cleared. If the specified flow is a Flow Generator flow, the statistics are retained.

- You are not asked to confirm the deletion of a flow unless you use **all** as the flow name and do *not* use the **-force** keyword. For example: **flow --delete all**.

For more information on the **flow --delete** command, refer to the *Brocade Fabric OS Command Reference*.

Deleting flows

Flow Vision allows you to delete either a single flow or all flows.

Deleting a single flow

To delete any Flow Vision flow, complete the following steps.

1. Connect to the switch and log in using an account with admin permissions.
2. Enter **flow --delete** *flow_name*.
   The named flow is immediately deleted and cannot be recovered.

The following example deletes a Flow Monitor flow named "Flow1".

```
switch:admin> flow --delete Flow1
```

Deleting all flows at one time

To delete all Flow Vision flows at one time, complete the following steps.

1. Connect to the switch and log in using an account with admin permissions.
2. Enter **flow --delete all**.
   You are then prompted to confirm this action.
3. Enter **y**.
   All user-defined Flow Vision flows will be deleted and cannot be recovered. Predefined flows will not be deleted, but they will be deactivated.

   NOTE
   You can compel the deletion of flows by adding the **-force** keyword to the command. Using this keyword causes Flow Vision to not issue a confirmation prompt.

The following example deletes all flows without prompting you for confirmation.

```
switch:admin> flow --delete all -force
```

Resetting flow statistics

Flow Vision allows you to clear (reset) the flow statistics record for each feature individually or as a group.

NOTE
Clearing the Flow Mirror statistics for a flow also clears the mirrored frames.

Clearing the statistics for all flow features

To clear all the statistics for a flow, complete the following steps.

1. Connect to the switch and log in using an account with admin permissions.

2. Enter **flow --reset** *flow_name* **-feature all**.
   You will **not** be asked to confirm this action.

The following example clears all the statistics for the flow named "Flow4."

```
switch:admin> flow --reset Flow4 -feature all
```

Clearing the statistics for a specific flow feature

To clear the statistics for specified features of a flow, complete the following steps.

1. Connect to the switch and log in using an account with admin permissions.
2. Enter **flow --reset** *flow_name* **-feature** *feature_list*. Replace *feature_list* with either an individual feature or a comma-separated list of features (for example, "generator,monitor" or "mir,mon").
   You will **not** be asked to confirm this action.

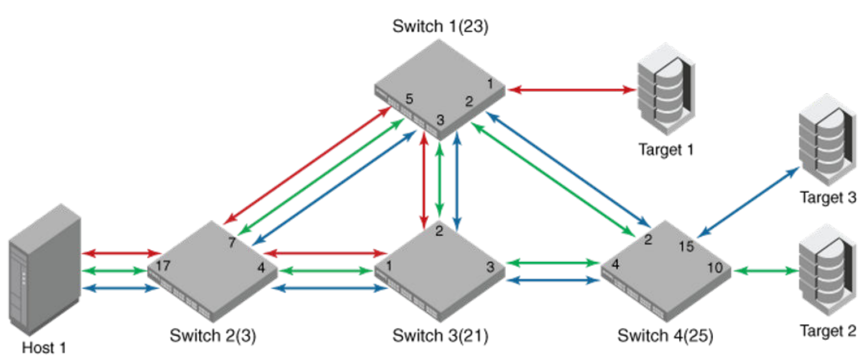
The following example clears only the Flow Monitor statistics for the flow named "Flow4."

```
switch:admin> flow --reset Flow4 -feature monitor
```

Understanding the Fabric Flow Dashboard

The following illustration displays a sample fabric that is used as a foundation for the Fabric Flow Dashboard explanation. In this discussion, the **flow** command is executed on Switch2. In the switch identification format Switch#(X), Switch# is the switch name, and X is the Domain ID. If the SAN administrator sees a performance issue affecting Host 1, he or she can then check the Fabric Flow Dashboard to determine all the targets connected to Host 1. Once the targets are known, then each of them can be checked to determine the problematic target.

FIGURE 2 Sample fabric



For the figure above, the Fabric Flow Dashboard identifies the flows as follows; notice that there are multiple potential routes for each flow.

- Flow 1: Target1 to Host1 (Red flow)
- Flow 2: Target2 to Host1 (Green flow)
- Flow 3: Target3 to Host1 (Blue flow)

The topology information for the identified flows uses the following format:

```
[SrcDev: DeviceID] <-> (Domain1/inPort, Domain1/outPort) <-> (D2/inP, D2/outP) <-> (D3/inP, D3/outP) … <->
[DstDev: DeviceID]
```

For Flow 1 in the example above, the paths are:

- `[SrcDev:Target1] <-> (SW1/F1, SW1/E5) <-> (SW2/E7, SW2/F17) <-> [DstDev:Host1]`

For Flow 2 and Flow 3, the paths are:

- `srcDev(Target2, Target3), dstDev(Host1)`
- `[srcDev: *] <-> (SW4/*, SW4/E2) <-> (SW1/E1, SW1/E5) <->(SW2/E7, SW2/17#) <-> [dstDev: Host1]`
- `[srcDev: *] <-> (SW4/*, SW4/E3) <-> (SW1/E3, SW1/E1#) <->(SW2/E4, SW2/17#) <-> [dstDev: Host1]`
- `[srcDev: *] <-> (SW4/*, SW4/E4) <-> (SW1/E4, SW1/E1#) <->(SW2/E4, SW2/17#) <-> [dstDev: Host1]`

# indicates that one or more MAPS rule violations are present in a port.

The following extract from a Fabric Flow Dashboard provides a typical example of a situation where multiple hosts/targets are connected to the source and destination switches.

```
srcDev (0x170500), dstDev(0x031100)
[srcDev: 23/1] <-> (23/1#, 23/5) <-> (3/7, 3/17) <-> [dstDev: 3/0x031100]

srcDev (0x190a00), dstDev(0x031100, 0x031200, 0x031300, 0x031400, 0x031500)
srcDev (0x190b00), dstDev(0x031100, 0x031200, 0x031300, 0x031400, 0x031500)
srcDev (0x190c00), dstDev(0x031100, 0x031200, 0x031300, 0x031400, 0x031500)
srcDev (0x190d00), dstDev(0x031100, 0x031200, 0x031300, 0x031400, 0x031500)
srcDev (0x190e00), dstDev(0x031100, 0x031200, 0x031300, 0x031400, 0x031500)
[srcDev: 25/*] <-> (25/*, 25/2) <-> (23/2#, 23/5) <->(3/7, 3/*) <-> [dstDev: 3/*]
[srcDev: 25/*] <-> (25/*, 25/3) <-> (21/3, 21/1#) <->(3/4, 3/*) <-> [dstDev: 3/*]
[srcDev: 25/*] <-> (25/*, 25/4) <-> (21/4, 21/1#) <->(3/4, 3/*) <-> [dstDev: 3/*]
```

NOTE

If the flow definition contains an E_Port as an ingress or egress port, the path will contain information starting from the E_Port until reaching the destination. For the figure above, if the **flow -show -feature fabinfo** command is run on Switch1 with the aim to identify all paths from Target 2 to Host 1 using egress port 5, the paths are identified as follows:

```
[srcDev:Target2] <-> (SW1/*, SW1/E5) <-> (SW2/E7, SW2/F17) <-> [dstDev:Host1]
```

Viewing fabric information with the Fabric Flow Dashboard

The Flow Vision Fabric Flow Dashboard provides SAN administrators with the ability to have a consolidated view of a flow across a Fabric OS fabric, allowing them to quickly isolate network issues.

You can view the Fabric Flow Dashboard by specifying **fabinfo** as the feature for the **flow --show** command, for example:

```
flow --show -feature fabinfo -srcdev "*" -egrport 17
```

The output of this command has two parts.

- **Summary topology**: The first part presents a summary topology of all the paths taken by the flows in the flow definition.
- **Switch-specific information and MAPS rule violations:** The second part presents information and MAPS rule violations for each of the switches the flow passes through. While reporting the MAPS data, preference is given to the data for ports which are in the flow path. If there are no violations on any ports in the flow path, only the category and violation count is displayed without additional details. Additional information about MAPS history data and about fenced, decommissioned, and quarantined ports is displayed only for switches running Fabric OS 8.0.x and 8.1.0.

The following example shows the results of running the command described above. The column headings in the example have been edited slightly to allow it to display clearly.

```
switch:admin> flow --show –feature fabinfo –srcdev "*" -egrport 17
Flow Dashboard Information:
==========================
Topology Data:
-------------
srcDev(0x170500), dstDev(0x31100)
srcDev(0x170500) <-> (23/1#, 23/5) <-> (3/7, 3/17) <-> dstDev(0x31100)

srcDev(0x190a00), dstDev(0x31100)
srcDev(0x190a00)<-> (25/10, 25/2)<->(23/2#, 23/5)<->(3/7, 3/17#)<-> dstDev(0x31100)
srcDev(0x190a00)<-> (25/10, 25/3)<->(21/3, 21/1#)<->(3/4, 3/17#)<-> dstDev(0x31100)
srcDev(0x190a00)<-> (25/10, 25/4)<->(21/4, 21/1#)<->(3/4, 3/17#)<-> dstDev(0x31100)

# "Indicates there are MAPS violation on these ports"

MAPS Violations on source and destination ports:
Category(RuleCnt)|RptCnt|Rule Name                |Execution Time   |Object|Triggered
                                                                            Value    |
--------------------------------------------------------------------------------------
(Source port: 23/1)
Port Health(176) |3     | defALL_TARGET_PORTSCRC_1|12/18/16 06:03:45|Port1 | 2       |
(Destination port: 3/17)
Port Health(24)  |6     | defALL_HOST_PORTSCRC_2  |12/18/16 09:13:24|Port17| 5       |
Port Health(15)  |2     | defNON_E_F_PORTSLF_0    |12/18/16 09:18:24|Port17| 3       |

Switch Specific Data:
==============================================================================
SwitchDomain: 23 (0x17)
Name: sw0    Model: 66.1    Uptime: (2 days 2 hrs 37 mins)
FirmwareVersion : v8.1.0
OperationalStatus: MARGINAL    Reason : BAD_PWR
RebootReason: Reboot
Fenced Ports         : 24
Decommissioned Ports : None
Quarantined Ports    : None
Category(RuleCnt)|RptCnt|Rule Name                |Execution Time   |Object|Triggered|
                 |      |                         |                 |      |Value    |
--------------------------------------------------------------------------------------
Port Health(176) | 3    | defALL_TARGET_PORTSCRC_1|12/18/16 06:03:45|Port1 | 2       |
                          |14    | defNON_E_F_PORTSLF_0    |12/17/16 09:13:24|Port 5| 6       |

==============================================================================

SwitchDomain: 3 (0x3)
Name: switch 3    Model: 121.3    Uptime: (41 days 0 hrs 02 mins)
FirmwareVersion : v7.4.1
OperationalStatus: Healthy
RebootReason: Reboot
Fenced Ports         : None
Decommissioned Ports : None
Quarantined Ports    : None
Category(RuleCnt)|RptCnt|Rule Name                |Execution Time   |Object|Triggered|
                 |      |                         |                 |      |Value    |
--------------------------------------------------------------------------------------
Port Health(24)  |6     | defALL_HOST_PORTSCRC_2  |12/17/16 09:13:24|Port17| 5       |
Port Health(15)  |2     | defNON_E_F_PORTSLF_0    |12/17/16 09:18:24|Port17| 3       |

==============================================================================

SwitchDomain: 25 (0x19)
Name: switch 25    Model: 121.3    Uptime: (40 days 21 hrs 12 mins)
FirmwareVersion : v7.4.0
OperationalStatus: CRITICAL    Reason : FAULTY_BLADE
RebootReason: Reboot
Fenced Ports         : None
Decommissioned Ports : None
Quarantined Ports    : None
Category(RuleCnt)|RptCnt|Rule Name                |Execution Time   |Object|Triggered|
```

```
                 |      |                          |                 |      |Value    |
--------------------------------------------------------------------------------------
Port Health(16)  | No violations on the ports in the flow path

=============================================================================

SwitchDomain: 21 (0x15)
Name: switch 21    Model: 121.3    Uptime: (40 days 03 hrs 10 mins)
FirmwareVersion : v8.0.1
OperationalStatus: Healthy
RebootReason: Reboot
Fenced Ports         : None
Decommissioned Ports : None
Quarantined Ports    : None

No MAPS violations
=============================================================================
```

Viewing additional MAPS history data

Using Fabric OS 8.1.0, administrators can specify a domain name as an option of the **fabinfo** feature, for example:

```
flow --show -feature fabinfo -srcdev 0x011100
            -egrport 17 -verbose -domain 3
```

When the domain name is specified, only data corresponding to the domain is displayed. Data for other domains is not displayed. The data displayed includes MAPS history data. Showing statistics for only those ports in the path of the flow definition, MAPS history data also provides statistics about events that have not crossed the MAPS violation threshold but still have non-zero values. Also, the data covers multiple durations.

Example of displaying MAPS history data

The following example shows the results of running the command described above. Be aware that the column headings in the example have been edited slightly to allow it to display clearly.

```
switch:admin> flow --show -feature fabinfo -srcdev 0x011100
                          -egrport 17 -verbose -domain 3

Flow Dashboard Information:
==========================

Switch Specific Data:

=============================================================================

SwitchDomain: 3 (0x3)
Name: sw0 Model: 121.3 Uptime: (41 days 0 hrs 02 mins)
FirmwareVersion : v8.1.0
OperationalStatus: Healthy
RebootReason: Reboot
Fenced Ports  : None
Decommissioned Ports : None
Quarantined Ports    : None

MAPS violation:
---------------
Port Health(24)   |6    | defALL_HOST_PORTSCRC_2|12/17/16 09:13:24|Port17| 5 |
Port Health(15)   |2    | defNON_E_F_PORTSLF_0  |12/17/16 09:18:24|Port17| 3 |

MAPS history data:
------------------

Stats   Current   12/16/16  12/15/16    12/14/16  12/13/16  12/12/16  12/11/16
(Units) Port(val) Port(val) Port(val)   Port(val) Port(val) Port(val) Port(val)
-----------------------------------------------------------------------------
CRC       32(3)     32(2)     46(34)      45(7)     32(1)     -         32(7)
          42(2)     42(2)     43(33)      44(6)     33(1)     -         33(7)
          12(1)     12(1)     42(31)      42(5)     34(1)     -         34(7)
          06(1)     06(1)     47(20)      46(5)     35(1)     -         35(7)

ITW       32(11)    32(8)     45(2300)    45(539)   32(3)     0(1)      32(21)
          33(3)     33(3)     40(2282)    40(522)   33(1)     -         33(7)
          34(3)     34(3)     44(2276)    41(519)   34(1)     -         34(7)
          35(3)     35(3)     41(2269)    44(517)   35(1)     -         35(7)
```

Limitations and restrictions on Fabric Flow Dashboard flows

The following limitations and restrictions apply to a Fabric Flow Dashboard flow.

- The Fabric Flow Dashboard command (**flow --show -feature fabinfo**) can only be run on a switch running Fabric OS 7.4.0 or later.
- This feature is not available on switches running in Access Gateway mode.
- The Fabric Flow Dashboard flow definition can only use E_Ports or F_Ports.
- EX_Ports and XISL ports are not supported in the definition.
- Only Master trunk ports are supported — Slave ports are not supported.
- The maximum number of flows that can be discovered is the maximum number of sub-flows supported for a flow. This count will not be affected by the maximum number of sub-flows supported across the chassis.
- Only the following MAPS categories are considered for the MAPS violations shown in the Fabric Flow Dashboard report:
  - Port Health
  - Traffic Performance
  - Fabric Performance Impact

- Peer zones and Target Driven zones are not considered.
- TI zones are not supported.
- The frame options **-lun** and **-frametype** are not supported for the flow definition.
- The **flowname** and **-sortby** options are not supported.
- In a fabric that includes versions of Fabric OS earlier than 7.4.0, the Fabric Flow Dashboard functionality is limited. While topology and switch data will be shown for all the switches, MAPS violation details are not available for those switches running earlier versions of Fabric OS.
- MAPS history data and back-end port data are not available for switches running a Fabric OS version earlier than 8.0.1.

Fabric Flow Dashboard use cases

The following section provides a use case for the Fabric Flow Dashboard.

Use case: Slow drain device

If you suspect that there is a slow-draining device in your fabric as evidenced by a significant number of class 3 timeouts (C3TXTO) on E_Port 5. You can quickly set up a flow to check all the flows passing through this E_Port and view the data in the flow dashboard. In this case you can specify the flow as follows:

flow -show -feature fabinfo -srcdev `"*"` -dstdev `"*"` -egrport 5

In this command example, you have specified that you want to view all flows by using `"*"` for both the source and destination devices, and identified port 5 as the egress port you are interested in.

If you see “Fabric Performance Impact” violations on any target port(s) in the output, this suggests that the E_Port timeouts are probably because of a slow draining device attached to the destination F_Port. You can then look more closely at the device(s) attached to that F_Port.

System event handling

Flow Vision handles the following system events:

- When an E_Port or F_Port comes online (PORT_ONLINE), if the resources are available, any flows specifying that port will be installed in the ASIC and made active. If the resources are not available, the flow will stay deactivated.
- When an E_Port is changed (EPORT_CHANGE), if the resources are available, any flows specifying that port will be installed in the ASIC and made active. If the resources are not available (for example, if there is already a flow using that port), the flow will stay deactivated.
- When an F_Port is changed (FPORT_CHANGE), if the resources are available, any flows specifying that port will be installed in the ASIC and made active. If the resources are not available (for example, if there is already a flow using that port), the flow will stay deactivated.

  An F_Port trunk has same trunk index for both the master and slave ports (as displayed in the output for **switchshow**, below). In order to create the same flow definition on the master and slave ports when portIdMode is set to “index”, the domain/Index can be obtained by using the **porttrunkarea –show all** command (as shown below). If you create a flow using the trunk index, then depending upon whether the trunk index maps to the master or the slave port the flow may or may not be installed. If the slave port becomes the master, it will be installed.

```
switch:admin> switchshow
(output truncated)
Index Port Address  Media Speed  State   Proto
==================================================
```

```
4     4    010400   id    N8      Online   FC  F-Port (Trunk master)
4     5    010400   id    N8      Online   FC  F-Port (Trunk port, master is Port 4)
(output truncated)

switch:admin> porttrunkarea --show all
Port  Type   State   Master   TI  DI
-------------------------------------
  0    --              --     --   0
  1    --              --     --   1
  2    --              --     --   2
  3    --              --     --   3
  4   F-port  Master    4      4   4
  5   F-port  Slave     4      4   5
```

- If a PID is changed for a WWN flow and the new PID does not match the PID in the flow definition, the active flow will be uninstalled and then reinstalled with the updated PID in the flow definition.

High Availability and Flow Vision

High Availability (HA) preserves only the Flow Vision configuration settings through an HA failover, HA reboot, or a power cycle and reboot. Such a failover does not save feature-related data (for example, statistics). User-defined or system-defined (predefined) flows are not restored if failed over to a version of Fabric OS that does not support these flows.

If a standby command processor (CP) with a downgrade revision code comes online and any flows (active or non-active) are configured, the HA will be out of sync. If a standby CP with a downgrade revision code comes online and no flows (active or non-active) are configured, HA will be in sync but flow creation will fail.

Refer to the following sections for information on how each Flow Vision feature is treated under HA:

- Flow Monitor and High Availability on page 98
- Flow Generator and High Availability on page 116
- Flow Mirror and High Availability on page 138

Flow Vision integration with MAPS

Statistics generated using Flow Vision can be monitored with the Monitoring and Alerting Policy Suite (MAPS) threshold service.

Refer to the individual features for information on how that feature interacts with MAPS, and the *Brocade Monitoring and Alerting Policy Suite Configuration Guide* for more details on MAPS in general.

Flow Monitor

Overview of Flow Monitor

To provide more effective fabric design and capacity planning, storage administrators require the ability to monitor a complete fabric. Flow Monitor allows you to monitor using predefined flows, static flows, or learning flows. It provides Fibre Channel-level statistics and SCSI IOPS performance counters. With Gen 6 hardware, a new set of counters lets you monitor flow latency at the SCSI I/O exchange level.

Flow Monitor enables you to monitor in real time all traffic passing through fabric E_Ports, EX_Ports, F_Ports, and XISL_Ports using any hardware-supported flow parameters. It also lets you define your own monitoring flows using combinations of ingress and egress ports, source and destination devices, logical unit numbers (LUNs), and frame types to create a flow definition for a specific use case.

Flow Monitor lets you monitor the following traffic using learning and static flows:

- All traffic passing through all E_Ports and F_Ports.
- Traffic flows through the fabric for virtual machine (VM) instances that originate from a single N_Port ID (PID) to destination targets.
- Flows monitoring edge-to-edge traffic, edge-to-backbone traffic, and backbone-to-edge traffic passing through EX_Ports.
- Flows monitoring traffic inside logical fabrics and inter-fabric (routed) traffic passing through XISL_Ports.
- Flows monitoring inter-fabric traffic and backbone traffic passing through backbone E_Ports.
- In Fabric OS 7.4.0 and later versions, Flow Monitor provides support for system-generated flows to automatically learn traffic through all active F_Ports, including trunked F_Ports. However, statistics are gathered only on the flow associated with the master port.

Flow Monitor provides you with the following abilities:

- Monitoring an application flow (for example, a flow within a fabric from a Host to a Target/LUN) at a given port.
- Obtaining comprehensive visibility into application flows in a fabric, including the ability to learn (discover) flows automatically.

- Capturing statistics on VM flows through the fabric from the host to target devices to help isolate performance issues and plan migration of VMs for load balancing.
- Gaining insights into application performance through the capture of statistics for specified flows. These statistics include transmitted and received frame counts, transmitted and received frame throughput rates, SCSI Read and SCSI Write frame counts, the number of SCSI Reads and Writes per second (IOPS), as well as others.

  A sample use case would be to monitor throughput statistics for inbound traffic between a source device and a destination device. Monitoring LUN-level statistics on page 97 provides an example of the command and the results for this use case.
- Monitoring various frame types at a switch port to provide deeper insights into storage I/O access patterns at a LUN, reservation conflicts, and I/O errors. Examples of the frame types that can be monitored include SCSI Aborts, SCSI Read, SCSI Write, SCSI Reserve, all rejected frames, and many others. Refer to Flow frametype parameters on page 24 for a list and description of the frame types that can be monitored.
- Obtaining SCSI Read/Write Frame Count and SCSI Read/Write Data statistics. Support is provided only for F_Ports for any flow configuration where either srcdev or dstdev exists on the switch, and the flow is defined using a combination of srcdev, dstdev, ingrport, or egrport (with or without bidir), or a combination of srcdev, dstdev, lun, ingrport, or egrport.
- Performing threshold-based monitoring and alerting based on flows through the integration with the Monitoring and Alerting Policy Suite (MAPS) service. Refer to the *Brocade Monitoring and Alerting Policy Suite Configuration Guide* for more information on integration with MAPS.

  NOTE
  In Fabric OS 7.1.x and earlier, the Advanced Performance Monitor (APM) provided the following monitors: End-to-End, Frame-based, ISL, and Top Talker. Although APM is not available in Fabric OS 7.4.0 and later versions, Flow Monitor provides equivalent functionality. Refer to Replicating APM monitors using Flow Monitor on page 48 for details.

Replicating APM monitors using Flow Monitor

Advanced Performance Monitor (APM) functionality was removed from Fabric OS 7.4.0 and later. However, it is possible to replicate standard APM functionality using Flow Monitor.

For information on replicating standard APM functionality using Flow Monitor, refer to the following links:

- Creating an end-to-end monitor equivalent on page 95
- Creating a frame monitor equivalent on page 95
- Creating an ingress or egress Top Talker monitor equivalent on page 96
- Creating an end-to-end monitor equivalent on page 95

Creating Flow Monitor flows

To create a Flow Monitor flow, enter the **flow --create** *flowname* **-feature monitor** *parameters* command using the parameters listed in the following table. The figure Frame and port parameters on page 22 illustrates how the frame and port parameters apply to a flow.

TABLE 10 Flow Monitor flow parameter combinations

Parameters	Field names	Description
Port	ingrport egrport	• One field only must be specified. • Values must be explicit. • Can be an F_Port, E_Port, or EX_Port on a local switch.
Frame	srcdev	• At least one field must be specified.

TABLE 10 Flow Monitor flow parameter combinations (continued)

Parameters	Field names	Description
	dstdev lun frametype	• Values for srcdev and dstdev can be explicit or "*" ("*" indicates learned flows). • Values for lun and frametype must be explicit.
VM flow	srceid dstdev ingrport	• At least one field must be specified. • Values for srceid must be explicit. Values for dstdev can be explicit or "*" ("*" indicates learned flows). • Values for ingrport must be explicit.
Optional keyword parameters		
	-bidir	Adding this keyword makes the application monitor traffic in both directions.
	-noactivate	Adding this keyword creates the flow without activating it.
	-noconfig	Adding this keyword creates the flow without saving the flow to the configuration.

Parameter usage exceptions

The following restrictions apply to parameter usage in Flow Monitor flow definitions:

- The **create** and **delete** options are not supported for the sys_mon_all_fports flow.
- The **-lun** and **-bidir** parameters cannot be used together in a flow definition.
- The egrport and srcdev parameters are not used for VM Insight flows. The srceid parameter is only used for VM Insight flows.
- Flow Monitor does not support learning flows using the **-frametype**, **-lun** or **-bidir** parameters.

The following example creates a Flow Monitor flow named “Flow1” that monitors all traffic flowing from device 010403 to device 020504 ingressing through port 10 on the switch on which this command was run.

```
switch:admin> flow --create Flow1 -feature monitor -ingrport 10 -srcdev 010403 -dstdev 020504
```

When you create a flow, it is automatically activated unless you use the **-noactivate** keyword as part of the **flow --create** command. Refer to Creating an inactive flow in Flow Monitor on page 49 for an example of this option.

Creating an inactive flow in Flow Monitor

The reason to create an inactive flow is to have it ready for future use. To create an inactive Flow Monitor flow, enter **flow --create** *flowname* **-feature** *feature_list flow_parameters* **-noactivate**.

The following example creates an inactive Flow Monitor flow named “sflow128” from device 020a00 to device 01c000 ingressing through port 10.

```
switch:admin> flow --create sflow128 -feature mirror -ingrport 10 -srcdev 0x020a00 -dstdev 0x01c000 -
noactivate
```

For information on activating an inactive Flow Monitor flow, refer to Activating Flow Monitor flows on page 50.

Activating Flow Monitor flows

To activate an inactive Flow Monitor flow, enter **flow --activate** *flowname* **-feature monitor**. Activating a flow automatically clears all the flow statistics for that flow.

The following example activates the Flow Monitor flow named "Flow1":

```
switch:admin> flow --activate Flow1 -feature monitor
```

Automatic activation of a Flow Monitor flow

Flow Monitor automatically activates monitoring flows under the following conditions:

- On flow creation, unless the flow is created using the **-noactivate** keyword.
- On slot power-on, if any of the ports or devices defined in the flow are on the slots being powered on. This assumes that the flow was active when the slots were powered off.
- On a High Availability (HA) failover, HA reboot, or a power cycle, if the flow was active when the event occurred.

Deactivating Flow Monitor flows

You can deactivate Flow Monitor flows without deleting them. This allows you to create and store a "library" of flows that you can activate when needed without having to recreate them every time they are needed.

To deactivate a Flow Monitor flow, enter **flow --deactivate** *flow_name* **-feature monitor**.

> NOTE
> You can deactivate a single feature even though the flow is defined for multiple features. For example, if a flow had been defined using "-feature monitor,generator", you can deactivate only the monitoring feature, while leaving the generator feature active.

The following example deactivates the Flow Monitor flow named "Flow1".

```
switch:admin> flow --deactivate Flow1 -feature monitor
```

Automatic deactivation of a Flow Monitor flow

Flow Vision automatically deactivates Flow Monitor flows and stops monitoring if any of these conditions occur:

- Slot is powered off for the ingress or egress ports defined in the flow.
- Slot is powered off for the source or destination devices defined in the flow.
- The ingress or egress port type changes to anything other than an E_Port, EX_Port, XISL_Port, F_Port, or SIM port for a learned flow (`"*"`). The flow will not automatically reactivate if the port type is changed back. You must manually reactivate such a flow.
- The ingress or egress port type changes to anything other than an E_Port, EX_Port, XISL_Port, F_Port, or SIM port for a flow that has the lun or frametype value specified. The flow will not automatically reactivate if the port type is changed back. You must manually reactivate such a flow.

Viewing Flow Monitor flows

To display Flow Monitor flows, enter **flow --show** *flowname* **-feature monitor**.

You can also repeat and sort the output of a Flow Monitor flow. For information on these tasks, refer to Repeating flow output on page 28 and Sorting flow output on page 31.

Viewing summary flow data for a specific device pair

The following example creates a Flow Monitor flow gathering statistics for frames ingressing through port 30 between device 011e00 and device 010100, and then displays the results. The point of interest can be an E_Port, EX_Port, or F_Port; in this example it is port 30, which is also the physical port number of the SrcDev.

```
switch:admin> flow --create sumflow1 -feature monitor
                    -ingrport 30 -srcdev 011e00 -dstdev 010100

switch:admin> flow --show
------------------------------------------------------------------------------------------
Flow Name      | Feature     | SrcDev          | DstDev          |IngrPt|EgrPt |BiDir| LUN  | FrameType|
------------------------------------------------------------------------------------------
 sumflow1      |mon+         |011e00           |010100           |30    |-     |no   |-     |-         |
 -----------------------------------------------------------------------------------------
  + Denotes feature is currently activated for the flow

switch:admin> flow --show sumflow1 -feature monitor
=============================================================================================
Name      : sumflow1 Features: mon(Active) noConfig: Off
Definition: IngrPort(30),SrcDev(0x011e00),DstDev(0x010100)
Flow Monitor (Activated):
Monitor time:   | Mon May 16 22:06:32 CLT 2016 |
---------------------------------------------------------
---------------------------------------------------------------------------------------------
| Rx Frames Count | Rx Frames per Sec. |  Rx Bytes Count | Rx Throughput(Bps)  | Avg Rx Frm Sz(Bytes)|
---------------------------------------------------------------------------------------------
|       4.83G     |        10.62M      |      617.07G    |         1.34G       |         140         |
---------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------
|      I/O Count          |   I/O Per Sec.(IOPS)    | I/O bytes Transferred   |   I/O bytes Per Sec.   |
| Reads / Writes/ Total   | Reads / Writes/ Total   | Reads / Writes/ Total   | Reads / Writes/ Total  |
---------------------------------------------------------------------------------------------
| 4.88G/     0 /   4.88G | 10.62M/     0 / 10.62M | 21.79T/     0 / 21.79T | 559.59M/    0 / 559.59M |
---------------------------------------------------------------------------------------------
=============================================================================================
```

Learning in Flow Monitor flows

You can apply learning in Flow Monitor flows in order to discover flow traffic. To apply learning to a Flow Monitor flow, you enter an asterisk inside quotation marks (`"*"`) as the parameter value for the parameter to be learned.

The following list outlines the support for learning on various port types:

- **F_Ports**: Learning and static monitoring is supported.
- **E_Ports**: Learning and static monitoring is supported.
- **EX_Ports**: Learning and static monitoring is supported.
- **XISL_Ports**: Fabric wide learning and static monitoring is supported.
- **Backbone E_Ports**: Learning and static monitoring is supported.
- **Trunked ports**: Learning is supported, although statistics will be gathered only on the flow associated with the master port. These statistics are the cumulative total of all the ports on the trunk.

- **VM instances**: Learning and static monitoring is supported.

Notes on learning in Flow Monitor flows

The following items apply to learning in Flow Monitor flows:

- Only Gen 6 (32 Gbps-capable) and Gen 5 (16 Gbps-capable) Fibre Channel platforms and 8 Gbps Enhanced blades have the capability to learn flows on a specified port.
- Only one active flow per ASIC can be a learning flow.
- Using two flows to monitor traffic ingressing and egressing on the same ASIC is only supported if both flows are static. If one flow is a learning flow, only the static flow will count frames.
- You can use either a WWN or a PID for the srcdev and dstdev values when creating learning flows for Flow Monitor.
- Hardware resources for a learned flow are limited. Consequently, the number of actual sub-flows for a flow definition might exceed the number of sub-flows permitted by limitations. This might cause a situation where a one-time traffic bottleneck occupies a hardware resource and an important sub-flow does not get monitored. To avoid first-come-first-serve scenarios and give priority to recent sub-flows, Flow Monitor refreshes the sub-flows when the following conditions are met:
  - The limit for sub-flows is reached.
  - The number of frames not counted by the hardware is more than or equal to the total of counted frames.

  When the sub-flows are refreshed, the previous counters are cleared and learning is started for fresh sub-flows.

Creating Flow Monitor learning flows

Using learning flows in Flow Monitor allows you to discover flow traffic without having to identify specific devices or ports.

Refer to Monitoring flows using the learning functionality on page 54 to view how to display the data captured using a learning flow.

To create a leaning flow in Flow Monitor, complete the following steps.

1. Connect to the switch and log in using an account with admin permissions.
2. Enter **flow --create** *flow_name* **-feature monitor** *port_values* , using an asterisk in quotes (`"*"`) for those port and device values you want to be learned.

The following example creates a Flow Monitor learning flow named “ingressTT” ingressing through port 30 (an E_Port).

```
switch:admin> flow --create ingressTT -feature monitor -ingrport 30 -srcdev "*" -dstdev "*"
```

The following example creates a Flow Monitor learning flow named “ex_lrn_ingr” which captures traffic ingressing through port 219 headed for device 20:13:00:05:33:88:a3:90.

```
switch:admin> flow --create ex_lrn_ingr  -feature monitor -ingrport 219 -srcdev "*" -dstdev
20:13:00:05:33:88:a3:90
```

Learning flow creation on offline or slave ports

Flows can be created that specify either an inactive or slave port in their definitions, but activation is controlled not only by the port status, but by available resources and flow count limits.

- When a flow is created on an offline port, the flow is considered inactive until the port comes online. However, the flow would only be installed in the ASIC and made active if the resources are available. If the resources are not available the flow would remain deactivated.

- When a flow is created on a slave port, the flow is considered inactive until the port becomes the master port. However, the flow would only be installed in the ASIC and made active if the resources are available. If the resources are not available, the flow would remain deactivated.

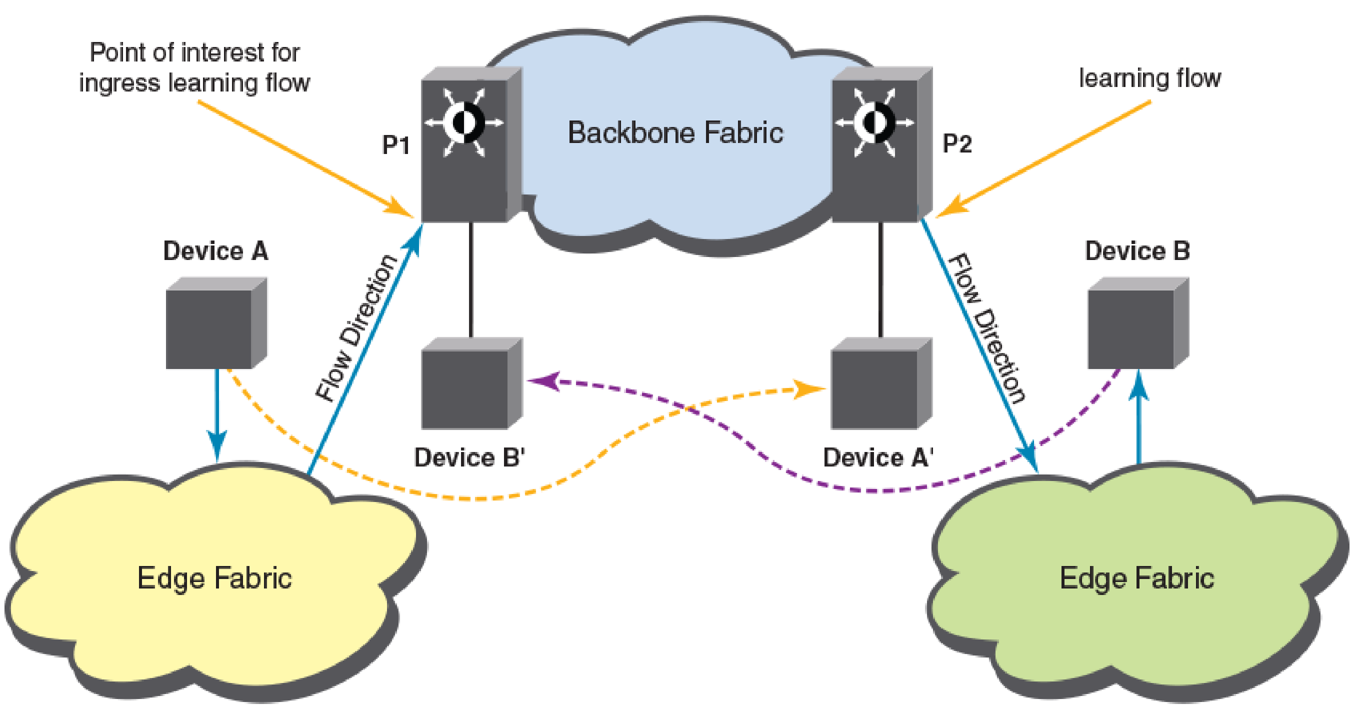
Flow Monitor learning on E_Ports and EX_Ports

Flow Monitor supports learning and monitoring all the traffic passing through both E_Ports and EX_Ports. The captured statistics are similar to the existing learning support provided for F_Ports.

