Storage on the QFX Series

Release
13.2X52

Published: 2014-07-15

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Storage on the QFX Series
13.2X52

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at http://www.juniper.net/support/eula.html. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

List of Figures

List of Tables

About the Documentation

Documentation and Release Notes

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at http://www.juniper.net/techpubs/.

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at http://www.juniper.net/books.

Supported Platforms

For the features described in this document, the following platforms are supported:

- QFabric System

Using the Examples in This Manual

If you want to use the examples in this manual, you can use the **load merge** or the **load merge relative** command. These commands cause the software to merge the incoming configuration into the current candidate configuration. The example does not become active until you commit the candidate configuration.

If the example configuration contains the top level of the hierarchy (or multiple hierarchies), the example is a *full example*. In this case, use the **load merge** command.

If the example configuration does not start at the top level of the hierarchy, the example is a *snippet*. In this case, use the **load merge relative** command. These procedures are described in the following sections.

Merging a Full Example

To merge a full example, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration example into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following configuration to a file and name the file **ex-script.conf**. Copy the **ex-script.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   system {
     scripts {
       commit {
         file ex-script.xsl;
       }
     }
   }
   interfaces {
     fxp0 {
       disable;
       unit 0 {
         family inet {
           address 10.0.0.1/24;
         }
       }
     }
   }
   ```

2. Merge the contents of the file into your routing platform configuration by issuing the **load merge** configuration mode command:

   ```
   [edit]
   user@host# load merge /var/tmp/ex-script.conf
   load complete
   ```

Merging a Snippet

To merge a snippet, follow these steps:

1. From the HTML or PDF version of the manual, copy a configuration snippet into a text file, save the file with a name, and copy the file to a directory on your routing platform.

   For example, copy the following snippet to a file and name the file **ex-script-snippet.conf**. Copy the **ex-script-snippet.conf** file to the **/var/tmp** directory on your routing platform.

   ```
   commit {
     file ex-script-snippet.xsl; }
   ```

2. Move to the hierarchy level that is relevant for this snippet by issuing the following configuration mode command:

```
[edit]
user@host# edit system scripts
[edit system scripts]
```

3. Merge the contents of the file into your routing platform configuration by issuing the **load merge relative** configuration mode command:

```
[edit system scripts]
user@host# load merge relative /var/tmp/ex-script-snippet.conf
load complete
```

For more information about the **load** command, see the *CLI User Guide*.

Documentation Conventions

Table 1 on page xxi defines notice icons used in this guide.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.
	Tip	Indicates helpful information.
	Best practice	Alerts you to a recommended use or implementation.

Table 2 on page xxi defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the **configure** command: user@host> **configure**

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
`Fixed-width text like this`	Represents output that appears on the terminal screen.	`user@host>` **show chassis alarms** `No alarms currently active`
Italic text like this	• Introduces or emphasizes important new terms. • Identifies guide names. • Identifies RFC and Internet draft titles.	• A policy *term* is a named structure that defines match conditions and actions. • *Junos OS CLI User Guide* • RFC 1997, *BGP Communities Attribute*
Italic text like this	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# **set system domain-name** *domain-name*
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	• To configure a stub area, include the **stub** statement at the **[edit protocols ospf area area-id]** hierarchy level. • The console port is labeled **CONSOLE**.
< > (angle brackets)	Encloses optional keywords or variables.	**stub <default-metric *metric*>;**
| (pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	**broadcast | multicast** **(*string1* | *string2* | *string3*)**
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	**rsvp { # Required for dynamic MPLS only**
[ ] (square brackets)	Encloses a variable for which you can substitute one or more values.	**community name members [*community-ids*]**

Indention and braces ({ })	Identifies a level in the configuration hierarchy.	[edit]
		routing-options {
		  static {
		    route default {
		      nexthop *address*;
		      retain;
		    }
		  }
		}

; (semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		
Bold text like this	Represents graphical user interface (GUI) items you click or select.	• In the Logical Interfaces box, select **All Interfaces**. • To cancel the configuration, click **Cancel**.

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select **Protocols>Ospf**.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can provide feedback by using either of the following methods:

- Online feedback rating system—On any page at the Juniper Networks Technical Documentation site at http://www.juniper.net/techpubs/index.html, simply click the stars to rate the content, and use the pop-up form to provide us with information about your experience. Alternately, you can use the online feedback form at https://www.juniper.net/cgi-bin/docbugreport/.
- E-mail—Send your comments to techpubs-comments@juniper.net. Include the document or topic name, URL or page number, and software version (if applicable).

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/

- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: http://kb.juniper.net/InfoCenter/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://tools.juniper.net/SerialNumberEntitlementSearch/

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at http://www.juniper.net/cm/.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see http://www.juniper.net/support/requesting-support.html.

PART 1

Overview

CHAPTER 1

Software Features Overview

Overview of Fibre Channel on the QFX Series

Fibre Channel (FC) is a high-speed network technology that interconnects network elements and allows them to communicate with one another. The International Committee for Information Technology Standards (INCITS) T11 Technical Committee sets FC standards.

FC networks provide high-performance characteristics such as lossless transport combined with flexible network topology. FC is primarily used in storage area networks (SANs) because it provides reliable, lossless, in-order frame transport between initiators and targets. FC components include initiators, targets, and FC-capable switches that interconnect FC devices and may also interconnect FC devices with Fibre Channel over Ethernet (FCoE) devices. Initiators originate I/O commands. Targets receive I/O commands. For example, a server can initiate an I/O request to a storage device target.

Juniper Networks QFX Series can function as an FCoE-FC gateway or as an FCoE transit switch.

FCoE transports native FC frames over an Ethernet network by encapsulating the unmodified frames in Ethernet. It also provides protocol extensions to discover FCoE devices through the Ethernet network. FCoE requires that the Ethernet network support data center bridging (DCB) extensions that ensure lossless transport and allow the Layer 2 Ethernet domain to meet the requirements of FC transport.

The FCoE-FC gateway functionality is a licensed feature on the QFX Series. As an FCoE-FC gateway, the switch connects FCoE devices on an Ethernet network to a SAN FC switch.

You do not need a license to use the switch as an FCoE transit switch. As an FCoE transit switch, the switch:

- Is a Layer 2 data center bridging (DCB) switch that can transport FCoE frames.
- Implements FCoE Initialization Protocol (FIP) snooping.
- Connects multiple FCoE endpoints to the FC network.

NOTE: **The QFX3600 switch does not support FC because it does not support native FC port configuration. The QFX3600 switch supports FCoE.**

QFX3500 and QFX3600 Virtual Chassis switches do not support FC or FCoE .

This topic describes:

Fibre Channel Transport Protocol

The Fibre Channel Protocol is a transport protocol that consists of five layers as shown in Table 3 on page 5:

Table 3: Fibre Channel Protocol Layers

FC Protocol Layer	Description
FC-0	Physical (cabling, connectors, and so on)
FC-1	Data link layer
FC-2	Network layer (defines the main protocols)
FC-3	Common services
FC-4	Protocol mapping

The FC protocol layers are generally split into three groups:

- FC-0 and FC-1 are the physical layers.
- FC-2 is the protocol layer, similar to OSI Layer 3.
- FC-3 and FC-4 are the services layers.

The FCoE-FC gateway operates the physical layers and the protocol layer, and provides FIP and service redirection at the services layer.

How FC Works on the QFX Series

The switch connects devices that support FC and Ethernet (such as FCoE servers on an Ethernet network) to an FC SAN, thus converging the Ethernet and FC networks on a single physical network infrastructure. The switch provides the class-of-service (CoS) features needed to handle the different types of traffic appropriately.

To converge FC and Ethernet networks, you can configure the switch as an:

- FCoE-FC Gateway on page 5
- FCoE Transit Switch on page 6
- FCoE VLANs on page 6

FCoE-FC Gateway

When the switch functions as an FCoE-FC gateway, the switch aggregates FCoE traffic and performs the encapsulation and de-encapsulation of native FC frames in Ethernet as it transports the frames between FCoE devices in the Ethernet network and the FC switch. In effect, the switch translates Ethernet to FC and FC to Ethernet.

The gateway receives FC frames encapsulated in Ethernet from FCoE devices through an FCoE VLAN interface composed of one or more 10-Gigabit Ethernet interfaces. The gateway removes the Ethernet encapsulation from the FC frames, and then sends the native FC frames to the FC switch through a native FC interface.

The gateway receives native FC frames from the FC switch on the gateway's native FC interfaces. The gateway encapsulates the native FC frames in Ethernet, and then sends the encapsulated frames to the appropriate FCoE device through the FCoE VLAN interface.

To FCoE devices, the gateway behaves like an FC switch and can present multiple virtual F_Ports (VF_Ports) on a single interface. To an FC switch, the gateway behaves like an FC node that is doing N_Port ID virtualization (NPIV).

FCoE Transit Switch

When the switch functions as an FCoE transit switch, it forwards traffic (including FCoE traffic) based on Layer 2 media access control (MAC) forwarding and is a normal DCB-enabled Layer 2 switch that also performs FIP snooping. The switch aggregates FCoE traffic and passes it through to an FCF. The switch does not remove the Ethernet encapsulation from the FC frames, but it does preserve the class of service (CoS) required to transport FC frames.

The switch inspects (snoops) FIP information in order to create filters that permit only valid FCoE traffic to flow through the switch between FCoE devices and the FCF. The switch does not use native FC ports because the FC frames are encapsulated in Ethernet when they flow between the FCoE devices and the FCF. Virtual point-to-point links between each FCoE device and the FCF pass transparently through the switch, so the switch is not seen as a terminating point or an intermediate point by FCoE devices or by the FCF.

FCoE VLANs

On the QFX Series, all FCoE traffic must travel in a VLAN dedicated to transporting only FCoE traffic. Only FCoE interfaces should be members of an FCoE VLAN. Ethernet traffic that is not FCoE or FIP traffic must travel in a different VLAN.

NOTE: **On a QFX3500 or QFabric system Node device, the same VLAN cannot be used in both transit switch mode and FCoE-FC gateway mode.**

NOTE: FCoE VLANs (any VLAN that carries FCoE traffic) support only Spanning Tree Protocol (STP) and link aggregation group (LAG) Layer 2 features.

FCoE traffic cannot use a standard LAG because traffic might be hashed to different physical LAG links on different transmissions. This breaks the (virtual) point-to-point link that Fibre Channel traffic requires. If you configure a standard LAG interface for FCoE traffic, FCoE traffic might be rejected by the FC SAN.

QFabric systems support a special LAG called an FCoE LAG, which enables you to transport FCoE traffic and regular Ethernet traffic (traffic that is not FCoE traffic) across the same link aggregation bundle. Standard LAGs use a hashing algorithm to determine which physical link in the LAG is used for a transmission, so communication between two devices might use different physical links in the LAG for different transmissions. An FCoE LAG ensures that FCoE traffic uses the same physical link in the LAG for requests and replies in order to preserve the virtual point-to-point link between the FCoE device converged network adapter (CNA) and the FC SAN switch across the QFabric system Node device. An FCoE LAG does not provide load balancing or link redundancy for FCoE traffic. However, regular Ethernet traffic uses the standard hashing algorithm and receives the usual LAG benefits of load balancing and link redundancy in an FCoE LAG.