As it is for F_Ports, learning is supported on trunked E_Ports and EX_Ports, and statistics are gathered only on the flow associated with the master port. These statistics are the cumulative total from all of the trunked ports. For this reason, you should create and use the same flow definition for each of the ports in the trunk. You can use either a WWN or a PID value for the srcdev and dstdev identifiers when creating learning flows.

While the source fabric ID to destination fabric ID option is allowed on any E_Port, it is needed only for Backbone E_Ports and the XISL_Ports. If the source fabric ID to destination fabric ID combination is configured on an edge E_Port learning flow, it will display the same source fabric ID to destination fabric ID combination for all learned flows.

FIGURE 3 Learning support in a Fibre Channel Routing fabric



The fixed parameters of a learning flow on an E_Port or an EX_Port should use the following combinations when defining flows using specific ingress or egress ports:

- For flows using an ingress port (ingrport), the real source ID of the source device and the proxy ID of the destination device.
- For flows using an egress port (egrport), the proxy ID of the source device and the real destination ID of the destination device.

The **flow --show** output for an E_Port or an EX_Port will use the flow definition; in this output the proxy ID of the destination device will be reported for flows defined using the ingress port, and the proxy ID of the source device will be reported for flows defined using the egress port.

Monitoring flows using the learning functionality on page 54 provides examples of both the flow command and the flow output.

Monitoring flows using the learning functionality

The following example illustrates using the learning functionality for flow monitoring on an E_Port. The defined flow will monitor for frames ingressing on port 30 between all devices. The example then shows the flow output.

```
switch:admin> flow --create ingressTT -feature monitor
                    -ingrport 30 -srcdev "*" -dstdev "*"

switch:admin> flow --show ingressTT
-----------------------------------------------------------------------------------------------
Flow Name     |Feature     |SrcDev        |DstDev        |IngrPt|EgrPt |BiDir| LUN   | FrameType|
-----------------------------------------------------------------------------------------------
ingressTT     |mon+        |*             |*             |30    |-     |no   |-      |-         |
-----------------------------------------------------------------------------------------------
   + Denotes feature is currently activated for the flow

switch:admin> flow --show ingressTT -feature monitor
===============================================================================================
Name      : ingresstt Features: mon(Active) noConfig: Off
Definition: IngrPort(30),SrcDev(*),DstDev(*)
Flow Monitor (Activated):
Monitor time:  | Tue Apr 12 06:12:12 UTC 2016 |
---------------------------------------------------------
-------------------------------------------------------------------------------------
|SID(*)|DID(*)|Rx Frames Cnt|Rx Frames    |Rx Bytes Cnt |Rx Through-   |Avg Rx Frm   |
|      |      |             |  per Sec.   |             | put(Bps)     | Sz (Bytes)  |
-------------------------------------------------------------------------------------
|010000|010100|   9.92G     | 201.22k     |  18.87T     |  399.93M     |   2092      |
|010000|010200|   9.92G     | 201.23k     |  18.87T     |  399.93M     |   2092      |
|010000|010300|   9.92G     | 201.23k     |  18.87T     |  399.93M     |   2092      |
|010000|010400|   9.92G     | 201.22k     |  18.87T     |  399.93M     |   2092      |
-------------------------------------------------------------------------------------
|  *   |  *   | 39.69G      | 804.91k     |  75.48T     |    1.56G     |   2092      |
-------------------------------------------------------------------------------------
=====================================================================================
```

The following example illustrates using the learning functionality for flow monitoring on an EX_Port. The defined flow will monitor for frames ingressing on port 219 headed for device 20:02:00:11:0d:51:00:00. The example then shows the flow output.

```
DCX_edge2:admin> flow --create ex_lrn_ingr -feature monitor -ingrport 219 -srcdev "*" -dstdev 20:02:00:11:0d:
51:00:00

DCX_edge2:admin> flow --show ex_lrn_ingr
==================================================================================================
Name      : ex_lrn_ingr   Features: mon(Active)
Definition: IngrPort(219),SrcDev(*),DstDev(*)


Flow Stats (Active):
Stats time:   | Tue Apr 12 14:59:58 UTC 2016 |
--------------------------------------------------------------------------------------------------
|  SID | DID    |Rx Frames Cnt|Rx Frames per Sec.|Rx Bytes Cnt|Rx Throughput(Bps)|Avg Rx Frm Sz(Bytes)|
--------------------------------------------------------------------------------------------------
|220200| 01f001 |  2.85G      |   8.45M          |   387.88G  |     1.12G        |    132             |
--------------------------------------------------------------------------------------------------
|220200| 02f001 |  2.85G      |   6.00M          |   350.00G  |     1.00G        |    232             |
--------------------------------------------------------------------------------------------------
|   *  |    *   |  5.60G      |  14.45M          |   737.88G  |     2.12G        |    152             |
==================================================================================================
```

Learning for all F_Ports using the sys_mon_all_fports flow

Flow Monitor supports learning flows on all F_Ports in a fabric through the predefined sys_mon_all_fports flow.

The sys_mon_all_fports flow supports F_Port and provides the following:

- Continuous and non-intrusive monitoring of every F_Port in the edge switches of the fabric.
- Provisioning and capacity planning with ranking on top and bottom bandwidth-consuming flows.
- Measuring device performance on all F_Ports in the switch.

Working with the sys_mon_all_fports flow

You can activate, deactivate, and display the output of the sys_mon_all_fports flow, as well as resetting the flow logs.

The following table provides the commands for activating, deactivating, resetting the flow logs, and displaying the output of the sys_mon_all_fports flow.

TABLE 11 Valid commands for the sys_mon_all_fports flow

Action	Command
Activate flow	**flow -activate sys_mon_all_fports -feature monitor**
Deactivate flow	**flow -deactivate sys_mon_all_fports -feature monitor**
Reset flow logs	**flow -reset sys_mon_all_fports -feature monitor**
Show flow logs	**flow -show sys_mon_all_fports -feature monitor**

Because sys_mon_all_fports is a predefined flow, the **create** and **delete** options for the **flow** command are not supported for it.

The **flow --show** output for all F-port learning includes data about all the F-ports that are first discovered for monitoring statistics.

The output for all F_Port learning flows includes the following data about all the F_Ports that are discovered.

- Frame Statistics: Frame count and rate for the flow-defined frame type
- Throughput Statistics: Word count and throughput (bytes per second)
- I/O Statistics: I/O count, I/O per second, and I/O data transferred on a read/write basis
- Learn Statistics: All learned (`"*"`) flows on a given port and the throughput and frame statistics for each learned flow on 16 GBps-capable and 32 GBps-capable Fibre Channel platforms

Limitations and restrictions for the sys_mon_all_fports flow

Learning for all F_Ports—including trunked F_Ports—using the sys_mon_all_fports flow is subject to the following restrictions:

- The sys_mon_all_fports flow is inactive by default; the flow must be explicitly activated.
- The sys_mon_all_fports flows are supported only on Gen 5 and Gen 6 platforms. Refer to Supported hardware and software on page 12 for a list of Gen 5 and Gen 6 platforms.
- User-defined Flow Monitor static flows cannot be active when sys_mon_all_fports is active.
- User-defined Flow Monitor learning flows cannot be active on any partition in the chassis when sys_mon_all_fports is active.
- User-defined Flow Monitor static flows and learning flows are not supported for AE ports in the Fabric OS switch platforms.
- Learning flows cannot be active when sys_mon_all_fports is active.
- The sys_mon_all_fports flow can be active on only one logical switch in the entire chassis at a time.
- The sys_mon_all_fports flow monitors only traffic ingressing through F_Ports. However, currently, a sys_mon_all_fports flow is unable to collect statistics for the F_Port trunk.

- The sys_mon_all_fports flow inherits the current configured port ID mode, device ID mode, and other control parameters active for the defined application.
- Activation, deactivation, reset, show, and control operations are supported for the sys_mon_all_fports flow. Other flow commands (including flow creation and deletion) are not supported for this flow.
- When downgrading to a version of Fabric OS earlier than 7.4.1, the sys_mon_all_fports flows are not replayed.

Limitations when monitoring SIM traffic

Monitoring of SIM traffic on a single Brocade G610 switch or Brocade G620 switch is not supported in Fabric OS 8.1.0. These switches do not update LNK statistics for discarded frames. There are some configurations that support SIM port monitoring. However, using Flow Monitor in combination with Flow Generator is limited.

The following table shows the supported use of Flow Monitor and Flow Generation in combination:

TABLE 12 SIM traffic monitoring using Flow Monitor with Flow Generator

Switch	Ingress monitoring with source and destination SIM ports on:		Egress monitoring
	Same chip	Different chip	
Brocade X6 Director	Yes	Yes	No
Brocade G610 Switch	No	Yes (from to Brocade G610 to Brocade G610 only)	No
Brocade G620 Switch	No	Yes (from Brocade G620 to Brocade G620 only)	No

Examples of flow definitions for SIM traffic monitoring

SIM traffic monitoring is supported for certain configurations. The following examples illustrate flow definitions that are supported and are not supported for SIM traffic monitoring.

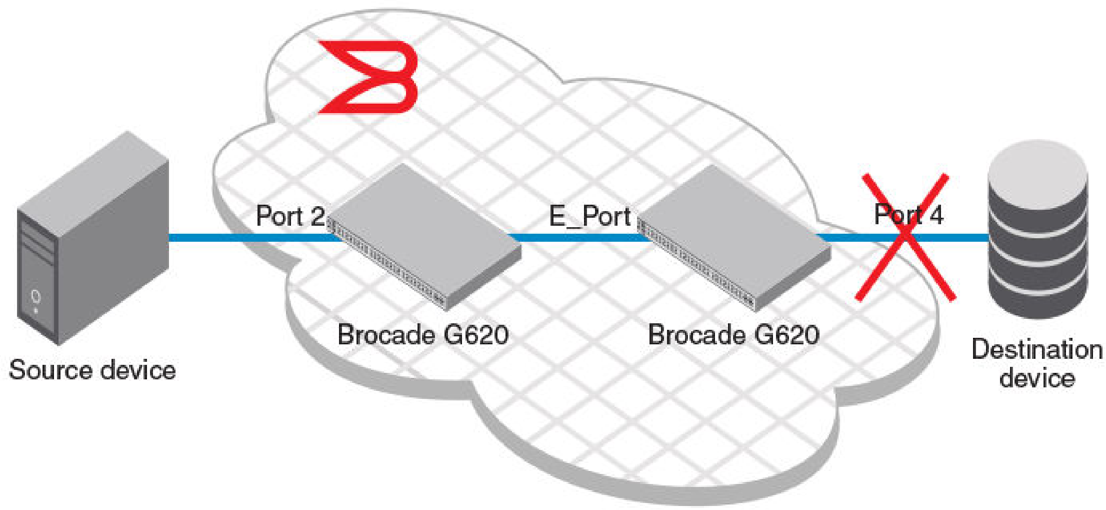
SIM traffic monitoring on a single Brocade G620 switch

On a single Brocade G620 switch, ingress flow monitoring and egress flow monitoring of SIM traffic are not supported.

SIM traffic monitoring for connected Brocade G620 switches

When two Brocade G620 switches are connected, ingress flow monitoring of SIM traffic is supported on Port 2 of the switch connected to the source device; however, egress flow monitoring of SIM traffic is *not* supported on Port 4 of the switch connected to the destination device.

FIGURE 4 Supported configuration with two connected Brocade G620 switches

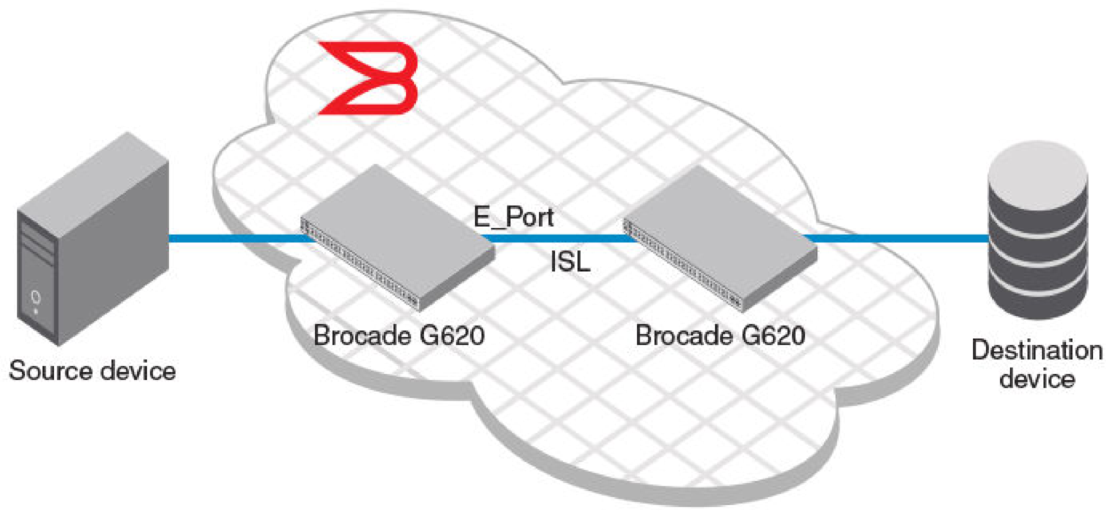


SIM traffic monitoring on Brocade G620 switches connected by an ISL

When an inter-switch link (ISL) connects two Brocade G620 switches, the following applies:

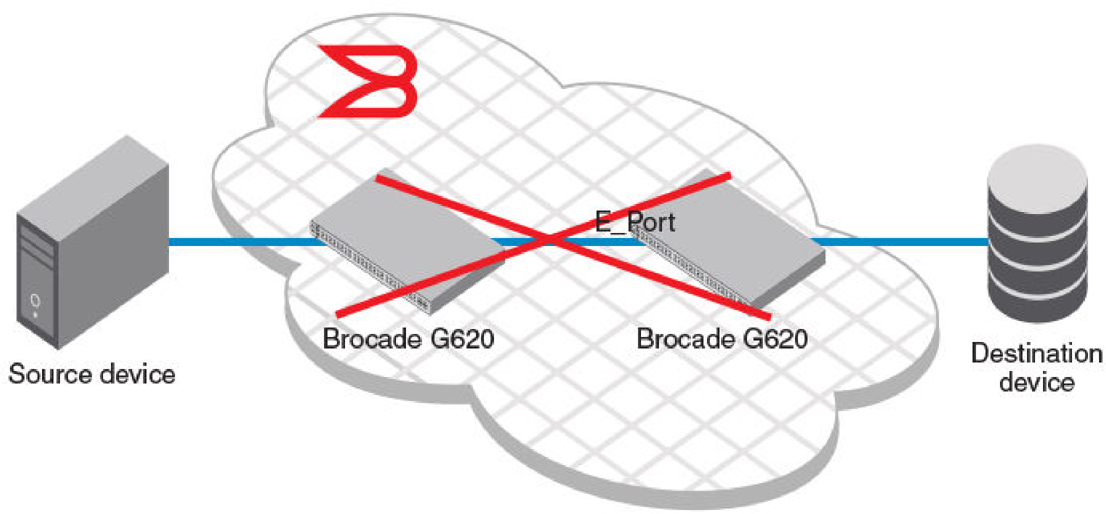
- **Supported configuration:** When traffic is going out of the E_Port, the LNK statistics update in the E_Port flow.

FIGURE 5 Supported configuration for ISL connection



- **Unsupported configuration:** When traffic is going into the E_Port, frames are dropped by the target Brocade G620 and LNK statistics are not updated. Statistics about the E_Port do not appear.

FIGURE 6 Unsupported configuration with ISL connection



SIM traffic monitoring for the Brocade X6 Director

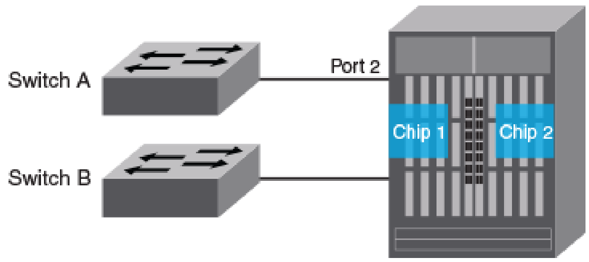
For the Brocade X6 Director:

- Ingress flow monitoring of SIM traffic on the same chip or between two different chips is supported.
- Egress flow monitoring of SIM traffic is not supported.

The following examples illustrate configurations that are supported or not supported for monitoring SIM traffic for the Brocade X6 Director:

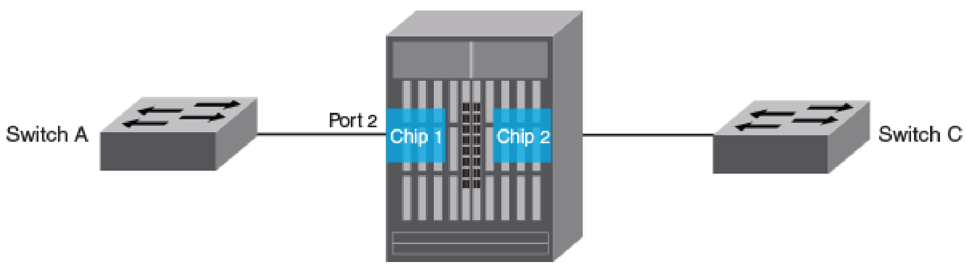
- **Supported configuration:** Ingress flow monitoring of SIM traffic is supported on the same chip when using Port 2 as the ingress port from the source device (A).

FIGURE 7 Supported configuration using same chip on the Brocade X6 Director



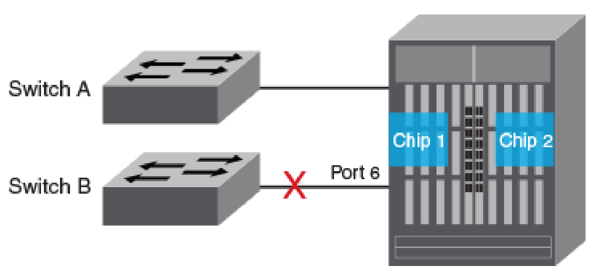
- **Supported configuration:** Ingress flow monitoring of SIM traffic is supported on different chips when using Port 2 as the ingress port on the chip connected to the source device (A) .

FIGURE 8 Supported configuration using different chips on the Brocade X6 Director



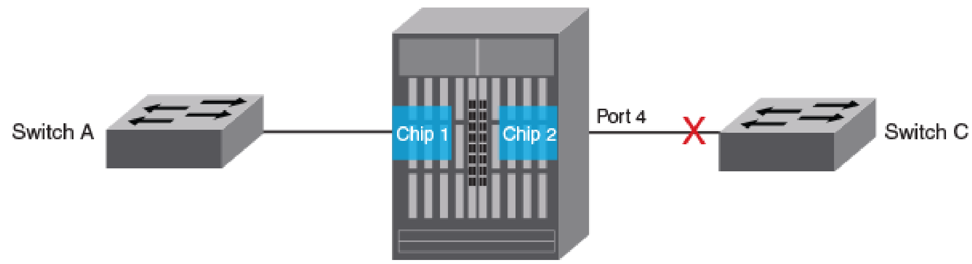
- **Unsupported configuration:** Egress flow monitoring of SIM traffic is *not* supported on the same chip when using Port 6 as the egress port to the destination device (B).

FIGURE 9 Unsupported configuration using the same chip on the Brocade X6 Director



- **Unsupported configuration:** Egress flow monitoring of SIM traffic is *not* supported on different chips when using Port 4 as the egress port to the destination device (C).

FIGURE 10 Unsupported configuration using different chips on the Brocade X6 Director



Duplicate flows and the sys_mon_all_fports flow

The sys_mon_all_fports flow has the following features:

- The flow can actively coexist with a duplicate predefined flow as long as each is for a different application (Flow Generator or Flow Mirror).
- The flow cannot actively coexist with a duplicate predefined flow for a Flow Monitor application active in any other logical switch in the chassis.
- The flow cannot be activated if user-defined monitor flows are active.

  When the sys_mon_all_fports flow is active, user-defined static and learning flows are considered as duplicate flows.

System event handling for the sys_mon_all_fports flow

The following table displays the sys_mon_all_fports flow behaviors for various system events.

TABLE 13 sys_mon_all_fports flow system event handling

System event	sys_mon_all_fports behavior
F_Port offline	Stops monitoring the flows related to that particular F_Port. All existing sub-flows related to that particular F_Port are deleted.
F_Port online	Starts monitoring the flows related to that particular F_Port. New sub-flows related to that particular F_Port are displayed in output.
Switch offline	Stops monitoring the flows on the F_Ports on that switch. All existing sub-flows related to that particular switch are deleted.
Switch online	Starts monitoring the flows on the F_Ports on that switch. New sub-flows related to that particular switch are displayed in output.
Slot offline	Stops monitoring the flows on the F_Ports on that particular slot. All existing sub-flows related to that particular slot are deleted.
Slot online	Starts monitoring the flows on the F_Ports on that particular slot. New sub-flows on the F_Ports related to that particular slot are displayed in output.

The sys_mon_all_fports flow does not respond to any offline event unless the port type is changed. No subflows will be deleted while the F_Port is offline.

Supported combinations for F_Port learning

The following table provides the valid supported combinations for F_Port learning.

TABLE 14 Supported combinations for F_Port learning

Srcdev	Dstdev	Port	Port	Port Type	Description
*	*	F	Ingr / Egr	F/E/Ex	Learns all active source device and destination device ID pairs, and collects statistics for each of those flows.
*	WWN/PID	F	Ingr / Egr	F/E/Ex	Learns all active flows from any source devices to the specified destination device, and collects statistics for each of those flows.
WWN/PID	*	F	Ingr / Egr	F/E/Ex	Learns all active flows to the destination device from the specified source device, and collects statistics for each of those flows.
*	Not specified	F	Ingr / Egr	F/E/Ex	Learns all source devices that are active and collects statistics for each of those flows. Cumulative data for all the destination devices will be displayed.
Not specified	*	F	Ingr / Egr	F/E/Ex	Learns all destination devices that are active and collects statistics for each of those flows. Cumulative data for all the destination devices will be displayed.
*	*	*	Ingr	F	Learns all the F_Ports on the logical switch and collects statistics for all the active source device and destination device pairs on all the learned ports.

Configuring Flow Monitor for a trunk group

Flow Monitor supports monitoring trunk ports subject to the following conditions:

- You must create the same flow on all trunk member ports.
- If you create a flow on a slave port without using the **-noactivate** keyword with the **flow** command, this flow is then automatically activated when the slave port becomes the master port.
- After a switch initialization or a recovery (cold or warm), existing flows are re-created on both master and slave ports, but only those flows associated with the master port are activated.

To configure Flow Monitor on a trunk group, use the following steps.

1. Identify your trunk group members using the **switchshow** command.
2. Create individual flow monitors for each member of the trunk group using the **flow -create** command.
3. Enter **flow --show** *flow_name* **-feature monitor** to view the Flow Monitor statistical data for the entire trunk group.

   NOTE
   The accumulated Flow Monitor statistical data for the entire trunk group is stored on the master port. If the master port changes, the data is transferred to the new master port. To view this data, you must run the **flow --show** command on a flow that is defined using the master port. Flow statistics are not displayed for slave trunk ports.

Example of creating Flow Monitor flows for a trunk group

The following example displays the trunked ports and then creates four flows, one for each member of the trunk group identified by the **switchshow** command.

```
switch:admin> switchshow

24  24  021800  id  N16  Online  FC E-Port 10:00:00:05:33:e5:3c:d4 "Odin" (downstream)(Trunk master)
25 25 021900 id  N16  Online FC E-Port (Trunk port, master is Port 24)
26 26 021a00 id  N16  Online FC E-Port (Trunk port, master is Port 24)
27 27 021b00 id  N16  Online FC E-Port (Trunk port, master is Port 24)

switch:admin> flow -create f1 -feature monitor -egrport 24 -srcdev 022b00 -dstdev 033a00
switch:admin> flow -create f2 -feature monitor -egrport 25 -srcdev 022b00 -dstdev 033a00
switch:admin> flow -create f3 -feature monitor -egrport 26 -srcdev 022b00 -dstdev 033a00
switch:admin> flow -create f4 -feature monitor -egrport 27 -srcdev 022b00 -dstdev 033a00
```

Example of creating a configured learned Flow Monitor flow in an F_Port

The following example shows the commands to configure a learned flow in an F_Port trunk and to display the created flow.

```
switch:admin> flow --create mirflow -fea mir,mon -srcdev "*" -dstdev "*" -ingrport 10/38
Monitor,Mirror feature(s) have been activated.

switch:admin> flow --show mirflow
=====================================================================================
Name      : mirflow    Features: mon(Activated),mir(Activated)    noConfig: Off
Definition: IngrPort(10/38),SrcDev(*),DstDev(*)


Flow Monitor (Activated):
Monitor time:  | Fri Apr 15 17:01:16 PDT 2016 |
---------------------------------------------------------
---------------------------------------------------------------------------------------
|SID(*)|DID(*)| Rx Frames | Rx Frames | Rx Bytes   | Rx Through- | Avg Rx Frm Sz(Bytes)|
|      |      | Count     | per Sec.  | Count      | put (Bps)   |                     |
---------------------------------------------------------------------------------------
|07de81|07d202|  1.28M    |   80.48k  |  2.52G     |   162.47M   |        2116         |
|07de81|07d201|  1.28M    |   80.03k  |  2.52G     |   161.56M   |        2116         |
|07de81|07d203|  1.28M    |   80.42k  |  2.52G     |   162.34M   |        2116         |
|07de81|07d200|  1.28M    |   81.30k  |  2.53G     |   164.12M   |        2116         |
|07de81|07d204|  1.02M    |   65.03k  |  2.02G     |   131.29M   |        2116         |
---------------------------------------------------------------------------------------
|     *|     *|  6.15M    |  387.28k  | 12.13G     |   781.80M   |        2116         |
---------------------------------------------------------------------------------------
```

Monitoring Fibre Channel routed fabrics

When you are monitoring Fibre Channel-routed fabrics, you should keep the following points in mind:

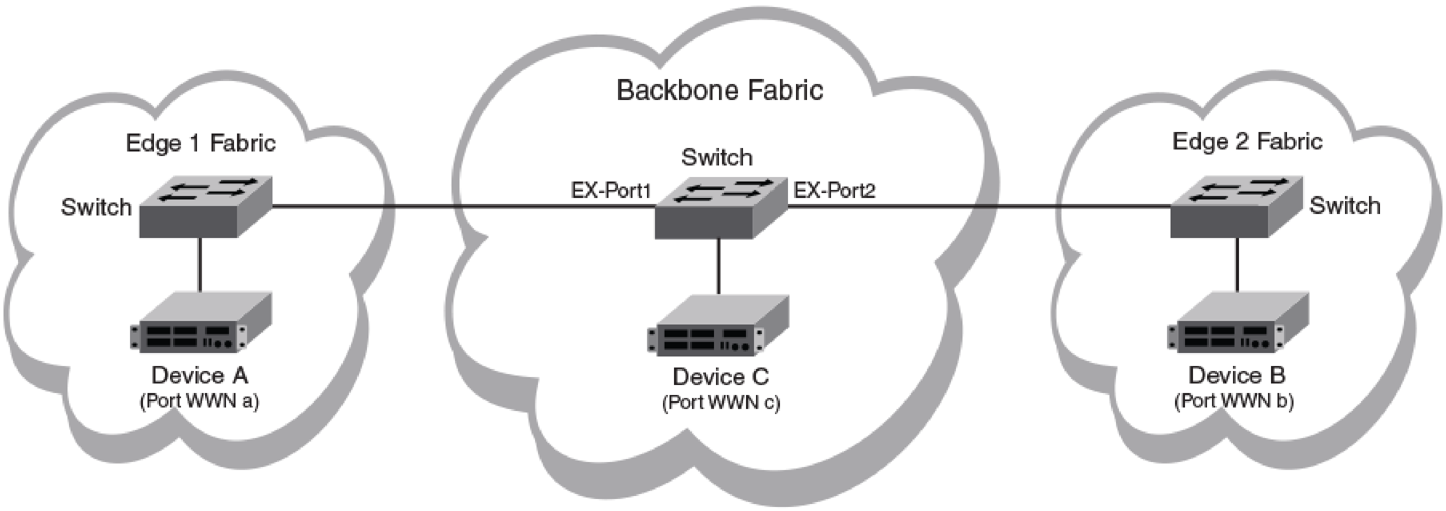
- When monitoring a FC-routed fabric, you may find it simpler to use port WWNs rather than proxy IDs in your flow definitions. This is because you do not need to locate and map the proxy IDs for the actual source and destination devices.
- When creating flow monitors on EX_Ports, you can use either a WWN or a Fibre Channel ID (FCID) for the source device (srcdev) and destination device (dstdev).
- Inter-Fabric Link (IFL) flows can be monitored only on 16 Gbps-capable EX_Ports in a Fibre Channel router.
- IFL flows are not supported on E_Ports or F_Ports.
- Even though a flow definition is always created in the backbone fabric, the perspective of the flow is from the edge fabric. That is, looking from the edge towards the backbone.

Monitoring FC router fabrics using port WWNs

The following figures and their descriptions illustrate how port IDs are assigned in Fibre Channel router fabrics using World Wide Names (WWNs). Use the **flow --control -deviceidmode wwn** command to set the mode to WWN.

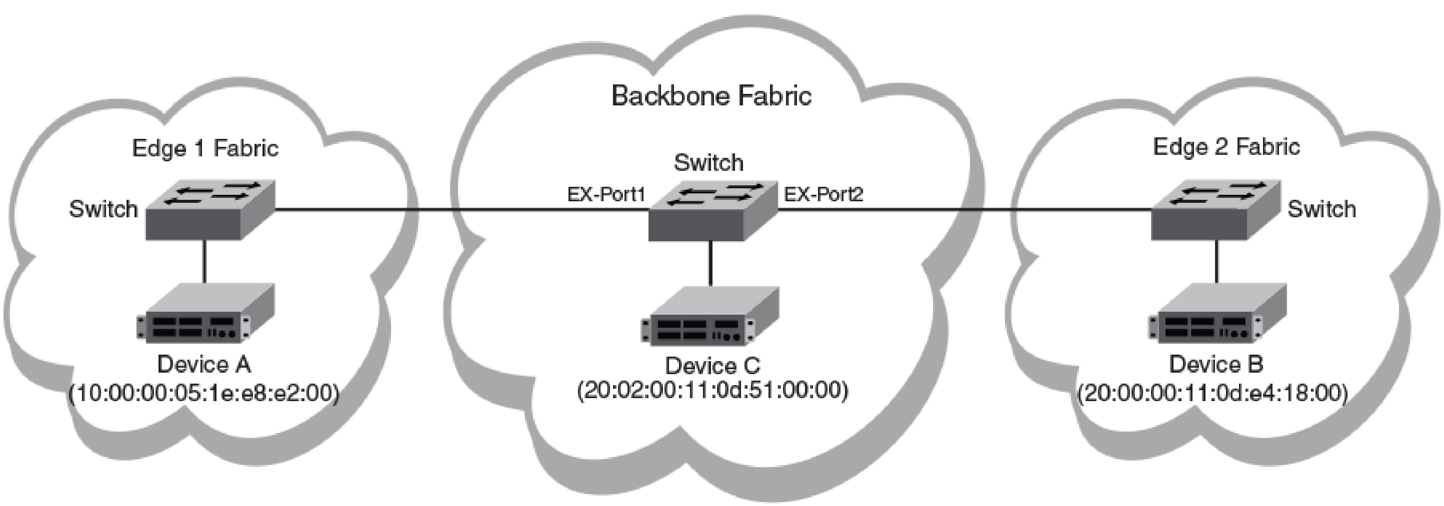
The following figure identifies the physical devices as A, B, and C, and shows that they have the port WWNs a, b, and c, respectively.

FIGURE 11 A Fibre Channel router fabric



The following figure provides the port WWN values for the physical devices and port WWNs.

FIGURE 12 A Fibre Channel router fabric annotated with port WWN values



Monitoring an edge-to-edge flow through an ingress port identified by a WWN

Flow Vision allows you to monitor an edge-to-edge flow passing through an ingress port identified by a World Wide Name (WWN) ID.

In a network set up as shown in Figure 12 on page 63, for a flow passing from Device A to Device B that is ingressing through EX_Port1, the source device (srcdev) is port "WWN a", the destination device (dstdev) is port "WWN b", and the ingress port (ingrport) is EX_Port1. (Traffic in the figure is running from right to left, and the flow definitions are based on the Edge 1 Fabric's perspective.)

The following example creates a flow that filters frames passing from one edge fabric to another edge fabric using a specific ingress port on the backbone. The first command shows the available ports and the available Fibre Channel routers. The second command creates a Flow Monitor flow named "e2e_src_dcx_wwn" between device 10:00:00:05:1e:e8:e2:00 and device 20:00:00:11:0d:e4:18:00 ingressing through port 219, and the last command displays the results of the flow.

NOTE
The slash character (\) in the example indicates a break inserted because the output is too long to display here as a single line.

```
DCX_Backbone128:admin> switchshow |grep Port

Index Slot  Port Address Media  Speed  State  Proto
  37    3    5   012500   id    N16    Online FC EX-Port 10:00:00:05:33:ef:f1:1c \
  47    3   15   012f00   id    N8     Online FC F-Port  20:02:00:11:0d:51:00:00 \
 219   10   27   01db00   id    N16    Online FC EX-Port 10:00:00:05:33:ee:d0:a5 \
                                                E-Port  50:00:51:e4:91:9e:0f:28 \
\ "Wasp_e2" (fabric id=50)(Trunk master)
\
\ "Gnat_e1" (fabric id=100)(Trunk master)
\ "fcr_xd_2_100"

DCX_Backbone128:admin> flow --create e2e_src_dcx_wwn -feature monitor -ingrport 219
-srcdev 10:00:00:05:1e:e8:e2:00 -dstdev 20:00:00:11:0d:e4:18:00

DCX_Backbone128:admin>flow --show
------------------------------------------------------------------------------------------ \
      Flow Name      |     Feature     |         SrcDev          |          DstDev        | \
------------------------------------------------------------------------------------------ \
e2e_src_dcx_wwn      |mon+            |10:00:00:05:1e:e8:e2:00  |20:00:00:11:0d:e4:18:00 | \
------------------------------------------------------------------------------------------ \

\ -------------------------------------
\ IngrPt|EgrPt |BiDir| LUN  |FrameType|
\ -------------------------------------
\ 219   |-     |no   |-     |-        |
\ -------------------------------------
\   + Denotes feature is currently activated for the flow

DCX_Backbone128:admin> flow --show e2e_src_dcx_wwn -feature monitor
==================================================================
Name      : e2e_src_dcx_wwn   Features: mon(Active) noConfig: Off
Definition: IngrPort(219),SrcDev(10:00:00:05:1e:e8:e2:00),
DstDev(20:00:00:11:0d:e4:18:00)

Flow Monitor (Activated):
Monitor time:  | Fri Apr 15 14:59:58 UTC 2016 |
------------------------------------------------------------------

------------------------------------------------------------------------------------------------------
| Rx Frames Count | Rx Frames per Sec. |  Rx Bytes Count | Rx Throughput(Bps) | Avg Rx Frm Sz(Bytes)|
------------------------------------------------------------------------------------------------------
|      2.85G      |        8.44M       |     387.88G     |        1.12G       |         132         |
------------------------------------------------------------------------------------------------------
```

Monitoring an edge-to-edge flow through an egress port identified by a WWN

Flow Vision allows you to monitor an edge-to-edge flow passing through an egress port identified by a World Wide Name (WWN) ID.