NOTE: IGMP snooping is enabled by default on all VLANs in all software versions before Junos OS R13.2. Disable IGMP snooping on FCoE VLANs if you are using software that is older than 13.2.

You can configure more than one FCoE VLAN, but any given virtual link must be in only one FCoE VLAN.

NOTE: All 10-Gigabit Ethernet interfaces that connect to FCoE devices must have a native VLAN configured in order to transport FIP traffic, because FIP VLAN discovery and notification frames are exchanged as untagged packets.

BEST PRACTICE: Only FCoE traffic is permitted on the FCoE VLAN. A native VLAN might need to carry untagged traffic of different types and protocols. Therefore, it is a good practice to keep the native VLAN separate from FCoE VLANs.

Supported FC Features and Functions

QFX Series supports the following features and functionality:

- As an FCoE-FC gateway:
  - DCB, including Data Center Bridging Capability Exchange protocol (DCBX), priority-based flow control (PFC), enhanced transmission service (ETS), and 10-Gigabit Ethernet interfaces
  - FCoE Initialization Protocol (FIP)
  - Proxy for FCoE devices when communicating with FC switches and acts as a proxy for FC switches when communicating with FCoE devices
  - Up to 12 native FC interfaces per QFX3500 switch (each interface can be configured as a 2-Gigabit, 4-Gigabit, or 8-Gigabit Ethernet interface)
- As an FCoE transit switch:
  - DCB functions
  - FIP snooping
  - Transparent Layer 2 MAC forwarding of FCoE frames

Lossless Transport Support

The QFX Series supports up to six lossless forwarding classes. For lossless transport, you must enable PFC on the IEEE 802.1p code point of lossless forwarding classes. The following limitations apply to support lossless transport:

- The external cable length from the QFX3500 or QFabric system Node device to other devices cannot exceed 300 meters.
- The internal cable length from the QFabric system Node device to the QFabric system Interconnect device cannot exceed 150 meters.
- For FCoE traffic, the interface maximum transmission unit (MTU) must be at least 2180 bytes to accommodate the packet payload, headers, and checks.

Related Documentation

- Understanding Fibre Channel on page 12
- Understanding an FCoE-FC Gateway on page 29
- Understanding FCoE Transit Switch Functionality on page 25
- Understanding FCoE on page 19
- Understanding DCB Features and Requirements on page 16
- Overview of FIP on page 9
- Understanding VN_Port to VF_Port FIP Snooping on an FCoE Transit Switch on page 80
- Understanding CoS Flow Control (Ethernet PAUSE and PFC) on page 121
- Understanding Interfaces on an FCoE-FC Gateway on page 50

- Understanding FCoE LAGs on page 158
- Understanding Fibre Channel Terminology on page 131

Overview of FIP

Fibre Channel over Ethernet (FCoE) Initialization Protocol (FIP) is a Layer 2 protocol that establishes and maintains Fibre Channel (FC) virtual links between pairs of FCoE devices such as server FCoE Nodes (ENodes) and FC switches. FIP can also establish and maintain virtual links between FCoE devices and an FCoE-FC gateway such as the QFX Series, where the gateway acts on behalf of the FC switch.

FIP enables FCoE devices to discover one another and to initialize and maintain virtual links over a physical Ethernet network. This allows FCoE devices in the Ethernet network to access storage devices in the FC storage area network (SAN).

FIP solves the problem presented by the FC requirement for point-to-point connections (FC does not permit point-to-multipoint connections) by creating a unique virtual link for each connection between an ENode VN_Port and an FC switch VF_Port. Multiple virtual links can use a single physical link and virtual links can traverse Ethernet transit (passthrough) switches while appearing to be direct point-to-point connections to the FC switch.

FIP has its own EtherType (0x8914) to distinguish its traffic from payload-carrying FCoE traffic and other Ethernet traffic. FIP operations occur on a per-VLAN basis.

For more details about FIP, see the Technical Committee T11 organization document *Fibre Channel Backbone - 5 (FC-BB-5) Rev 2.00* available at http://www.t11.org/ftp/t11/pub/fc/bb-5/09-056v5.pdf.

Related Documentation

- Overview of Fibre Channel on the QFX Series on page 4
- Understanding Fibre Channel on page 12
- Understanding FIP Functions on page 38
- Understanding FIP Implementation on page 42
- Understanding FIP Parameters on an FCoE-FC Gateway on page 46
- Understanding Fibre Channel Virtual Links on page 49
- Understanding FCoE on page 19
- Understanding an FCoE-FC Gateway on page 29
- Configuring FIP on an FCoE-FC Gateway on page 283
- Understanding Fibre Channel Terminology on page 131

CHAPTER 2

Fibre Channel, FCoE, and FIP

Understanding Fibre Channel

Fibre Channel (FC) is a serial I/O interconnect network technology capable of supporting multiple protocols. It is used primarily for storage area networks (SANs). The committee standardizing FC is the International Committee for Information Technology Standards (INCITS).

When configured as a Fibre Channel over Ethernet (FCoE)-FC gateway, the QFX Series supports the transport of native FC traffic between FC switches and the gateway's native FC interfaces.

NOTE: **The QFX3600 switch does not support FC because it does not support native FC port configuration. In addition, QFX3500 and QFX3600 Virtual Chassis switches do not support FC.**

FC concepts include:

- FC Fabrics on page 12
- FC Port Types on page 12
- FC Switches on page 13
- Adapters on page 13
- N_Port ID Virtualization (NPIV) on page 13
- FC Services on page 14

FC Fabrics

An FC fabric is a switched network topology that interconnects FC devices using FC switches, usually to create a SAN. An FC switch is a Layer 3 network switch that is compatible with the FC protocol, forwards FC traffic, and provides FC services to the components of the FC fabric. FC devices are usually servers or storage devices such as disk arrays.

Switches called FCoE forwarders (FCFs) perform a subset of FC switch functions. An FCF is a Layer 3 network switch that is compatible with the FC protocol and forwards FC traffic, but does not provide network services.

When configured as an FCoE-FC gateway, the QFX Series acts a proxy for the FCF functionality of an FC switch. The gateway provides FCoE devices on the Ethernet network access to the FC network without requiring the FC switches in the SAN to support Ethernet interfaces. The gateway is not an FCF and does not provide FC services.

FC network design often uses two fabrics (dual-rail topology) for redundancy. The two fabrics connect to edge devices but are otherwise unconnected, so that if one fabric goes down, the other fabric can continue to provide connectivity.

FC Port Types

The QFX Series supports the following FC port types:

- N_Port—An N_Port is a port on the node of an FC device such as a server or a storage device and is also known as a node port.
- F_Port—An F_Port is a port on an FC switch that connects to an FC device N_Port in a point-to-point connection. F_Ports are also known as fabric ports.

These port types are a subset of the existing FC port types that can be supported in an FC fabric.

FC Switches

FC switches provide FC services to the FC network. FC switches forward Layer 3 traffic. They may transport a combination of native FC traffic and other traffic, such as Internet Small Computer Systems Interface (iSCSI) or FCoE, or they may transport only native FC traffic. When an FC switch supports FCoE, it combines FCoE termination functions with the FC stack on an FC switching element. This is also known as a dual-stack switch.

When FC switches support FCoE, they present virtual FC interfaces in the form of virtual F_Ports (VF_Ports) to the FCoE nodes (ENodes) on FCoE devices. A VF_Port is an endpoint in a virtual point-to-point connection with an ENode virtual N_Port (VN_Port). A VF_Port emulates a native FC F_Port and performs similar functions. A VF_Port is an intermediate port in a connection between an FCoE device such as a server in the Ethernet network and a storage device in the FC SAN.

FC switches that support FCoE contain at least one lossless Ethernet media access controller (MAC) paired with an FCoE controller. The lossless Ethernet MAC implements Ethernet extensions to avoid frame loss due to congestion. The FCoE controller instantiates and terminates virtual port instances as they are needed. Each VF_Port instance has one unique virtual link to an ENode VN_Port.

FCoE support also requires one FCoE Link End Point (LEP) for each VF_Port connection. An FCoE LEP is a virtual FC interface mapped onto the physical Ethernet interface. It transmits and receives FCoE frames on the virtual link, and handles FC frame encapsulation for traffic going from the FC switch to the FCoE device and frame de-encapsulation of traffic received from the FCoE device.

When you configure the QFX Series as an FCoE-FC gateway, the gateway performs these FC-to-Ethernet and Ethernet-to-FC conversion functions so that the FC switch does not need Ethernet (FCoE) ports.

Adapters

FC host bus adapters (HBAs) in FC switches and devices perform functions similar to those of Ethernet adapters in Ethernet switches and devices. Switches that perform FCoE functions and FCoE devices have converged network adapters (CNAs) that support both native FC and Ethernet functionality.

N_Port ID Virtualization (NPIV)

FC requires a unique point-to-point link between the FC switch (F_Port) and each host N_Port. In order to avoid using one physical link for each F_Port to N_Port connection, the port connections must be virtualized so that they can share a physical link while maintaining logical separation.

FC accomplishes this by enabling you to create an independent virtual link for each FC session by mapping each session to a virtualized N_Port. This process is called N_Port ID virtualization (NPIV).

NPIV makes each virtual link look like a dedicated point-to-point link. In this way, multiple FC devices and multiple applications or virtual machines (VMs) on a single FC device can connect to an FC switch using one physical port instead of using a physical port for each connection. The virtual link creates a secure boundary between traffic from different sources on a single physical connection.

NPIV works by creating a unique virtual port identifier for each logical connection on a physical port. Conceptually, this is similar to splitting a single physical interface into multiple logical interfaces or subinterfaces. A virtual port identifier consists of the port's unique worldwide name (WWN) combined with a Fibre Channel ID (FCID) that the FC switch assigns to the virtual connection. This creates a virtual host bus adapter (HBA) for each virtual link that uniquely identifies the link to the FC switch.