In a network set up as shown in Figure 12 on page 63, for a flow passing from Device B to Device A that is egressing through EX_Port1, the source device (srcdev) is port "WWN b", the destination device (dstdev) is port "WWN a", and the egress port (egrport) is EX_Port1. (Traffic in the figure is running from left to right, and the flow definitions are based on the Edge 1 Fabric's perspective.)

The following example creates a flow that filters out frames passing from one edge fabric to another edge fabric using a specific egress port on the backbone. The first command shows the available ports and the available Fibre Channel routers. The second command creates a Flow Monitor flow named "e2e_dst_dcx" between device 20:00:00:11:0d:e4:18:00 and device 10:00:00:05:1e:e8:e2:00 egressing through port 219, and the last command displays the results of the flow.

NOTE
The slash character (\) in the example indicates a break inserted because the output is too long to display here as a single line.

```
DCX_Backbone128:admin> switchshow |grep Port
Index Slot Port Address Media  Speed      State      Proto
37    3     5    012500   id     N16       Online     FC EX-Port 10:00:00:05:33:ef:f1:1c \
47    3    15    012f00   id     N8        Online     FC F-Port  20:02:00:11:0d:51:00:00 \
219   10   27    01db00   id     N16       Online     FC EX-Port 10:00:00:05:33:ee:d0:a5 \
                                                         E-Port  50:00:51:e4:91:9e:0f:28 \

\ "Wasp_e2" (fabric id=50)(Trunk master)
\
\ "Gnat_e1" (fabric id=100)(Trunk master)
\ "fcr_xd_2_100"

DCX_Backbone128:admin> flow --create e2e_dst_dcx -feature monitor -egrport 219
-srcdev 20:00:00:11:0d:e4:18:00 -dstdev 10:00:00:05:1e:e8:e2:00

DCX_Backbone128:admin> flow --show
----------------------------------------------------------------------------- \
   Flow Name  |  Feature  |       SrcDev            |        DstDev          | \
----------------------------------------------------------------------------- \
e2e_dst_dcx   |mon+       |20:00:00:11:0d:e4:18:00 |10:00:00:05:1e:e8:e2:00 | \
----------------------------------------------------------------------------- \
\ ---------------------------------------
\ |IngrPt|EgrPt |BiDir|LUN    |FrameType|
\ ---------------------------------------
\ |-     |219   |no   |-      |-        |
\ ---------------------------------------
\   + Denotes feature is currently activated for the flow

DCX_Backbone128:admin> flow --show e2e_dst_dcx -feature monitor
===========================================================================
Name      : e2e_dst_dcx   Features: mon(Active) noConfig: Off
Definition: EgrPort(219),SrcDev(20:00:00:11:0d:e4:18:00),
DstDev(10:00:00:05:1e:e8:e2:00)

Flow Monitor (Activated):
Monitor time:   | Fri Apr 15 14:59:58 UTC 2016 |
---------------------------------------------------------
-----------------------------------------------------------------------------------------------
| Tx Frames Count| Tx Frames per Sec.| Tx Bytes Count| Tx Throughput(Bps)| Avg Tx Frm Sz(Bytes)|
-----------------------------------------------------------------------------------------------
|       2.85G    |         8.44M     |    387.88G    |         1.12G     |          132        |
-----------------------------------------------------------------------------------------------
```

Monitoring a backbone-to-edge flow identified by WWNs

Flow Vision allows you to monitor backbone-to-edge flows passing through ports identified by World Wide Name (WWN) IDs.

In a network set up as shown in Figure 12 on page 63, for a flow passing from Device C to Device A that is egressing through EX_Port1, the source device (srcdev) is port "WWN c", the destination device (dstdev) is port "WWN a", and the egress port (egrport) is EX_Port1. (Traffic in the figure is running from right to left, and the flow definitions are based on the Edge 1 Fabric's perspective.)

The following example creates a flow that filters out frames passing from the backbone fabric to an edge fabric using a specific egress port. The first command shows the available ports and the available Fibre Channel routers. The second command creates a Flow Monitor flow named "b2e_dst_dcx" between device 20:02:00:11:0d:51:00:00 and device 10:00:00:05:1e:e8:e2:00 egressing through port 219, and the last command displays the results of the flow.

NOTE

The slash character (\) in the example indicates a break inserted because the output is too long to display here as a single line.

```
DCX_Backbone128:admin> switchshow |grep Port
Index Slot Port Address Media  Speed      State      Proto
37    3    5    012500   id     N16       Online     FC EX-Port 10:00:00:05:33:ef:f1:1c \
47    3   15    012f00   id     N8        Online     FC F-Port  20:02:00:11:0d:51:00:00 \
219   10  27    01db00   id     N16       Online     FC EX-Port 10:00:00:05:33:ee:d0:a5 \
                                                        E-Port  50:00:51:e4:91:9e:0f:28 \

\ "Wasp_e2" (fabric id=50)(Trunk master)
\
\ "Gnat_e1" (fabric id=100)(Trunk master)
\ "fcr_xd_2_100"

DCX_Backbone128:admin> flow --create b2e_dst_dcx -feature monitor -egrport 219
-srcdev 20:02:00:11:0d:51:00:00 -dstdev 10:00:00:05:1e:e8:e2:00

DCX_Backbone128:admin> flow --show
------------------------------------------------------------------------------ \
  Flow Name   |  Feature  |      SrcDev             |      DstDev             | \
------------------------------------------------------------------------------ \
b2e_dst_dcx   |mon+       |20:02:00:11:0d:51:00:00 |10:00:00:05:1e:e8:e2:00 | \
------------------------------------------------------------------------------ \
\ ------------------------------------
\ |IngrPt|EgrPt|BiDir| LUN   | FrameType|
\ ------------------------------------
\ |-     |219  |no    |-      |-         |
\ ------------------------------------
  + Denotes feature is currently activated for the flow

DCX_Backbone128:admin> flow --show b2e_dst_dcx -feature monitor
============================================================================
Name      : b2e_dst_dcx   Features: mon(Active) noConfig: Off
Definition: EgrPort(219),SrcDev(20:02:00:11:0d:51:00:00),
DstDev(10:00:00:05:1e:e8:e2:00)

Flow Monitor (Activated):
Monitor time:  | Fri Apr 15 15:59:58 UTC 2016 |
----------------------------------------------------------
-------------------------------------------------------------------------------------------------
| Tx Frames Count| Tx Frames per Sec.| Tx Bytes Count| Tx Throughput(Bps)| Avg Tx Frm Sz(Bytes)|
-------------------------------------------------------------------------------------------------
|     142.93M    |    3.74M          |    26.97G     |       724.78M     |          204        |
-------------------------------------------------------------------------------------------------
```

Monitoring an edge-to-backbone flow identified by WWNs

Flow Vision allows you to monitor edge-to-backbone flows passing through ports identified by World Wide Name (WWN) IDs.

In a network set up as shown in Figure 12 on page 63, for a flow passing from Device A to Device C that is ingressing through EX_Port1, the source device (srcdev) is port “WWN a”, the destination device (dstdev) is port “WWN c”, and the ingress port (ingrport) is EX_Port1. (Traffic in the figure is running from right to left, and the flow definitions are based on the Edge 1 Fabric’s perspective.)

The following example creates a flow that filters out frames passing from an edge fabric to the backbone fabric using a specific ingress port. The first command shows the available ports and the available Fibre Channel routers. The second command creates a Flow Monitor flow named “e2b_src_dcx” between device 10:00:00:05:1e:e8:e2:00 and device 20:02:00:11:0d:51:00:00 ingressing through port 219, and the last command displays the results of the flow.

NOTE

The slash character (\) in the example indicates a break inserted because the output is too long to display here as a single line.

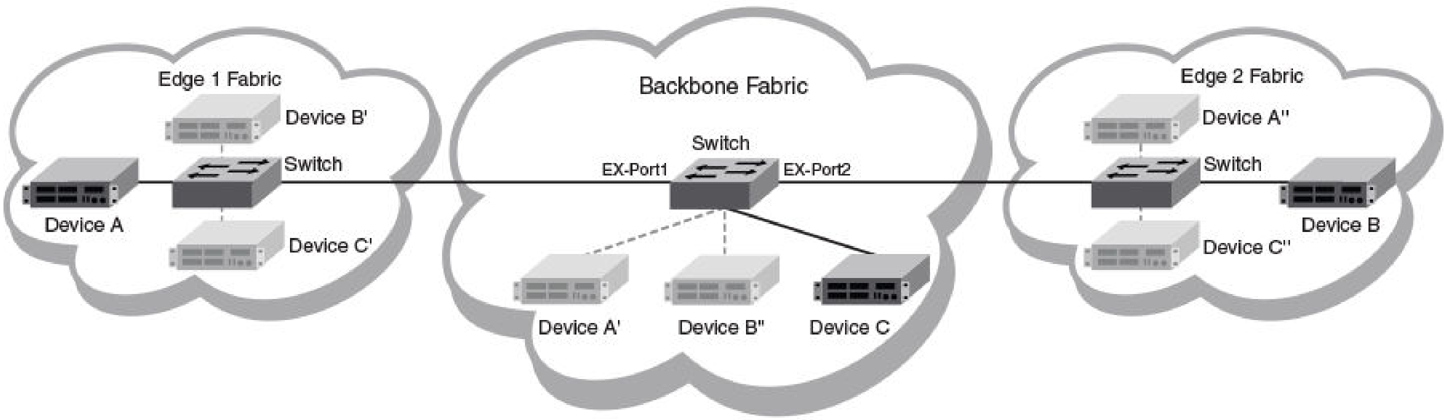
```
DCX_Backbone128:admin> switchshow |grep Port
Index Slot Port Address Media  Speed      State       Proto                         \
37    3    5    012500  id     N16        Online      FC EX-Port 10:00:00:05:33:ef:f1:1c \
47    3   15    012f00  id     N8         Online      FC F-Port  20:02:00:11:0d:51:00:00 \
219   10  27    01db00  id     N16        Online      FC EX-Port 10:00:00:05:33:ee:d0:a5 \
                                                         E-Port  50:00:51:e4:91:9e:0f:28 \
\
\ "Wasp_e2" (fabric id=50)(Trunk master)
\
\ "Gnat_e1" (fabric id=100)(Trunk master)
\ "fcr_xd_2_100"

DCX_Backbone128:admin> flow --create e2b_src_dcx -feature monitor -ingrport 219
-srcdev 10:00:00:05:1e:e8:e2:00 -dstdev 20:02:00:11:0d:51:00:00

DCX_Backbone128:admin>flow --show
---------------------------------------------------------------------------- \
  Flow Name    | Feature  |      SrcDev             |      DstDev            | \
---------------------------------------------------------------------------- \
e2b_src_dcx    |mon+      |10:00:00:05:1e:e8:e2:00 |20:02:00:11:0d:51:00:00 | \
---------------------------------------------------------------------------- \
\ --------------------------------------
\ |IngrPt|EgrPt|BiDir| LUN  |FrameType |
\ --------------------------------------
\ |219   |-    |no   |-     |-         |
\ --------------------------------------
     + Denotes feature is currently activated for the flow

DCX_Backbone128:admin> flow --show e2b_src_dcx -feature monitor
==========================================================================
Name: e2b_src_dcx  Features: mon(Active) noConfig: Off
Definition: IngrPort(219),SrcDev(10:00:00:05:1e:e8:e2:00),
DstDev(20:02:00:11:0d:51:00:00)

Flow Monitor (Activated):
Monitor time:   | Mon Jul 17 15:59:58 UTC 2013 |
---------------------------------------------------------
-------------------------------------------------------------------------------------------------
| Rx Frames Count| Rx Frames per Sec.| Rx Bytes Count| Rx Throughput(Bps)| Avg Rx Frm Sz(Bytes)|
-------------------------------------------------------------------------------------------------
|     142.93M    |         3.74M     |     26.97G     |     724.78M       |          204        |
-------------------------------------------------------------------------------------------------
```

Monitoring Fibre Channel router fabrics using proxy IDs

The following figures and their descriptions illustrate how port IDs (PIDs) are assigned in Fibre Channel router fabrics using proxy IDs. Use the **flow --control -deviceidmode pid** command to set the mode to PID mode.

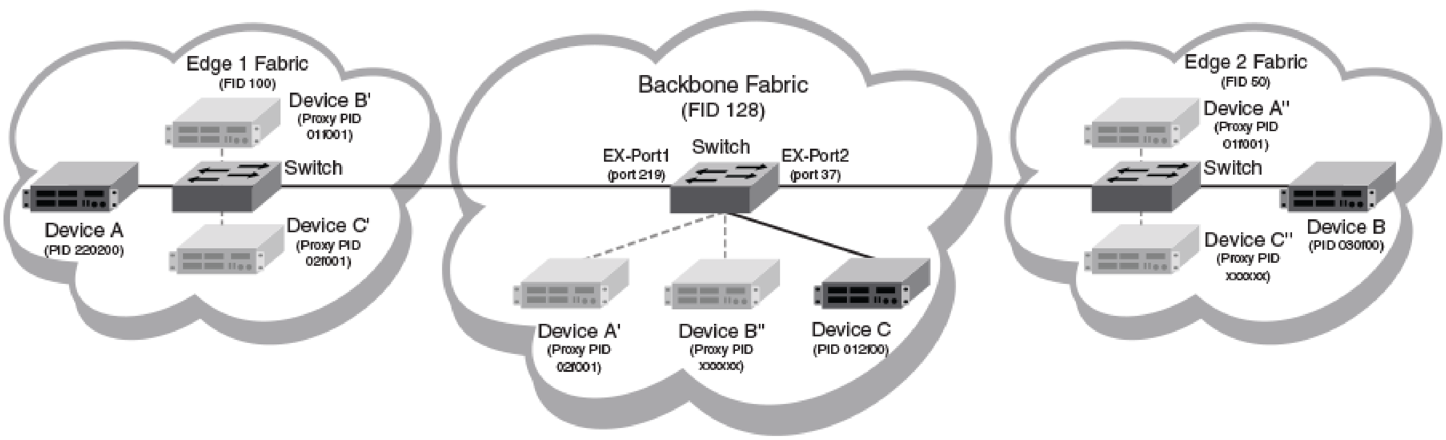
In the following figure, the physical devices are labeled A, B, and C. The proxy devices are the devices labeled A′, B′, C′, A″, B″, and C″, representing the physical devices A, B, and C, respectively.

FIGURE 13 An FC router fabric



The following figure provides the port ID, fabric ID, and proxy port ID values for the following examples.

FIGURE 14 An FC router fabric annotated with PID, FID, and proxy PID values



NOTE
The proxy port ID values for devices B″ and C″ were not generated for the examples, and so are marked “xxxxxx” in this figure.

Monitoring an edge-to-edge flow through an ingress port identified by a proxy ID

In a network set up as shown in Figure 14 on page 68, for a flow passing from Device A to Device B that is ingressing through EX_Port1, the source device (srcdev) is Device A, the destination device (dstdev) is Device B′, and the ingress port (ingrport) is EX_Port1. (Traffic is running from left to right, and the flow definitions are based on the Edge 1 Fabric’s perspective.)

The following example creates a flow that filters frames passing from one edge fabric to another edge fabric using a specific ingress port on the backbone. Notice that this is running in port WWN (portwwn) mode rather than device ID (deviceid) mode. The first two commands show the available ports and the available Fibre Channel routers. The third command creates a Flow Monitor flow named "e2e_src_dcx_wwn" between device 220200 and device 01f001 ingressing through port 219, and the last command displays the results of the flow.

NOTE
The slash character (\) in the example indicates a break inserted because the output is too long to display here as a single line.

```
DCX_Backbone128:admin> switchshow |grep Port
 Index Slot Port Address Media  Speed     State    Proto                       \
  37    3    5   012500   id    N16       Online   FC EX-Port 10:00:00:05:33:ef:f1:1c \
  47    3   15   012f00   id    N8        Online   FC F-Port  20:02:00:11:0d:51:00:00 \
 219   10   27   01db00   id    N16       Online   FC EX-Port 10:00:00:05:33:ee:d0:a5 \
                                                      E-Port  50:00:51:e4:91:9e:0f:28 \
\
\"Wasp_e2" (fabric id=50)(Trunk master)
\
\"Gnat_e1" (fabric id=100)(Trunk master)
\"fcr_xd_2_100"

DCX_Backbone128:admin> fcrproxydevshow

  Proxy           WWN             Proxy      Device   Physical    State
 Created                           PID       Exists     PID
in Fabric                                  in Fabric
-------------------------------------------------------------------------
    50   10:00:00:05:1e:e8:e2:00  01f001      100      220200   Imported
   100   20:00:00:11:0d:e4:18:00  01f001       50      030f00   Imported
   100   20:02:00:11:0d:51:00:00  02f001      128      012f00   Imported
   128   10:00:00:05:1e:e8:e2:00  02f001      100      220200   Imported
Total devices displayed: 4

DCX_Backbone128:admin> flow --create e2e_src_dcx_wwn -feature monitor -ingrport 219
-srcdev 220200 -dstdev 01f001

DCX_Backbone128:admin> flow --show
------------------------------------------------------------------------- \
      Flow Name    |   Feature    |       SrcDev       |        DstDev      \
------------------------------------------------------------------------- \
e2e_src_dcx_wwn   |mon+          |220200             |01f001              \
------------------------------------------------------------------------- \

\ --------------------------------------
\ |IngrPt|EgrPt|BiDir| LUN  | FrameType|
\ --------------------------------------
\ |219   |-    |no   |-     |-         |
\ --------------------------------------
    + Denotes feature is currently activated for the flow

DCX_Backbone128:admin> flow --show e2e_src_dcx_wwn -feature monitor

===========================================================================
Name      : e2e_src_dcx_wwn   Features: mon(Active) noConfig: Off
Definition: IngrPort(219),SrcDev(220200),DstDev(01f001)

Flow Monitor (Activated):
Monitor time:  | Fri Apr 15 14:59:58 UTC 2016 |
---------------------------------------------------------
-------------------------------------------------------------------------------------------------
| Rx Frames Count| Rx Frames per Sec.|  Rx Bytes Count| Rx Throughput(Bps)| Avg Rx Frm Sz(Bytes)|
-------------------------------------------------------------------------------------------------
|       2.85G    |          8.44M    |     387.88G    |         1.12G     |              132    |
-------------------------------------------------------------------------------------------------
```

Monitoring an edge-to-edge flow through an egress port identified by a proxy ID

In a network set up as shown in Figure 14 on page 68, for a flow passing from Device B to Device A that is egressing through EX_Port1, the source device (srcdev) is Device B′, the destination device (dstdev) is Device A, and the egress port (egrport) is EX_Port1. (Traffic is running from left to right.)

The following example creates a flow that filters out frames passing from one edge fabric to another edge fabric using a specific egress port on the backbone. The first two commands show the available ports and the available Fibre Channel routers. The third command creates a Flow Monitor flow named “e2e_dst_dcx” between device 01f001 and device 220200 egressing through port 219, and the last command displays the results of the flow.

NOTE

The slash character (\) in the example indicates a break inserted because the output is too long to display here as a single line.

```
DCX_Backbone128:admin> switchshow |grep Port
Index Slot Port Address Media  Speed     State     Proto                         \
37    3    5    012500   id    N16       Online    FC EX-Port 10:00:00:05:33:ef:f1:1c \
47    3   15    012f00   id    N8        Online    FC F-Port  20:02:00:11:0d:51:00:00 \
219   10  27    01db00   id    N16       Online    FC EX-Port 10:00:00:05:33:ee:d0:a5 \
                                                      E-Port  50:00:51:e4:91:9e:0f:28 \
\
\ "Wasp_e2" (fabric id=50)(Trunk master)
\
\ "Gnat_e1" (fabric id=100)(Trunk master)
\ "fcr_xd_2_100"

DCX_Backbone128:admin> fcrproxydevshow

  Proxy           WWN             Proxy      Device   Physical    State
 Created                           PID       Exists     PID
in Fabric                                   in Fabric
--------------------------------------------------------------------------
    50   10:00:00:05:1e:e8:e2:00  01f001      100      220200   Imported
   100   20:00:00:11:0d:e4:18:00  01f001       50      030f00   Imported
   100   20:02:00:11:0d:51:00:00  02f001      128      012f00   Imported
   128   10:00:00:05:1e:e8:e2:00  02f001      100      220200   Imported
Total devices displayed: 4

DCX_Backbone128:admin> flow --create e2e_dst_dcx -feature monitor -egrport 219
-srcdev 01f001 -dstdev 220200

DCX_Backbone128:admin>flow --show
------------------------------------------------------------------------- \
   Flow Name      |  Feature  |       SrcDev      |        DstDev         \
------------------------------------------------------------------------- \
e2e_dst_dcx       |mon+       |01f001             |220200                 \
------------------------------------------------------------------------- \

\ --------------------------------------
\ |IngrPt|EgrPt|BiDir| LUN  | FrameType|
\ --------------------------------------
\ |-     |219  |no   |-     |-         |
\ --------------------------------------
     + Denotes feature is currently activated for the flow

DCX_Backbone128:admin> flow --show e2e_dst_dcx -feature monitor

==========================================================================
Name      : e2e_dst_dcx   Features: mon(Active) noConfig: Off
Definition: EgrPort(219),SrcDev(0x01f001),DstDev(0x220200)

Flow Monitor (Activated):
Monitor time:  | Fri Apr 15 14:59:58 UTC 2016 |
---------------------------------------------------------
-------------------------------------------------------------------------------------------------
| Tx Frames Count| Tx Frames per Sec.| Tx Bytes Count| Tx Throughput(Bps)| Avg Tx Frm Sz(Bytes)|
-------------------------------------------------------------------------------------------------
|       2.85G    |         8.44M     |     387.88G    |         1.12G |         132          |
-------------------------------------------------------------------------------------------------
```

Monitoring a backbone-to-edge flow identified by proxy IDs

In a network set up as shown in Figure 14 on page 68, for a flow passing from Device C to Device A that is egressing through EX_Port1, the source device (srcdev) is Device C', the destination device (dstdev) is Device A, and the egress port (egrport) is EX_Port1. (Traffic is running from left to right, and the flow definitions are based on the Edge 1 Fabric's perspective.)

The following example creates a flow that filters out frames passing from the backbone fabric to an edge fabric using a specific egress port. The first two commands show the available ports and the available Fibre Channel routers. The third command creates a Flow Monitor flow named "b2e_dst_dcx" between device 02f001 and device 220200 egressing through port 219, and the last command displays the results of the flow.

NOTE

The slash character (\) in the example indicates a break inserted because the output is too long to display here as a single line.

```
DCX_Backbone128:admin> switchshow |grep Port
Index Slot Port Address Media  Speed      State       Proto                         \
37    3     5    012500   id     N16       Online      FC EX-Port 10:00:00:05:33:ef:f1:1c \
47    3    15    012f00   id     N8        Online      FC EX-Port 10:00:00:05:33:ef:f1:1c \
219   10   27    01db00   id     N16       Online      FC F-Port  20:02:00:11:0d:51:00:00 \
                                                       FC EX-Port 10:00:00:05:33:ee:d0:a5 \
                                                       E-Port   50:00:51:e4:91:9e:0f:28    \
\
\ "Wasp_e2" (fabric id=50)(Trunk master)
\ "Wasp_e2" (fabric id=50)(Trunk master)
\ "Gnat_e1" (fabric id=100)(Trunk master)
\ "fcr_xd_2_100"

DCX_Backbone128:admin> fcrproxydevshow
  Proxy           WWN              Proxy      Device   Physical    State
 Created                            PID       Exists     PID
in Fabric                                    in Fabric
-------------------------------------------------------------------------
    50   10:00:00:05:1e:e8:e2:00  01f001      100      220200   Imported
   100   20:00:00:11:0d:e4:18:00  01f001       50      030f00   Imported
   100   20:02:00:11:0d:51:00:00  02f001      128      012f00   Imported
   128   10:00:00:05:1e:e8:e2:00  02f001      100      220200   Imported
Total devices displayed: 4

DCX_Backbone128:admin> flow --create b2e_dst_dcx -feature monitor
-egrport 219 -srcdev 02f001 -dstdev 220200

DCX_Backbone128:admin> flow --show
------------------------------------------------------------------------- \
  Flow Name        |  Feature      |      SrcDev         |      DstDev        \
------------------------------------------------------------------------- \
b2e_dst_dcx       |mon+          |02f001              |220200             \
------------------------------------------------------------------------- \
\ --------------------------------------
\ |IngrPt|EgrPt|BiDir| LUN   | FrameType|
\ --------------------------------------
\ |-     |219  |no    |-      |-         |
\ --------------------------------------
    + Denotes feature is currently activated for the flow

DCX_Backbone128:admin> flow --show b2e_dst_dcx -feature monitor
=========================================================================
Name      : b2e_dst_dcx   Features: mon(Active) noConfig: Off
Definition: EgrPort(219),SrcDev(0x02f001),DstDev(0x220200)
Flow Monitor (Activated):
Monitor time:  | Fri Apr 15 15:59:58 UTC 2016 |
---------------------------------------------------------
-----------------------------------------------------------------------------------------------
| Tx Frames Count| Tx Frames per Sec.| Tx Bytes Count| Tx Throughput(Bps)|Avg Tx Frm Sz(Bytes)|
-----------------------------------------------------------------------------------------------
|    142.93M     |    3.74M          |     26.97G    |     724.78M        |       204          |
-----------------------------------------------------------------------------------------------
```

Monitoring an edge-to-backbone flow identified by proxy IDs

In a network set up as shown in Figure 14 on page 68, for a flow passing from Device A to Device C that is ingressing through EX_Port1, the source device (srcdev) is Device A, the destination device (dstdev) is Device C′, and the ingress port (ingrport) is EX_Port1. (Traffic is running from left to right, and the flow definitions are based on the Edge 1 Fabric's perspective.)

The following example creates a flow that filters out frames passing from an edge fabric to the backbone fabric using a specific ingress port. The first two commands show the available ports and the available Fibre Channel routers. The third command creates a Flow Monitor flow named "e2b_src_dcx" between device 220200 and device 02f001 egressing through port 219, and the last command displays the results of the flow.

NOTE
The slash character (\) in the example indicates a break inserted because the output is too long to display here as a single line.

```
DCX_Backbone128:admin> switchshow |grep Port
Index Slot Port Address  Media  Speed      State      Proto                            \
37     3    5    012500   id     N16        Online     FC EX-Port 10:00:00:05:33:ef:f1:1c \
47     3   15    012f00   id     N8         Online     FC F-Port  20:02:00:11:0d:51:00:00 \
219   10   27    01db00   id     N16        Online     FC EX-Port 10:00:00:05:33:ee:d0:a5 \
                                                          E-Port  50:00:51:e4:91:9e:0f:28 \
\
\ "Wasp_e2" (fabric id=50)(Trunk master)
\
\ "Gnat_e1" (fabric id=100)(Trunk master)
\ "fcr_xd_2_100"

DCX_Backbone128:admin> fcrproxydevshow
  Proxy           WWN             Proxy      Device   Physical    State
 Created                           PID       Exists     PID
in Fabric                                   in Fabric
---------------------------------------------------------------------------
    50   10:00:00:05:1e:e8:e2:00  01f001      100      220200   Imported
   100   20:00:00:11:0d:e4:18:00  01f001       50      030f00   Imported
   100   20:02:00:11:0d:51:00:00  02f001      128      012f00   Imported
   128   10:00:00:05:1e:e8:e2:00  02f001      100      220200   Imported
Total devices displayed: 4

DCX_Backbone128:admin> flow --create e2b_src_dcx -feature monitor -ingrport 219
-srcdev 220200 -dstdev 02f001

DCX_Backbone128:admin> flow --show
-------------------------------------------------------------------------- \
  Flow Name       |  Feature        |      SrcDev          |      DstDev      \
-------------------------------------------------------------------------- \
e2b_src_dcx      |mon+             |220200               |02f001            \
-------------------------------------------------------------------------- \

\ ----------------------------------------
\ |IngrPt|EgrPt|BiDir| LUN  | FrameType|
\ ----------------------------------------
\ |219   |-    |no   |-     |-         |
\ ----------------------------------------
    + Denotes feature is currently activated for the flow

DCX_Backbone128:admin> flow --show e2b_src_dcx -feature monitor

===========================================================================
Name: e2b_src_dcx  Features: mon(Active) noConfig: Off
Definition: IngrPort(219),SrcDev(0x220200),DstDev(0x02f001)
Flow Monitor (Activated):
Monitor time:  | Apr 15 15:59:58 UTC 2016 |
---------------------------------------------------------
-------------------------------------------------------------------------------------------------
| Rx Frames Count| Rx Frames per Sec.| Rx Bytes Count| Rx Throughput(Bps)| Avg Rx Frm Sz(Bytes)|
-------------------------------------------------------------------------------------------------
|     142.93M    |   3.74M           |    26.97G     |      724.78M       |          204        |
-------------------------------------------------------------------------------------------------
```

XISL and Backbone E_Port monitoring

Flow Monitor provides support for both learning and static monitoring of fabric-wide statistics on both XISL_Ports and Backbone E_Ports.

Output created using the **flow --show** command for a flow using an XISL_Port or a Backbone E_Port displays actual device PIDs and the edge fabric FIDs for Edge-to-Edge traffic, and PIDs with reference to the backbone fabric and backbone fabric FID for Edge to Backbone traffic. You can use this data to estimate the logical fabric or inter-fabric utilization of an XISL_Port or a Backbone E_Port, as shown in the following figures.

The following restrictions apply to XISL and Backbone E_Port monitoring:

- IFL flows cannot be monitored if the source fabric ID and destination fabric IDs are not specified. If these are not specified, the devices will not be known in the backbone fabric and the flow will be treated as a flow with end devices “offline”.
- Flows without the source fabric ID and destination fabric IDs specified will work only for flows on the same fabric. IFL traffic will not be monitored (except in the case of Backbone-to-Edge flows, where the flow is deemed to be in same fabric).
- Only the combinations specified in the following table can be configured.

TABLE 15 XISL and Backbone E_Port monitoring combinations

Srcdev	Dstdev	SFID	DFID	Port	Description
Not specified	Not specified	Fixed	Fixed	Ingress or Egress	This combination monitors frame statistics for traffic using the specified port from the specified SFID to the specified DFID.
Not specified	Not specified	*	*	Ingress or Egress	This combination monitors frame statistics on the specified port for traffic from all SFIDs to all DFIDs.
*	*	*	*	Ingress or Egress	This combination monitors frame statistics on the specified port for traffic from all SFIDs to all DFIDs, and records the device IDs associated with the traffic.

FIGURE 15 Monitoring fabric statistics on an XISL_Port

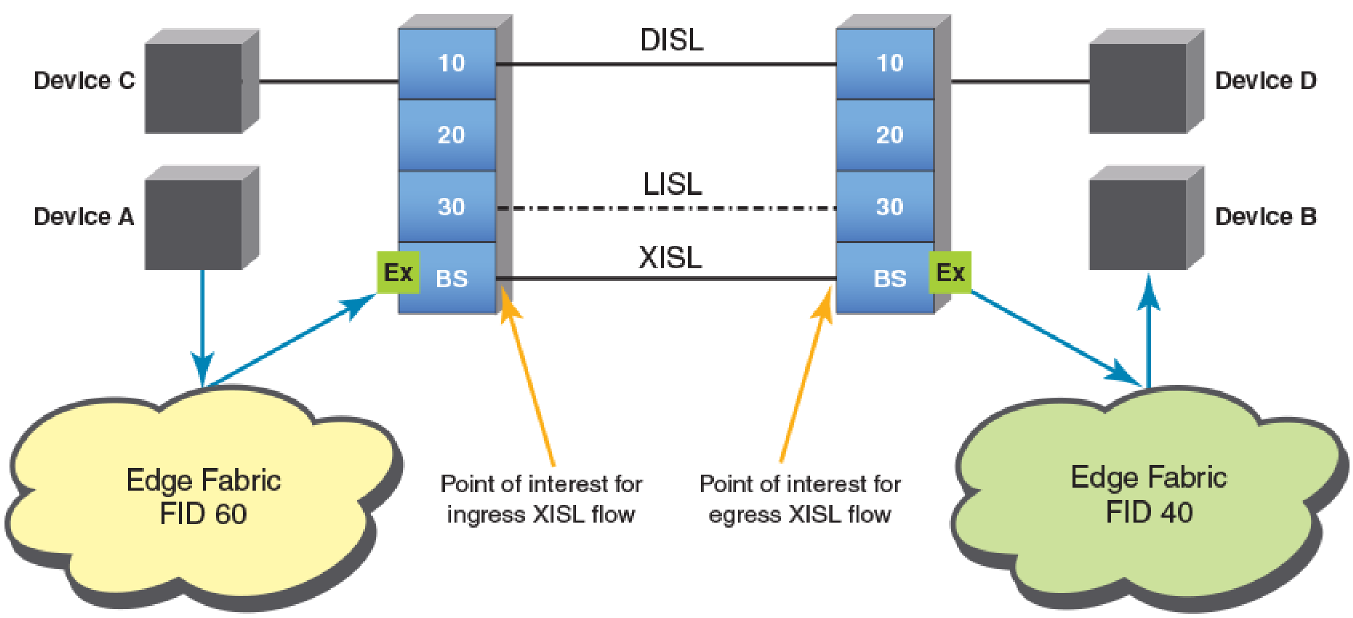
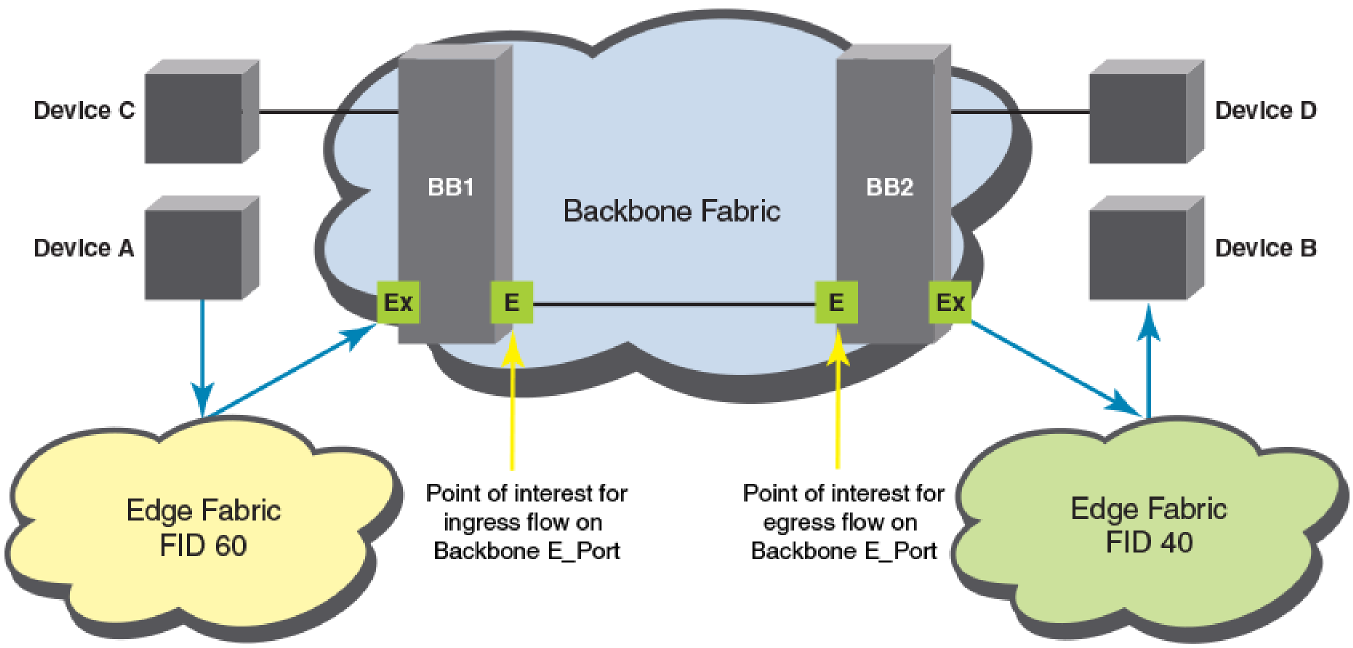


TABLE 16 Learning support for XISL_Ports and Backbone E_Ports

Learned traffic	Backbone E_Ports	XISL_Ports
Intra-fabric traffic	Only Backbone fabric traffic is learned.	Logical fabric traffic is learned.
Inter-fabric traffic	Edge-to-edge, Backbone-to-edge, and edge-to-Backbone traffic is learned.	Edge-to-edge traffic only is learned.