FC Services

When you configure the QFX Series as an FCoE-FC gateway, the gateway connects FCoE devices in the Ethernet network to the FC fabric. The gateway does not provide FC services directly. The gateway logs in to the FC fabric and obtains FC services from the FC fabric, including:

- Management servers
  - Zone server—Defines which devices can connect to each other in the FC fabric.
  - Fabric configuration server—Discovers FC fabric topology and attributes.
  - Policy server—Distributes the rules for administering, managing, and controlling access to FC fabric resources.
  - HBA management server—Registers HBA information with the FC fabric.
- Domain manager—Allocates domain IDs to virtual switches.
- Fabric login server—Provides login services to the gateway so that the native FC ports on the gateway can perform initial fabric login (FLOGI) to the FC fabric and subsequent fabric discovery (FDISC) logins for the physical and virtual ports on the FCoE devices in the Ethernet network. This includes allocating Fibre Channel IDs (FCIDs) to ports.
- Name server—Discovers, registers, and unregisters N_Port attributes, including the attributes of the native FC ports on the gateway that connect to the FC fabric.
- Event server—Validates incoming events to ensure transaction integrity.
- Time server—Maintains a common time for devices in the FC fabric.
- Fabric controller
  - Fabric Shortest Path First (FSPF)—The FC fabric provides link-state path selection to the gateway.

- State change notification (SCN) / registered state change notification server (RSCN)—Notifies the appropriate nodes when new devices come online, when other nodes fail, or when changes on an online node affect system operation.

Related Documentation

- Overview of Fibre Channel on the QFX Series on page 4
- Understanding FCoE on page 19
- Understanding an FCoE-FC Gateway on page 29
- Understanding Fibre Channel Terminology on page 131

Understanding DCB Features and Requirements

Data center bridging (DCB) is a set of enhancements to the IEEE 802.1 bridge specifications. DCB modifies and extends Ethernet behavior to support I/O convergence in the data center. I/O convergence includes but is not limited to the transport of Ethernet LAN traffic and Fibre Channel (FC) storage area network (SAN) traffic on the same physical Ethernet network infrastructure.

Video: **What is Data Center Bridging?**

A converged architecture saves cost by reducing the number of networks and switches required to support both types of traffic, reducing the number of interfaces required, reducing cable complexity, and reducing administration activities.

The Juniper Networks QFX Series supports the DCB features required to transport converged Ethernet and FC traffic while providing the class-of-service (CoS) and other characteristics FC requires for transmitting storage traffic. To accommodate FC traffic, DCB specifications provide:

- A flow control mechanism called priority-based flow control (PFC, described in IEEE 802.1Qbb) to help provide lossless transport.
- A discovery and exchange protocol for conveying configuration and capabilities among neighbors to ensure consistent configuration across the network, called Data Center Bridging Capability Exchange protocol (DCBX), which is an extension of Link Layer Data Protocol (LLDP, described in IEEE 802.1AB).
- A bandwidth management mechanism called enhanced transmission selection (ETS, described in IEEE 802.1Qaz).
- A congestion management mechanism called quantized congestion notification (QCN, described in IEEE 802.1Qau).

The switch supports the PFC, DCBX, and ETS standards but does not support QCN. The switch also provides the high-bandwidth interfaces (10-Gbps minimum) required to support DCB and converged traffic.

This topic describes the DCB standards and requirements the switch supports:

- Lossless Transport on page 16
- ETS on page 17
- DCBX on page 18

Lossless Transport

FC traffic requires lossless transport (defined as no frames dropped because of congestion). Standard Ethernet does not support lossless transport, but the DCB extensions to Ethernet along with proper buffer management enable an Ethernet network to provide the level of class of service (CoS) necessary to transport FC frames encapsulated in Ethernet over an Ethernet network.

This section describes these factors in creating lossless transport over Ethernet:

- PFC on page 17
- Buffer Management on page 17
- Physical Interfaces on page 17

PFC

PFC is a link-level flow control mechanism similar to Ethernet PAUSE (described in IEEE 802.3x). Ethernet PAUSE stops all traffic on a link for a period of time. PFC enables you to divide traffic on a link into eight priorities and stop the traffic of a selected priority without stopping the traffic assigned to other priorities on the link.

Pausing the traffic of a selected priority enables you to provide lossless transport for traffic assigned that priority and at the same time use standard lossy Ethernet transport for the rest of the link traffic.

Buffer Management

Buffer management is critical to the proper functioning of PFC, because if buffers are allowed to overflow, frames are dropped and transport is not lossless.

For each lossless flow priority, the switch requires sufficient buffer space to:

- Store frames sent during the time it takes to send the PFC pause frame across the cable between devices.
- Store the frames that are already on the wire when the sender receives the PFC pause frame.

The propagation delay due to cable length and speed, as well as processing speed, determines the amount of buffer space needed to prevent frame loss due to congestion.

The switch automatically sets the threshold for sending PFC pause frames to accommodate delay from cables as long as 150 meters (492 feet) and to accommodate large frames that might be on the wire when the switch sends the pause frame. This ensures that the switch sends pause frames early enough to allow the sender to stop transmitting before the receive buffers on the switch overflow.

Physical Interfaces

The switch supports 10-Gbps, full-duplex interfaces. The switch enables DCB capability only on 10-Gbps (or faster) Ethernet interfaces.

ETS

PFC divides traffic into up to eight separate streams (priorities, configured on the switch as forwarding classes) on a physical link. ETS enables you to manage the link bandwidth by:

- Grouping the priorities into priority groups (configured on the switch as forwarding class sets).

- Specifying the bandwidth available to each of the priority groups as a percentage of the total available link bandwidth.
- Allocating the bandwidth to the individual priorities in the priority group.

The available link bandwidth is the bandwidth remaining after servicing strict-high priority flows. We recommend that you always configure a shaping rate to limit the amount of bandwidth a strict-high priority flow can consume by including the **shaping-rate** statement in the **[edit class-of-service schedulers]** hierarchy on the strict-high priority scheduler. This prevents a strict-high priority from starving other queues on the port.

Managing link bandwidth with ETS provides several advantages:

- There is uniform management of all types of traffic on the link, both congestion-managed traffic and standard Ethernet traffic.
- When a priority group does not use all of its allocated bandwidth, other priority groups on the link can use that bandwidth as needed.

  When a priority in a priority group does not use all of its allocated bandwidth, other priorities in the group can use that bandwidth.

  The result is better bandwidth utilization, because priorities that consist of bursty traffic can share bandwidth during periods of low traffic transmission instead of consuming their entire bandwidth allocation when traffic loads are light.
- You can assign traffic types with different service needs to different priorities so that each traffic type receives appropriate treatment.
- Strict priority traffic retains its allocated bandwidth.

DCBX

DCB devices use DCBX to exchange configuration information with directly connected peers (switches and endpoints such as servers). DCBX is an extension of LLDP. If you disable LLDP on an interface, that interface cannot run DCBX. If you attempt to enable DCBX on an interface on which LLDP is disabled, the configuration commit fails.

DCBX can:

- Discover the DCB capabilities of peers.
- Detect DCB feature misconfiguration or mismatches between peers.
- Configure DCB features on peers.

You can configure DCBX operation for PFC, ETS, and for Layer 2 and Layer 4 applications such as FCoE and iSCSI. DCBX is enabled or disabled on a per-interface basis.

Related Documentation

- Overview of Fibre Channel on the QFX Series on page 4
- Understanding FCoE on page 19
- *Understanding CoS Hierarchical Port Scheduling (ETS)*
- Understanding CoS Flow Control (Ethernet PAUSE and PFC) on page 121

- Understanding DCBX on page 108
- Understanding Fibre Channel Terminology on page 131
- Example: Configuring CoS PFC for FCoE Traffic on page 186

Understanding FCoE

Fibre Channel over Ethernet (FCoE) is a method of supporting converged Fibre Channel (FC) and Ethernet traffic on a data center bridging (DCB) network. FCoE encapsulates unmodified FC frames in Ethernet to transport the FC frames over a physical Ethernet network. The T11 Technical Committee, which is the International Committee for Information Technology Standards (INCITS) committee responsible for FC interfaces, developed the FCoE standard to provide a method for transporting FC frames over a DCB network. The T11 document *Fibre Channel Backbone - 5 (FC-BB-5) Rev 2.00* at http://www.t11.org/ftp/t11/pub/fc/bb-5/09-056v5.pdf provides details about the FCoE version 1 standard.

NOTE: The switch does not support T11 Annex F *FCoE Pre-FIP Virtual Link Instantiation Protocol*.

To the Ethernet network, an FCoE frame is the same as any other Ethernet frame because the Ethernet encapsulation provides the header information needed to forward the frames. However, to achieve the lossless behavior that FC transport requires, the Ethernet network must conform to DCB standards.

DCB standards create an environment over which FCoE can transport native FC traffic encapsulated in Ethernet while preserving the mandatory class of service (CoS) and other characteristics that FC traffic requires.

Supporting FCoE in a DCB network requires that the FCoE devices in the Ethernet network and the FC switches at the edge of the SAN network handle both Ethernet and native FC traffic. To handle Ethernet traffic, an FC switch does one of two things:

- Incorporates FCoE interfaces.