The following figure shows fabric-based monitoring on a Backbone E_Port between BB1 and BB2.

FIGURE 16 Fabric-based monitoring on a Backbone E_Port



Refer to XISL_Port or Backbone E_Port flow examples on page 74 to see examples of the flow creation and output for this type of flow.

XISL_Port or Backbone E_Port flow examples

The following examples show the creation and typical results of flows for using XISL_Ports or Backbone E_Ports.

The following example creates three flows on one XISL_Port or Backbone E_Port.

```
switch10:FID128:admin> flow --create fm219 -ingrport 219 -sfid 10 -dfid 20 -feature monitor

switch10:FID128:admin> flow --create fm219 -ingrport 219 -sfid "*" -dfid "*" -feature monitor

switch10:FID128:admin> flow --create fm219 -ingrport219
                             -srcdev "*"  -dstdev "*" -sfid "*" -dfid "*" -feature monitor
```

The following example shows the result of entering these commands.

```
switch10:FID128:admin> flow --show fm219 -feature monitor
============================================================================
Name      : fm  Features: gen(Activated),mon(Activated) noConfig: Off
```

```
Definition: EgrPort(12),SFID(80),DFID(90)

Flow Monitor (Activated):
Monitor time:  | Thu Apr 28 23:48:30 UTC 2016 |
-------------------------------------------------------------------------------------------
|SFID|DFID|Tx Frames |Tx Frames    |Tx Bytes Count |Tx Throughput(Bps) |Avg Tx Frm Sz(Bytes)|
|    |    | Count    |  per Sec.   |               |                   |                    |
-------------------------------------------------------------------------------------------
|80  |90  | 16.29M   | 402.46k     |   31.61G      |   799.88M         |      2084          |

switch10:FID128:admin> flow --show fm -feature monitor
===========================================================================
Name      : fm  Features: gen(Activated),mon(Activated) noConfig: Off
Definition: EgrPort(11),SFID(70),DFID(70)

Flow Monitor (Activated):
Monitor time:  | Thu Apr 28 23:48:30 UTC 2016 |
-------------------------------------------------------------------------------------------
|SFID|DFID|Tx Frames |Tx Frames    |Tx Bytes Count |Tx Throughput(Bps) |Avg Tx Frm Sz(Bytes)|
|    |    | Count    |  per Sec.   |               |                   |                    |
-------------------------------------------------------------------------------------------
|70  |70  | 16.29M   | 402.46k     |   31.61G      |   799.88M         |      2084          |


switch10:FID128:admin>  flow --show fm219 -feature monitor
===========================================================================
Name      : fm  Features: gen(Activated),mon(Activated) noConfig: Off
Definition: IngrPort(8),SID(*),DID(*), SFID(*),DFID(*)

Flow Monitor (Activated):
Monitor time:  | Thu Apr 28 23:48:30 UTC 2016 |
-----------------------------------------------------------------------------------------------------
|SFID|DFID|SID   |DID   |RxFramesCount|RxFrames perSec|RxBytesCount|RxThroughput(Bps)|AvG Rx Sz(Bytes)|
-----------------------------------------------------------------------------------------------------
|7   |11  |021200|031300|  16.29M     |     402.46k   |    31.61G  |    799.88M      |      2084      |
|7   |9   |021700|041200|  16.29M     |     402.46k   |    31.61G  |    799.88M      |      2084      |
|10  |10  |057200|057300|  18.00M     |     420.50k   |    36.11G  |    799.88M      |      2084      |
|20  |20  |061200|061300|  19.30M     |     450.50k   |    41.61G  |    799.88M      |      2084      |
-----------------------------------------------------------------------------------------------------
|*   |*   | *    | *    |  69.88M     |     1674.96k  |   126.46G  |      3.12G      |      2084      |
```

Virtual E_Port monitoring

Flow Monitor provides support for both learning and static monitoring for all the traffic passing through the given virtual E_Port (VE_Port) used in FCIP fabrics.

The scalability, metrics, and statistics for VE_Ports are similar to that of E_Port monitoring.

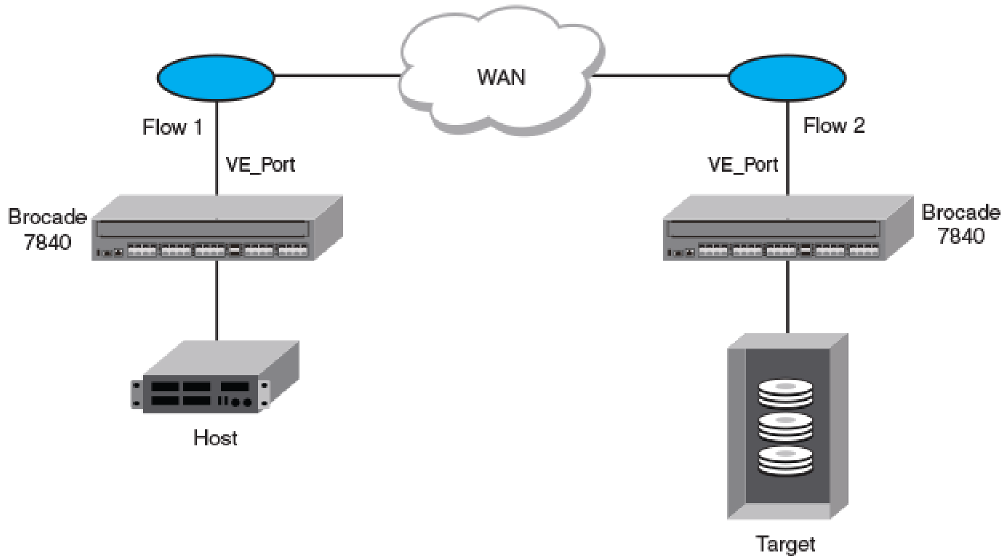
Frame type or bi-direction options can be added for the static flow combinations.

Statistics are gathered and displayed individually on each VE_Port. It is recommended that you create the same flow definition on each of the VE_Ports if they are connected to the same switch.

All metrics that are supported for E_Port monitoring are supported for VE_Port monitoring.

All existing commands used for E_Port monitoring can be used for VE_Port monitoring.

Static and learning support provided for VE_Port types



The follow table summarizes the static and learning support provided for VE_Port types.

TABLE 17 Static and learning support for VE_Ports

Port	SID	DID	Description
Ingress/Egress	WWN/PID	WWN/PID	Provides the statistics for the specified source and destination devices.
Ingress/Egress	WWN/PID	Not Specified	Collects the statistics for the specified source device and learns all active destination devices and provides the cumulative statistics of all device pairs.
Ingress/Egress	Not Specified	WWN/PID	Collects the statistics for the specified destination device and learns all active source devices and provides the cumulative statistics of all device pairs.
Ingress/Egress	WWN/PID	*	Collects the statistics for the specified source device and learns all destination devices and provides the itemized statistics for device pairs.
Ingress/Egress	*	WWN/PID	Collects the statistics for the specified destination device and learns all source devices and provides the itemized statistics for all device pairs.
Ingress/Egress	*	*	Learns all source and destination devices and provides the itemized statistics for all device pairs.

VE_Port monitoring restrictions

The following configurations and restrictions apply for VE_Port monitoring:

- Flow Generator and Flow Mirror features are not supported. Only the Flow Monitor feature is supported for VE_Ports.
- Only one learning flow can be active for a chip.
- TheBrocade 7840 switch and the Brocade SX6 Extension Blade are the only supported extension platforms.
- No support is available for legacy extension platforms (Brocade 7800 switch and FX8-24).
- If the data processing (DP) complex of the switch or blade is offline, all corresponding flows are uninstalled. Flows are reinstalled when DP is online.
- There is no Flow Vision support for VEX_Ports.
- LUN level monitor is not supported for VE_Ports.

- Trunking (FCIP Tunnels) is not exposed to VE_Ports.

Examples of VE_Port monitoring

The following examples shows the sample output while monitoring VE_Ports.

Example 1: VE_Port monitoring

For the flow, the following is specified: the source (SrcDev) is fixed, the target (DstDev) is **"*"**, and the ingress port (IngrPort) is fixed.

```
switch:admin> flow --create learn_ve27_sid755b00 -feature monitor
                    -ingrport 27 -srcdev 0x755b00 -dstdev "*"
Monitor feature(s) have been activated.
switch:admin>
switch:admin> flow --show learn_ve27_sid755b00
==================================================================================
Name      : learn_ve27_sid755b00        Features: mon(Activated)      noConfig: Off
Definition: IngrPort(27),SrcDev(0x755b00),DstDev(*)

Flow Monitor (Activated):
Monitor time:  | Thu Jan 14 11:17:47 UTC 2016 |
----------------------------------------------------------------------------------
|DID(*)| Rx Frames | Rx Frames | Rx Bytes | Rx Throughput(Bps)|Avg Rx Frm Sz(Bytes)|
|      | Count     | per Sec.  | Count    |                   |                    |
----------------------------------------------------------------------------------
|c21400|    350    |    8      |  23.24k  |       566         |         68         |
|c21500|    350    |    8      |  23.24k  |       566         |         68         |
|c21700|    340    |    6      |  22.57k  |       453         |         68         |
|c21800|    340    |    6      |  22.57k  |       453         |         68         |
|c21600|    140    |    6      |   9.29k  |       453         |         68         |
---------------------------------------------------------------------------------
|     *|  1.52k    |   34      | 100.93k  |     2.43k         |          68        |
----------------------------------------------------------------------------------
```

Example 2: VE_Port monitoring

For the flow, the following is specified: the source (SrcDev) is fixed, the target (DstDev) is fixed, the ingress port (IngrPort) is fixed, and the frame type is "srd" (SCSI Read).

```
switch:admin> flow --create ve_port27_srd -feature monitor
                    -ingrport 27 -srcdev 0x755b00 -dstdev 0xc21600 -frametype srd
Monitor feature(s) have been activated.
switch:admin>
switch:admin> flow --show ve_port27_srd
===========================================================================
Name      : ve_port27_srd      Features: mon(Activated)        noConfig: Off
Definition: IngrPort(27),SrcDev(0x755b00),DstDev(0xc21600),FrameType(srd)

Flow Monitor (Activated):
Monitor time:  | Wed Jan 13 12:48:39 UTC 2016 |
-----------------------------------------------
----------------------------------------
| Rx Frames Count | Rx Frames per Sec. |
----------------------------------------
|       10        |         0          |
----------------------------------------
```

Flow monitoring with IO Insight capability

Flow Monitor supports Small Computer Serial Interface (SCSI) Input/Output (I/O) command latency monitoring. SCSI I/O latency is the time it takes for an I/O operation to complete between an initiator and a target.

FPM allows the flexibility to select from predefined flows, static flows, or learning flows for monitoring purposes. It provides both Fibre Channel-level statistics and SCSI I/O operations per second (IOPS) performance counters. Fabric OS software enables flow latency monitoring at the SCSI I/O exchange level using a new set of counters. Flow Monitor supports the metrics of I/O latency only on Gen 6-based F_Ports

I/O statistics counters

Flow Monitor provides the following counters for monitoring I/O statistics:

- **First data time**: The time difference of either of these situations:
  - When the read command frame is transmitted and the first data frame is received.
  - When the write command frame is transmitted and the first Xfer-ready frame is received.
- **Status time**: The time difference of either of these situations:
  - When the read command frame is transmitted and the status frame is received.
  - When the write command frame is transmitted and the status frame is received.
- **Pending I/Os (Queue depth)**: The average number of outstanding I/O operations to be completed.
- **Completed I/Os**: The number of I/O operations that have good status responses.

Flow types supporting SCSI I/O latency statistics

There are two types of flows:

- Initiator-Target (IT) flow: IO Insight metrics are provided for SCSI READ and WRITE commands with SID and DID parameters.
  - An IT flow monitors traffic between any source device and a Gen 6 destination device.
  - The source or destination device can be in an unspecified condition (DC).
  - Flow learning is not supported.
  - A fully qualified IT flow will use fixed **srcdev** and fixed **dstdev** options.
  - A fully qualified IT flow will show complete IOPS metrics, but partially defined IT flows will not show SCSI statistics counters.
- Initiator-Target-Lun (ITL) flow: IO Insight metrics are provided for SCSI READ and WRITE commands with SID, DID, and LUN parameters.
  - An ITL flow monitors traffic between a source device and an individual LUN on a destination device.
  - The source or destination device can be in unspecified condition.
  - Flow learning is not supported.
  - ITL flows will not have Fibre Channel statistics.
  - A fully qualified ITL flow will use fixed **srcdev**, fixed **dstdev**, and fixed LUN options.
  - A fully qualified ITL flows will show complete IOPS metrics, but partially defined ITL flows will not show SCSI statistics counters.

SCSI I/O operation classification

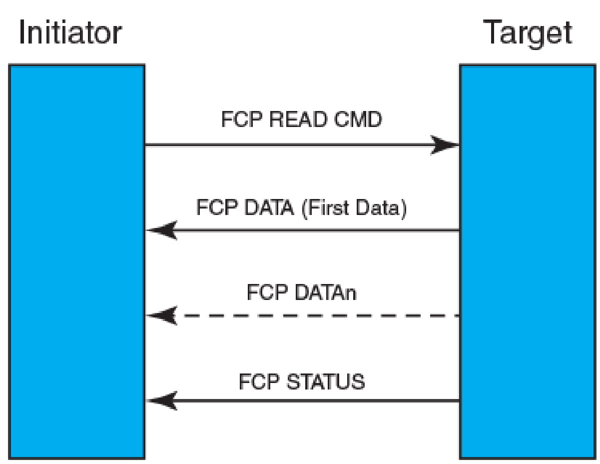
I/O operations are classified based on type and size, resulting in eight combinations for which I/O latency metrics are provided:

- **Operation type**: The type can be either READ or WRITE.
- **Operation size**: The size can be one of the following groups:
  - Less than 8K
  - 8K or more, but less than 64K
  - 64K or more, but less that 512K
  - 512K or more

SCSI Read I/O

The following figure shows the frame exchange for a SCSI Read I/O sequence.

FIGURE 17 SCSI Read I/O Latency

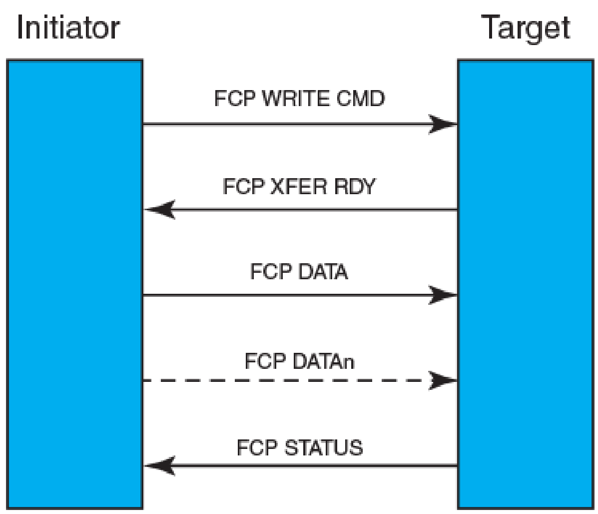


- **Read First Data Time:** The time difference between when the read command frame is transmitted and when the first data frame is returned.
- **Read Status Time:** The time difference between when the read command frame is transmitted and when the status frame is returned.
- **Pending I/Os (Queue depth):** The average number of outstanding read operations to be completed.
- **Completed I/Os:** The number of read operations that have good status responses.

SCSI Write I/O

The following figure shows the frame exchange for a SCSI Write I/O sequence.

FIGURE 18 SCSI Write I/O Latency



- **Write First Data Time:** The time difference between when the write command frame is transmitted and when the Xfer Ready frame is returned.
- **Write Status Time:** The time difference between when the write command frame is transmitted and when the status frame is returned.
- **Pending I/Os (Queue depth):** The average number of outstanding write operations to be completed.
- **Completed I/Os:** The number of write operations that have good status responses.

SCSI I/O latency configurations

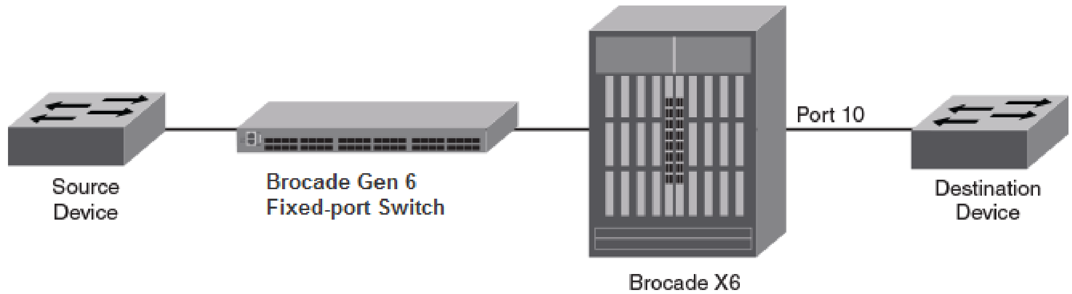
SCSI I/O latency is supported only on the F_Port and in the following situations:

- **Gen 6-based directors**
  - Both IT and ITL are supported.
  - Flows on both the ingress and egress ports are supported.
- **Gen 6-based fixed-port switches**
  - Only IT flows on the egress port support I/O latency.

Supported configurations for SCSI I/O latency

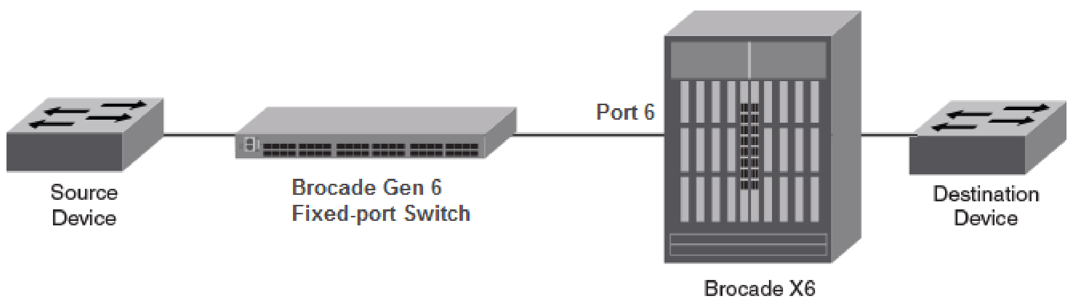
The following are examples of supported configurations:

FIGURE 19 Supported configuration for SCSI I/O latency: Egress port on director device



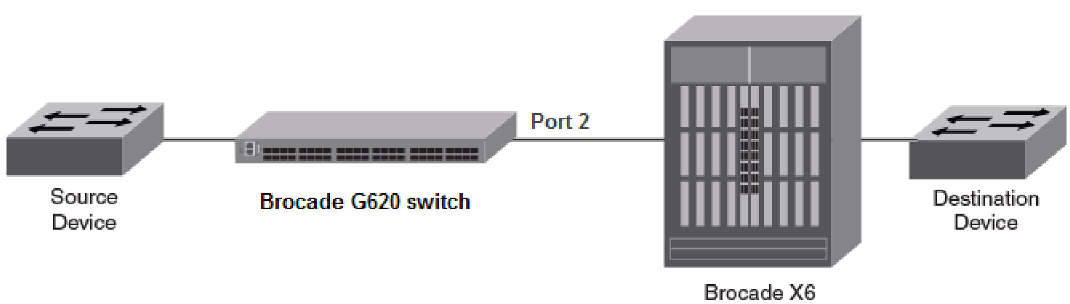
Because the point of interest (egress Port 10 on the Brocade X6-4 Director) is a Gen 6-based port, I/O latency statistics are supported for this flow. The support of I/O latency statistics is not affected if the source device is not Gen 6-based.

FIGURE 20 Supported configuration for SCSI I/O latency: Ingress port on director device



Because the point of interest (ingress Port 6 on the Brocade X6-4 Director) is a Gen 6-based port, I/O latency statistics are supported for this flow. The support of I/O latency statistics is not affected if the source device is not Gen 6-based.

FIGURE 21 Supported configuration for SCSI I/O latency: Egress port on fixed-port switch

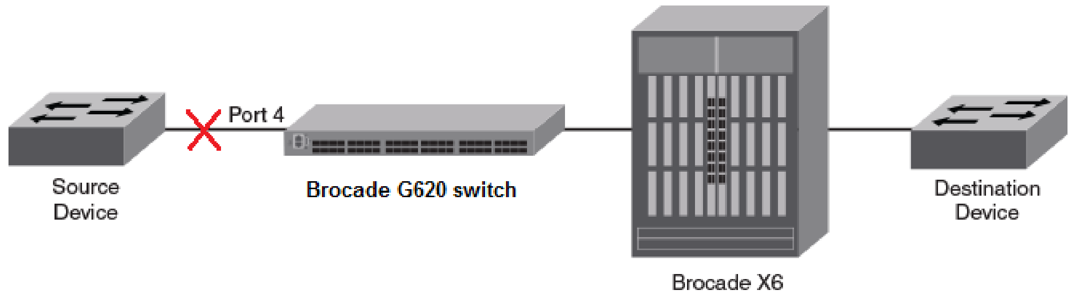


Because the point of interest (egress Port 6 on a Brocade) Gen 6-based fixed-port switch) is a Gen 6-based port, I/O latency statistics are supported for this flow. The support of I/O latency statistics is not affected if the source device is not Gen 6-based.

Unsupported configurations for SCSI I/O latency

The following is an example of a configuration that is **not** supported:

FIGURE 22 Unsupported configuration for SCSI I/O latency: Ingress port on fixed-port switch



Although the point of interest (ingress Port 4 on a Brocade) Gen 6-based fixed-port switch) is a Gen 6-based port, I/O latency statistics are **not** supported for this flow.

Statistics supported for configurations

For the following configurations, certain statistics are supported:

- **FC statistics:**
- **SCSI statistics:**
- **I/O statistics:**

TABLE 18 Statistics supported for configurations

	srcdev, dstdev,ingrport	srcdev, dstdev,egrport	srcdev, dstdev, lun (both ingrport and egrport)	srcdev alone or dstdev alone	srcdev alone with lun or destdev alone with lun
Brocade X6-4 Director or Brocade X6-8 Director	FC statistics SCSI statistics I/O statistics	FC statistics SCSI statistics I/O statistics	SCSI statistics I/O statistics	FC statistics I/O statistics	I/O statistics
Brocade G620 Switch	FC statistics SCSI statistics	FC statistics SCSI statistics I/O statistics	SCSI statistics	FC statistics	Flow not supported
Gen 5-based devices	FC statistics SCSI statistics	FC statistics SCSI statistics	SCSI statistics	FC statistics	Flow not supported

Flow definition metrics for IT and ITL flows

The following tables define which metrics are supported for IT and ITL flow definitions.

NOTE
Metrics supported for VM Insight flows are similar to those supported for other types of flows, except statistics are tracked by entity ID instead of port ID. For metrics used in VM flow definitions, refer to Flow definition metrics for VM Insight on page 91

TABLE 19 IT flow definition metrics

Description	Srcdev	Dstdev	Ingrport	Egrport
Monitor FC and I/O latency statistics on the ingress port for traffic from srcdev to any dstdev.	F	0	F	0

TABLE 19 IT flow definition metrics (continued)

Description	Srcdev	Dstdev	Ingrport	Egrport
Monitor FC and I/O latency statistics on the ingress port for traffic going to dstdev.	O	F	F	O
Monitor FC, IOPS and I/O latency statistics on the ingress port for traffic between srcdev-dstdev pair.	F	F	F	O
Monitor FC and I/O latency statistics on the egress port for traffic from srcdev to any dstdev.	F	O	O	F
Monitor FC and I/O latency statistics on the egress port for traffic going to any dstdev.	O	F	O	F
Monitor FC, IOPS, and I/O latency statistics on the egress port for traffic between a srcdev-dstdev pair.	F	F	O	F

TABLE 20 ITL flow definition metrics

Description	Srcdev	Dstdev	Ingrport	Egrport	Lun
Monitor only I/O latency statistics on the ingress port for traffic from srcdev on the given lun.	F	O	F	O	F
Monitor only I/O latency statistics on the ingress port for traffic going to dstdev on the given lun.	O	F	F	O	F
Monitor IOPS and I/O latency statistics on the ingress port for traffic between a srcdev-dstdev pair on the given lun.	F	F	F	O	F
Monitor only I/O latency statistics on the egress port for traffic from srcdev on the given lun.	F	O	O	F	F
Monitor only I/O latency statistics on the egress port for traffic going to dstdev on the given lun.	O	F	O	F	F
Monitor IOPS and I/O latency statistics on the egress port for traffic between a srcdev-dstdev pair on the given lun.	F	F	O	F	F

Flow latency metrics restrictions and limitations

- Flow latency is supported only on Gen 6 platforms.
- Flow latency is supported only on F_Ports. However, it is not supported on F_Ports on a backbone if the flow is an inter-fabric flow.
- I/O latency statistics are not supported in the Access Gateway platform.

Example of IT flow metrics

The following example shows the Fibre Channel, IOPS, and I/O performance and latency metrics in the output of the **flow --show** command for IT flows for which the ingress port (IngPort), the source device (SrcDev), and the target or destination device (DstDev) are fixed.

I/O latency metrics are recorded in units of milliseconds (m) and microseconds (u).

NOTE
If you install the monitor on the ingress port of the Gen 6-based fixed-port switch, the I/O performance metrics and the I/O latency metrics are not displayed. However, the Fibre Channel statistics and the SCSI statistics are displayed.

```
switch:admin> flow --show iostest
==============================================================================================================
======
Name      : iostest  Features: mon(Activated)   noConfig: Off
Definition: IngPort(12),SrcDev(0x030d00),DstDev(0x030c00)


Flow Monitor (Activated):
Monitor time:  | Mon Aug 15 13:09:51 UTC 2016 |
-----------------------------------------------
---------------------------------------------------------------------------------------------------
| Tx Frames Count | Tx Frames per Sec.| Tx Bytes Count | Tx Throughput(Bps)| Avg Tx Frm Sz(Bytes)|
---------------------------------------------------------------------------------------------------
|      1.78G      |        81.11k     |      2.06T     |        98.16M     |         1280        |
---------------------------------------------------------------------------------------------------

---------------------------------------------------------------------------------------------------
|      I/O Count        |    I/O Per Sec.(IOPS)   | I/O bytes Transferred |   I/O bytes Per Sec.   |
| Reads / Writes/ Total | Reads / Writes/ Total | Reads / Writes/ Total | Reads / Writes/ Total  |
---------------------------------------------------------------------------------------------------
|356.33M/356.33M/712.67M| 16.91k/ 16.28k/ 33.20k|  3.99T/  3.99T/  7.98T| 99.11M/ 95.42M/194.54M |
---------------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------------
| I/O Performance:                                                                                 |
---------------------------------------------------------------------------------------------------
|          Metric      |   IO Size   |  I/O Count   |      Max(IOPS)       |       Avg(IOPS)        |
|                      |             |     All      |      All             |   6 sec    /     All   |
---------------------------------------------------------------------------------------------------
| RD IO Count          | <8K         |    66.37k    |      8.59k           |    2.47k    /     2.13k  |
|                      | 8K - <64K   |       0      |        0             |      0      /       0    |
|                      | 64K - <512K|       0      |        0             |      0      /       0    |
|                      | >=512K      |       0      |        0             |      0      /       0    |
|                      | ALL         |    66.37k    |      8.59k           |    2.47k    /     2.13k  |
---------------------------------------------------------------------------------------------------
| WR IO Count          | <8K         |    66.83k    |      8.70k           |    2.43k    /     2.14k  |
|                      | 8K - <64K   |       0      |        0             |      0      /       0    |
|                      | 64K - <512K|       0      |        0             |      0      /       0    |
|                      | >=512K      |       0      |        0             |      0      /       0    |
|                      | ALL         |    66.83k    |      8.70k           |    2.43k    /     2.14k  |
---------------------------------------------------------------------------------------------------



------------------------------------------------------------------------------------------------
| I/O Latency Metrics:                                                                          |
------------------------------------------------------------------------------------------------
|          Metric             |   IO Size  |            Max             |           AVG          |
|                             |            |    6 sec   /     All       |   6 sec   /     All    |
------------------------------------------------------------------------------------------------
| RD CMD -> Status Time       | <8K        |      68u   /     2.86m     |     39u   /       39u  |
|                             | 8K - <64K  |       0u   /        0u     |      0u   /        0u  |
|                             | 64K - <512K|       0u   /        0u     |      0u   /        0u  |
|                             | >=512K     |       0u   /        0u     |      0u   /        0u  |
|                             | ALL        |      68u   /     2.86m     |     39u   /       39u  |
```

```
-----------------------------------------------------------------------------------------------
| WR CMD -> Status Time      | <8K          |     77u  /     2.84m  |       44u  /       44u  |
|                            | 8K - <64K    |      0u  /        0u  |        0u  /        0u  |
|                            | 64K - <512K|        0u  /        0u  |        0u  /        0u  |
|                            | >=512K       |      0u  /        0u  |        0u  /        0u  |
|                            | ALL          |     77u  /     2.84m  |       44u  /       44u  |
-----------------------------------------------------------------------------------------------
| RD CMD -> 1st Data Time    | <8K          |     47u  /     2.83m  |       21u  /       20u  |
|                            | 8K - <64K    |      0u  /        0u  |        0u  /        0u  |
|                            | 64K - <512K|        0u  /        0u  |        0u  /        0u  |
|                            | >=512K       |      0u  /        0u  |        0u  /        0u  |
|                            | ALL          |     47u  /     2.83m  |       21u  /       20u  |
-----------------------------------------------------------------------------------------------
| WR CMD -> 1st XFER_RDY Time| <8K          |     55u  /     2.80m  |       19u  /       19u  |
|                            | 8K - <64K    |      0u  /        0u  |        0u  /        0u  |
|                            | 64K - <512K|        0u  /        0u  |        0u  /        0u  |
|                            | >=512K       |      0u  /        0u  |        0u  /        0u  |
|                            | ALL          |     55u  /     2.80m  |       19u  /       19u  |
-----------------------------------------------------------------------------------------------
| RD Pending IOs             | <8K          |      4   /        4   |        1   /        3   |
|                            | 8K - <64K    |      0   /        0   |        0   /        0   |
|                            | 64K - <512K|        0   /        0   |        0   /        0   |
|                            | >=512K       |      0   /        0   |        0   /        0   |
-----------------------------------------------------------------------------------------------
| WR Pending IOs             | <8K          |      4   /        4   |        1   /        3   |
|                            | 8K - <64K    |      0   /        0   |        0   /        0   |
|                            | 64K - <512K|        0   /        0   |        0   /        0   |
|                            | >=512K       |      0   /        0   |        0   /        0   |
-----------------------------------------------------------------------------------------------
===============================================================================================
```

Example of ITL flow metrics

The following example shows the Fibre Channel, Type 5 IOPS, and I/O performance and latency metrics in the output of the **flow --show** command for ITL flows for which the ingress port (IngPort), the source device (SrcDev), the target or destination device (DstDev), and the lun (LUN) are fixed.

I/O latency metrics are recorded in units of milliseconds (m) and microseconds (u).

NOTE
ITL flows can only be defined for director devices, not for Gen 6-based fixed-port switches.

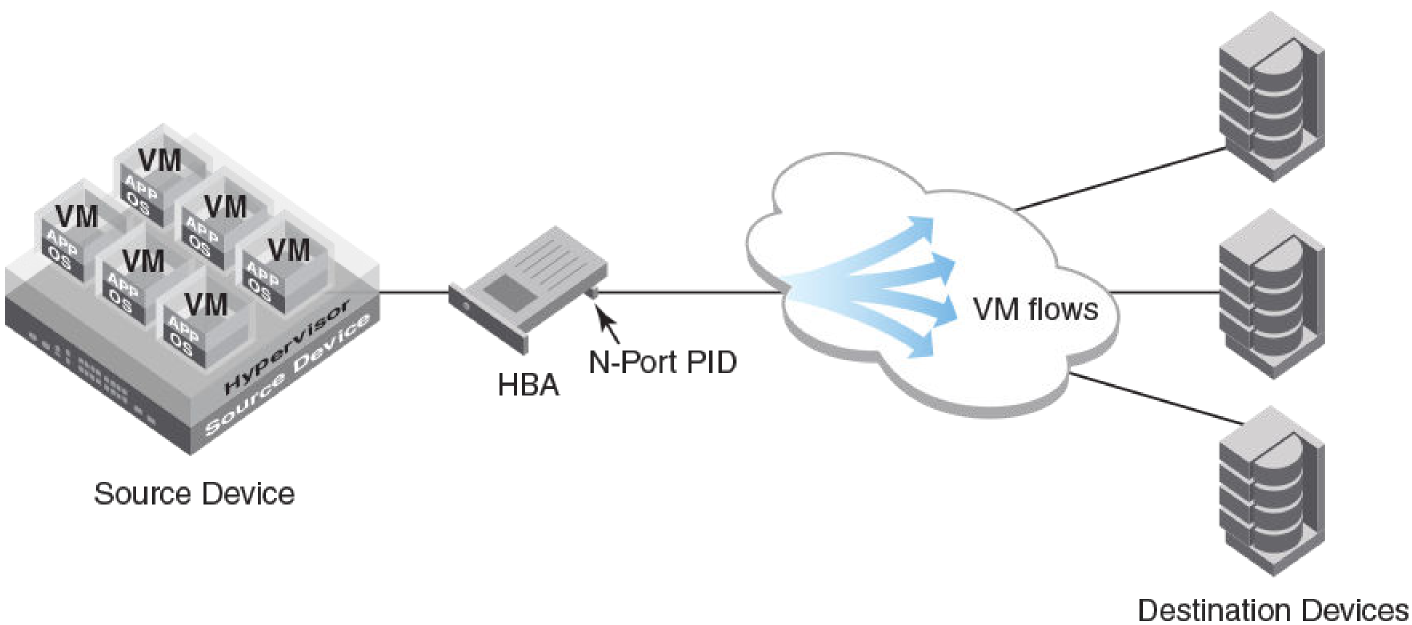
```
switch:admin> flow --show iostest
=======================================================================================
Name      : iostest  Features: mon(Activated)   noConfig: Off
Definition: IngPort(12),SrcDev(0x030d00),DstDev(0x030c00), LUN(5)


Flow Monitor (Activated):
Monitor time:   | Mon Aug 15 13:09:51 UTC 2016 |
-----------------------------------------------

-------------------------------------------------------------------------------------------------
|      I/O Count         |   I/O Per Sec.(IOPS)   | I/O bytes Transferred |   I/O bytes Per Sec.   |
| Reads / Writes/ Total | Reads / Writes/ Total | Reads / Writes/ Total | Reads / Writes/ Total  |
-------------------------------------------------------------------------------------------------
|356.33M/356.33M/712.67M| 16.91k/ 16.28k/ 33.20k|  3.99T/  3.99T/  7.98T| 99.11M/ 95.42M/194.54M |
-------------------------------------------------------------------------------------------------

-------------------------------------------------------------------------------------------------
| I/O Performance:                                                                              |
-------------------------------------------------------------------------------------------------
|          Metric        |   IO Size   |  I/O Count  |     Max(IOPS)     |        Avg(IOPS)       |
|                        |             |     All     |       All         |    6 sec    /    All   |
-------------------------------------------------------------------------------------------------
| RD IO Count            | <8K         |    66.37k   |      8.59k        |    2.47k    /    2.13k |
|                        | 8K - <64K   |        0    |          0        |        0    /        0 |
|                        | 64K - <512K|        0    |          0        |        0    /        0 |
|                        | >=512K      |        0    |          0        |        0    /        0 |
|                        | ALL         |    66.37k   |      8.59k        |    2.47k    /    2.13k |
-------------------------------------------------------------------------------------------------
| WR IO Count            | <8K         |    66.83k   |      8.70k        |    2.43k    /    2.14k |
|                        | 8K - <64K   |        0    |          0        |        0    /        0 |
|                        | 64K - <512K|        0    |          0        |        0    /        0 |
|                        | >=512K      |        0    |          0        |        0    /        0 |
|                        | ALL         |    66.83k   |      8.70k        |    2.43k    /    2.14k |
-------------------------------------------------------------------------------------------------

----------------------------------------------------------------------------------------------
| I/O Latency Metrics:                                                                       |
----------------------------------------------------------------------------------------------
|          Metric             |   IO Size   |          Max           |          AVG           |
|                             |             |    6 sec   /     All   |    6 sec   /     All   |
----------------------------------------------------------------------------------------------
| RD CMD -> Status Time       | <8K         |      68u   /     2.86m  |      39u   /       39u  |
|                             | 8K - <64K   |       0u   /       0u   |       0u   /        0u  |
|                             | 64K - <512K|       0u   /       0u   |       0u   /        0u  |
|                             | >=512K      |       0u   /       0u   |       0u   /        0u  |
|                             | ALL         |      68u   /     2.86m  |      39u   /       39u  |
----------------------------------------------------------------------------------------------
| WR CMD -> Status Time       | <8K         |      77u   /     2.84m  |      44u   /       44u  |
|                             | 8K - <64K   |       0u   /       0u   |       0u   /        0u  |
|                             | 64K - <512K|       0u   /       0u   |       0u   /        0u  |
|                             | >=512K      |       0u   /       0u   |       0u   /        0u  |
|                             | ALL         |      77u   /     2.84m  |      44u   /       44u  |
----------------------------------------------------------------------------------------------
| RD CMD -> 1st Data Time     | <8K         |      47u   /     2.83m  |      21u   /       20u  |
|                             | 8K - <64K   |       0u   /       0u   |       0u   /        0u  |
```

```
|                               | 64K - <512K|       0u   /       0u   |       0u   /       0u   |
|                               | >=512K     |       0u   /       0u   |       0u   /       0u   |
|                               | ALL        |      47u   /    2.83m   |      21u   /      20u   |
-----------------------------------------------------------------------------------------------
| WR CMD -> 1st XFER_RDY Time| <8K         |      55u   /    2.80m   |      19u   /      19u   |
|                               | 8K - <64K  |       0u   /       0u   |       0u   /       0u   |
|                               | 64K - <512K|       0u   /       0u   |       0u   /       0u   |
|                               | >=512K     |       0u   /       0u   |       0u   /       0u   |
|                               | ALL        |      55u   /    2.80m   |      19u   /      19u   |
-----------------------------------------------------------------------------------------------
| RD Pending IOs                | <8K        |       4    /       4    |       1    /       3    |
|                               | 8K - <64K  |       0    /       0    |       0    /       0    |
|                               | 64K - <512K|       0    /       0    |       0    /       0    |
|                               | >=512K     |       0    /       0    |       0    /       0    |
-----------------------------------------------------------------------------------------------
| WR Pending IOs                | <8K        |       4    /       4    |       1    /       3    |
|                               | 8K - <64K  |       0    /       0    |       0    /       0    |
|                               | 64K - <512K|       0    /       0    |       0    /       0    |
|                               | >=512K     |       0    /       0    |       0    /       0    |
-----------------------------------------------------------------------------------------------
===============================================================================================
```

Flow monitoring with VM Insight capability

The VM Insight feature allows you to monitor individual traffic flows from source to destination target for multiple virtual machine (VM) instances that originate from a single N_Port ID (PID). The following figure illustrates these flows.