- Uses an FCoE-FC gateway such as a QFX Series to de-encapsulate FCoE traffic from FCoE devices into native FC and to encapsulate native FC traffic from the FC switch into FCoE and forward it to FCoE devices through the Ethernet network.

NOTE: QFX3500 and QFX3600 Virtual Chassis switches do not support FCoE .

FCoE concepts include:

- FCoE Devices on page 20
- FCoE Frames on page 21

• Virtual Links on page 22
• FCoE VLANs on page 22

FCoE Devices

Each FCoE device has a converged network adapter (CNA) that combines the functions of an FC host bus adapter (HBA) and a lossless Ethernet network interface card (NIC) with 10-Gbps Ethernet ports. The portion of the CNA that handles FCoE traffic is called an FCoE Node (ENode). An ENode combines FCoE termination functions and the client part of the FC stack on the CNA.

ENodes present virtual FC interfaces to FC switches in the form of virtual N_Ports (VN_Ports). A VN_Port is an endpoint in a virtual point-to-point connection called a virtual link. The other endpoint of the virtual link is an FC switch (or FCF) port. A VN_Port emulates a native FC N_Port and performs similar functions: handling the creation, detection, and flow of messages to and from the FC switch. A single ENode can host multiple VN_Ports. Each VN_Port has a separate, unique virtual link with a FC switch.

ENodes contain at least one lossless Ethernet media access controller (MAC). Each Ethernet MAC is paired with an FCoE controller. The lossless Ethernet MAC is a full-duplex Ethernet MAC that implements Ethernet extensions to avoid frame loss due to congestion and supports frames of at least 2500 bytes. The FCoE controller instantiates and terminates VN_Port instances dynamically as they are needed for FCoE sessions. Each VN_Port instance has a unique virtual link to an FC switch.

NOTE: **A *session* is a fabric login (FLOGI) or fabric discovery (FDISC) login to the FC SAN fabric. Session does not refer to end-to-end server-to-storage sessions.**

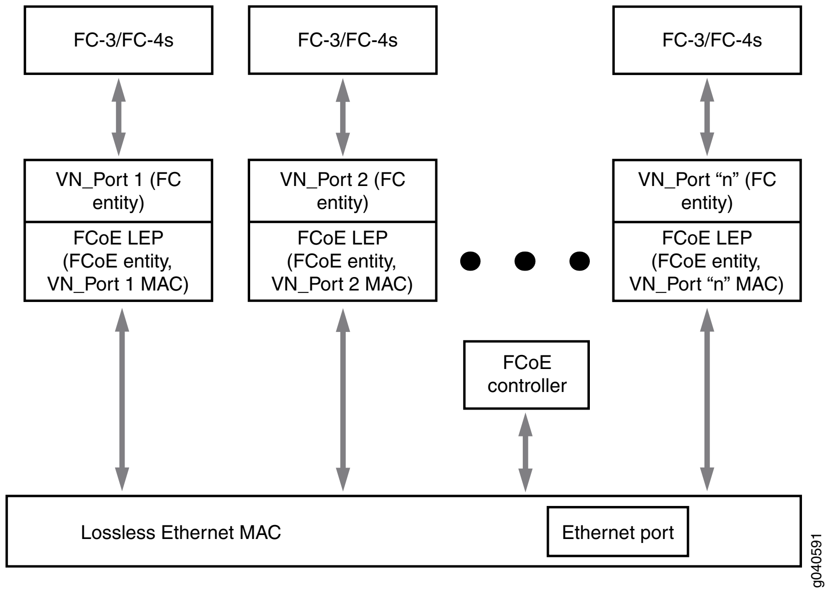
ENodes also contain one FCoE link end point (LEP) for each VN_Port connection. An FCoE LEP is a virtual FC interface mapped onto the physical Ethernet interface.

An FCoE LEP:

- Transmits and receives FCoE frames on the virtual link.
- Handles FC frame encapsulation for traffic going from the server to the FC switch.
- Performs frame de-encapsulation of traffic received from the FC switch.

Figure 1 on page 21 shows a block diagram of the major ENode components.

Figure 1: ENode Components



FCoE Frames

The FCoE protocol specification replaces the FC0 and FC1 layers of the FC stack with Ethernet, but retains the FC frame header. Retaining the FC frame header enables the FC frame to pass directly to a native FC SAN after de-encapsulation. The FCoE header carries the FC start of file (SOF) bits and end of file (EOF) bits in an encoded format. FCoE supports two frame types, control frames and data frames. FCoE Initialization Protocol (FIP) carries all of the discovery and fabric login frames.

FIP control frames handle FCoE device discovery, initializing communication, and maintaining communication. They do not carry a data payload. FIP has its own EtherType (0x8914) to distinguish FIP traffic from FCoE traffic and other Ethernet traffic. To establish communication, the ENode uses the globally unique MAC address assigned to it by the CNA manufacturer.

After FIP establishes a connection between FCoE devices, the FCoE data frames handle the transport of the FC frames encapsulated in Ethernet. FCoE also has its own EtherType (0x8906) to distinguish FCoE frames from other Ethernet traffic and ensure the in-order frame handling that FC requires. FCoE frames include:

- 2112 bytes FC payload
- 24 bytes FC header
- 14 bytes standard Ethernet header
- 14 bytes FCoE header
- 8 bytes cyclic redundancy check (CRC) plus EOF
- 4 bytes VLAN header
- 4 bytes frame check sequence (FCS)

The payload, headers, and checks add up to 2180 bytes. Therefore, interfaces that carry FCoE traffic should have a configured maximum transmission unit (MTU) of 2180 or larger. An MTU size of 2180 bytes is the minimum size; some network administrators prefer an MTU of 2240 or 2500 bytes.

Virtual Links

Native FC uses point-to-point physical links between FC devices. In FCoE, virtual links replace the physical links. A virtual link emulates a point-to-point link between two FCoE device endpoints, such as a server VN_Port and an FC switch (or FCF) VF_Port.

Each FCoE interface can support multiple virtual links. The MAC addresses of the FCoE endpoints (the VN_Port and the VF_Port) uniquely identify each virtual link and allow traffic for multiple virtual links to share the same physical link while maintaining data separation and security.

A virtual link exists in one FCoE VLAN and cannot belong to more than one VLAN. Although the FC switch and the FCoE device detect a virtual link as a point-to-point connection, virtual links do not need to be direct connections between a VF_Port and a VN_Port. A virtual link can traverse one or more transit switches, also known as passthrough switches. A transit switch can transparently aggregate virtual links while still appearing and functioning as a point-to-point connection to the FCoE devices. However, a virtual link must remain within a single Layer 2 domain.

FCoE VLANs

On the QFX Series, all FCoE traffic must travel in a VLAN dedicated to transporting only FCoE traffic. Only FCoE interfaces should be members of an FCoE VLAN. Ethernet traffic that is not FCoE or FIP traffic must travel in a different VLAN.

NOTE: On a QFX3500 or QFabric system Node device, the same VLAN cannot be used in both transit switch mode and FCoE-FC gateway mode.

NOTE: FCoE VLANs (any VLAN that carries FCoE traffic) support only Spanning Tree Protocol (STP) and link aggregation group (LAG) Layer 2 features.

FCoE traffic cannot use a standard LAG because traffic might be hashed to different physical LAG links on different transmissions. This breaks the (virtual) point-to-point link that Fibre Channel traffic requires. If you configure a standard LAG interface for FCoE traffic, FCoE traffic might be rejected by the FC SAN.

QFabric systems support a special LAG called an FCoE LAG, which enables you to transport FCoE traffic and regular Ethernet traffic (traffic that is not FCoE traffic) across the same link aggregation bundle. Standard LAGs use a hashing algorithm to determine which physical link in the LAG is used for a transmission, so communication between two devices might use different physical links in the LAG for different transmissions. An FCoE LAG ensures that FCoE traffic uses the same physical link in the LAG for requests and replies in order to preserve the virtual point-to-point link between the FCoE device converged network adapter (CNA) and the FC SAN switch across the QFabric system Node device. An FCoE LAG does not provide load balancing or link redundancy for FCoE traffic. However, regular Ethernet traffic uses the standard hashing algorithm and receives the usual LAG benefits of load balancing and link redundancy in an FCoE LAG.

NOTE: IGMP snooping is enabled by default on all VLANs in all software versions before Junos OS R13.2. Disable IGMP snooping on FCoE VLANs if you are using software that is older than 13.2.

You can configure more than one FCoE VLAN, but any given virtual link must be in only one FCoE VLAN.

NOTE: All 10-Gigabit Ethernet interfaces that connect to FCoE devices must have a native VLAN configured in order to transport FIP traffic, because FIP VLAN discovery and notification frames are exchanged as untagged packets.

On QFX5100 switches that use the Enhanced Layer 2 Software (ELS) CLI, it is not sufficient only to configure the native VLAN on the interface, the interface must also be configured as a member of the native VLAN. (This is because the ELS CLI does not support tagged-access interface mode, so interfaces that are members of FCoE VLANs must use trunk mode, and trunk port interfaces must be explicitly included as members of a native VLAN.)

In addition, the VLAN ID must match the native VLAN ID that you configure on the physical interface. For example, to configure a native VLAN with an ID of 20 on interface xe-0/0/15 that is a member of an FCoE VLAN, you must include both of the following statements in the configuration:

1. Configure the native VLAN on the interface:

   `user@switch#` **set interfaces xe-0/0/15 native-vlan-id 20**

   (The equivalent configuration statement on a non-ELS device such as a QFX3500 or QFX3600 switch would be **set interfaces xe-0/0/15 unit 0 family ethernet-switching native-vlan-id 20**.)

2. Configure the port as a member of the native VLAN (this step is not required on QFX3500 and QFX3600 switches):

   `user@switch#` **set interfaces xe-0/0/15 unit 0 family ethernet-switching vlan members 20**

BEST PRACTICE: Only FCoE traffic is permitted on the FCoE VLAN. A native VLAN might need to carry untagged traffic of different types and protocols. Therefore, it is a good practice to keep the native VLAN separate from FCoE VLANs.

Related Documentation

- Overview of Fibre Channel on the QFX Series on page 4
- Understanding Fibre Channel on page 12
- Understanding DCB Features and Requirements on page 16
- Understanding FCoE Transit Switch Functionality on page 25
- Understanding VN_Port to VF_Port FIP Snooping on an FCoE Transit Switch on page 80
- Understanding FCoE LAGs on page 158
- Understanding CoS Flow Control (Ethernet PAUSE and PFC) on page 121
- Understanding Fibre Channel Terminology on page 131
- Configuring VLANs for FCoE Traffic on an FCoE Transit Switch on page 289
- Configuring an FCoE LAG on page 302

• Example: Configuring CoS PFC for FCoE Traffic on page 186

Understanding FCoE Transit Switch Functionality

You can use the QFX Series as a Fibre Channel over Ethernet (FCoE) transit switch. An FCoE transit switch is a Layer 2 data center bridging (DCB) switch that can transport FCoE frames and implements FCoE Initialization Protocol (FIP) snooping. A DCB switch transports both FCoE and Ethernet LAN traffic over the same network infrastructure while preserving the class of service (CoS) that Fibre Channel (FC) traffic requires.

An FCoE transit switch does not encapsulate or de-encapsulate FC frames in Ethernet. It is an access switch that transports FC frames that have already been encapsulated in Ethernet between FCoE initiators such as servers and a storage area network (SAN) FC switch that supports both Ethernet and native FC traffic on its interfaces. The transit switch acts as a passthrough switch and is transparent to the FC switch, which detects each connection to an FCoE device as a direct point-to-point link.

When a QFX Series acts as a transit switch, the VLANs you configure for FCoE traffic can use any of the switch ports on the QFX Series or QFabric system Node device because the traffic in both directions is standard Ethernet traffic, not native FC traffic.

NOTE: The Ethernet interfaces that connect to FCoE devices must include a native VLAN to transport FIP traffic, because FIP VLAN discovery and notification frames are exchanged as untagged packets. It is a good practice to keep the native VLAN separate from the VLANs that carry FCoE traffic. FCoE VLANs should carry only FCoE traffic, but other types of untagged traffic might use the native VLAN.

QFX3500 and QFX3600 switches and QFabric system Node devices only require that you configure the native VLAN on the FCoE interfaces that belong to the FCoE VLAN by including the [**set interfaces** ***interface-name*** **unit** ***unit*** **family ethernet-switching native-vlan-id** ***native-vlan-id***] statement in the configuration.

QFX5100 switches require that you include two statements in the configuration to configure a native VLAN on FCoE interfaces. Include the [**set interfaces** ***interface-name*** **native-vlan-id** ***vlan-id***] statement in the configuration to configure the native VLAN on the interface, and also include the [**set interfaces** ***interface-name*** **unit** ***unit*** **family ethernet-switching native-vlan-id** ***vlan-id***] statement in the configuration to configure the port as a member of the native VLAN.

FCoE traffic should use a VLAN dedicated only to FCoE traffic. Do not mix FCoE traffic with standard Ethernet traffic on a VLAN on the switch.

NOTE: FCoE VLANs (any VLAN that carries FCoE traffic) support only Spanning Tree Protocol (STP) and link aggregation group (LAG) Layer 2 features.

FCoE traffic cannot use a standard LAG because traffic might be hashed to different physical LAG links on different transmissions. This breaks the (virtual) point-to-point link that Fibre Channel traffic requires. If you configure a standard LAG interface for FCoE traffic, FCoE traffic might be rejected by the FC SAN.

QFabric systems support a special LAG called an FCoE LAG, which enables you to transport FCoE traffic and regular Ethernet traffic (traffic that is not FCoE traffic) across the same link aggregation bundle. Standard LAGs use a hashing algorithm to determine which physical link in the LAG is used for a transmission, so communication between two devices might use different physical links in the LAG for different transmissions. An FCoE LAG ensures that FCoE traffic uses the same physical link in the LAG for requests and replies in order to preserve the virtual point-to-point link between the FCoE device converged network adapter (CNA) and the FC SAN switch across the QFabric system Node device. An FCoE LAG does not provide load balancing or link redundancy for FCoE traffic. However, regular Ethernet traffic uses the standard hashing algorithm and receives the usual LAG benefits of load balancing and link redundancy in an FCoE LAG.

NOTE: IGMP snooping is enabled by default on all VLANs in all software versions before Junos OS R13.2. Disable IGMP snooping on FCoE VLANs if you are using software that is older than 13.2.

NOTE: On a QFX3500 switch or on a QFabric system Node device, the same VLAN cannot be used in both transit switch mode and FCoE-FC gateway mode. If you configure both a transit switch and an FCoE-FC gateway on the same QFX3500 switch or QFabric system Node device, configure different FCoE VLANs for the transit switch and the FCoE-FC gateway.

Transit switch architecture differs from FCoE-FC gateway architecture. As an FCoE-FC gateway, the system transports traffic to the FC SAN as native FC frames, and the VLAN must use an FCoE VLAN interface and native FC interfaces to transport that traffic. As a transit switch, the system forwards Ethernet traffic, and requires DCB configuration for lossless transport of that traffic and FIP snooping at FCoE device access ports, but not the FCoE-FC gateway features necessary for transporting FC traffic.

The QFX Series complies with DCB standards for ensuring lossless transport and low latency, and provides 10-Gbps ports for FCoE traffic. For lossless transport to function correctly, you must use priority-based flow control (PFC, described in IEEE 802.1Qbb) to create bandwidth reservations and ensure proper CoS for FCoE traffic.

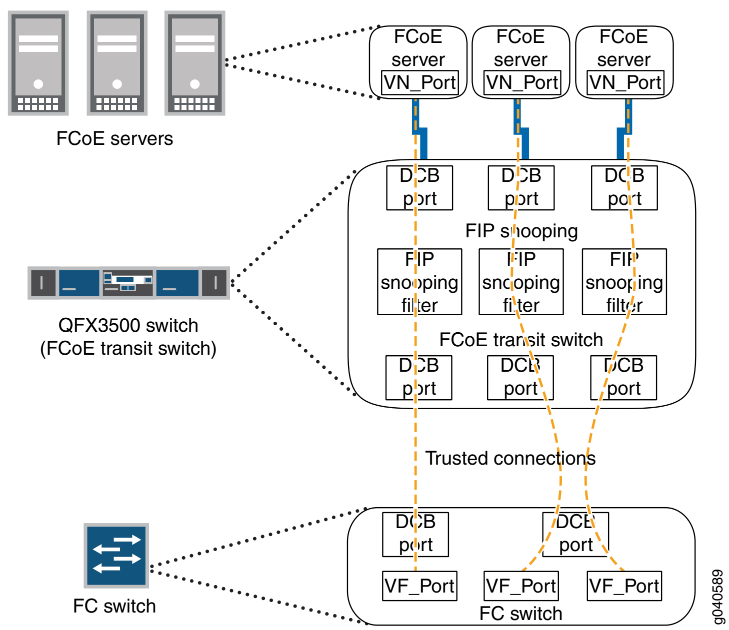
FIP snooping adds security by filtering access so that only traffic from servers that have successfully logged in to the FC network passes through the transit switch and reaches the FC network. The Technical Committee T11 organization specifications describe two types of FIP snooping:

- The FC-BB-5 specification describes VN_Port to VF_Port (VN2VF_Port) FIP snooping, which provides security for communication between FCoE device VN_Ports on the Ethernet network and FCF or FC switch VF_Ports.
- The FC-BB-6 specification describes VN_Port to VN_Port (VN2VN_Port) FIP snooping, which provides security for communication between FCoE device VN_Ports on the Ethernet network.

To accommodate the larger size of Ethernet-encapsulated frames, FCoE interfaces should be configured with a maximum transmission unit (MTU) size of at least 2180 bytes.

The transit switch transparently connects FCoE-capable devices such as servers in an Ethernet LAN to an FC switch or to a gateway switch (hereafter referred to as the FC switch), as shown in Figure 2 on page 27. The transit switch acts as a transparent DCB access layer between FCoE servers and the FC switch.

Figure 2: FCoE Transit Switch Connecting FCoE Devices to an FC Switch



The transit switch performs FIP snooping at the ports connected to the FCoE devices. For VN2VF_Port FIP snooping, at the SAN edge, the FC switch must be able to convert the FCoE traffic to native FC traffic. (VN2VN_Port FIP snooping switches traffic between VN_Ports directly through the transit switch, without going through the FC switch, so no conversion of FCoE traffic to native FC traffic is needed.)

Encapsulated FCoE traffic flows through the transit switch to the FCoE ports on the FC switch. The FC switch removes the Ethernet encapsulation from the FCoE frames to restore the native FC frames. Native FC traffic travels out native FC ports to storage devices in the FC SAN.

Native FC traffic from storage devices flows to the FC switch FC ports, and the FC switch encapsulates that traffic in Ethernet as FCoE traffic. The FCoE traffic flows through the transit switch to the appropriate FCoE device.

NOTE: The FC switch and FC fabric apply appropriate zoning checks on traffic to and from each ENode and provide FC services (for example, name server, fabric login server, or event server).

NOTE: The QFX3500 switch supports VN_Port to VN_Port FIP snooping to allow FCoE initiators and targets to communicate directly through the switch without going through an FCoE forwarder (FCF) or an FC switch. An FCoE VLAN can support either VN2VF_Port FIP snooping (FC-BB-5) or VN2VN_Port FIP snooping (FC-BB-6), but not both. The same QFX3500 switch can have multiple FCoE VLANs configured, some FCoE VLANs for VN2VF FIP snooping traffic and others for VN2VN FIP snooping traffic.

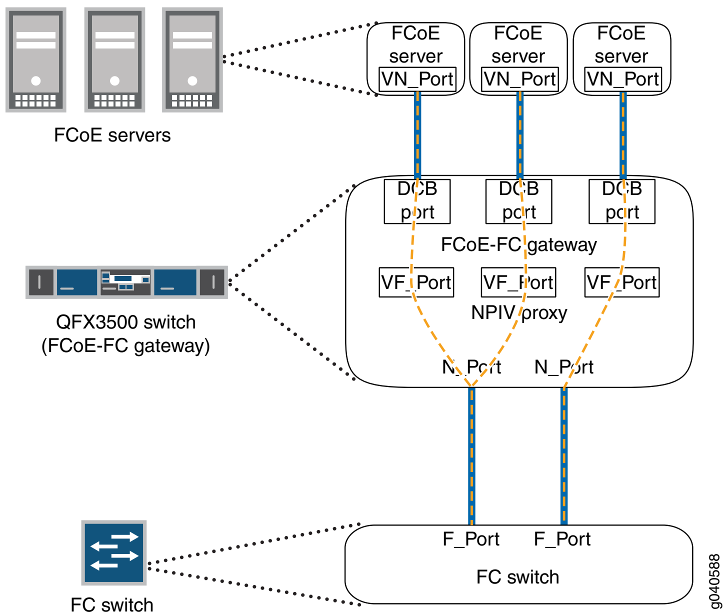
Related Documentation

- Overview of Fibre Channel on the QFX Series on page 4
- Understanding DCB Features and Requirements on page 16
- Understanding an FCoE-FC Gateway on page 29
- Understanding FCoE on page 19
- Understanding FCoE LAGs on page 158
- Understanding VN_Port to VF_Port FIP Snooping on an FCoE Transit Switch on page 80
- Understanding VN_Port to VN_Port FIP Snooping on an FCoE Transit Switch on page 87
- Understanding Fibre Channel Terminology on page 131
- Configuring VLANs for FCoE Traffic on an FCoE Transit Switch on page 289
- Disabling Enhanced FIP Snooping Scaling on page 298
- Configuring an FCoE LAG on page 302

Understanding an FCoE-FC Gateway

A Fibre Channel over Ethernet (FCoE)-Fibre Channel (FC) gateway connects FCoE devices on an Ethernet network to an FC switch in an FC storage area network (SAN) as shown in Figure 3 on page 29. To FCoE devices such as servers, the FCoE-FC gateway presents virtual fabric ports (VF_Ports) and appears to be an FCoE forwarder (FCF). To the FC switch, the FCoE-FC gateway presents a proxy node port (NP_Port) and appears to be an FC device.

Figure 3: FCoE-FC Gateway Topology



The FCoE-FC gateway handles FCoE Initialization Protocol (FIP) and FCoE traffic on the interfaces connected to FCoE devices. The gateway forwards native FC traffic on the interfaces to the FC switch. The gateway does not provide FC services (such as fabric login server or name server). It is a proxy for an FCF, not an FCF or an FC switch. The gateway transparently substitutes for the FC switch when communicating with FCoE devices and transparently substitutes for FCoE devices when communicating with the FC switch.

The gateway does not use an FC domain ID, so it extends the SAN fabric while saving domain resources. Using the gateway also means that the FC switch does not have to handle FCoE traffic (and therefore requires no FCoE blades or ports). The gateway converges Ethernet and FC backbones to leverage existing resources.

- Gateway FC Fabric on page 30
- Fabric Services on page 31
- FCoE-FC Gateway Traffic Switching on page 31

Gateway FC Fabric

A gateway FC fabric is a QFX Series configuration construct. It is not the same thing as an FC fabric in the SAN; the gateway FC fabric is local to the switch. It creates associations that connect FCoE devices with converged network adapters (CNAs) on the Ethernet network to an FC switch on the Fibre Channel network. A gateway FC fabric consists of:

- A unique fabric name.
- A unique fabric ID.
- At least one dedicated VLAN for FCoE traffic. VLANs that carry FCoE traffic should not carry any other type of traffic.

  NOTE: **On a QFX3500 or QFabric system Node device, the same VLAN cannot be used in both transit switch mode and FCoE-FC gateway mode.**

- At least one FCoE VLAN interface (Layer 3 VLAN interface) that includes one or more 10-Gigabit Ethernet interfaces connected to FCoE devices. The FCoE VLANs transport traffic between the FCoE servers and the FCoE-FC gateway. Each FCoE VLAN must carry only FCoE traffic. You cannot mix FCoE traffic and standard Ethernet traffic on the same VLAN.

  The 10-Gigabit Ethernet interfaces that connect to FCoE devices must include a native VLAN to transport FIP traffic because FIP VLAN discovery and notification frames are exchanged as untagged packets.

  Each FCoE VLAN interface can present multiple VF_Port interfaces to the FCoE network.

  NOTE: **Storm control must be disabled on all Ethernet interfaces that belong to the FCoE VLAN to prevent FCoE traffic from being dropped.**