FIGURE 23 Monitoring VM flows



Statistics gathered by monitoring the application-to-target performance path in the fabric using predefined and user-defined flows allows you to perform the following tasks:

- Isolate VMs that are being impacted by slow-draining devices, problematic cables, faulty SFPs, or other factors.
- Determine whether VM performance issues are originating in the host, fabric, or storage array.

- Gather detailed statistics on target devices to plan the allocation of VMs per node or to determine when VMs must be migrated for load balancing based on input/output operations per second (IOPS) or historical latency measurements.

VM Insight identifies VM traffic flows by tagging frames in each flow with an application header containing a unique *application ID*. This ID is based on the entity ID assigned to the VM instance by the hypervisor, such as the UUID assigned by VMware, and the PID assigned by the fabric to the HBA initiator N_Port.

NOTE
When defining a flow, you use the entity ID for the VM instance and not the application ID.

The application server is a key component of VM Insight. Based on a new FC standard (FC-GS), this service allows entities (VMs) to be registered with the fabric. A component of the management server, the application server stores entity IDs for virtual machine (VM) instances registered by HBAs and allocates application IDs to HBAs for identifying VM traffic in the fabric. The application server is a distributed database, similar to the name server and management server, and is propagated to all switches in the fabric that support the application server.

You can use the **appserver** command to display contents of the application server database including N_Port IDs, application IDs, PIDs, and domain information associated with registered entity IDs. Refer to the *Brocade Fabric OS Command Reference* for details.

The following steps summarize the process used by the fabric to identify VM traffic frames with the application ID:

1. The HBA (initiator) logs into the fabric and the fabric assigns a PID to the N_Port.
2. The HBA fetches the entity IDs for each VM instance from the VM traffic flows and registers these entity IDs with the application server.
3. The application server returns a unique application ID for each VM instance to the HBA.
4. The HBA tags each traffic flow for a specific VM with an application header containing the application ID.

Created VM flows can also be imported to MAPS, which can monitor statistics and generate alerts for defined out-of-bound values. For details, refer to the *Brocade Monitoring and Alerting Policy Suite Configuration Guide*.

VM Insight requirements

The following are requirements for VM Insight:

- The initiator HBA must support VM Insight by communicating with the application server to register VM entity IDs and tagging VM traffic with an application header containing unique application IDs. Contact your Brocade representative for the latest information on HBA products and firmware with VM Insight support.
- The target must support VM Insight by consuming a frame tagged with the application header containing the application ID and sending responses using this ID as the destination. Contact your Brocade representative for the latest information on storage devices and firmware with VM Insight support.
- Hypervisor support must be through VMware VSphere Hypervisor 5.5 and 6.X and later. Contact your Brocade representative for the latest support details.

VM Insight flow monitoring

Create VM Insight flows using the following command format.

flow --create *flowname* **-feature monitor -ingrport** *Source_ID* **-srceid** *Entity_ID* **-dstdev** *Destination_ID*

Components of a flow definition for VM Insight include the following:

- *flowname*: Use a unique string composed of a maximum of 20 alphanumeric or underscore characters. For example, **monitor_port2_vms**.
- ingrport: For *Source_ID*, use the PID or PWWN of the F_Port connected to the hypervisor managing the VM instance of interest.
- srceid: For *Entity_ID*, use the hexadecimal or ASCII entity ID of the VM instance for the ID of the source device.
- dstdev: For *Destination_ID*, use the PID or PWWN of the target device.

To view metrics output for the created flow, use the following format:

flow --show *flowname*

The following is a summary of available VM Insight flow monitors, flow monitor command definitions, monitor purpose, and monitor benefits.

TABLE 21 Summary of static and learning flow support for VM Insight

Flow monitor type	Dstdev	Srceid	Create command	Purpose	Benefits
Predefined learning	Not specified	Specified	Flow creation is not required. Activate predefined flow using **--activate sys_mon_all_vms**.	Monitors overall health and performance of the virtualized fabric.	Metrics can determine most-used and least-used applications for capacity planning and VM migration purposes. Statistics are reported for each registered instance across all communicating storage devices.
User-defined learning	*	Specified	**flow --create** *flowname* **-feature monitor -srceid** *Entity_ID* **-dstdev "*"**	Monitors traffic flow from a known VM instance to all communicating target devices.	Use metrics for both capacity planning and to troubleshoot performance issues for traffic flowing to targets from a known VM instance.
User-defined static	Specified	Specified	**flow --create** *flowname* **-feature monitor -ingrport** *Source_ID* **-srceid** *Entity_ID* **-dstdev** *Destination_ID*	Monitor application to storage. Monitors traffic flow from a known VM instance to a known target device. Note that you could use a destination ID discovered through a user-defined learning flow to obtain detailed metrics for traffic between the communicating target device and known VM instance.	Monitors traffic from an over- demanding or under-performing application to troubleshoot performance issues.

Considerations when configuring VM Insight flows

You should be aware of the following considerations when configuring VM Insight flows:

- To learn destination devices that are communicating with a VM instance that you want to monitor, determine that the entity ID for the VM instance is registered on the application server using the **appserver --show** command (refer to Creating a VM Insight flow monitor on page 91). You can cut and paste the ID from the **appserver --show** output for the --srceid operand into the **flow --create** command.

- The entity ID for the srceid operand can be a hexadecimal or ASCII value. You can use either format. If you use an ID that begins with 0x, the value is interpreted as hexadecimal.

Creating a VM Insight flow monitor

Use the following process to creating a VM Insight flow monitor.

1. Using your hypervisor application, determine the entity ID for the VM instance for which you want to monitor traffic flow.
2. Enable your HBA to use VM Insight using instructions in your HBA user documentation.
3. Enter the **appserver --show -all** command on the switch to display all entity IDs registered with the application server. The entity ID for the VM that you want to monitor must be registered.

   Following is an example of the output of the **appserver --show -all** command . Depending on the HBA, the ID displays as either a hexadecimal number or an ASCII value. You can use either format when creating a flow definition. If you use an ID that begins with 0x, the value is interpreted as hexadecimal.

   ```
   switch:admin>appserver --show -all
   ----------------------------------------------------------
   Displaying results for Fabric
   ----------------------------------------------------------
   N_Port ID         : 010200
   Entity ID (ASCII) : 52 fc ef 53 8b ed 5a 32-10 5b 72 77 e7 df d8 83
   Entity ID (Hex)   : 0x353220666320656620353332038622065
                       642035612033322d3130203562203732
                       2037372065372064662064382033833
   Application ID    : 0x00000209h (521)
   ```

   Options are available for the **appserver --show -all** command to filter the output from the application server database by a specific PID, entity ID, or domain ID. Refer to the *Brocade Fabric OS Command Reference* for details.
4. Record the entity ID or copy it to a flow definition to use as the *srceid* operand when required.
5. Using flow definitions outlined in VM Insight flow monitoring on page 89, create a learning, static, or predefined flow monitor on each switch where the VM traffic that you want to monitor will enter the fabric.

   NOTE
   You can only configure monitors in a per-switch basis.
6. Activate the flow.
7. View the metrics output for the flow using **flow --show** *flowname* .

Flow definition metrics for VM Insight

In general, the same type of statistics are displayed for VM flows as other Flow Monitor flows, but statistics are tracked by entity ID instead of port ID. The following metrics are reported from VM Insight flows.

Fibre Channel statistics:

- **Frame**: Frame count, rate, and average frame size.
- **Throughput:** Frame count and throughput (bytes per sec).

SCSI I/O statistics: Brief or detailed I/O statistics are reported from VM Insight flows, depending on the platform (fixed-port switch or chassis) and the configured flow monitor:

- **Detailed statistics**: Classified by the type of operation (READ/WRITE) and size of the data block.
- **Brief statistics**: Output for learning flows defined with an asterisk (*) for a target device. Only maximum and average I/O data are reported for the last 6 seconds and the lifetime of the flows. Statistics for various sizes of data blocks are not reported.
- **I/O performance statistics**: I/O count and I/O per second (IOPS). Includes average and maximum data. Latency is measured from 6 seconds to the lifetime of flow.
- **I/O latency statistics**: Queue depth. Command status latency and command-first response latency. Includes average and maximum data. Latency is measured from 6 seconds to the lifetime of flow.

The following is a summary of Fibre Channel and I/O statistics displayed for VM Insight flow types by chassis and fixed-port switch type.

TABLE 22 Statistics displayed for VM Insight flows

Flow type	Create command	Fibre Channel	I/O statistics	I/O performance	I/O latency
User-defined static	**flow --create** *flowname* **-feature monitor -ingrport** *Entity_ID* **-srceid** *Source_ID* **-dstdev** *Destination_ID*	**Chassis and fixed-port switch:** frame count, frames per second, bytes count, and throughput per average frame size.	**Chassis and fixed-port switch:** I/O count, I/O per second (IOPS), I/O bytes transferred, and I/O bytes per second (IOPS) for read commands, write commands, and total commands.	**Chassis switch:** Read and write I/O size, I/O count, maximum IOPS, and average IOPS. **Fixed-port switch:** Statistics not displayed.	**Chassis switch:** Read and write command status time, read command-first data time, write command first XFER_RDY time, and read and write command pending I/Os per I/O size, maximum latency, and average latency. **Fixed-port switch:** Statistics not displayed.
User-defined learning	**flow --create** *flowname* **-feature monitor -srceid** *Source_ID* **-dstdev "*"**	**Chassis and fixed-port switch:** frame count, frames per second, bytes count, and throughput per average frame size.	**Chassis switch:** Brief statistics are displayed for each destination device including device ID, read and write command maximum completion and first response time, average IOPS, and maximum pending IOs (queue depth). **Fixed-port switch:** Statistics not displayed.	**Chassis and fixed-port switch:** Statistics not displayed.	**Chassis and fixed-port switch:** Statistics not displayed.
Predefined learning	Flow create is not required. Activate predefined flow using **sys_mon_all_vms**.	**Chassis and fixed-port switch:** frame count, frames per second, bytes count, throughput, and average frame size for each entity ID.	**Chassis switch:** Brief statistics are displayed for each destination device including device ID, read and write command maximum completion time, maximum first response time, pending I/Os per second, and maximum IOPS. **Fixed-port switch:** Statistics not displayed.	**Chassis and fixed-port switch:** Statistics not displayed.	**Chassis and fixed-port switch:** Statistics not displayed.

VM Insight limitations and considerations

The following are limitations and considerations for using VM Insight.

- The application server is only supported by Brocade GEN 6 platforms running Fabric OS 8.1.0. It will not attempt inter-switch communication with Brocade GEN 4 or Brocade GEN 5 platforms.
- An entity can be registered at multiple N_Ports within a fabric.
- The application server supports up to 1,008 entity ID registrations per PID. This limit is not configurable. A message is issued when the limit has been reached.
- Flows to monitor VM traffic using VM Insight can only be configured on Brocade Gen 6 switches using Fabric OS 8.1.0.
- Brocade Gen 4 and Brocade Gen 5 devices can reside in the fabric to pass through VM traffic tagged with the application header, but flows cannot be monitored at these devices.
- VM Insight monitors can be configured on each switch where a VM flow can enter the fabric. VM Insight monitors can only be configured on a per-switch basis.
- Flow monitoring is only supported on ingress traffic from F_ports.
- If downgrading from Fabric OS 8.1.0, any defined flows will be automatically removed. Flow data cannot be obtained from the downgraded platform.
- A single-mode firmware download (firmwaredownload -s) will cause devices to log out of the local switch and disrupt traffic. A nondisruptive firmware downgrade will occur if local devices have logged out of the local switch or their VMs have deregistered from the application server before the firmware download. To avoid disruption before a download, you can perform one of the following actions:
  - Disable ports connected to the local devices.
  - Disable the VM Insight feature on the HBA.
  - Issue a **firmwaredownload -s** command, which causes all devices to log out and log back in.
- If a fabric hosts VMs with more than one instance of a hypervisor, traffic from different VM instances might propagate in the fabric with duplicate entity IDs. The fabric does not attempt to resolve this conflict and will treat exchanges with the same entity ID as if from a single VM.
- VM Insight can only monitor an entity ID from a single source or destination. It can track VMs or containers, but not both at the same time for the same PID. In cases where a VM hosts one or more containers, and all are registering for the same entity ID, your HBA, hypervisor, or container application must determine which entity ID should be placed in a frame. Refer to your application documentation for information or configuration instructions.
- For trunked F_Ports, create and activate user-defined flows on all ports of the trunk group.
- Devices connecting to a fabric using the Access Gateway can register with the application server of the hosting Fabric OS switch.

Extension platform support for VM Insight

The following table summarizes support on Brocade extension platforms for the application header. Note the following about the indicated support:

- **No support:** The header will not properly forward from the tunnel out of the FC egress port.
- **Pass-through:** Data containing the header should flow through the tunnel without modification.
- **Supported:** FCIP will correctly generate emulation sequences that include the established header.

TABLE 23 Extension platform support for VM Insight

Tunnel type	Brocade 7800 and FX8-24 (all Fabric OS levels)	Brocade 7840 and SX6 (Fabric OS before 8.1.0)	Brocade 7840 and SX6 (Fabric OS 8.1.0)
FCP emulating	No support	No support	Supported
FICON emulating	No support	No support	No support
Nonemulating	Pass-through	Pass-through	Pass-through

Adding SRR frame type for flow monitoring

Flow Monitor provides support for error detection and recovery through the Sequence Retransmission Request (SRR) frame type.

The traffic in Fibre Channel fabrics is susceptible to link errors, CRC errors, and timeouts that can affect application response times. To mitigate the effects of these errors, FC-Tape backup, Read Exchange Concise (REC), and SRR are used for error detection and recovery. The amount of time to recover from a link error at the FCP level is less than the time needed for FC-Tape backup applications. However, the Gen 6 I/O statistics module does not detect the error recovery operation. To identify these conditions, you must add an SRR frame type.

The SRR frame type has the following limitations:

- The SRR frame type is supported only on Gen 5 and Gen 6 platforms.
- It is supported only on an F_Port.

To add the SRR frame type, enter **flow --create srr21 -feature mon** , and specify the source device, destination device egress port, and frame type. The following example creates the SRR frame type for Flow Monitor.

```
switch:admin> flow --create srr21 -feature mon -srcdev 0x01ba00
                             -dstdev 0x01c000 -egrport 21 -frametype srr

switch:admin>  flow --show srr21
==============================================================================================================
============
Name      : srr21 Features: mon(Active)
Definition: IngrPort(21),FrameType(SequenceRetransmissionRequest)


Flow Stats (Active):
Stats time:  | Thu Jan 14 04:22:30 UTC 2016 |
----------------------------------------------------------
----------------------------------------
| Rx Frames Count | Rx Frames per Sec. |
----------------------------------------
|        5       |          1         |
----------------------------------------
```

Legacy use case monitoring

You can create monitoring flows in Flow Vision that provide similar functionality to those available through Advanced Performance Monitor (APM). The following examples cover creating Flow Monitor equivalents for an end-to-end monitor, a frame monitor, and an ingress or egress Top Talker monitor.

Creating an end-to-end monitor equivalent

You can use the **-bidir** keyword with the **flow** command to create the equivalent to an end-to-end monitor.

The following example creates a bidirectional Flow Monitor flow between device 02d8c0 and device 023a00 egressing port 4/10 of the switch on which the command is running.

```
switch:admin> flow --create e2eflow -feature monitor -egrport 4/10
                                    -dstdev 023a00 -srcdev 02d8c0 -bidir

switch:admin> flow --show e2eflow
==========================================================================
Name      : e2eflow      Features: mon(Activated)      noConfig: Off
Definition: EgrPort(4/10),SrcDev(0x02d8c0),DstDev(0x023a00),BiDir
Flow Monitor (Activated):
Monitor time:  | Fri Apr 15 22:52:35 UTC 2016 |
--------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------
|    Frame Count       | Frames Per Sec.     |    Byte count         | Throughput(Bps)     |Frame      |
|                      |                     |                       |                     |Size(Bytes)|
|   Tx  /   Rx   / Total |  Tx  /    Rx / Total | Tx  /    Rx   / Total  |  Tx  /    Rx  / Total| Tx  / Rx  |
--------------------------------------------------------------------------------------------------------
|65.11M/129.63M/194.75M|962.52k/653.84k/1.61M|99.49G/185.44G/284.93G|958.32M/957.77M/1.87G|1632 /1528 |
--------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------
|      I/O Count        |   I/O Per Sec.(IOPS)     | I/O bytes Transferred    |   I/O bytes Per Sec.   |
| Reads / Writes/ Total  | Reads / Writes/ Total     | Reads / Writes/ Total     | Reads / Writes/ Total  |
--------------------------------------------------------------------------------------------------------
|  3.19M/ 24.14M/ 27.33M | 481.13k/481.13k/962.27k | 176.56G/  1.30T/ 1.47T   | 2.61G/ 2.61G/ 1.22G    |
--------------------------------------------------------------------------------------------------------
====================================================================================================
```

Creating a frame monitor equivalent

You can use the **-frametype** parameter as part of the **flow** command to create the equivalent to an APM monitor created using the **fmmonitor** command.

The following example creates a Flow Monitor flow that counts SCSI Read-Write (scsirw) frames egressing port 2 of the switch on which the command is running.

```
switch:admin> flow --create scsirw -feature monitor -egrport 2 -frametype scsirw
Monitor feature(s) have been activated.

switch:admin> flow --show
---------------------------------------------------------------------------------------------------
Flow Name    |  Feature     |    SrcDev        |     DstDev    |IngrPt|EgrPt|BiDir| LUN   | FrameType |
---------------------------------------------------------------------------------------------------
f1          |mon          |030300           |030400         |3      |-     |no    |0      |-          |
lrn0        |mon+         |*                |*              |0      |-     |no    |-      |-          |
lrn89       |mon+         |*                |*              |89     |-     |no    |-      |-          |
scsirw      |mon+         |-                |-              |-      |2     |no    |-      |srdwr      |
---------------------------------------------------------------------------------------------------

switch:admin> flow --show scsirw
============================================================================
Name      : scsirw      Features: mon(Activated)        noConfig: Off
Definition: EgrPort(2),FrameType(srdwr)
Flow Monitor (Activated):
Monitor time:  | Fri Apr 15 23:31:04 UTC 2016 |
----------------------------------------------------------
-----------------------------------------
| Tx Frames Count | Tx Frames per Sec.  |
-----------------------------------------
|      10.27M     |       327.55k       |
-----------------------------------------
============================================================================
```

NOTE
"0x0f" is the bitmask to monitor SOFx. Use "0x0f" as bitmask while creating user defined frame type to monitor SOFx.

Creating an ingress or egress Top Talker monitor equivalent

You can use the learning flow (`"*"`) parameter to create the equivalent to a legacy Top Talker monitor. A Top Talker monitor is used to identify high-volume flows passing a port.

The following example creates both an ingress and an egress Top Talker monitor. The first command creates a Flow Monitor learning flow named "ingresstt" for all frames between any devices ingressing through port 41 of the switch on which the command is running. The second command creates a Flow Monitor learning flow named "egresstt" for all frames between any devices egressing through port 30 of the switch on which the command is running.

```
switch:admin> flow --create ingresstt -feature monitor -ingrport 41 -srcdev "*" -dstdev "*"
Monitor feature(s) have been activated.

switch:admin> flow --create egresstt -feature monitor -egrport 30 -srcdev "*" -dstdev "*"
Monitor feature(s) have been activated.

switch:admin> flow --show
-------------------------------------------------------------------------------------------
Flow Name      | Feature     | SrcDev    | DstDev    |IngrPt|EgrPt|BiDir| LUN   | FrameType|
-------------------------------------------------------------------------------------------
ingresstt      |mon+         |*          |*          |41    |-    |no   |-      |-         |
egresstt       |mon+         |*          |*          |30    |-    |no   |-      |-         |
-------------------------------------------------------------------------------------------
  + Denotes feature is currently activated for the flow
=========================================================================

switch:admin> flow --show ingresstt
=========================================================================
Name      : ingresstt    Features: mon(Activated)        noConfig: Off
Definition: IngrPort(41),SrcDev(*),DstDev(*)
Flow Monitor (Activated):
Monitor time:  | Fri Apr 15 23:44:58 UTC 2016 |
-------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
|SID(*)|DID(*)|Rx Frames Cnt|Rx Frames    |Rx Bytes Cnt|Rx Throughput(Bps)| Avg Rx Frm Sz(Bytes)|
|      |      |             | per Sec.    |            |                  |
|
----------------------------------------------------------------------------------------------------
|012900|051e00|   3.97M     | 112.68k     |   5.68G    |    165.06M       |         1536        |
|012900|020600|   7.95M     | 225.64k     |  15.09G    |    438.13M       |         2036        |
|012900|010700|   4.52M     | 128.62k     |   4.36G    |    127.08M       |         1036        |
|012900|020900|   7.90M     | 224.21k     |  11.31G    |    328.43M       |         1536        |
----------------------------------------------------------------------------------------------------
|  *   |  *   |  24.36M     | 691.16k     |  36.44G    |      1.03G       |         1606        |
----------------------------------------------------------------------------------------------------

switch:admin> flow --show egresstt
=========================================================================
Name      : egresstt     Features: mon(Activated)        noConfig: Off
Definition: EgrPort(0/30),SrcDev(*),DstDev(*)
Flow Monitor (Activated):
Monitor time:  | Fri Apr 15 23:25:35 UTC 2016 |
-------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------
|SID(*)|DID(*)|Tx Frames Cnt|Tx Frames    |Tx Bytes Cnt|Tx Throughput(Bps)| Avg Tx Frm Sz(Bytes)|
|      |      |             | per Sec.    |            |                  |
|
----------------------------------------------------------------------------------------------------
|022900|051e00|   7.99M     |   1.86k     |    9.19G   |        2.19M     |         1236        |
|012c00|051e00|   4.50M     |   1.04k     |    4.35G   |        1.03M     |         1036        |
|022800|051e00|   7.99M     |   1.86k     |    7.71G   |        1.83M     |         1036        |
|012a00|051e00|   2.25M     |     524     |    3.22G   |      787.35k     |         1532        |
```

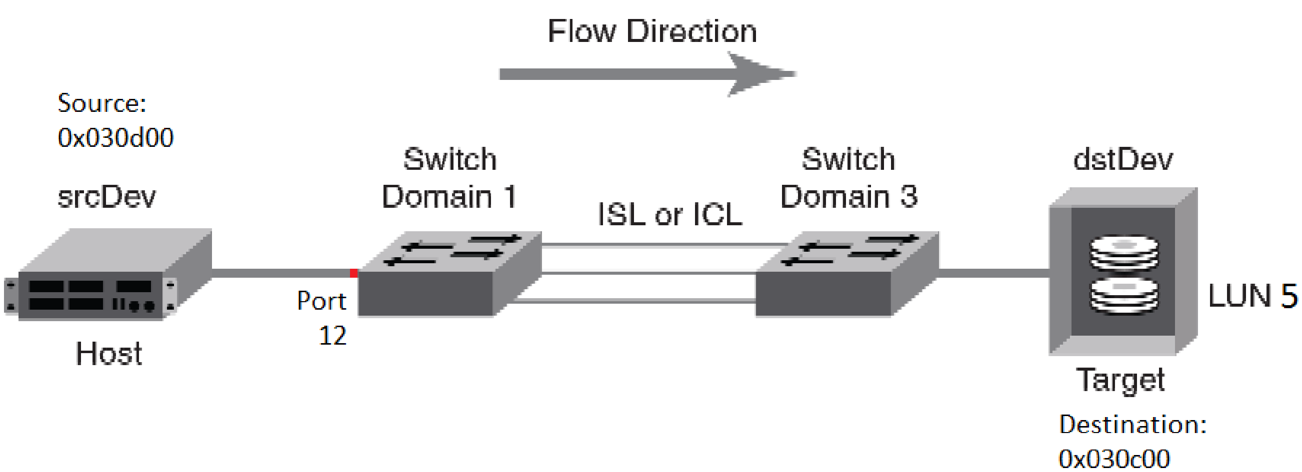
```
|022a00|051e00|   7.38M     |   1.71k     |      14G   |         3.33M      |         2032        |
|012b00|051e00|   4.50M     |   1.04k     |    4.35G   |         1.03M      |         1036        |
|012900|051e00|   2.25M     |     524     |    4.27G   |         1.01M      |         2032        |
-------------------------------------------------------------------------------------------------
|  *   |  *   |  36.89M     |   8.58k     | 47.11G     |        11.23M      |         1368        |
-------------------------------------------------------------------------------------------------

Note: Statistics are provided for the aggregate traffic generated to the specified SIM-port.
No traffic is actually transmitted out on the SIM-port.
=======================================================================
```

Monitoring LUN-level statistics

A common use of flow monitors is to monitor traffic flowing from a particular ingress port to a specified LUN.

The following example creates a flow named "lunFlow11" that monitors traffic ingressing on port 12 between source device 0x030d00 and destination device 0x030c00 using LUN 5, and then displays the results of that flow. The figure provides an illustration of what is happening in the example.

FIGURE 24 A LUN monitoring flow




```
switch:admin> flow --create lunFlow11 -feature monitor
                    -ingrport 12 -srcdev 0x030d00 -dstdev 0x030c00 -lun 5

switch:admin> flow --show lunFlow11
Name      : lunflow11  Features: mon(Activated) noConfig: off
Definition: IngPort(12),SrcDev(0x030d00),DstDev(0x030c00), LUN(5)

Flow Monitor (Activated):
Monitor time:  | Tue Mar 15 13:09:51 UTC 2016 |
-----------------------------------------------
----------------------------------------------------------------------------------------------
|      I/O Count         |   I/O Per Sec.(IOPS)   | I/O bytes Txfrd      | I/O bytes Per Sec.   |
| Reads / Writes/ Total  | Reads / Writes/ Total  | Reads/Writes/ Total  | Reads /Writes/ Total |
----------------------------------------------------------------------------------------------
|356.33M/356.33M/712.67M| 16.91k/ 16.28k/ 33.20k| 3.99T/ 3.99T/ 7.98T | 99.11M/95.42M/194.54M|
----------------------------------------------------------------------------------------------


-------------------------------------------------------------------------------------
| I/O Performance:                                                                  |
-------------------------------------------------------------------------------------
```

```
|      Metric    |   IO Size    |  I/O Count   |  Max(IOPS)   |       Avg(IOPS)            |
|                |              |       All    |   All        |   6 sec   /      All       |
-------------------------------------------------------------------------------------------
| RD IO Count    | <8K          |    66.37k    |  8.59k       |    2.47k     /      2.13k  |
|                | 8K - <64K    |         0    |      0       |        0     /          0  |
|                | 64K - <512K|          0    |      0       |        0     /          0  |
|                | >=512K       |         0    |      0       |        0     /          0  |
|                | ALL          |    66.37k    |  8.59k       |    2.47k     /      2.13k  |
-------------------------------------------------------------------------------------------
| WR IO Count    | <8K          |    66.83k    |  8.70k       |    2.43k     /      2.14k  |
|                | 8K - <64K    |         0    |      0       |        0     /          0  |
|                | 64K - <512K|          0    |      0       |        0     /          0  |
|                | >=512K       |         0    |      0       |        0     /          0  |
|                | ALL          |    66.83k    |  8.70k       |    2.43k     /      2.14k  |
-------------------------------------------------------------------------------------------

------------------------------------------------------------------------------------------
| I/O Latency Metrics:                                                                   |
------------------------------------------------------------------------------------------
|           Metric      |  IO Size  |            Max            |          AVG           |
|                       |           |  6 sec   /      All       |  6 sec   /     All     |
------------------------------------------------------------------------------------------
| RD CMD -> Status      | <8K       |    68u   /      2.86m     |    39u   /        39u  |
|           Time        | 8K - <64K |     0u   /         0u     |     0u   /         0u  |
|                       | 64K - <512K|    0u   /         0u     |     0u   /         0u  |
|                       | >=512K    |     0u   /         0u     |     0u   /         0u  |
|                       | ALL       |    68u   /      2.86m     |    39u   /        39u  |
------------------------------------------------------------------------------------------
| WR CMD -> Status      | <8K       |    77u   /      2.84m     |    44u   /        44u  |
|           Time        | 8K - <64K |     0u   /         0u     |     0u   /         0u  |
|                       | 64K - <512K|    0u   /         0u     |     0u   /         0u  |
|                       | >=512K    |     0u   /         0u     |     0u   /         0u  |
|                       | ALL       |    77u   /      2.84m     |    44u   /        44u  |
------------------------------------------------------------------------------------------
| RD CMD -> 1st Data    | <8K       |    47u   /      2.83m     |    21u   /        20u  |
|           Time        | 8K - <64K |     0u   /         0u     |     0u   /         0u  |
|                       | 64K - <512K|    0u   /         0u     |     0u   /         0u  |
|                       | >=512K    |     0u   /         0u     |     0u   /         0u  |
|                       | ALL       |    47u   /      2.83m     |    21u   /        20u  |
------------------------------------------------------------------------------------------
| WR CMD -> 1st         | <8K       |    55u   /      2.80m     |    19u   /        19u  |
|           XFER_RDY    | 8K - <64K |     0u   /         0u     |     0u   /         0u  |
|           Time        | 64K - <512K|    0u   /         0u     |     0u   /         0u  |
|                       | >=512K    |     0u   /         0u     |     0u   /         0u  |
|                       | ALL       |    55u   /      2.80m     |    19u   /        19u  |
------------------------------------------------------------------------------------------
| RD Pending IOs        | <8K       |     4    /         4      |     1    /         3   |
|                       | 8K - <64K |     0    /         0      |     0    /         0   |
|                       | 64K - <512K|    0    /         0      |     0    /         0   |
|                       | >=512K    |     0    /         0      |     0    /         0   |
------------------------------------------------------------------------------------------
| WR Pending IOs        | <8K       |     4    /         4      |     1    /         3   |
|                       | 8K - <64K |     0    /         0      |     0    /         0   |
|                       | 64K - <512K|    0    /         0      |     0    /         0   |
|                       | >=512K    |     0    /         0      |     0    /         0   |
------------------------------------------------------------------------------------------
```

Flow Monitor and High Availability

When a High Availability (HA) failover, High Availability reboot, or a power cycle occurs, all flows are deactivated, and statistics for all Flow Monitor flows are not retained. Flow Monitor will begin to gather statistics again when the standby control processor becomes active. After the device is back online, only the first 64 Flow Monitor sub-flows that can be learned are reactivated. Flow Monitor always recreates these sub-flows based on the order in which the switch learns the flows.

Refer to High Availability and Flow Vision on page 45 for more information.

Flow monitors and MAPS

Flow Monitor statistics can be used by the Monitoring and Alerting Policy Suite (MAPS) service. This can help you identify critical administrative information such as traffic patterns, bottlenecks, and slow drains. Refer to the "Importing flows" section in *Brocade Monitoring and Alerting Policy Suite Configuration Guide* for more details.

Flow monitors on Access Gateways

Access Gateways support flow monitoring on ingress F_Ports only. The CLI and outputs are exactly the same as for a switch.

NOTE
Enabling WWN device ID mode is blocked on Access Gateways. This means that the **-deviceIdMode WWN** keyword is not permitted as part of the **flow** command, so the device ID mode always remains “PID” on Access Gateways.

Flow Monitor limitations

The following limitations apply to all Flow Monitor flows:

- Only one active learned flow is supported per ASIC.
- Learning is supported only on Gen 5 and Gen 6 Fibre Channel platforms.
- The frame type parameters scsiread, scsiwrite, and scsirdwr monitor only SCSI 6-, 10-, 12-, and 16-bit Read and Write values. Read Long and Write Long values cannot be monitored.
- The predefined learning flow, sys_mon_all_fports, cannot monitor SIM-port traffic on Gen 6 platforms.
- Flow Monitor is not supported on ports with Encryption or Compression enabled.
- Flow Monitor is not supported on Gen 6 ingress traffic for local SIM-port destination device.
- IFL flows can be monitored only on EX_Ports in a Fibre Channel router.
- Flow Monitor cannot monitor Inter-Fabric Link (IFL) flows on E_Ports or F_Ports.
- Flow Monitor cannot monitor flows that are using frame redirection for encryption.
- Flow Monitor flows cannot be converted to Fabric OS 7.1.x flow performance monitors.
- Flow Monitor does not support monitoring frames that are mirrored to a local or remote mirror port.
- Flow Monitor is not supported on AE ports.
- Flows based on E_Ports cannot be monitored at the same time as flows based on EX_Ports.
- The calculated Rx and Tx frame size values displayed in the output are accurate within a range of -4 through +8 bytes. For example, a frame size value of 256 bytes may actually be anywhere from 252 to 260 bytes in size.
- For Flow Monitor flows passing through the base switch in a VF-enabled fabric the source fabric ID (SFID) and destination fabric ID (DFID) values must be specified when the flow is defined.
- When defining flow monitoring on E_Ports, neither the source device for the ingress port flow nor the destination device for the egress port flow should match the port PID of the E_Port (otherwise the flow will be activated but not enforced). The PIDs for the flow definition should be fport or npiv PIDs.

The following limitations apply to Gen 5 and Gen 6 Fibre Channel platforms:

- They support a maximum of two flows defined using a combination of ingress port and frame type parameters per ASIC chip.
- Each port supports a maximum of 12 flows defined using both egress port and frame type parameters.

- Flow Monitor can only monitor flows that are using EX_Ports.

The following limitations apply to 8 Gbps-capable Fibre Channel platforms and blades:

- They do not support monitoring flows using both ingress port and frame type parameters.
- Each port supports a maximum of 12 flows defined using both egress port and frame type parameters, except for the Brocade 300, 5300, 5410, 5424, 543x, 5450, 5460, 5470, 5480, and 7800 platforms, which support a maximum of eight flows per port.
- They cannot show statistics for SIM ports.
- They do not support learning.

The following limitations apply to 16 Gbps-capable switches:

- You **can** mirror a uni-directional flow from a 16 Gbps-capable device port to a 16 Gbps mirror port.
- You **cannot** mirror a bi-directional flow from a 16 Gbps-capable device port to a 16 Gbps mirror port.

  You must speed lock the 16 Gbps-capable device to 8 Gbps or slower so that the device port physical speed is less than or equal to half of the mirror port's physical speed. It is the physical port speeds that is important, not the amount of bandwidth that the device is actually using.

Flow Generator

Overview of Flow Generator

Flow Generator is a test traffic generator that allows you to pre-test a SAN infrastructure (including internal connections) for robustness before deploying it.

CAUTION
You should not use Flow Generator in an active production environment, because the traffic created by Flow Generator can saturate the links, which will have a negative impact on any other traffic sharing the same links.

Flow Generator lets you perform the following:

- Configure a 16 Gbps Gen 5 or 32 Gbps Gen 6 Fibre Channel-capable port as a simulated device that can transmit frames at a full 16 Gbps or 32 Gbps line rate.
- Emulate a 32 Gbps SAN without actually having any 32 Gbps hosts or targets or SAN-testers.
- Pre-test the entire SAN fabric at the full line rate, including optics and cables on ISLs as well as internal connections within a switch.

Flow Generator uses simulation mode (SIM) ports. SIM ports behave like normal F_Ports, but are used only for testing. By using SIM ports, Flow Generator traffic is terminated at the destination port and does not leave the fabric. Refer to SIM port attributes and configuration on page 114 for more information on SIM ports.

Flow Generator can generate standard frames or create custom frames with sizes and patterns that you specify. A sample use case would be to create a traffic flow from a Source ID (SID) to a Destination ID (DID) to validate routing and throughput. Creating a flow from a specific source ID to a specific destination ID on page 110 provides an example of the command and the results for this use case.

Flow Generator supports predefined flows to generate traffic between all configured SIM ports. Fabric OS 7.3.0 and later support 39 Virtual Channels (VCs) and balance the traffic generated from a SIM port to multiple destinations by transmitting traffic using the same number of VCs for each destination.

After you activate a Flow Generator flow, the flow stays active until you deactivate the flow. The flow stays active even though the SID and DID SIM-ports are offline. As soon as SID and DID SIM-ports are online, traffic starts.

Flow Vision also supports generating simulated traffic to a remote fabric. To generate traffic, you specify the imported proxy ID or the real PWWN of the device. Flow Generator support is similar to Flow Monitor support, except Flow Generator only supports flows from edge to edge but not from backbone to edge or from edge to backbone. For additional information, refer to Monitoring Fibre Channel router fabrics using proxy IDs on page 67.

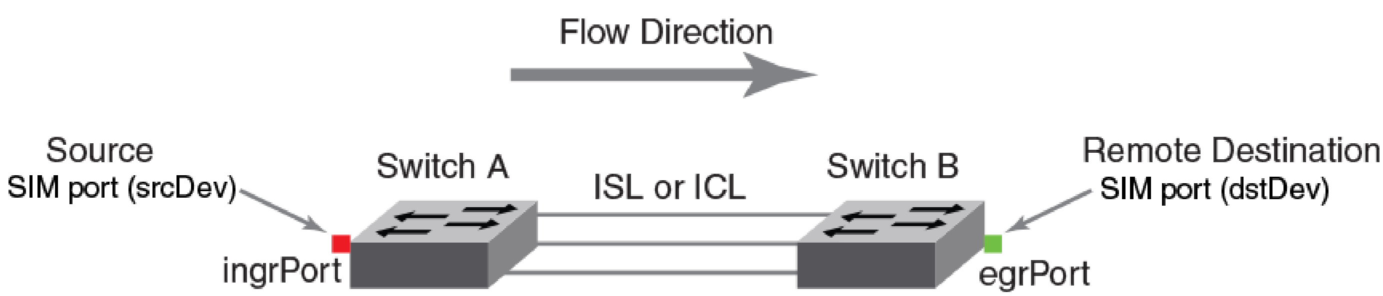
Flow Generator setup

Flow Generator offers several flow control options that you can configure, including the ability to specify both the frame size and the frame payload pattern. Header parameters and other control parameters can also be added as part of the definition. The OXID value for frames cannot be user-specified.

Flow Generator generates and receives traffic only from simulated ingress and egress ports (SIM ports) which emulate device entries in the Name Server database, so that they are treated as real devices and can be used to evaluate various switch and fabric operations such as QoS and Traffic Isolation. For more information on working with SIM ports, refer to SIM port attributes and configuration on page 114.

Flow Generator flows are defined using a combination of the source device (srcdev), destination device (dstdev), ingress port (ingrport), and egress port (egrport) parameters. All of these must be SIM ports. The source device is the origination point of the test traffic. The destination device is the destination of the test traffic; for Flow Generator flows it may be remote from the switch. The port that transmits the simulation traffic must be a 16 Gbps-capable or 32 Gbps-capable Fibre Channel port. The port that receives the simulated traffic can be either an 8 Gbps-capable, 16 Gbps-capable, or 32 Gbps-capable Fibre Channel port. The following figure illustrates this concept.

FIGURE 25 A Flow Generator flow structure



Predefined Flow Generator flows

Predefined flows are system-defined flows that support most common testing scenarios. These flows are automatically created and defined based on your network structure as part of the upgrade to Fabric OS 7.3.0 or later. These flows use the port ID and device ID modes and other control parameters configured for Flow Generator.

Currently Fabric OS supports one predefined flow named sys_gen_all_simports. In this flow, traffic is generated from each SIM port on a switch to all the other SIM ports configured on that switch. This allows you to easily stress-test the hardware, ASIC path, backplanes, front-end, switch, and fabric operations. The stress-testing operation is handled in iterations. At each iteration, all SIM ports generate traffic to a set of destination ports, determined by their order in the physical chip interleave, so that all the chips are tested at the beginning of the test. In this flow, every SIM port sends traffic to four destinations using nine Virtual Channels (VCs) for each destination, but does not generate traffic to the originating port. This uses 36 VCs; the remaining three VCs are not used by the flow.

You can configure this flow to run on all the ports on the switch, or on a specific slot, port range, or list of ports.

- To run the test on all the ports on the local switch, set all the ports on the switch to SIM ports.
- To run the test on one slot only, set only the ports of the slot to SIM ports.
- To run the test on any other set of ports, set these specific ports to SIM ports.

The frame payload size and pattern control parameters apply to sys_gen_all_simports. The control parameters can be the default, user-defined default, or specific to the sys_gen_all_simports flow. Refer to Customizing Flow Generator flows on page 109 for more details.

To start this flow, you must first enable it (specifying the ports you want used), and then you must activate it. The following example enables SIM ports 10 through 20 and then activates the flow.

```
switch:admin> flow --control -simport 10-20 -enable
```

```
switch:admin> flow --activate sys_gen_all_simports -feature generator
```

NOTE
Once started, sys_gen_all_simports will run until you deactivate the flow.

To stop the flow, enter **flow --deactivate sys_gen_all_simports -feature generator**.

To reset the flow metrics, enter **flow --reset sys_gen_all_simports -feature generator**. This resets the ingress and egress counts for all the tested SIM ports.

The following restrictions apply specifically to this stress-test flow:

- You must configure the SIM ports you are going to test before activating the flow.
- There is a limit of 256 flows (64 flows at four flows per port) that can be generated per iteration.
- This flow bypasses all zoning, as SIM ports do not need to be zoned for the flows to be generated.
- Activating the predefined flow is mutually exclusive with all other active Flow Generator flows. If there are any static flows or sub-flows activated on the switch, you cannot activate the predefined flow. If the predefined flow is activated, no other Flow Generator flow can be activated.

Entering **flow --show sys_gen_all_simports** displays:

- The SID-DID pairs tested
- The number of complete runs where all the SID-DID pairs have been tested
- The completion percentage of the current run
- The total number of frames generated from all the ingress ports

The following example shows the typical results of the **flow --show** command when the sys_gen_all_simports flow is active:

```
switch:admin> flow --show
--------------------------------------------------------------------------------------
Flow Name           | Feature   | SrcDev | DstDev |IngrPt|EgrPt |BiDir| LUN |FrameType|
--------------------------------------------------------------------------------------
f1                  |gen+,mon+ |*       |-       |-     |5     |no   |-    |-        |
sys_gen_all_simports|gen+      |*       |*       |-     |*     |*    |-    |-        |
--------------------------------------------------------------------------------------
  + Denotes feature is currently activated for the flow
  The flow name with prefix sys_ denotes predefined flow
```

The following example shows the results of **flow --show sys_gen_all_simports**.

```
switch:admin> flow --show sys_gen_all_simports
===============================================================================
Name      : sys_gen_all_simports   Features: gen(Activated)      noConfig: Off

Number of complete runs: 2
Percent complete of the current run: 33

Flow Generator (Activated):
-----------------------
|  SrcDev   |  DstDev  |
-----------------------
| 0x014000 | 0x014100 |
-----------------------
| 0x014000 | 0x014200 |
-----------------------
| 0x014000 | 0x014300 |
-----------------------
| 0x014000 | 0x014400 |
-----------------------
| 0x014100 | 0x014000 |
-----------------------
| 0x014100 | 0x014200 |
-----------------------
| 0x014100 | 0x014300 |
-----------------------
| 0x014100 | 0x014400 |


    (output truncated)

Number of frames generated from IngrPorts :   9.20G
```

Notes on predefined flows

The following items should be kept in mind when working with predefined flows:

- If the sys_gen_all_simports flow was active before a reboot, it will be replayed after the reboot.
- If the port, SIM device, slot, or switch goes offline, the subflows on the offline ports will be stopped. Traffic will continue to run on the online ports.
- If all SIM ports in the test go offline, the flow will stay active but traffic will not start, and entering **flow --show** will return "no sim devices". In this case, the flow traffic will resume when any SIM ports in the flow come online.
- To prevent dropped frames when offline SIM ports come back on line, those ports included in the test flow are only added back to the test flow during the pause between iterations.
- Activation, deactivation, reset, show, and control operations are supported for predefined flows, but creation and deletion operations are not supported.
- In those cases where the device ID (DID) is equal to the source (SID), that DID will be skipped and the next DID (from the next iteration) will be selected to run. This way, every SID will send traffic to four destinations at a time for every iteration.

Refer to Flow Generator and MAPS on page 116 for information on how sys_gen_all_simports behaves with MAPS, and Flow Generator and High Availability on page 116 for information on how sys_gen_all_simports behaves in HA situations.

Determining how long one pass of sys_gen_all_simports should take

The following calculation can be used to estimate how long one full pass should take if all the possible pairings of Source ID and Destination ID are tested. There are always four flows per SIM port, and each iteration takes one minute. The number of iterations that equals a full run is the determined by taking the number of SIM ports, subtracting one, and then dividing the remainder by four. This value is rounded up to the next whole number. The number of iterations needed to complete one full pass is the number of iterations plus 1, and the time needed to run one full pass is the number of iterations for a full run multiplied by the time needed for each iteration. Therefore, the time needed to run a full pass of sys_gen_all_simports for 128 SIM ports is: `{[(128-1)/4]+1} * 1`, or 33 minutes.

In other words, subtracting 1 from the port count is 127. Dividing this value by 4 (the number of flows per port) produces 31.75. Rounding this value up to the next whole number makes it 32. Adding 1 produces a full-pass iteration count of 33. Multiplying this by the iteration time (one minute) results in the time for one pass of sys_gen_all_simports in minutes (33).

Creating Flow Generator flows

To create a Flow Generator flow, enter the **flow --create** *flow_name* **-feature generator** *parameters* command.

The following table lists the parameters available to Flow Generator.

TABLE 24 Flow Generator supported flow parameter combinations

Parameter	Field name	Description
Port	ingrport egrport	• One field only must be specified. • Values must be explicit. • Must be a SIM port local to the switch for a flow to generate traffic.
Frame	srcdev dstdev	• At least one field must be specified. • Values for srcdev and dstdev can be explicit or `"*"` (`"*"` indicates learned flows). • Must be a SIM port PID for a flow to generate traffic. • The parameters lun and frametype are not supported.
Optional keyword parameters		
	-bidir	Not supported directly. To emulate this function you must create two flows (one in each direction).
	-noactivate	Adding this keyword creates the flow without activating it.
	-noconfig	Adding this keyword creates the flow without saving the flow to the configuration.

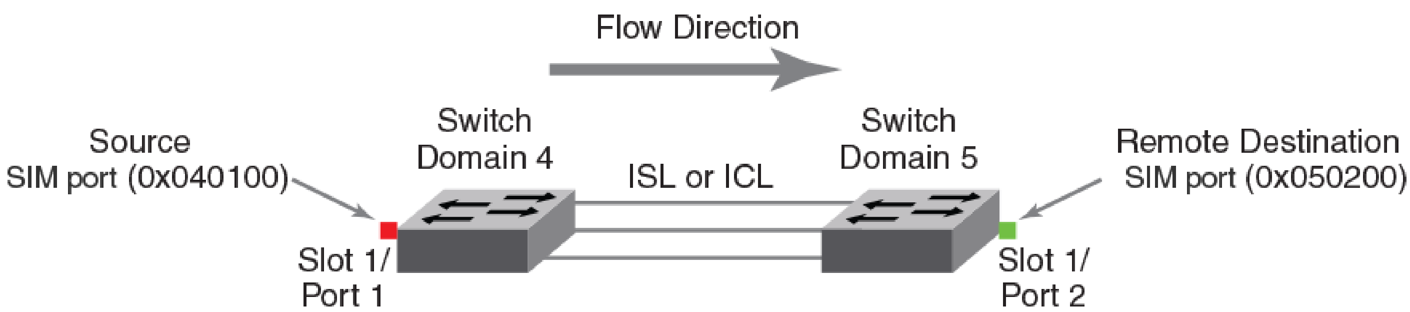
Refer to Customizing Flow Generator flows on page 109 for information on changing the flow frame size and contents.

Parameter usage exceptions

When you create a flow, it is automatically activated unless you use the **-noactivate** keyword for the **flow --create** command. Refer to Creating an inactive flow in Flow Generator on page 106 for an example.

The following illustrated example creates a Flow Generator flow named “Flow11” and generates traffic using the ingress SIM port 1/1 from device 040100 (Domain 4) to device 050200 (Domain 5).

FIGURE 26 A Flow Generator flow between two switches




```
switch123:admin> flow --create Flow11 -feature generator -ingrport 1/1 -srcdev 040100 -dstdev 050200
```

Restrictions

The following restrictions apply to Feature Generator parameter usage:

- If the srcdev and ingrport parameters are both defined, they must both be local to the switch and refer to the same source.
- If the dstdev and egrport parameters are both defined for the flow, they must both be local to the switch and represent the same destination.

  ATTENTION
  Flow creation is not allowed if Advanced Performance Monitor (APM) or Port Mirroring is enabled. Similarly, APM and Port Mirroring-related operations will not be allowed if any flow (active or defined) is present on the switch.

Creating an inactive flow in Flow Generator

To create an inactive Flow Generator flow, enter **flow --create** *flow_name* **-feature generator** *flow_parameters* **-noactivate**.

Refer to Activating Flow Generator flows on page 106 for information on activating a Flow Generator flow.

The following example creates an inactive Flow Generator flow named "superflow238" from device 020a00 to device 01c000 ingressing through SIM-Port 110.

```
switch:admin> flow --create superflow238 -feature generator -ingrport 110 -srcdev 0x020a00 -dstdev 0x01c000
-noactivate
```

Activating Flow Generator flows

When a flow is activated, traffic is generated by the ingress port or source device for that flow and any sub-flows associated with it as soon as all SIM ports and devices defined in the flow are online.

Activating a flow does not automatically clear the flow statistics for that flow; the existing statistic counters resume counting using the resumed flow data. If you are activating a learned flow, the subflows considered for the flow statistics are refreshed based on the first 39 Source ID-Destination ID SIM device pairs in the zone database that are registered in the Name Server. A flow can be activated that includes SIM ports that are either offline or that have not yet been created. If such a flow is activated, an alert message is displayed, noting that the activated flow is not enforced.

To activate an inactive Flow Generator flow, complete the following steps.

1. Connect to the switch and log in using an account with admin permissions.

2. Enter **flow --activate** *flow_name* **-feature generator**.

The following example activates the Flow Generator flow named "Flow1".

```
switch:admin> flow --activate Flow1 -feature generator
```

Automatic activation of a Flow Generator flow

Flow Generator automatically activates a generated flow under the following conditions:

- On flow creation, unless the flow is being created using the **-noactivate** keyword as part of the **flow** command. Refer to Creating an inactive flow in Flow Generator on page 106 for information on this procedure.
- On slot power-on, if the specified port belongs to the slot being powered on and the flow was active when the slot was powered off.
- On a High Availability (HA) failover, or a HA reboot, if the flow was active when the HA event occurred.

Learning in Flow Generator flows

To apply learning to a Flow Generator flow, use an asterisk inside of quotation marks (**"*"**) to specify the parameter to be learned.

When Flow Generator activates learned flows, it queries the Name Server database to identify source and destination devices that are zoned together. These pairings are not automatically changed if either member of the pair changes zones. If either member of the pair changes zones, you must deactivate the flow and then reactivate it to use the new zone values. Flow Generator allocates the flows per source ID to zoned destination IDs based on the maximum permitted number of flows. Once this limit has been reached, the rest of the destination IDs are not tested. For learned flows, zone enforcement is applied to both the source and destination SIM-Ports. Refer to Numbers of flows supported on page 25 for these limits.

In the following example, the only flows that will be activated are the ones where the destination devices share a zone with the source device (021800) and use ingress port 24. To view the data generated by this flow, refer to Viewing the output of a learned Flow Generator flow on page 108.

```
switch:admin> flow --create fgflow12 -feature generator -ingrport 24 -srcdev 0x021800 -dstdev "*"
```

Flow Generator learning flows on egress ports

Creating an active Flow Generator flow on an egress port starts a Flow Generator flow from the srcdev port to the designated egrport. This is the opposite of what happens when creating a Flow Generator flow on an ingress port. If the value for the srcdev is **"*"**, a Flow Generator flow will be initiated to the egress port from every SIM port on the switch that shares a zone with the egress port. This may result in a large number of flows from a large number of ports going to the egress port; the exact number depends on how many SIM ports share the zone with the egress port and the maximum permitted number of flows for the switch. Refer to Numbers of flows supported on page 25 for these limits.

Viewing Flow Generator flows

To display Flow Generator flows, enter **flow --show** *flowname* **-feature generator**. For root and static flows, this command shows the Source ID-Destination ID pairs and the cumulative frame count on the ingress or egress port specified in the flow definition.

Displaying the status of a single Flow Generator flow

The following example displays the status of the Flow Generator flow named "f2".

```
switch:admin> flow --show f2 -feature generator
================================================================================
Name      : f2       Features: gen(Active) noConfig: Off
Definition: IngrPort(1/9),SrcDev(0x010900),DstDev(0x01c100)
Flow Generator (Activated):
-----------------------
|  SrcDev   |  DstDev  |
-----------------------
| 0x010900 | 0x01c100 |
-----------------------
Number of frames generated from IngrPort : 595.41M
Note: More than 1 flow active on this port.
================================================================================
```

Viewing the output of a learned Flow Generator flow

To view the output of a learned Flow Generator flow, enter **flow --show** *flow_name* **-feature generator**.

When you view the output of a learned flow, the Name line displays the flow name and flow features with their respective states. The Definition line displays the port of interest and the device. In the table under Flow Generator, each row is an individual sub-flow, and the column shows the individual device IDs. The last line displays the number of frames (in units of 1000 (K), 1,000,000 (M), or 1,000,000,000 (G)) that have passed through that port measured from the time the port became active.

The following example shows the output of the Flow Generator flow named "fgflow12":

```
switch:admin>flow --show fgflow12 -feature generator
Name : fgflow12 Features: gen(Active) noConfig: Off
Definition: IngrPort(4),DstDev(*)
Flow Generator (Activated):
-----------------------
| SrcDev   | DstDev   |
-----------------------
| 0x204000 | 0x040700 |
-----------------------
| 0x204000 | 0x040800 |
-----------------------
| 0x204000 | 0x050900 |
-----------------------
| 0x204000 | 0x051000 |
-----------------------
Number of frames generated from IngrPort : 82.21M
```

Notes on displaying the status of a Flow Generator flow

- If you want to see the per-flow frame count on a port with multiple flows, you must include the flow monitoring feature in the flow definition (**flow --create** *flow_name* **-feature generator**,**monitor**).
- Flow Generator will append a note at the bottom of the output if there is more than one static flow or a learned flow active on the port.

Deactivating Flow Generator flows

You can deactivate Flow Generator flows without deleting them. This allows you to have a "library" of flows that you can activate as needed without having to recreate them.

> **NOTE**
> When a flow is deactivated, traffic stops for that flow and any sub-flows associated with it. When a flow is deactivated, the definition remains but Flow Generator does not populate it with traffic.

To manually deactivate a Flow Generator flow, complete the following steps.

1. Connect to the switch and log in using an account with admin permissions.
2. Enter **flow --deactivate** *flow_name* **-feature generator**.

The following example deactivates the Flow Generator flow named "Flow1".

```
switch:admin> flow --deactivate Flow1 -feature generator
```

Customizing Flow Generator flows

Flow Generator allows you to specify the frame payload size and pattern to be used for the Flow Generator flows by using the **flow --control** command.

Frame payload size

Flow Generator allows you to define the frame payload size in bytes. The frame payload size value must be either 0 (which produces frames of random size using a multiple of 4 between 64 and 2048 bytes) or a multiple of 4 in the range from 64 through 2048. (64, 68, 72, 120, 140, 320, 512, and so on). The default payload size value is 2048.

There are two ways to change the frame payload size; you can change the default payload size or you can change the payload size for a single flow.

To see the current generic payload size, enter **flow --show -ctrlcfg**; to see the payload size for a specific flow, enter **flow --show** *flow_name* **-ctrlcfg**.

> **NOTE**
> There are limitations for Flow Generator learning flows and the sys_gen_all_simports flow; refer to Flow Generator limitations and considerations on page 117 for specific details.

Changing the default frame payload size

Assigning a frame payload size without defining a flow creates a default that applies to all flows created afterward. The following example sets the payload size default for all new flows to 512 bytes.

```
switch:admin> flow --control -feature generator -size 512
```

Changing the frame payload size for a single flow

Assigning a frame payload size explicitly to a flow affects only that flow and overrides the existing payload size for that flow. Changing a flow's payload size can only be done to an inactive flow. The following example changes the payload size for the flow "F1024" to 1024 bytes.

```
switch:admin> flow --control F1024 -feature generator -size 1024
```

Frame payload pattern

Flow Generator allows you to define the pattern to be used as the frame payload. The frame payload pattern must be an alphanumeric ASCII string between 1 and 32 characters in length. The default frame payload pattern value is 0, which produces a random pattern of alphanumeric ASCII characters with a variable string length between 1 and 32 characters.

There are two ways to change a frame payload pattern; you can change the default payload pattern, or you can change the payload pattern for a single flow.

To see the current generic payload pattern, enter **flow --show -ctrlcfg**; to see the payload pattern for a specific flow, enter **flow --show** *flow_name* **-ctrlcfg**.

Changing the default frame payload pattern

Assigning a frame payload pattern without defining a flow creates a default that applies to all subsequently-created flows. The following example sets the default payload pattern for all new flows to "TestFlow".

```
switch:admin> flow --control -feature generator -pattern "TestFlow"
```

Changing the frame payload pattern for a specific flow

Explicitly assigning a frame payload pattern to a flow overrides the existing frame payload pattern for that flow, and affects only that flow. Changing a payload pattern can only be done to an inactive flow. The following example sets the default payload pattern for the flow F2 to "a5a5a5".

```
switch:admin> flow --control F2 -feature generator -pattern "a5a5a5"
```

Flow Generator examples

The following examples describe how to work with Flow Generator flows.

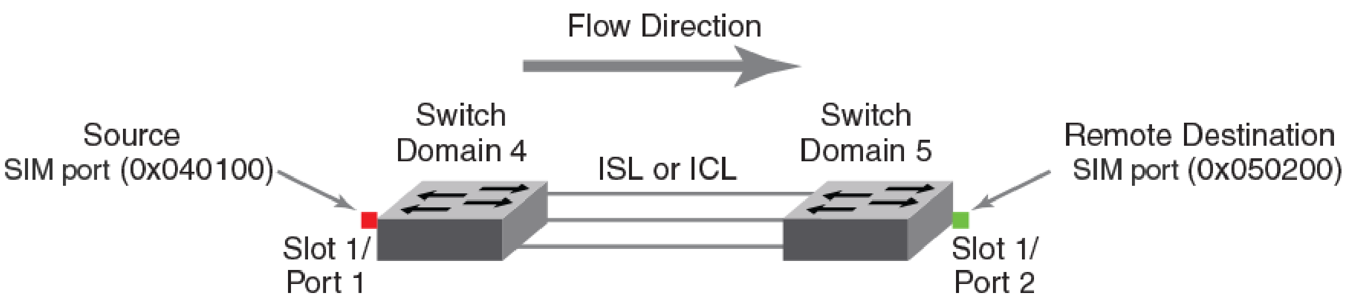
Creating a flow from a specific source ID to a specific destination ID

To create a flow between a specific source ID (SID) and a specific destination ID (DID), complete the following steps.

1. Create two SIM ports.

2. Create an active flow from the SID to the DID.

FIGURE 27 A flow from a specific source ID to a specific destination ID



The following example creates the flow as shown in the figure above. SIM port 1/1 is the source port and SIM port 1/2 is the destination device. The **flow --show flowCase1 -feature generator** command displays the SID frame count and then the DID frame count.

The second **flow --control** command must be run on the remote switch.

```
switch4:admin> flow --control -simport 1/1 -enable

switch5:admin> flow --control -simport 1/2 -enable   <--On the remote switch

switch4:admin> flow --create flowcase1 -feature generator
                     -ingrport 1/1 -srcdev 0x040100 -dstdev 0x050200

switch4:admin> flow --show flowCase1 -feature generator
Name: flowcase1 Features: gen(Activated) noConfig: Off
Definition: IngrPort(1/1),SrcDev(040100),DstDev(050200)
Flow Generator (Activated):
-----------------------
|  SrcDev  |  DstDev  |
-----------------------
| 0x040100 | 0x050200 |
-----------------------
Number of frames generated from ingrport:  19.46M

switch4:admin> flow --show -ctrlcfg
SimPort Information
-------|------|--------|-------------------------|--------------------|------------------|
 Slot  | Port |  PID   |          PWWN           |   SID Frame Count  | DID Frame Count  |
-------|------|--------|-------------------------|--------------------|------------------
   1   |  2   | 050200 | 20:02:00:05:1e:e2:8e:00 |        0K          |      19480K      |
-------|------|--------|-------------------------|--------------------|------------------|
Addressing mode information
Port Addressing Mode: index
Device Addressing Mode: PID
Flow Generator Information
Size: 2048
Pattern: Random (Default)
Flow mirror Information
enable_wrap
```

Integrating Flow Generator with Flow Monitor

Flow Generator flows can be monitored using Flow Monitor. For example, you can use a combination of Flow Generator flows and Flow Monitor flows to verify per-flow throughput at an ingress or egress port. This can be useful when more than one Flow Generator flow shares the same ingress or egress port. To do this, you must create a flow using both the Flow Generator feature and the Flow Monitor feature that share the ingress or egress port.

> **NOTE**
> With Fabric OS 8.0.1a and later releases, Flow Generator traffic cannot be monitored on Gen 6 fixed-port devices or chassis-based systems. Only ingress monitoring is supported.

The following example illustrates a flow that uses both the Flow Generator and Flow Monitor features. In this example, they share an ingress port.

```
switch:admin> flow --create flowCase3Source -feature generator,monitor
                    -ingrPort 1/1 -srcDev 0x010100 -dstDev "*"

switch:admin> flow --show flowCase3Source -feature generator,monitor
Name: flowCase3Source Features: gen(Activated),sts(Activated)
Definition: IngrPort(1/1),SrcDev(010100),DstDev(*)

Flow Generator (Activated):
-----------------------
| SrcDev   | DstDev   |
-----------------------
| 0x010100 | 0x010000 |
-----------------------
| 0x010100 | 0x015000 |
-----------------------
| 0x010100 | 0x015100 |
-----------------------

Number of frames generated from IngrPort : 1.63G

Flow Monitor (Activated):
---------------------------------------------------------------------------------------------
|DID(*)| Rx Frames Count | Rx Frames per Sec. | Rx Bytes Count | Rx Through- | Avg Rx Frm Sz|
|      |                 |                    |                |  put(Bps)   |  (Bytes)     |
---------------------------------------------------------------------------------------------
| 10000| 75.84M          | 482.95k            | 1568G          | 14.47M      | 2080         |
| 15000| 37.42M          | 241.47k            | 784.4G         | 50.23M      | 2080         |
| 15100| 37.42M          | 241.47k            | 784.4G         | 50.23M      | 2080         |
---------------------------------------------------------------------------------------------
| *    | 1.50G           | 965.90k            | 3137G          | 1.15G       | 2080         |
---------------------------------------------------------------------------------------------
```

Commands related to Flow Generator

The following commands affect or are affected by Flow Generator. Refer to the *Brocade Fabric OS Command Reference* for more detailed information.

portcfgshow

Entering **portcfgshow** for a specified port shows the configuration of the SIM port. (Called out for illustration.)

```
switch:admin>portcfgshow 8/0
Area Number:              8
Octet Speed Combo:        1(32G|16G|8G|4G)
Speed Level:              AUTO(HW)
AL_PA Offset 13:          OFF
Trunk Port                ON
Long Distance             OFF
VC Link Init              OFF
Locked L_Port             OFF
Locked G_Port             OFF
Disabled E_Port           OFF
```

```
Locked E_Port             OFF
ISL R_RDY Mode            OFF
RSCN Suppressed           OFF
Persistent Disable        OFF
LOS TOV mode              0(OFF)
NPIV capability           ON
QOS Port                  AE
Port Auto Disable:        OFF
EX Port                   OFF
Mirror Port               OFF
SIM Port                  OFF   <-- SIM Port status
Credit Recovery           ON
F_Port Buffers            OFF
E_Port Credits            OFF
Fault Delay:              0(R_A_TOV)
NPIV PP Limit:            126
NPIV FLOGI Logout:        OFF
CSCTL mode:               OFF
TDZ mode:                 OFF
D-Port mode:              OFF
D-Port over DWDM:         OFF
10G/16G FEC:              ON
16G FEC via TTS:          OFF
```

portperfshow

When you run the **portperfshow** command, an asterisk (*) represents traffic running on either an ingress port or an egress port from a Flow Generator flow. For the **portperfshow** command, the Transmit Throughput value represents the egress port, and the Receive Throughput value represents the ingress port.

portstatsclear

Entering **portstatsclear** for a port clears the Flow Generator frame count for all flows sharing this port. This is equivalent to the **flow --reset** command. Refer to Resetting flow statistics on page 38 for details.

portstatsshow

When you run the **portstatsshow** command, the Transmit Word Count value represents the egress port, and the Receive Word Count value represents the ingress port. The frame size used for **portstatsshow** is 2048 bytes, regardless of what value has been set for the flow.

slotstatsclear

Entering **slotstatsclear** for a slot clears the Flow Generator frame count for all flows sharing the ports on that slot.

switchshow

Entering **switchshow** generates output showing which ports are set as simulation mode ports (SIM ports) and displays the WWN for each emulated device.

```
switch:admin> switchshow | grep SIM
192  8  0  046000    id    16G    Online    FC  SIM-Port 20:c0:00:05:1e:99:61:00
193  8  1  046100    id    16G    Online    FC  SIM-Port 20:c1:00:05:1e:99:61:00
194  8  2  046200    id    16G    Online    FC  SIM-Port 20:c2:00:05:1e:99:61:00
195  8  3  046300    id    16G    Online    FC  SIM-Port 20:c3:00:05:1e:99:61:00
196  8  4  046400    id    16G    Online    FC  SIM-Port 20:c4:00:05:1e:99:61:00
```

SIM port attributes and configuration

Flow Vision requires that the source device and destination device ports be in simulation mode (SIM port mode) prior to activating the pre-defined or static test flows, and checks for this before activating the test flows.

Prior to creating and activating flows, you must enter **flow --control** on the local switch to set the source device (srcdev) and the destination device (dstdev) ports as SIM ports. This ensures that test flows are not unintentionally transmitted to real devices. After the source device and destination device ports are configured to be SIM ports, you can create and activate the flow.

The following restrictions will affect your use of SIM ports:

- Flow Generator supports up to 39 active flows per ingress SIM port, and takes 48 credits per SIM port from the ASIC.
- Zoning is bypassed on SIM ports for both pre-defined and static flows. In these cases, traffic will reach its destination regardless of the zoning configuration.
- Zones are used to gather the Source ID-Destination ID pairs for learning flows.

SIM port criteria

Flow Generator simulation (SIM) ports must meet the following criteria to be valid:

- SIM ports are supported on ASICs that support either 8 Gbps-capable, 16 Gbps-capable, or 32 Gbps-capable Fibre Channel ports. Source devices or ingress ports can only be on 16 Gbps-capable or 32 Gbps-capable Fibre Channel ports. Destination devices or egress ports can be on either 8 Gbps-capable, 16 Gbps-capable, or 32 Gbps-capable Fibre Channel ports.
- SIM ports cannot be in the base switch or Access Gateway.
- SIM ports cannot be configured on a port that is online and connected to a real device.

  NOTE
  If a port is connected to a real device, you can disable the port, configure the SIM port, and then re-enable the port. The port will be a SIM port; the real device will not join the fabric.

- Existing SIM ports are added to Device Connection Control (DCC) policies when created with a wildcard (*) but are not adhered to. These SIM port entries must be deleted if a new WWN is connected.
- SIM ports cannot be configured as any of the following port types; these restrictions also apply at the time a SIM port is enabled:
  - Any port running Encryption or Compression
  - Any F_Port connected to a real device (unless the port is disabled)
  - Any port configured with F_Port buffers
  - D_Port (Diagnostic Port)
  - E_Port
  - EX_Port
  - F_Port trunked
  - Fastwrite port
  - FCoE port
  - ICL port
  - L_Port
  - M_Port (Mirror Port)
  - VE port
- If a port is configured with CSCTL_mode and/or QoS enabled, you can configure it as a SIM port.
- If a port has an Ingress Rate Limit set, you can configure it as a SIM port.

- If a port is configured as a SIM port, you can set an Ingress Rate Limit.
- The following features of a SIM port are persistent across a reboot:
  - Each SIM port is assigned a PID and is displayed in a **switchShow** command.
  - Each SIM port's Port Worldwide Name by default is the switch Port Worldwide Name (PWWN), unless a user-defined Virtual Port Worldwide Name is assigned to it.
  - Each SIM port registers itself into the Name Server database.

Identifying SIM ports

In order to distinguish SIM ports from other devices, the Name Server commands show "Port Properties: SIM Port", as called out in the following example:

```
switch:admin> nscamshow
nscam show for remote switches:
Switch entry for 105
state  rev    owner  cap_available
known   v730  0xfffc6e 1
Device list: count 12
Type Pid     COS     PortName                NodeName
N    691000; 3;      10:00:00:00:00:0f:00:00;10:00:00:00:00:00:0f:00;
     Fabric Port Name: 20:10:00:05:1e:57:dc:b3
     Permanent Port Name: 10:00:00:00:00:0f:00:00
     Port Index: 16
     Share Area: No
     Device Shared in Other AD: No
     Redirect: No
     Partial: No   <— Port Properties value is not shown for non-SIM ports
N    691400; 2,3;    20:14:00:05:1e:57:dc:b3;20:14:00:05:1e:57:dc:b3;
     Fabric Port Name: 20:14:00:05:1e:57:dc:b3
     Permanent Port Name: 20:14:00:05:1e:57:dc:b3
     Port Index: 20
     Share Area: No
     Device Shared in Other AD: No
     Redirect: No
     Partial: No
     Port Properties: SIM Port   <— Port Properties value shows "SIM Port"
      (output truncated)
```

```
switch:admin> nsshow
Type Pid     COS    PortName                NodeName                 TTL(sec)
N    691000; 3;     10:00:00:00:00:0f:00:00;10:00:00:00:00:00:0f:00; na
    Fabric Port Name: 20:10:00:05:1e:57:dc:b3
    Permanent Port Name: 10:00:00:00:00:0f:00:00
    Port Index: 16
    Share Area: No
    Device Shared in Other AD: No
    Redirect: No
    Partial: No
    LSAN: No
    Device link speed: 8G
    Connected through AG: Yes
    Real device behind AG: Yes<— Port Properties value is not shown for non-SIM ports
N   691400; 2,3;   20:14:00:05:1e:57:dc:b3;20:14:00:05:1e:57:dc:b3; na
    Fabric Port Name: 20:14:00:05:1e:57:dc:b3
    Permanent Port Name: 20:14:00:05:1e:57:dc:b3
    Port Index: 20
    Share Area: No
    Device Shared in Other AD: No
    Redirect: No
    Partial: No
    LSAN: No
    Device link speed: 8G
```

```
Connected through AG: Yes
Real device behind AG: Yes
Port Properties: SIM Port   <-- Port Properties value shows "SIM Port"
 (output truncated)
```

Sending traffic using a Fabric-Assigned WWN

If you want to use a Fabric-Assigned WWN (FA-WWN), you need to set the FA-WWN on the SIM port using the Dynamic Fabric Provisioning command, **fapwwn -assign**. For details, refer to the "Dynamic Fabric Provisioning" section of the *Brocade Fabric OS Administration Guide*.