- One or more native FC interfaces. The native FC interfaces transport traffic between the gateway and the FC switch.

  TIP: **If the network does not use a dual-rail architecture for redundancy, configure more than one native FC interface for each FC fabric to create redundant connections between the FCoE devices and the FC switch. If one physical link goes down, any sessions it carried can log in again and connect to the FC switch on a different interface. Even in dual-rail architecture networks, creating redundant connections between the QFabric system and the FC switch is the best practice.**

You can also configure FIP parameters for the fabric or accept the default FIP parameters. VN_Port to VF_Port (VN2VF_Port) FIP snooping is automatically enabled on all server-facing ports because all ports are untrusted by default. You can disable VN2VF_Port FIP snooping on a port-by-port basis by marking a port as an FCoE trusted interface. You can disable VN2VF_Port FIP snooping on all Ethernet ports in an FC fabric by configuring the fabric as FCoE trusted.

Because the switch has 12 native FC ports and each FC fabric requires a minimum of one native FC port, the switch supports a maximum of 12 FC fabrics. However, as a best practice for redundancy, we recommend that you assign at least two native FC interfaces to each FC fabric.

On a QFabric system, all of the FC and FCoE traffic that belongs to a particular gateway FC fabric must ingress and egress the same gateway Node device. Gateway FC fabrics do not span across Node devices. All of the native FC interfaces and the Ethernet interfaces that belong to the FCoE VLAN must reside on the same gateway Node device to be included in an FC fabric on that Node device.

Traffic from FC and FCoE devices that are not in the same FC fabric remain separate and cannot communicate with each other through the gateway.

Fabric Services

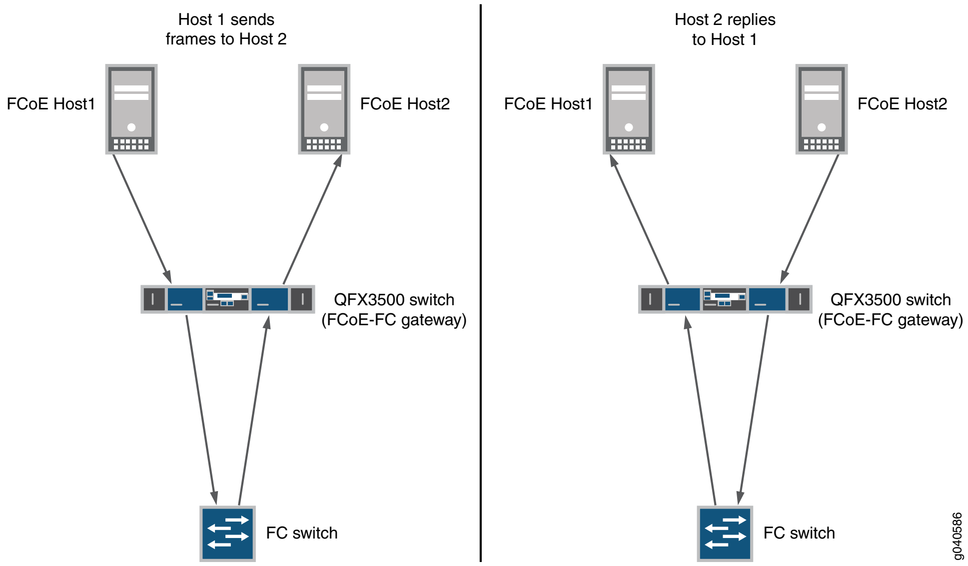
The FC switch provides all FC services (domain manager, name server, fabric login server, and so on) except FIP to the FCoE devices. The FC switch assigns all FCIDs (through N_Port ID virtualization) and fabric attributes to FCoE device VN_Ports.

The FCoE-FC gateway does not provide FC services (except FIP). The gateway relays communication between the FC switch and the FCoE devices, encapsulates and de-encapsulates native FC frames, converges Ethernet and FC backbones, and aggregates FCoE device VN_Port sessions.

FCoE-FC Gateway Traffic Switching

All traffic that flows through the gateway FC fabric is switched through the FC switch. Even if two hosts on the Ethernet FCoE network connect directly to the gateway, FCoE communication between them goes through the FC switch, as shown in Figure 4 on page 32.

Figure 4: Traffic Switching Between FCoE Hosts Connected to the FC Network by an FCoE-FC Gateway



For example, FCoE host server *Host1* sends frames destined for FCoE host server *Host2*. Both *Host1* and *Host2* are directly connected to the gateway. The communication path looks like this:

1. *Host1* sends FCoE frames destined for *Host2* to the gateway .
2. The gateway de-encapsulates the FCoE frames from *Host1* into native FC frames and switches them to the FC switch.
3. The FC switch processes the native FC frames and sends them back to the gateway destined for *Host2*.
4. The gateway encapsulates the FC frames in Ethernet and sends the resulting FCoE frames to *Host2*.
5. When *Host2* replies, the FCoE reply goes to the gateway. The gateway de-encapsulates the reply and switches it to the FC switch for processing. The FC switch then sends it back to the gateway, which encapsulates the FC frames and sends them to *Host1*.

Related Documentation

- Overview of Fibre Channel on the QFX Series on page 4
- Understanding Fibre Channel on page 12
- Understanding FCoE-FC Gateway Functions on page 33
- Overview of FIP on page 9
- Understanding Interfaces on an FCoE-FC Gateway on page 50
- Example: Setting Up Fibre Channel and FCoE VLAN Interfaces in an FCoE-FC Gateway Fabric on page 171
- Disabling Storm Control on FCoE Interfaces on an FCoE-FC Gateway on page 278

- Understanding Fibre Channel Terminology on page 131

Understanding FCoE-FC Gateway Functions

When it functions as a Fibre Channel over Ethernet (FCoE)-Fibre Channel (FC) gateway, the QFX Series provides the following functions:

Login and Logout

Each of the native FC interfaces on the gateway performs a fabric login (FLOGI) to the FC switch when each interface initializes. This establishes the link between each gateway FC interface and the FC switch.

When FCoE devices on the Ethernet network send an FCoE Initialization Protocol (FIP) login (FIP FLOGI) or FIP discovery (FIP FDISC) request to the gateway, the gateway acts on behalf of those devices and converts their FIP FLOGI and FIP FDISC requests to FC FDISC requests. The gateway then sends the FC FDISC requests to the FC switch. When the FC switch responds to an FDISC request, the gateway converts the FC response into a FIP response and sends it to the appropriate FCoE device.

The gateway also converts FIP logout (LOGO) requests from FCoE devices into FC LOGO requests to the FC switch, and converts the FC switch response into a FIP response for the FCoE device.

FCoE and FC Frame Handling

When it receives FCoE frames from FCoE devices, the gateway strips away the Ethernet encapsulation from the FC frame before sending the native FC frame to the FC switch.

When it receives native FC frames from the FC switch, the gateway encapsulates the native FC frames in Ethernet before sending the resulting FCoE frames to the appropriate VN_Port.

Data Center Bridging

The Ethernet ports connected to the FCoE devices are 10-Gbps Ethernet ports and support data center bridging (DCB) specifications:

- Priority-based flow control (PFC, described in IEEE 802.1Qbb)
- Data Center Bridging Capability Exchange protocol (DCBX), which is an extension of Link Layer Data Protocol (LLDP, described in IEEE 802.1AB)

- Enhanced transmission selection (ETS, described in IEEE 802.1Qaz)
- 10-Gigabit Ethernet ports

Disabling the Fabric WWN Verification Check

The gateway connects to a SAN fabric using the gateway NP_Ports (native FC ports). When the NP_Ports initialize, each port sends a FLOGI to the FC switch to which it is connected in the SAN fabric. The FC switch sends a FLOGI accept (FLOGI-ACC) message back to each NP_Port. The FLOGI-ACC message includes the SAN fabric worldwide name (WWN). The gateway uses the SAN fabric WWN in the multicast discovery advertisement (MDA) that the gateway sends to the ENodes in the FCoE network.

Some FC switches substitute their own WWN (often the FC switch's virtual WWN) for the SAN fabric WWN in the FLOGI-ACC message. When the FC switch substitutes its own WWN for the fabric WWN, gateway NP_Ports that log in to the same SAN fabric might receive different fabric WWNs in the FLOGI-ACC messages if the NP_Ports are connected to different FC switches in that SAN fabric. This creates a problem, because different fabric WWNs indicate different SAN fabrics. But in this scenario, the different fabric WWNs come from different FC switches in the same SAN fabric.

If the gateway receives different fabric WWNs on NP_Ports that are connected to the same SAN fabric, the gateway uses the first fabric WWN it receives in the MDA it sends to the ENodes. The gateway isolates the NP_Ports connected to that fabric that receive a different fabric WWN in the FLOGI-ACC message. No ENode sessions are assigned to the isolated NP_Ports. FC traffic is assigned only to NP_Ports that receive a fabric WWN that matches the fabric WWN received by the first NP_Port to log in to the FC fabric. (If an NP_Port receives a fabric WWN that does not match the fabric WWN received by the first NP_Port to log in to the FC fabric, it does not carry traffic to the SAN fabric.)

In summary, the scenario is:

1. The gateway has multiple NP_Ports connected to more than one FC switch in a SAN fabric.
2. When the NP_Ports initialize, each NP_Port sends a FLOGI to the FC switch to which it is connected.
3. The FC switches substitute their own WWNs for the fabric WWN in the FLOGI-ACC message, so different NP_Ports receive different fabric WWNs.
4. In the MDA the gateway sends to FCoE devices, the gateway uses the fabric WWN that the first NP_Port to log in to the fabric receives in the FLOGI-ACC message. If other NP_Ports receive a different fabric WWN from other FC switches in the SAN fabric, that fabric WWN is not advertised.
5. NP_Ports that receive a fabric WWN that does not match the first received fabric WWN are isolated, and the ENode sessions cannot use those ports.

To prevent this from happening, you can disable the gateway fabric WWN verification check so that all NP_Ports connected to a SAN fabric are used to carry traffic between the gateway and the FC switch, regardless of the fabric WWN the NP_Port receives in the FLOGI-ACC message.

NOTE: Disabling or enabling the fabric WWN verification check logs out all FCoE sessions.

Load Balancing

The switch performs automatic link load balancing for the connections between the gateway and the FC SAN and can also perform load balancing for the connections between the gateway and the FCoE devices in the Ethernet network. On the native FC links (NP_Ports) between the gateway and the FC SAN, the gateway can use one of the following three load-balancing algorithms:

- Simple load balancing—The switch assigns each ENode FLOGI session and VN_Port FDISC session to the least-loaded link. The switch can place FDISC sessions on a different link than the parent FLOGI session (an ENode FLOGI session and its subsequent FDISC sessions can be placed on different links). Simple load balancing is the default load-balancing algorithm. Rebalancing the link load disrupts only selected sessions to minimize the impact (the switch uses an algorithm to log out only the sessions that need to be moved to other links to balance the load when those sessions log in again).
- ENode-based load balancing—When an ENode logs in to the fabric, the switch places all subsequent VN_Port FDISC sessions associated with that ENode on the same link as the ENode FLOGI session, regardless of the link load. New ENode FLOGIs are placed on the least-loaded link. The switch calculates the link load based on the combined total of FLOGIs and FDISCs on each NP_Port link. Rebalancing the link load disrupts all sessions (all sessions log out and then log in again).
- FLOGI-based load balancing—Similar to ENode-based load balancing; when an ENode logs in to the fabric, the switch places all subsequent VN_Port FDISC sessions associated with that ENode on the same link as the ENode FLOGI session, regardless of the link load. New ENode FLOGIs are placed on the least-loaded link.