CAUTION
If the fapwwn command is used to assign a user-defined Port WWN to a SIM port, it is the responsibility of the person making the assignment to not assign a Port WWN that duplicates one already in the fabric. If there is a duplicated WWN, both entries will be removed from the Name Server database. This has a high probability of disrupting traffic.

Flow Generator and High Availability

On a High Availability (HA) failover, HA reboot, or a power cycle and reboot, both local and remote flows remain active. When SIM ports come back online from an HA failover or HA reboot event, local flows are re-created and reactivated, and local traffic is restarted. Inactive flows are re-created but not activated by an HA failover, HA reboot, or a power cycle and reboot.

ATTENTION
A HA failover, HA reboot, or a power cycle and reboot event may cause different sub-flows to be re-created, as the flow order depends on the zone database. If the number of zoned SID-DID pairs is greater than the maximum permitted number of flows, then any pairs that appear in the zone database after that limit is reached will be ignored. Refer to Numbers of flows supported on page 25 for these limits.

If the sys_gen_all_simports flow was active prior to a HA failover, the flow will be replayed after the HA failover with all the SIM ports that are configured on the switch.

Refer to High Availability and Flow Vision on page 45 and Notes on predefined flows on page 104 for more information.

Flow Generator and MAPS

The Brocade Monitoring and Alerting Policy Suite (MAPS) can be used to monitor SIM port traffic thresholds while Flow Generator flows are running.

MAPS treats SIM ports as F_Ports, so MAPS can issue warnings on these ports if threshold values are triggered. If you do not want to see MAPS warnings for SIM ports, you must disable MAPS monitoring for those ports.

Flow Generator traffic will also impact E_Ports; this may cause MAPS warnings for E_Port throughput levels. Refer to the *Brocade Monitoring and Alerting Policy Suite Configuration Guide* for more information about working with MAPS.

You can use MAPS to monitor traffic in the sys_gen_all_simports flow using the same default and custom policies as used for F_Ports. However, there will not be a check in MAPS for maximum throughput for SIM ports, because Flow Generator always runs at maximum throughput.

Flow Generator limitations and considerations

The following limitations apply specifically to Flow Generator:

- Flow Generator traffic competes with any other traffic on a fabric. Consequently, E_Ports and FCIP links can become congested when using Flow Generator, leading to throughput degradation. FCIP links are more prone to congestion than E_Ports. If used on a live production system, this can cause traffic issues.
- The maximum number of Flow Generator flows allowed per ingress port is 39.
- Activating more than five static flows on a fixed-port switch or five static flows per blade in a chassis switch might result in a "DMA buffer full" error message. Prior to upgrading, make sure that you will not exceed these limits after upgrading.
- Flow Generator flows can only be mirrored at the ingress port; they cannot be mirrored at the egress port.
- Flow Generator is not supported on Access Gateways.
- Flow Generator traffic over a VE_port is non-disruptive only if no other traffic is running on any of the other VE_ports on that blade or switch platform. If Flow Generator traffic is run over a VE_port and production traffic is run over the same VE_port or another VE_port, then the production traffic could be negatively affected.
- Frame redirection is not supported for SIM ports.
- If a SIM-port configuration is deleted while the port is online, Flow Vision automatically stops all Flow Generator data flowing to those ports, but the flows themselves are not deactivated.
- Zoning is not enforced. Sources and destinations can be in different zones.
- Flow Generator gathers source and destination pairs from the zoning database for learning flows only at the time the flow is activated. Subsequent changes to this database will not be registered until a flow is reactivated.
- Flow Generator learning flows and the sys_gen_all_simports flow support a frame size range from 64 to 128 bytes. You can set this value from 64 to 2048 using either BNA or the CLI. Use the command **flow --control** *flow_name* **-ctrlcfg** to display the active frame size. The frame size will be restricted internally when the flow is activated.

Considerations when using Flow Generator across FCR

The following considerations apply when you use Flow Generator across Fibre Channel Routing (FCR):

- For Fabric OS 8.1.0 and later versions, Flow Generator is supported across Fibre Channel Routing (FCR). However, for Fabric OS 8.0.1 and early versions, FCR does not exchange SIM-port parameters of proxy devices to the name server. Therefore, for those versions, Flow Generator is not supported across FCR.
- Only edge domains using FCR and running Fabric OS 8.1.0 and later versions can process and exchange SIM-port and FC4 feature parameters.
- In backbone fabrics, only FCR-enabled domains that are running Fabric OS 8.1.0 or a later version can sync SIM-port and FC4 feature parameters with each other.
- For Fabric OS 8.1.0, this capability is supported only for edge-to-edge topologies; it is not supported in backbone-to-edge topologies. Also, All the switches from edge to edge must be running Fabric OS 8.1.0 or later.

Flow Mirror

Overview of Flow Mirror

As storage networks get larger and more complicated, non-intrusive diagnostic tools are becoming increasingly important to help identify problems without disturbing the operating fabric. Flow mirroring is a diagnostic feature within Flow Vision that addresses this need.

With Flow Mirror, you can do the following:

- Non-disruptively create copies of application flows that can optionally be captured for deeper analysis.
- Define a traffic pattern and create a real-time copy of this traffic, allowing you to analyze a live system without disturbing existing connections. You can also use this feature to view traffic passing through a port.
- Use a MAPS logical port group for either an ingress or egress port. Note that logical port groups are not supported on CPU flow mirroring (CFM) or logical flow mirroring (LFM).
- Use a predefined flow with the **flow --show** command to mirror SCSI command, first response, status and ABTS frames from all the Gen 5 and Gen 6 F_Ports on a switch. For example: **flow --show sys_analytics_vtap**

Flow Mirror duplicates the specified frames in a user-defined flow, and sends them to a sink. This sink could be either:

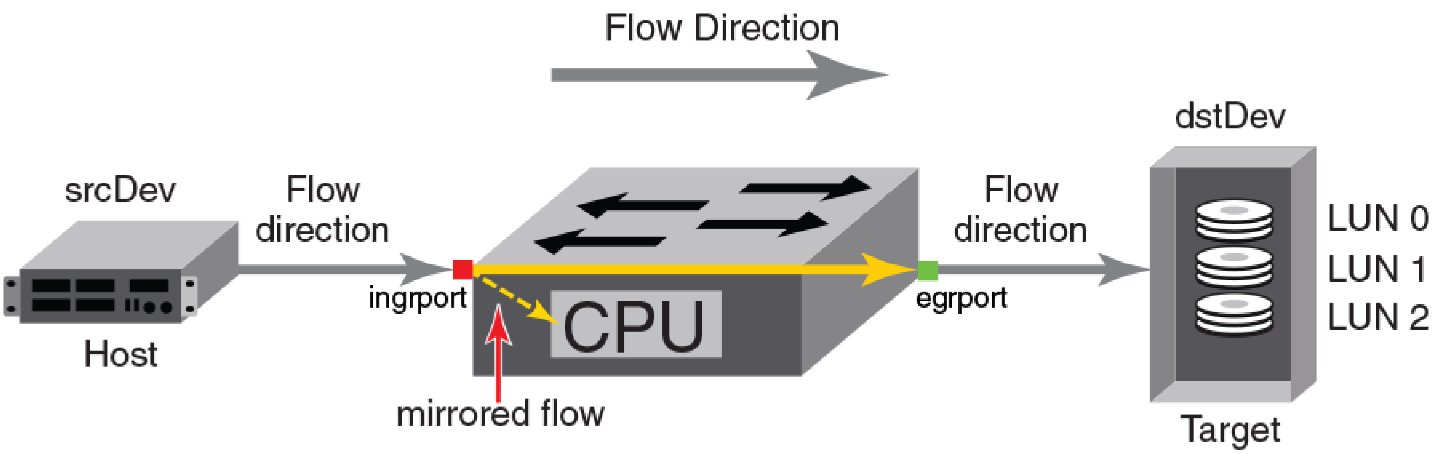
- The local switch control central processor unit (CPU); this form is called "CPU flow mirroring" (CFM), and has a limit of 256 frames per second.
- An analyzer/packet sniffer connected through a port in the metaSAN. The bandwidth limit for a flow using this type of link is the bandwidth of the mirror destination port. This form is called "Local flow mirroring" (LFM), and mirrors the flow to a port on the same physical switch. This requires that a loopback SFP be plugged in at the other end of the analyzer or on the port configured as a mirror port, which must be in the same domain.

> **NOTE**
> All forms of mirroring are mutually exclusive. Only one form of mirroring (LFM, CFM, or RFM) can be active at a time.

Flow Mirror flows can be in an active or inactive state. If the mirror flow is "active", mirroring starts immediately; if the flow is "inactive", the flow must be activated (by using the **flow --activate** command) for mirroring to start. Mirrored flows can be unidirectional or bidirectional. A sample use case would be to mirror the traffic flow from a slow-draining F_Port to see what is causing this condition. Diagnosing a slow-draining F_Port on page 135 provides an example of this use case.

The following figure provides a diagram of a flow that mirrors to the CPU the traffic passing into the switch through the ingress port. Flow Mirror can similarly mirror the egress port, but only one port (ingrport or egrport) can be mirrored per flow. To mirror from one port in both flow directions (left to right and right to left in the figure), the **-bidir** keyword must be used in the flow definition.

FIGURE 28 Illustration of CPU flow mirroring



Creating Flow Mirror flows

To create a Flow Mirror flow, use the **flow --create** *flow_name* **-feature mirror** *parameters* command.

When you create a flow, it is automatically activated unless you use the **-noactivate** keyword as part of the **flow --create** command. Refer to Creating an inactive flow in Flow Mirror on page 127 for an example.

Figure 1 on page 22 illustrates how the frame and port parameters apply to a flow. The following table shows the supported Flow Mirror flow parameter combinations.

TABLE 25 Flow Mirror-supported flow parameter combinations

Parameters	Field names	Description
Port	ingrport egrport	• One field only must be specified. • Values must be explicit. • Can only be an F_Port local to the switch, a Gen 5 (16 Gbps) or Gen 6 (32 Gbps) F_Port, or a Gen 5 or Gen 6 F_Port trunk on the local domain. • For CFM and LFM, the values for ingrport and egrport can only be a single F_Port or F_Port trunk. • For the predefined flow, sys_analytics_vtap, the values for ingrport and egrport can be a single F_Port, an F_Port trunk, an asterisk in quotation marks (`"*"`), or a MAPS logical port group. Mirroring is active only on the F_Ports, even though the port is specified as `"*"` or as a MAPS logical group.
Frame	srcdev dstdev lun frametype	• Only one field can be specified. • Values for srcdev and dstdev can be explicit or `"*"` (`"*"` indicates learned flows). • Values for lun and frametype must be explicit.
Optional keyword parameters		
	-bidir	Adding this keyword makes the application mirror traffic in both directions.
	-noactivate	Adding this keyword creates the flow without activating it.
	-noconfig	Adding this keyword creates the flow without saving the flow to the configuration.

The following example creates a Flow Mirror flow named “flowmirror14” that mirrors traffic flowing from device “080e00 “ to device “080f00” ingressing through the port 14 on the switch on which this command was run.

```
switch1234:admin> flow --create flowmirror14 -feature mirror -ingrport 14 -srcdev 080e00 -dstdev 080f00
Mirror feature(s) have been activated.
```

Flow Mirror limitations and restrictions

The following limitations and restrictions apply specifically to Flow Mirror flows and flow mirroring:

- Flow Mirror in Fabric OS 8.1.0 is backwardly compatible with the Flow Vision suite of applications in the following ways:
  - CFM flows are backwardly compatible to version 7.2.x of Fabric OS.
  - LFM flows are backwardly compatible to version 7.3.x of Fabric OS.
  - The predefined flow, sys_analytics_vtap, is backwardly compatible with versions 7.3.2a, 7.4.x, and 8.0.x of Fabric OS.
- Flow Mirror is supported only on Gen 5 and Gen 6 Fibre Channel platforms. Refer to Supported hardware and software on page 12 for a list of these platforms.
- Activation of the predefined Flow Mirror flow, sys_analytics_vtap, fails if the chosen AE_port is not a Gen 5 or Gen 6 port.
- The sys_analytics_vtap flow can be enabled simultaneously on multiple logical fabrics for Gen 5 and Gen 6 slotted chassis and on the Gen 5 Brocade 6520 platform.
- Before activating the RFM flow on Gen 5 and Gen 6 platforms, turn on the chassis-level configuration feature "vTAP and QoS High Priority Zone Compatibility mode." The configuration feature is required in order for RFM and QoS High Priority Zones to co-exist. Use the following configuration:

```
switch:admin> configurechassis

Configure...

  cfgload attributes (yes, y, no, n): [no]
  Custom attributes (yes, y, no, n): [no]
  system attributes (yes, y, no, n): [no]
  fos attributes (yes, y, no, n): [no] y

        Reboot needed to effect new CSCTL Mode
        CSCTL QoS Mode (0 = default; 1 = auto mode): (0..1) [0]
        Chassis SDDQ Limit: (0..32) [10]
        ChassisDisble followed by ChassisEnable needed to effect new Mode
        vTap and QOS High Priority Zone Compatibility Mode (on, off): [on]

switch:admin>
```

  NOTE
  The predefined flow sys_analytics_vtap is compatible on Brocade Analytics Monitoring Platform version 5.0 and later. If a Fabric OS 8.1.0 chassis is connected to Brocade Analytics Monitoring Platform version 4 or lower, then the AE links will be segmented.

- The mirror port is specified using -mirrport in the flow command. Not specifying this option indicates that the flow is a CFM flow. For an LFM flow, the -mirrorport is the index of a local port in the chassis. An analyzer can be connected on this port. For the "sys_analytics_vtap" flow, the -mirrorport refers to the logical Fabric AF_Port of the AMP that is connected to the logical switch or fixed-port switch.
- Of the three types of Flow Mirror flows—CFM, LFM, and RFM (predefined)—only one type of flow can be active, because these types are exclusive of each other.
- A mirror port cannot be either the ingress or egress port specified in an LFM flow.
- The flow ingress or egress ports must refer to ports on the current switch or a MAPS logical group.

- You can use the following options of the **flow --control** command only for CPU flow mirroring:
  - **flow --control enable_wrap**
  - **flow --control disable_wrap**
- For an LFM flow, if the combined rate of bidirectional traffic exceeds the limit of the mirror port traffic limit, then the Flow Mirror flow is not permitted. (For example, the combined rate of a bidirectional flow from a 16-Gbps port is 32 Gbps, and so it cannot be mirrored to a 16-Gbps mirror port.)
- On Gen 5 platforms, if a Flow Monitor flow that was defined using the **-frametype** keyword as part of the **flow** command is installed on an ingress port, and a matching Flow Mirror flow (user-defined or predefined) is installed on an egress port, then traffic egressing through the egress port is not mirrored.
- On Gen 5 platforms, if a user-defined or predefined flow is active for both Flow Monitor and Flow Mirror, then frames matching the following definitions will be monitored but will not be mirrored:
  - Definitions using a combination of the **-frametype** and **-ingrport** keywords.
  - Definitions using a combination of **-frametype**, **-egrport**, and **bidir** keywords.
- On Gen 5 platforms, if a user-defined or predefined Flow Monitor flow is active on a blade or fixed-port switch, then Flow Mirror frames matching any active flow definitions defined using the combinations below are not mirrored if they use ports on the same blade or fixed-port switch as defined by the following flow definitions:
  - Flow definitions using a combination of the **-frametype** and **-ingrport** keywords.
  - Flow definitions using a combination of **-frametype**, **-egrport**, and **bidir** keywords.
- On Gen 5 platforms, if a predefined Flow Mirror flow is active, then Flow Monitor can support only 11 flows defined using the following flow definitions:
  - Definitions using a combination of frame type and egress port.
  - Definitions using a combination of frame type, ingress port, and bidir.
- On Gen 5 platforms, if the Host and Target reside on a fixed-port switch or on the same blade, then only a single directional flow is mirrored by the predefined Flow Mirror flow.
- On Gen 6 platforms, if a Flow Monitor flow that was defined using the **-frametype** keyword as part of the flow command is installed on an ingress port, and a matching Flow Mirror flow is installed on an egress port, then traffic egressing through the egress port is mirrored but *not* monitored.
- On Gen 6 platforms, if a flow was created for both Flow Monitor and Flow Mirror that was defined using either a combination of the **-frametype** and **-ingrport** keywords or a combination of **-frametype**, **-egrport**, and **-bidir** keywords, then frames matching these definitions are mirrored but *not* monitored.
- On Gen 6 platforms, if a Flow Monitor flow was created on a blade or fixed-port switch that was defined using either a combination of the **-frametype** and **-ingrport** keywords or a combination of **-frametype**, **-egrport**, and **-bidir** keywords, then Flow Mirror frames matching any flow definitions using the **-ingrport** keyword that use ports on the same blade or fixed-port are mirrored but *not* monitored.
- On Gen 6 platforms, Flow Monitor supports only two flows defined using either a combination of the **-frametype** and **-ingrport** keywords or a combination of **-frametype**, **-egrport**, and **-bidir** keywords.
- On Gen 6 platforms, if a Flow Monitor flow is activated that uses frametype and ingress port keywords or frametype, egress port, and bidir keywords as part of the flow command, and a matching Flow Mirror flow is active on the port, then that traffic is mirrored but not monitored.
- The predefined Flow Mirror flow is not supported in the base switch. You must deactivate the predefined Flow Mirror flow before you convert a logical switch to a base switch.
- Mirroring of exported and imported devices is supported only on edge fabrics, not on backbone or base fabrics.
- Flow Mirror is not supported on SIM ports that are specified as ingress or egress ports in the flow definition.
- When using Flow Mirror in Access Gateway mode, you cannot configure internal ports on SAN I/O modules as mirror ports.

- Flow Mirror cannot mirror:
  - Frames belonging to device-switch communication (for example, a FLOGI or PLOGI action).
  - Link Primitives, discarded frames, frames from a remote Control Unit Port (CUP), Link Control Frames, or frames containing domain controller addresses used as source IDs.
- For flows mirrored to a CPU, only the first 256 frames of a second are mirrored. If a greater number of frames that match the flow definition within a second are identified, those later frames are not mirrored.

  For example, on a fixed-port switch, if 500 frames meet the flow definition in the first second of the Flow Mirror operation, only the first 256 frames are mirrored. In the next second, frame mirroring will begin with the 501st frame that matches the flow definition. The intervening frames will not be mirrored, even though they match the flow definition.
- Although the system attempts to handle misconfigurations of zoning, it is recommended that you do not place mirror ports in zones with any other port or device.

Remote Flow Mirroring limitations and restrictions

In addition to the above limitations and restrictions, the following restrictions and limitations apply to Remote Flow Mirroring (RFM).

- RFM can only mirror frames to the Brocade Analytics Monitoring Platform.
- RFM is not supported in Access Gateway mode.
- RFM is supported on Gen 6 SX6 FCIP blade.
- No user-defined flow creation is supported for RFM. You must use the existing predefined **sys_analytics_vtap** flow.
- Before activating an RFM flow, you must turn on the chassis-level configuration feature "vTAP and QoS High Priority Zone Compatibility Mode." With this feature activated, RFM/vTAP and QoS High Priority can co-exist.
- Because sys_analytics_vtap is a flow that is pre-defined in the system, it cannot be deleted nor can a new one be created. The permitted operations on this flow are "show," "modify," "deactivate," "activate," and "reset."
- When set up as a learning flow, the predefined **sys_analytics_vtap** flow uses the first mirror port it learns about as the remote mirror port.
- Remote Flow mirroring does not function if the remote domain or AF_Port is not accessible.
- You can only modify the ingress port and mirror port of the predefined **sys_analytics_vtap** flow. The ingress port for a RFM can be `"*"`, a single F_Port, or a MAPS logical port group.
- For Gen 5 platforms that also support Multi-VF RFM, the sys_analytics_vtap flow is not activated if online LISLs exist in the partition. If the sys_analytics_vtap flow is active, and LISLs come online in that partition, then the sys_analytics_vtap flow is deactivated.
- RFM traffic always uses only VC14 for reaching a Brocade Analytics Monitoring Platform.
- Flow Mirror supports using only Gen 5 or Gen 6 ports as AE_Ports to connect a switch and a Brocade Analytics Monitoring Platform. Brocade 7840 Extension Switch does not support Remote Flow mirroring.
- RFM multi-VF feature is not supported on fixed-port switches, except the Brocade 6520 switch.
- RFM multi-VF feature requires AE_Port DISLs in each logical switch for which a predefined **sys_analytics_vtap** flow is supposed to be activated. This feature is not supported with LISLs.
- RFM uses only a single AE_Port DISL per fabric, that is, DPS is not supported for mirrored VTAP frames. When multiple AE_Ports are present in the logical switch, only one AE_Port is picked up to transport mirrored frames. However, an AE_Port trunk is supported.
- Activation of features for the predefined **sys_analytics_vtap** flow cannot be enforced if the AF_port is either unreachable or offline.

- On Brocade 6520 and DCX 8510 running Fabric OS 8.1.1, if the vTap and Encryption/Compression Coexistence Mode is turned on, encryption/compression and RFM can coexist on the same ASIC. To enable the co-existence mode, use the following chassis-wide configuration. If the coexistence mode is not turned on, then the previous release version behavior is reflected.
  1. Run the **chassisdisable** command.
  2. Run the following commands:

```
switch:admin> configurechassis

Configure...

  cfgload attributes (yes, y, no, n): [no]
  Custom attributes (yes, y, no, n): [no]
  system attributes (yes, y, no, n): [no]
  fos attributes (yes, y, no, n): [no] y

        Reboot needed to effect new CSCTL Mode
        CSCTL QoS Mode (0 = default; 1 = auto mode): (0..1) [0]
        Chassis SDDQ Limit: (0..32) [10]
        ChassisDisble followed by ChassisEnable needed to effect new Mode
        vTap and QOS High Priority Zone Compatibility Mode (on, off): [on]
        vTap and Encryption/Compression Coexistence Mode (on, off):[on]

switch:admin>
```

     NOTE
     The vTap and Encryption/Compression Coexistence Mode configuration choice is not available when you run **configurechassis** on unsupported platforms.

  3. Run the **chassisenable** command.

  If the coexistence mode is turned on and if the mirrored traffic exceeds 250K IOPS on the ASIC, then RFM is unenforced on the F_Ports of the ASIC and a raslog [FV-1006] is also generated. This raslog specifies the port range of the F_Ports on which RFM is unenforced.

  NOTE
  The vTap and Encryption/Compression Coexistence Mode should not be turned on when the defzone is configured with "allaccess mode" and no zone exists in the active configuration.

  The following limitations apply when using in-flight encryption or compression with RFM:

  – When in-flight encryption and compression are not enabled on any of the ports on an ASIC, then RFM functions at full capacity on that ASIC.
  – When in-flight encryption or compression is enabled and RFM exists on the same ASIC, and the number of frames being mirrored exceeds 250K IOPS, then the RFM is unenforced on the F_Ports of the ASIC.
  – When the coexistence mode is enabled, the CPU originated frames are neither encrypted nor compressed when they egress through the encrypted/compressed ISLs.
  – When the coexistence mode is enabled, and RFM is also activated and enforced, if in-flight encryption and compression are disabled on all the ports of the ASIC, then RFM functions at full capacity.

  NOTE
  Certain versions of Fabric OS support connections to the Brocade Analytics Monitoring Platform. For more information about the Brocade Analytics Monitoring Platform, including which versions of Fabric OS can interact with it, refer to the Brocade Analytics Monitoring Platform documentation or MyBrocade.com.

CFM frame rates and frame capacity

The following table shows the maximum frame rate and mirroring capacity for each platform type.

TABLE 26 Flow Mirror CFM frame rates and frame capacity

Platform type	Maximum rate (frames per second)	Maximum capacity (frames)
Fixed-port switch	256	1280
Chassis-based systems	256	5120

Local flow mirroring

Local flow mirroring (LFM) allows you to mirror a flow to a port in the same domain that the flow has been defined in. This mirrored data can then be analyzed through an external analyzer/frame sniffer connected to the port.

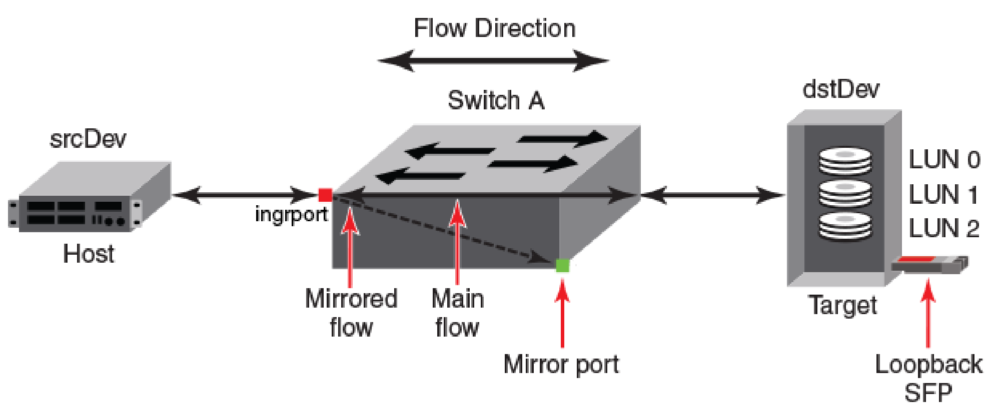
To mirror traffic to a local port, the port must be configured as a mirror port before it is used in the flow definition. A loopback SFP should be plugged in at the other end of the analyzer or on the port configured as a mirror port.

Here is the pattern for a command creating a flow definition to mirror a local flow:

- First: Use the **portcfg mirrorport** *port_num* **--enable** command to configure the mirror port.
- Next: Use the **flow --create** *flow_name* **–feature mirror –srcDev** *SrcDevID* **-dstDev** *DestDevID* **-ingrport** *portID* **-bidir –mirrorport** *port_num* to create the flow that is mirrored to that port.

A typical configuration for local flow mirroring is shown below.

FIGURE 29 Sample local flow mirroring configuration



The following example configures port 16 as a mirror port, then creates a flow for all traffic flowing from device 010e00 to device 010f00 ingressing through port 14 that sends the mirrored frames to the mirror port, and finally displays a list of the flows followed by the results of the flow.

```
switch:admin> portcfg mirrorport 16 --enable

switch:admin> flow --create fmtest -feature mirror -srcdev 010e00 -dstdev 010f00
                      -ingrport 14 -mirrorport 16
Mirror feature(s) have been activated.

switch:admin> flow --show
--------------------------------------------------------------------------------------------
Flow Name           |Feature| SrcDev | DstDev |IngrPt|EgrPt|BiDir|LUN|FrameType|SFID|DFID|MirPt|
--------------------------------------------------------------------------------------------
sys_gen_all_simports|gen    |*       |*       |*     |*     |no   |-  |-        |-   |-   |-    |
fmtest              |mir+   |010e00  |010f00  |14    |-     |no   |-  |-        |-   |-   |16   |
```

```
-----------------------------------------------------------------------------------------------
  + Denotes feature is currently activated for the flow
  The flow name with prefix sys_ denotes predefined flow

switch:admin> flow --show fmtest -feature mirror
===============================================================================================
Name      : fmtest      Features: mir(Activated)        noConfig: Off
Definition: IngrPort(14),SrcDev(0x010e00),DstDev(0x010f00),MirPort(16)

Flow Mirror (Activated):
-----------------------------------------------------------------------------------------------
No of Mirrored Frames : 91134, No of RX Mirrored Frames : 91134, No of TX Mirrored Frames : 0
-----------------------------------------------------------------------------------------------
===============================================================================================
```

After a port is configured as a mirror port, the **switchshow** command output looks similar to the following example:

```
switch:admin> switchshow|grep Port

Index Port Address Media Speed       State   Proto
   3   3   010300   id    N16      Online      FC  G-Port
   4   4   010400   id    16G      Online      FC  SIM-Port 20:04:00:27:f8:93:6a:23
   5   5   010500   id    16G      Online      FC  SIM-Port 20:05:00:27:f8:93:6a:23
   6   6   010600   id    16G      Online      FC  SIM-Port 20:06:00:27:f8:93:6a:23
   7   7   010700   id    16G      Online      FC  SIM-Port 20:07:00:27:f8:93:6a:23
   8   8   010800   id    N8       Online      FC  G-Port
  14  14   010e00   id    N8       Online      FC  F-Port  50:08:01:60:09:15:7a:03
  15  15   010f00   id    N8       Online      FC  F-Port  50:08:01:60:09:15:7a:02
  16  16   011000   id    8G       Online      FC  Mirror Port <— Mirror port identification
```

After a port is configured as a mirror port, the **portcfgshow** command output looks similar to the following example:

```
switch:admin> portcfgshow 16
Area Number:             16
Speed Level:             AUTO(SW)
AL_PA Offset 13:         OFF
Trunk Port               ON
Long Distance            OFF
VC Link Init             OFF
Locked L_Port            OFF
Locked G_Port            OFF
Disabled E_Port          OFF
Locked E_Port            OFF
ISL R_RDY Mode           OFF
RSCN Suppressed          OFF
Persistent Disable       OFF
LOS TOV enable           OFF
NPIV capability          ON
QOS Port                 AE
Port Auto Disable:       OFF
Rate Limit               OFF
EX Port                  OFF
Mirror Port              ON <— Mirror port identification
SIM Port                 OFF
Credit Recovery          ON
F_Port Buffers           32
E_Port Credits           OFF
Fault Delay:             0(R_A_TOV)
NPIV PP Limit:           126
NPIV FLOGI Logout:       OFF
CSCTL mode:              OFF
D-Port mode:             OFF
D-Port over DWDM:        OFF
FEC:                     ON
FEC via TTS:             OFF
Non-DFE:                 OFF
```

Creating an inactive flow in Flow Mirror

To create an inactive Flow Mirror flow, enter the **flow --create** *flow_name* **-feature mirror** *flow_parameters* **-noactive** command.

The following example creates an inactive Flow Mirror flow named "mirroredflow120" from device 020a00 to device 01c000 ingressing through port 120.

```
switch:admin> flow --create mirroredflow120 -noactivate -feature mirror -ingrport 120 -srcdev 0x020a00 -
dstdev 0x01c000
```

Refer to Activating Flow Mirror flows on page 127 for information on activating a Flow Mirror flow.

Remote Flow Mirroring

Remote Flow Mirroring (RFM) permits you to route a flow from a switch to a Brocade Analytics Monitoring Platform so that the flow frame data can be monitored and analyzed.

> NOTE
> This feature is only supported when utilizing the Brocade Analytics Monitoring Platform, and the mirrored frames can be sent only to the analytics platform. RFM is not supported in Fabric OS 8.0.0.

To correctly route the mirrored frames, Remote Flow Mirroring must be activated on the source switch F_Port, and the destination port for the mirrored flow must be a Brocade Analytics Monitoring Platform analytics port (AF_Port). The RFM flow can monitor either one F_Port, all F_Ports, or a specific group of F_Ports defined using a MAPS logical group. Flow Vision will automatically learn the RFM mirror port based on the connected analytics platform. The mirrored traffic uses VC14.

IF RFM is already active in the logical switch, then the RFM filter flow is updated to include the new F_Port. The VTAP frames belonging to the new F_Port being mirrored can be viewed as soon as it is online.

Remote flow monitoring also allows for simultaneous mirroring of VTAP frames (SCSI CMD, STS, 1st XFER_RDY, and 1st RD_RSP) that belong to multiple logical fabrics (VFs). Simultaneous mirroring is only supported onBrocade 6520 Gen 5 platform and on Gen 6 platforms, such as Brocade DCX 8510-4, Brocade DCX 8510-8, Brocade 6520, Brocade X6-4, and Brocade X6-8. Simultaneous mirroring is not supported on Brocade 6510, Brocade 6505, or Brocade G620.

For more information on how to enable RFM, its limitations and restrictions, and to learn more about the Brocade Analytics Monitoring Platform, including which versions of Fabric OS can interact with it, refer to the Brocade Analytics Monitoring Platform documentation or MyBrocade.com.

Activating Flow Mirror flows

To activate an inactive Flow Mirror flow, complete the following steps.

1. Connect to the switch and log in using an account with admin permissions.
2. Enter **flow --activate** *flow_name* **-feature mirror**.

The following example activates the Flow Mirror flow named "Flow1".

```
switch:admin> flow --activate Flow1 -feature mirror
```

> NOTE
> Activating a Flow Mirror flow automatically clears all the flow statistics for that flow.

Automatic activation of a Flow Mirror flow

Flow Mirror automatically activates a mirroring flow under the following conditions:

- On flow creation, unless the flow is created using the **-noactivate** keyword as part of the **flow** command.
- On slot power-on, if the port specified is part of the slot being powered on and the flow was active when the slot was powered off.
- On a High Availability (HA) failover, HA reboot, or a power cycle, if the flow was active when the HA event occurred.

  NOTE
  Flow Mirror will not automatically re-activate a flow if the port types are other than Gen 5 (16 Gbps) or Gen 6 (32 Gbps) F_Ports or F_Port trunks.

Viewing Flow Mirror flows

The following sections describe the different ways of viewing frames mirrored by Flow Mirror.

Summary information view of a Flow Mirror flow

To display the summary view of a Flow Mirror flow, enter **flow --show** *flow_name* **-feature mirror**.

In the summary information view, the first output line lists the flow name and the flow features; the second line lists the source and destination devices and ports, and the flow's directionality; the third line identifies the active features. The following section presents a table with a line for each mirrored frame listing the destination ID, the OXID (originator exchange identifier) of the flow, the RXID (responder exchange identifier) of the flow, the start of frame and end of frame values, the frame type, the LUN, the direction of the frame, and the time stamp of the frame. A learned field column (indicated by an asterisk (*) adjacent to the column name) is added to the output only if the flow definition does not contain the keyword associated with the field for this column. The last data line displays the total number of frames sent to the CPU, the total number of received frames and the total number of transmitted frames.

The following example displays the summary information recorded for the Flow Mirror flow named "fmshow". The LUN column is a learned field in this example.

```
switch:admin> flow --show fmshow -feature mirror
===================================================================================================
Name      : fmshow  Features: mir(Activated) noConfig: Off
Definition: IngrPort(2),SrcDev(*)
Flow Mirror (Activated):
------------------------------------------------------------------------------------------------
| SID(*)  | DID(*)   | OXID | RXID | SOF    | EOF  |Frame_type| LUN(*)  | Dir | Time-Stamp         |
--------------------------------------------------------------------------------------------------
| 040200  | 040f00   | 0ca0 | ffff | SOFi3  | EOFt | SCSIRead | 0000    | Rx  | Jul 12 06:29:13:639|
| 040200  | 040f00   | 0bea | 0be1 | SOFn3  | EOFn | Data     | ----    | Rx  | Jul 12 06:29:13:639|
| 040200  | 040f00   | 0f42 | ffff | SOFi3  | EOFt | SCSIWrite| 0000    | Rx  | Jul 12 06:29:13:639|
| 040200  | 040f00   | 0dc0 | ffff | SOFi3  | EOFt | SCSIRead | 0000    | Rx  | Jul 12 06:29:13:639|
| 040200  | 040f00   | 0592 | ffff | SOFi3  | EOFt | SCSIRead | 0000    | Rx  | Jul 12 06:29:13:639|
......
| 000000  | fffffe   | 801e | ffff | SOFi3  | EOFt | FLOGI    | ----    | Rx  | Jul 12 06:29:45:292|
| 040200  | fffffc   | 8024 | ffff | SOFi3  | EOFt | PLOGI    | ----    | Rx  | Jul 12 06:29:49:411|
| 040200  | fffffd   | 8028 | ffff | SOFi3  | EOFt | ELSframe | ----    | Rx  | Jul 12 06:29:49:411|
| 040200  | fffc04   | 88b9 | ffff | SOFi3  | EOFt | 01       | ----    | Rx  | Jul 12 06:29:51:614|
| 040200  | fffc04   | 802c | ffff | SOFi3  | EOFt | PRLI     | ----    | Rx  | Jul 12 06:29:51:614|
--------------------------------------------------------------------------------------------------
 No of Mirrored Frames : 528, No of RX Mirrored Frames : 528, No of TX Mirrored Frames : 0
-------------------------------------------------------------------------------------------
===================================================================================================
```

Verbose information view of a Flow Mirror flow

To display all the information recorded for a flow, enter **flow --show** *flow_name* **-feature mirror -verbose**.

In the verbose information view, the first output line lists the flow name and the flow features; the second line lists the source and destination devices and ports; the third line identifies the active features. The following lines list for each frame the time stamp of the frame, the direction of the frame, the start of frame and end of frame values, the frame type, and the first 64 bytes (16 words) of the frame. The last data line displays the number of frames sent to the switch Control Processor Unit (CPU), the number of received frames and the number of transmitted frames. If any learned field (indicated by an asterisk (*) adjacent to the column name) is part of the flow definition, then that field is displayed in the show output (there will be a column in the output representing this field).

The following example displays all the information recorded for the Flow Mirror flow named “fmshow”. The LUN column is a learned field in this example, and the “Frame Contents” column has been trimmed.

```
switch:admin> flow --show fmshow -feature mirror -verbose
=========================================================
Name      : fmshow Features: mir(Activated) noConfig: Off
Definition: IngrPort(2),SrcDev(*)
Flow Mirror (Activated):
-------------------------------------------------------------------------------------------------
| Time-Stamp          | Dir  | SOF    | EOF  | Frame_type | LUN(*)  | Frame Contents             |
-------------------------------------------------------------------------------------------------
| Jul 12 06:29:13:637 | Rx   | SOFn3  | EOFn | Data       | ----    | 01040f00 00040200 08000008 … |
| Jul 12 06:29:51:614 | Rx   | SOFi3  | EOFt | PRLI       | ----    | 22fffc04 00040200 01290000 … |
| Jul 12 06:29:51:622 | Rx   | SOFi3  | EOFt | 01         | ----    | 23fffc04 00040200 01990000 … |
| Jul 12 06:29:51:625 | Rx   | SOFi3  | EOFt | 01         | ----    | 23fffc04 00040200 01990000 … |
| Jul 12 06:30:10:951 | Rx   | SOFi3  | EOFt | Abort      | ----    | 81fffffc 00040200 00090000 … |
| Jul 12 06:30:12:970 | Rx   | SOFi3  | EOFt | 20         | ----    | 02fffffc 00040200 20290000 … |
     (output truncated)
-------------------------------------------------------------------------------------------
 No of Mirrored Frames : 530, No of RX Mirrored Frames : 530, No of TX Mirrored Frames : 0
-------------------------------------------------------------------------------------------
=================================================================================================
```

Viewing a Flow Mirror flow in time blocks

To display all the information recorded for a Flow Mirror flow blocked out using a specific time interval, enter **flow --show** *flow_name* **-feature mirror -t** *num*.

The “-t” parameter applies only to Flow Mirror flows. The *num* value is the number of seconds between samples, and can be 6, 7, 8, 9, or 10. The default value is 7. Using this parameter updates the output on the console at the specified time interval until you press **Ctrl+C**. In time interval output, only frames that were mirrored in the time window between t and t+t (for example at 6, 12, 18, 24... seconds) are displayed.

The following examples show the frame rates for the Flow Mirror flow named “fmshow” at 6-second intervals. The time stamp is for the frame, not the time block. At the bottom of each interval block, a data line displays the following:

- The total number of mirrored frames sent to the CPU
- The total number of received mirrored frames
- The total number of frames mirrored during the previous interval

Example of mirroring the flow to the CPU (CFM)

This example shows the result of mirroring the flow to the CPU (CFM).

```
switch:admin> flow --show fmtime_cfm -feature mirror -time 6
==============================================================================
Name      : fmtime_cfm    Features: mir(Activated)    noConfig: Off
Definition: EgrPort(15),SrcDev(0x010e00),DstDev(0x010f00),BiDir
```

```
Flow Mirror (Activated):
-------------------------------------------------------------------------------------
| OXID | RXID | SOF    | EOF       | Frame_type  | LUN(*)  | Dir | Time-Stamp         |
-------------------------------------------------------------------------------------
| 0001 | ffff | SOFn3  | EOFn      | Data        | ----    | Tx  | Jun 05 09:45:34:100 |
| 0044 | ffff | SOFn3  | EOFn      | Data        | ----    | Rx  | Jun 05 09:45:34:100 |
 (output truncated)
| 0001 | ffff | SOFn3  | EOFn      | Data        | ----    | Tx  | Jun 05 09:45:38:110 |
| 0044 | ffff | SOFn3  | EOFn      | Data        | ----    | Rx  | Jun 05 09:45:38:110 |
| 0001 | ffff | SOFn3  | EOFn      | Data        | ----    | Tx  | Jun 05 09:45:38:110 |
-------------------------------------------------------------------------------------------
 No of Mirrored Frames : 1280, No of RX Mirrored Frames : 640, No of TX Mirrored Frames : 640
-------------------------------------------------------------------------------------------
======================================================================================
======================================================================================
Name      : fmtime_cfm    Features: mir(Activated)    noConfig: Off
Definition: EgrPort(15),SrcDev(0x010e00),DstDev(0x010f00),BiDir

Flow Mirror (Activated):
-------------------------------------------------------------------------------------
| OXID | RXID | SOF    | EOF       | Frame_type  | LUN(*)  | Dir | Time-Stamp         |
-------------------------------------------------------------------------------------
| 0001 | ffff | SOFn3  | EOFn      | Data        | ----    | Tx  | Jun 05 09:45:41:100 |
| 0044 | ffff | SOFn3  | EOFn      | Data        | ----    | Rx  | Jun 05 09:45:41:100 |
 (output truncated)
| 0001 | ffff | SOFn3  | EOFn      | Data        | ----    | Tx  | Jun 05 09:45:45:109 |
| 0044 | ffff | SOFn3  | EOFn      | Data        | ----    | Rx  | Jun 05 09:45:45:109 |
-------------------------------------------------------------------------------------------
 No of Mirrored Frames : 1280, No of RX Mirrored Frames : 640, No of TX Mirrored Frames : 640
-------------------------------------------------------------------------------------------
======================================================================================
======================================================================================
Name      : fmtime_cfm    Features: mir(Activated)    noConfig: Off
Definition: EgrPort(15),SrcDev(0x010e00),DstDev(0x010f00),BiDir


Flow Mirror (Activated):
-------------------------------------------------------------------------------------
| OXID | RXID | SOF    | EOF       | Frame_type  | LUN(*)  | Dir | Time-Stamp         |
-------------------------------------------------------------------------------------
| 0001 | ffff | SOFn3  | EOFn      | Data        | ----    | Tx  | Jun 05 09:45:47:100 |
| 0044 | ffff | SOFn3  | EOFn      | Data        | ----    | Rx  | Jun 05 09:45:47:100 |
| 0001 | ffff | SOFn3  | EOFn      | Data        | ----    | Tx  | Jun 05 09:45:47:100 |
 (output truncated)
| 0001 | ffff | SOFn3  | EOFn      | Data        | ----    | Tx  | Jun 05 09:45:51:109 |
| 0044 | ffff | SOFn3  | EOFn      | Data        | ----    | Rx  | Jun 05 09:45:51:109 |
-------------------------------------------------------------------------------------------
 No of Mirrored Frames : 1280, No of RX Mirrored Frames : 640, No of TX Mirrored Frames : 640
-------------------------------------------------------------------------------------------
======================================================================================

(output truncated)
```

Example of mirroring the flow to a local port (LFM)

This example shows the result of mirroring the flow to a local port (LFM).

```
switch:admin> flow --show fmtime_lfm -feature mirror -time 6
======================================================================================
Name      : fmtime_lfm    Features: mir(Activated)    noConfig: Off
Definition: EgrPort(15),SrcDev(0x010e00),DstDev(0x010f00),MirPort(16)

Flow Mirror (Activated):
-------------------------------------------------------------------------------------------
 No of Mirrored Frames : 481442, No of RX Mirrored Frames : 0, No of TX Mirrored Frames : 481442
-------------------------------------------------------------------------------------------
======================================================================================
======================================================================================
Name      : fmtime_lfm    Features: mir(Activated)    noConfig: Off
Definition: EgrPort(15),SrcDev(0x010e00),DstDev(0x010f00),MirPort(16)
```

```
Flow Mirror (Activated):
---------------------------------------------------------------------------------------------
 No of Mirrored Frames : 716244, No of RX Mirrored Frames : 0, No of TX Mirrored Frames : 716244
---------------------------------------------------------------------------------------------
========================================================================================
========================================================================================
Name      : fmtime_lfm    Features: mir(Activated)    noConfig: Off
Definition: EgrPort(15),SrcDev(0x010e00),DstDev(0x010f00),MirPort(16)

Flow Mirror (Activated):
---------------------------------------------------------------------------------------------
 No of Mirrored Frames : 951046, No of RX Mirrored Frames : 0, No of TX Mirrored Frames : 951046
---------------------------------------------------------------------------------------------
========================================================================================
(output truncated)
```

Learning in Flow Mirror flows

Flow Mirror supports learning for both source and destination devices. To specify learning in a Flow Mirror flow, identify the parameter to be learned by using an asterisk inside quotation marks (**"*"**) for the device identifier.

For a flow using learning, if the frame type is specified in the flow definition, the value for **-frametype** must be a fixed value for the flow to work. Refer to Flow frametype parameters on page 24 for a list of valid **-frametype** values.

The following example creates a Flow Mirror flow using the learning capability to mirror traffic from any device to any device that is ingressing through port 1/20.

```
switch:admin> flow --create fmir_learn1-20 -feature mirror -ingrport 1/20 -srcdev "*" -dstdev "*"
```

Deactivating Flow Mirror flows

Flow Mirror flows can be deactivated without deleting them. This allows you to have a “library” of flows that you can activate as needed without having to create them repeatedly.

To manually deactivate a Flow Mirror flow, enter **flow --deactivate** *flow_name* **-feature mirror**.

The following example deactivates the Flow Mirror flow named “Flow1”:

```
switch:admin> flow --deactivate Flow1 -feature mirror
```

Automatic deactivation of a Flow Mirror flow

Flow Vision automatically deactivates a user-defined Flow Mirror flow if any of the following changes to a port defined as part of the mirrored flow occur:

- An ingress or egress port defined in the flow has the port type change to other than an F_Port or F_Port Trunk. You must correct the port type error and then manually reactivate the flow.
- Slot power being powered off for ingress or egress ports. Reactivation occurs automatically when the power is restored.

Customizing Flow Mirror CFM flow frame retention

You can change how frames mirrored to a CPU are retained in the Flow Mirror buffer when it is full.

To have the Flow Mirror buffer overwrite existing frames in the buffer on a first-in-first-out basis when full (replacing the oldest frames with newer ones), enter **flow --control -feature mirror -enable_wrap**.

To have the Flow Mirror buffer discard any additional mirrored frames once the buffer is full, enter **flow --control -feature mirror -disable_wrap**.

The **-enable_wrap** and **-disable_wrap** keywords affect only Flow Mirror flows, but they apply to all Flow Mirror flows, so you cannot specify a flow name. By default, **-enable_wrap** is active.

> ATTENTION
> All Flow Mirror flows must be inactive to use **-enable_wrap** or **-disable_wrap** as part of a **flow --control** command. If any Flow Mirror flow is active when you run the command, it will fail and an error message will be displayed in the interface.

To see the current buffer setting, enter **flow --show -ctrlcfg** (the buffer status appears in the last line of the following example, and is called out).

```
switch:admin> flow --show -ctrlcfg
SimPort Information
-----|------|--------|-------------------------|------------------|------------------
Slot | Port |  PID   |          PWWN           | SID Frame Count  | DID Frame Count |
-----|------|--------|-------------------------|------------------|------------------
  1  |  2   | 050200 | 20:02:00:05:1e:e2:8e:00 |      0K          |      19480K     |
-----|------|--------|-------------------------|------------------|------------------
Addressing mode information
Port Addressing Mode : index
Device Addressing Mode: PID
Flow Generator Information
Size: 2048
Pattern: Random (Default)
Flow mirror Information
  enable_wrap <- current buffer setting
```

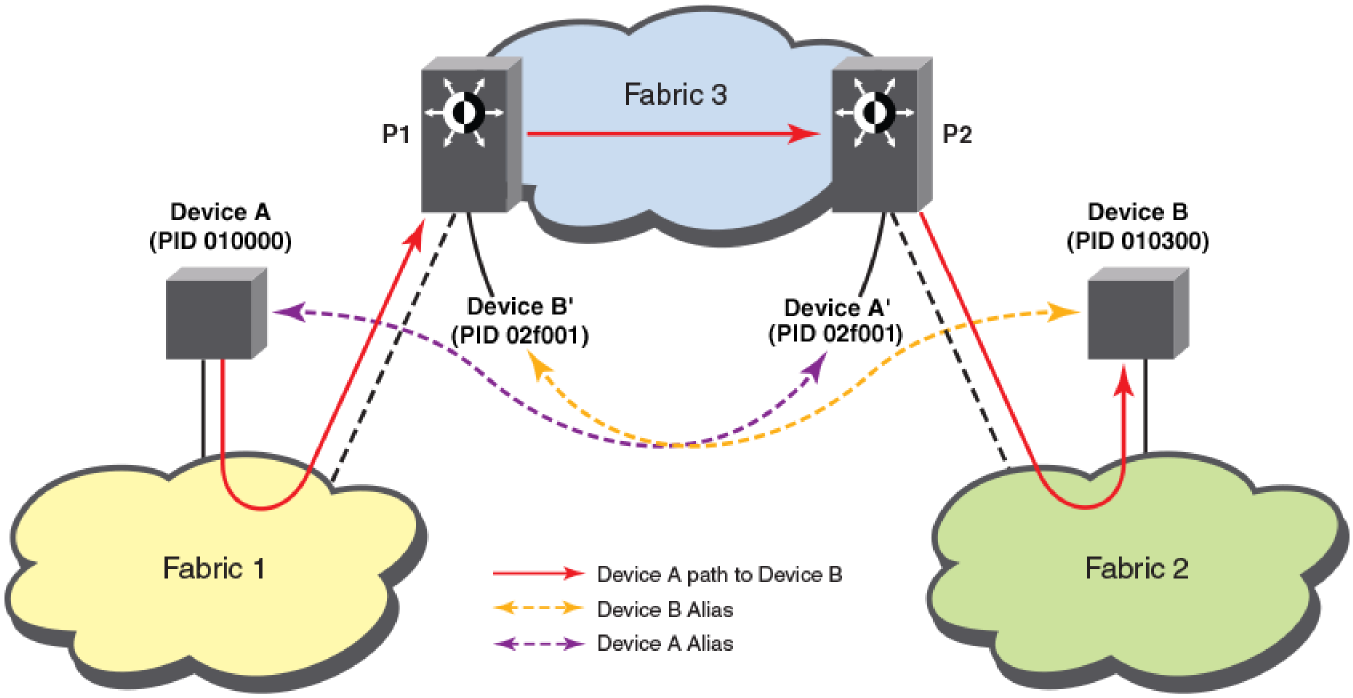
Mirroring traffic flowing to remote fabrics

Flow Vision allows you to mirror traffic that is flowing to a remote fabric.

To mirror traffic that is flowing to a remote fabric, in the flow definition you must specify the imported port ID or real PWWN of the device in the other fabric.

The following illustration shows how such a mirroring situation might be constructed. In this illustration, if the flow definition is made on Device A (a switch in Fabric 1), then you must use the imported port ID for B′ to mirror to a port on device B in Fabric 2. If the flow definition is given in WWN mode, then the real PWWN of Device B can be used instead.

FIGURE 30 Mirroring traffic flowing to a remote fabric



The following example shows the proxy identifications and output of such a flow.

```
switchP1:admin> fcrproxydevshow
  Proxy           WWN              Proxy      Device   Physical    State
 Created                            PID       Exists     PID
in Fabric                                    in Fabric
----------------------------------------------------------------------------
     1    20:00:00:11:0d:07:00:00  02f001        2      010300    Imported
Total devices displayed: 1

switchP2:admin> fcrproxydevshow
  Proxy           WWN              Proxy      Device   Physical    State
 Created                            PID       Exists     PID
in Fabric                                    in Fabric
----------------------------------------------------------------------------
     2    10:00:8c:7c:ff:25:9b:00  02f001        1      010000    Imported
Total devices displayed: 1


switchP2:admin> flow --create fm_edge1_edge2 -srcdev 010000 -dstdev 02f001
                      -feature mirror -ingrport 3/22 -mirrorport 3/43
Mirror feature(s) have been activated.

switch2:admin> flow --show
--------------------------------------------------------------------------------------------------
Flow Name           |Feature| SrcDev  | DstDev  |IngrPt|EgrPt|BiDir|LUN|FrameType  |SFID|DFID|MirPt |
--------------------------------------------------------------------------------------------------
sys_gen_all_simports|gen    |*        |*        |*     |*     |no   |-  |-          |-   |-   |-     |
fm_edge1_edge2      |mir+   |010000   |02f001   |3/22  |-     |no   |-  |-          |-   |-   |3/43  |
--------------------------------------------------------------------------------------------------
  + Denotes feature is currently activated for the flow


switchP2:admin> flow --show fm_edge1_edge2 -feature mirror
=================================================================================================
Name       : fm_edge1_edge2      Features: mir(Activated)        noConfig: Off
```

```
Definition: IngrPort(3/22),SrcDev(0x010000),DstDev(0x02f001),MirPort(3/43)

Flow Mirror (Activated):
---------------------------------------------------------------------------------------------
No of Mirrored Frames : 1903144, No of RX Mirrored Frames : 1903144, No of TX Mirrored Frames : 0
---------------------------------------------------------------------------------------------
==================================================================================================
```

Using Flow Mirror with Access Gateway

With Fabric OS 8.1.0 and later, Access Gateway supports Flow Mirror for Brocade Gen 5 and Brocade Gen 6 switches. Flow Mirror is supported for N_Ports and F_Ports as ingress or egress ports in the flow definition.

Enable a mirror port on the Access Gateway, and then configure a flow to the mirrored port from an Access Gateway port exactly as you would for any switch. Refer to Local flow mirroring on page 125 for general steps.

The following are limitations and considerations for configuring Flow Mirror on Access Gateways:

- Local flow mirroring (LFM) and CPU flow mirroring (CFM) flows are supported in AG mode. Remote flow mirroring (RFM) is not supported.
- A port should be in disabled state before configuring it as a mirror port. The mirror port configuration will be blocked if the port is enabled.
- You do not need to clear the N_Port to F_Port mappings when a port is configured as a mirror port.
- The mirror port can be specified by its port or index number, because both are identical in AG mode. Display ports and their index numbers using the **switchShow** command.
- You cannot configure internal ports on SAN I/O modules as mirror ports.
- Enabling and disabling AG mode is blocked if flows are configured on the switch.
- During a high-availability failover, all flows will be deleted and then reinitiated after failover. If any Flow Mirror flow is active, then mirroring resumes after the failover.
- If an online port is part of the flow definition, Flow Mirror will be enforced.
- If an offline port is part of the flow definition, Flow Mirror is activated, but not enforced.

For more information on Flow Mirror related to Access Gateway, refer to the *Brocade Access Gateway Administration Guide*.

Troubleshooting using Flow Mirror

The following use cases describe how to use Flow Mirror to troubleshoot typical fabric performance problems.

Diagnosing excessive SCSI reserve and release activity

If there is excessive SCSI reserve and release activity in a virtualized environment, you can use Flow Mirror to identify the affected LUNs.

The following example creates a flow to mirror all the SCSI release frames from multiple servers to LUNs on the target on port 1/20. You can then analyze the mirrored frames to determine the impacted LUNs.

```
switch:admin> flow --create flow_scsi -feature mirror -egrport 1/20
                    -srcdev "*" -dstdev "*" -frametype scsiresrel
---------------------------------------------------------------------------------------------------
| SID(*)   | DID(*)  | OXID | RXID | SOF    | EOF    | Frame_type | LUN(*)| Dir | Time-Stamp          |
---------------------------------------------------------------------------------------------------
| b28600   | a2bd00  | 0f27 | ffff | SOFi3  | EOFt   | SCSI3_Rel  | 0003  | Tx  | Jul 16 17:21:47:253|
| b28000   | a2bd00  | 09de | ffff | SOFi3  | EOFt   | SCSI3_Rel  | 0002  | Tx  | Jul 16 17:21:47:253|
```

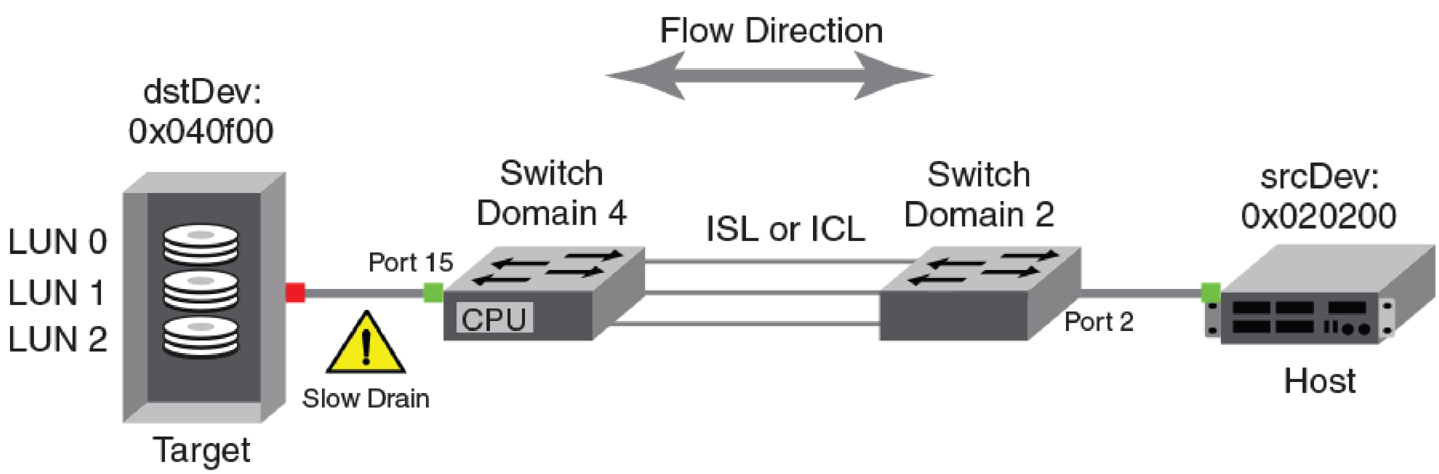
```
| b2c680   | a2bd00 | 0afe | ffff | SOFi3 | EOFt  | SCSI3_Rel  | 0001  | Tx  | Jul 16 17:21:47:253|
| b28600   | a2bd00 | 0f51 | ffff | SOFi3 | EOFt  | SCSI3_Rel  | 0005  | Tx  | Jul 16 17:21:47:253|
| b28000   | a2bd00 | 09f0 | ffff | SOFi3 | EOFt  | SCSI3_Res  | 0002  | Tx  | Jul 16 17:21:47:253|
| b28600   | a2bd00 | 0f1f | ffff | SOFi3 | EOFt  | SCSI3_Rel  | 0004  | Tx  | Jul 16 17:21:47:253|
       (output truncated)
```

Diagnosing a slow-draining F_Port

A slow-draining F_Port can be challenging to diagnose. Bottleneck detection allows you to identify the slow draining F_Port, and Flow Mirroring helps you identify the affected LUN on that F_Port.

The following example creates a flow to mirror traffic passing in both directions from device 0x020200 to F_Port 15 on device 0x040F00, and then displays the output. (The "Frame Contents" column has been trimmed in this example.) The following image provides a illustration of what is happening in the example.

FIGURE 31 A Flow Mirror revealing a slow drain




```
switch:admin> flow --create slwdrn -feature mirror -egrport 15
                    -dstdev 0x040f00 -srcdev 0x020200 -bidir

switch:admin> flow --show slwdrn -feature mirror -verbose
Name      : flow_slowdrain  Features: mir(Active)
Definition: EgrPort(15),SrcDev(0x020200),DstDev(0x040f00),BiDir
Flow Mirror (Active):
-------------------------------------------------------------------------------------------------
| Time-Stamp            | Dir | SOF   | EOF   | Frame_type | LUN(*)  | Frame Contents             |
-------------------------------------------------------------------------------------------------
| Jul 12 06:29:13:637 | Rx   | SOFn3 | EOFn | Data        | ----    | 01040f00 00020200 08000008 … |
| Jul 12 06:29:13:639 | Rx   | SOFi3 | EOFt | SCSIRead    | 0000    | 06040f00 00020200 08290000 … |
| Jul 12 06:29:13:639 | Rx   | SOFi3 | EOFt | SCSIWrite   | 0000    | 06040f00 00020200 08290000 … |
| Jul 12 06:29:13:639 | Rx   | SOFi3 | EOFt | SCSIRead    | 0000    | 06040f00 00020200 08290000 … |
| Jul 12 06:29:13:639 | Rx   | SOFn3 | EOFn | Data        | ----    | 01040f00 00020200 08000008 … |
| Jul 12 06:29:13:639 | Rx   | SOFi3 | EOFt | SCSIRead    | 0000    | 06040f00 00020200 08290000 … |
| Jul 12 06:29:49:398 | Rx   | SOFi3 | EOFt | FLOGI       | ----    | 22fffffe 00000000 01290000 … |
| Jul 12 06:29:49:411 | Rx   | SOFi3 | EOFt | PLOGI       | ----    | 22fffffc 00020200 01290000 … |
| Jul 12 06:29:49:411 | Rx   | SOFi3 | EOFt | ELSframe    | ----    | 22fffffd 00020200 01290000 … |
| Jul 12 06:29:51:614 | Rx   | SOFi3 | EOFt | PRLI        | ----    | 22fffc04 00020200 01290000 … |
     (output truncated)
--------------------------------------------------------------------------------------------
 No of Mirrored Frames : 530, No of RX Mirrored Frames : 530, No of TX Mirrored Frames : 0
--------------------------------------------------------------------------------------------
```

Tracking SCSI commands

There are many reasons why you might want to see the SCSI commands being initiated by a host. For example, such a flow could be used to find all the targets that the host is communicating with. Getting this data can help identify the favorite targets of a host, which would then allow you to provide additional privileges like QoS or TI path creation between those devices.

If you want to see all the SCSI frames being initiated by a host device (for example, the host H1 connected to port F1 on switch SW Dom1 in the following figure), you would create a pair of flow definitions such as the following on the switch SW Dom 2:

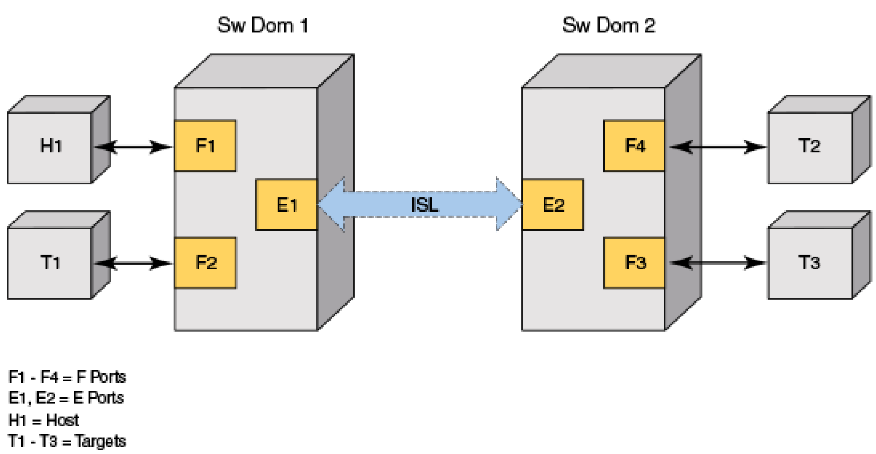
```
switch:admin> flow -create SCSIflow1 -feature mirror -egrport F3 -srcdev H1 -frametype scsicmd
switch:admin> flow -create SCSIflow1 -feature mirror -egrport F4 -srcdev H1 -frametype scsicmd
```

Then you would create a flow definition similar to the following on the switch SW Dom 1:

```
switch:admin> flow -create SCSIflow2 -feature mirror -ingrport F1 -frametype scsicmd
```

This flow will mirror all the frames containing SCSI commands that ingress through port F1. As you can have only one mirror flow active at a time for a chassis or fixed port switch, you cannot trap all the frames initiated by H1 to both T2 and T3 at the same time. This makes it almost impossible to capture all the SCSI command frames initiated by H1. However, by using the **–frametype** option along with **–ingrport** option, you can still track frames between a given initiator and target pair. This will help you view any of the SCSI frames which are being sent out by H1 to any target in the zone.

FIGURE 32 SCSI command tracking

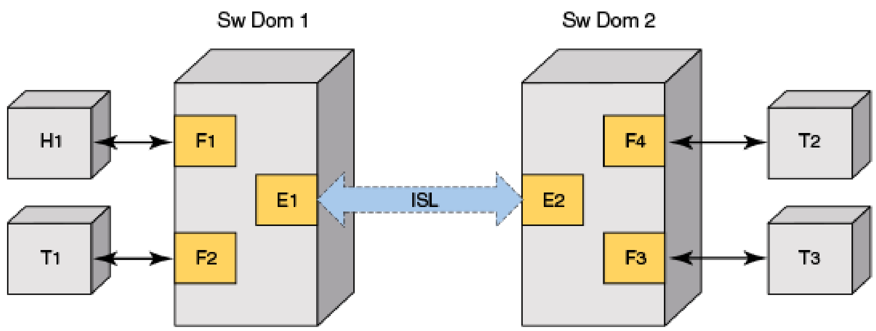


Tracking latency between a host and all connected targets

In order to smooth out application performance, you can track the latency of SCSI Initiator-Target pairs so that you can load balance them.

The following examples capture all the SCSI commands and their status frames initiated by device H1 ingressing through port F1, as illustrated by the following figure. This will capture all the SCSI commands and status frames that can be viewed using Flow Vision. You can then deduce the latency between a SCSI command and its respective status frame for a CPU-mirrored flow at the host port by viewing the output of the Flow Mirror flow using the **flow --show** command, and for a local mirrored flow by viewing the mirrored frame data.

FIGURE 33 SCSI command tracking for latency



Tracking latency using a CPU-mirrored flow

To track the latency using a CPU-mirrored flow, you create the flow and then use the **flow --show** command to view the captured results. In this case, the latency is an approximate latency (best effort latency), because the latency is an approximate latency block completion time. You can use the time stamps on the appropriate frames and deduce the latency from those values. The following example shows a typical command for this purpose; be aware that the output is dependent on the fabric topology.

```
switch:admin> flow --create fm_scsicmdsts -feature mirror
                    -srcdev 10:00:8c:7c:ff:25:aa:00 -ingrport 3/11 -frametype scsicmdsts
Mirror feature(s) have been activated.

switch:admin> flow --show fm_scsicmdsts -feature mirror
===============================================================================================
Name      : fm_scsicmdsts    Features: mir(Activated)    noConfig: Off
Definition: IngrPort(3/11),SrcDev(10:00:8c:7c:ff:25:aa:00),FrameType(scmdsts)

Flow Mirror (Activated):
-----------------------------------------------------------------------------------------------
| DID(*)   | OXID | RXID | SOF    | EOF        | Frame_type    | LUN(*)  | Dir | Time-Stamp          |
-----------------------------------------------------------------------------------------------
| 072300   | 0772 | ffff | SOFi3  | EOFt       | SCSIRead      | 0000    | Rx  | Jun 05 12:23:50:334 |
| 072300   | 0b5c | ffff | SOFi3  | EOFt       | SCSIRead      | 0000    | Rx  | Jun 05 12:23:50:334 |
(output truncated)
| 072b00   | 0c30 | 0d3d | SOFi3  | EOFt       | SCSIGoodSts | ----    | Tx  | Jun 05 12:24:09:347 |
| 07cac0   | 056c | ffff | SOFi3  | EOFt       | SCSIWrite     | 0000    | Rx  | Jun 05 12:24:09:347 |
-----------------------------------------------------------------------------------------------
 No of Mirrored Frames : 5120, No of RX Mirrored Frames : 2567, No of TX Mirrored Frames : 2553
-----------------------------------------------------------------------------------------------
===============================================================================================
```

Tracking latency using a local mirrored flow

To track the latency using a locally mirrored flow, you must first specify the mirror port, and then create the flow using that mirror port. You can then deduce the latency by viewing the mirrored frames. In this case, the latency is close to the exact latency, because the latency I/O Block completion time is close to the actual I/O Block time. The following example shows a typical command for this purpose; the output can be viewed on an analyzer.

```
switch:admin> portcfg mirrorport 9 --enable

switch:admin> flow --create LocalMirrorflow -feature mirror -ingrport F1 -srcdev H1
                    -frametype scsicmdsts -mirrorport 9
```

Troubleshooting protocol errors

You can use Flow Mirror to mirror protocol error frames by creating a flow that tracks the error frames you want to know about.

The following example mirrors only SCSI abort (ABTS) frames egressing through port 1/20. The Flow Mirror output provides you with samples of the ABTS frames for detailed analysis.

```
switch:admin> flow --create fprotocol_errors -feature mirror -egrport 1/20
                    -srcdev "*" -dstdev "*" -frametype abts
```

NOTE
This can also be set up to mirror frames based on the total ABTS count provided by Flow Monitor. The following example creates such a flow to the CPU and then shows the output. If you wanted to mirror to a local port (LFM), you would add **-mirrorport** *port_ID* at the end of the command.

```
switch:admin> flow --create fm_abts -feature mirror -srcdev "*" -dstdev "*"
                    -ingrport 3/4 -frametype abts
Mirror feature(s) have been activated.

switch:admin> flow --show fm_abts
====================================================================================
Name      : fm_abts    Features: mir(Activated)    noConfig: Off
Definition: IngrPort(3/4),SrcDev(*),DstDev(*),FrameType(abts)

Flow Mirror (Activated):
--------------------------------------------------------------------------------------------------
| SID(*)   | DID(*)  | OXID | RXID | SOF   | EOF   | Frame_type | LUN(*)  | Dir | Time-Stamp          |
--------------------------------------------------------------------------------------------------
| 072400   | 07ce40  | f59b | ffff | SOFn3| EOFn | Abort      | ----    | Rx  | Jun 05 13:18:24:982 |
| 072400   | 07ce40  | f59b | ffff | SOFn3| EOFn | Abort      | ----    | Rx  | Jun 05 13:18:24:982 |
| 072400   | 07ce40  | f59b | ffff | SOFn3| EOFn | Abort      | ----    | Rx  | Jun 05 13:18:24:982 |
     (output truncated)
| 072400   | 07ce40  | f59b | ffff | SOFn3| EOFn | Abort      | ----    | Rx  | Jun 05 13:18:43:983 |
| 072400   | 07ce40  | f59b | ffff | SOFn3| EOFn | Abort      | ----    | Rx  | Jun 05 13:18:43:983 |
--------------------------------------------------------------------------------------------------
 No of Mirrored Frames : 5120, No of RX Mirrored Frames : 5120, No of TX Mirrored Frames : 0
--------------------------------------------------------------------------------------------
===================================================================================================
```

Flow Mirror and High Availability

On High Availability (HA) failover, HA reboot, or a power cycle, Flow Mirror will stop mirroring frames until the system recovers; at which point it will resume mirroring. This could be as early as when the **hashow** command indicates that both control processor units (CPUs) are in sync, but it could occur after HA sync, in which case **switchshow** output would then indicate the correct switch port status. All flow statistics are cleared and reset after a failover recovery. Refer to High Availability and Flow Vision on page 45 for more information.