NOTE: Changing the load-balancing algorithm when FCoE sessions are running forces the FCoE sessions to log out, then log in again.

Related Documentation

- Understanding Fibre Channel on page 12
- Understanding an FCoE-FC Gateway on page 29
- Understanding DCB Features and Requirements on page 16
- Understanding Load Balancing in an FCoE-FC Gateway Proxy Fabric on page 63
- Understanding Interfaces on an FCoE-FC Gateway on page 50
- Disabling the Fabric WWN Verification Check on page 267
- Monitoring Fibre Channel Interface Load Balancing on page 383

Understanding FCoE and FIP Session High Availability

The QFX Series provides high availability features to maintain storage network sessions when a system process is terminated and during certain types of upgrades:

- High Availability for Fibre Channel Process Termination (FCoE-FC Gateway Mode) on page 36
- High Availability for FIP Snooping on page 36
- Nonstop Software Upgrade (QFabric Systems) on page 37

High Availability for Fibre Channel Process Termination (FCoE-FC Gateway Mode)

In FCoE-FC gateway mode, the QFX3500 switch provides high availability to restore the FCoE sessions running on the switch in case the Fibre Channel (FC) process is terminated. A session is a fabric login (FLOGI) or fabric discovery (FDISC) login to the FC SAN fabric, not an end-to-end server-to-storage session.

The switch stores FCoE session data in a persistent storage module. If the FC process terminates, the switch restores the existing FCoE sessions on the same interfaces that they were on before the FC process terminated. Data traffic for existing sessions is not affected during session restoration.

For a brief time, the system does not process control traffic because of the FC process restart and session restoration. During this brief time, no new FCoE sessions can be established, and no existing sessions can log out.

NOTE: During the restoration process, if the FC process does not receive an *interface up* notification from a particular interface within a certain time, the switch times out the restore operation and discards the data on that interface. The previously existing FCoE sessions on that interface are not restored, and the ENodes must log in again.

NOTE: An FC process restart and session restoration resets the Fibre Channel statistics.

If the FC process terminates repeatedly, the operating system disables the process until you manually restart it. To restart the FC process manually, issue the **restart fibre-channel** command.

High Availability for FIP Snooping

You can configure the system to perform FIP snooping on Ethernet interfaces that are connected to FCoE devices that have ENodes. The QFX Series provides high availability to restore running FIP snooping sessions in case the Ethernet switching process is terminated.

The Ethernet switching process stores the FIP snooping state in a persistent storage module. If the Ethernet switching process terminates, the QFX Series restores the existing FIP snooping sessions on the same interfaces that they were on before the Ethernet switching process terminated. The high availability features preserve:

- Logged in ENodes
- Discovered FCFs
- Existing sessions
- Existing FIP snooping filters

The complete restoration process, including reconciling all valid states, takes a maximum of 8 seconds. During the restoration process, the switch can learn a new FCF or a new FC switch, and new ENodes can log in to the FC network. However, FDISC messages from an ENode that is already logged in to the network might be dropped if the ENode has not yet been restored.

When the Ethernet switching process terminates ungracefully, the FIP keepalive timer is reset to the normal initial value, not the value at the time of the Ethernet switching process termination.

In the event of an Ethernet switching process termination, ENodes remain logged in, and existing sessions are not interrupted.

NOTE: An Ethernet switching process restart and session restoration resets the FIP snooping statistics.

Nonstop Software Upgrade (QFabric Systems)

On QFabric system Node groups that have more than one Node device, nonstop software upgrade (NSSU) enables you to upgrade the Node devices with minimal packet loss and maximum uptime. NSSU automates software upgrades on the QFabric system components in an orderly and consistent manner to maximize system uptime.

The system upgrades components with redundant architectures, such as redundant server Node groups and network Node groups that have two or more members, in stages. While the system upgrades one component, the redundant component continues to function.

For example, while one member of a redundant server Node group is upgraded, the other member continues to forward traffic. When the first Node group member completes the upgrade, it comes online while the system upgrades the second member.

NSSU provides high availability for the lossless traffic forwarding required to support storage networks. If your system design includes redundancy (redundant Node devices in Node groups, LAGs, and so on) so that an alternate traffic path is available, when you upgrade a Node device, traffic is not impacted.

In fully redundant topologies, NSSU preserves FIP session, FIP snooping filter, VN2VF_Port session, and VN2VN_Port session information and prevents traffic loss in most cases. An exception is that Node devices that are directly connected to ENodes experience momentary traffic loss when the Node device reboots.

Related Documentation

- Understanding an FCoE-FC Gateway on page 29
- Understanding FCoE on page 19
- *Understanding Nonstop Software Upgrade for QFabric Systems*
- *Performing a Nonstop Software Upgrade on the QFabric System*

Understanding FIP Functions

Fibre Channel over Ethernet (FCoE) Initialization Protocol (FIP) performs four major functions:

- FIP VLAN discovery: FCoE device FCoE nodes (ENodes) discover the FCoE VLANs on which to transmit and receive FIP and FCoE traffic.
- FIP discovery: FCoE devices discover Fibre Channel (FC) switches to which they can connect.
- Initialization: FCoE devices perform fabric login (FLOGI) and fabric discovery (FDISC) to create a virtual link with an FC switch.
- Maintenance: The QFX Series ensures that the virtual link between the FCoE device and the FC switch remains valid, and also that the link termination logout (LOGO) functions properly.

When you configure the switch as an FCoE-FC gateway, it converts FIP requests and information from FCoE devices into FC requests and information and relays them to the FC switch. To FCoE devices, the gateway appears to be an FCoE forwarder (FCF) and presents virtual fabric port (VF_Port) interfaces to the server ENode. To FC switches, the gateway appears to be an FC device that supports N_Port ID virtualization (NPIV) and presents an N_Port interface to the FC switch F_Port interface. When you configure the QFX Series as an FCoE transit switch, you do not configure FIP parameters on the switch.

FIP FLOGI, FDISC, and LOGO are similar to the same processes in the native FC protocol.

This topic describes:

FIP VLAN Discovery

The gateway supports FIP VLAN discovery. Host ENodes use FIP VLAN discovery to discover the FCoE VLANs on which they will send and receive FIP and FCoE traffic and on which they will establish a virtual link with the FC switch. This means FCoE devices do not need manually configured FCoE VLANs.

FIP VLAN discovery and notification takes place on the native VLAN that the FCoE device uses for Ethernet traffic:

1. The ENode sends a FIP VLAN discovery request to a multicast address called *ALL-FCF-MACs* to which all FC switches and FCFs on the VLAN listen.
2. The FC switches and FCFs respond on the native VLAN with a list of the FCoE VLANs that are available for login.
3. The ENode selects an FCoE VLAN and continues the FIP process on that VLAN.

Except for FIP VLAN discovery, all other FIP and FCoE traffic runs on an FCoE VLAN.

BEST PRACTICE: **Only FCoE traffic is permitted on the FCoE VLAN. A native VLAN might need to carry untagged traffic of different types and protocols. Therefore, it is a good practice to keep the native VLAN separate from FCoE VLANs.**

FIP Discovery

The FIP discovery process allows an FCoE device ENode MAC to locate (discover) the FC switches in the FCoE VLAN to which it belongs. The ENode selects an FC switch to log in to from the available FC switches. Either the ENode MAC or the FC switch can initiate the FIP discovery process.

Server ENode MACs initiate FIP discovery:

1. When an ENode MAC comes online, it sends a multicast discovery solicitation message on its FCoE VLAN to a multicast address called *ALL-FCF-MACs* to which all FCFs (including the FCF functionality of FC switches) on the VLAN listen. The discovery solicitation message includes the ENode's addressing mode and the maximum protocol data unit (PDU) size the ENode MAC uses for FCoE traffic.

   The ENode uses the globally unique ENode MAC address assigned to it by the converged network adapter (CNA) manufacturer as an identifier in the FIP frame header.
2. The FCFs on the VLAN that have a similar supported addressing mode, match the maximum FCoE size, and can accept a login from the ENode reply to the discovery solicitation message by sending a solicited unicast discovery advertisement message to the soliciting ENode MAC.
3. The ENode MAC compiles a list of FCFs that are available for login, selects an FCF (the FCF with the highest priority setting), and is then ready to log in to the FCF.

The FIP discovery process is similar when the FC switch or FCF initiates discovery:

1. FCF MACs periodically send unsolicited multicast discovery advertisements on the FCoE VLAN to the *ALL-ENode-MACs* multicast address, to which all ENode MACs on the VLAN listen. The FIP keepalive advertisement period timer (FKA_ADV_PERIOD) controls the interval between multicast discovery advertisements. The multicast discovery advertisements inform ENodes on the VLAN that FCF VF_Ports are available for establishing virtual links with ENode VN_Ports.
2. ENodes on the FCoE VLAN create an entry for the FCF-MAC in their FCF-MAC lists.
3. An ENode can respond to the unsolicited multicast discovery advertisement with a unicast discovery solicitation message to the FCF.
4. Upon receiving the ENode's unicast discovery solicitation, the FCF replies with a unicast discovery advertisement sent to the ENode MAC.

After the ENode MAC selects an FCF to log in to, FIP initialization begins. To proceed from discovery to initialization, the server ENode addressing mode must match the FCF addressing mode and maximum FCoE size. In addition, the FCF must be configured to allow FIP FLOGI from that ENode.

FIP FLOGI

FIP initialization is the server ENode login process to the FCF after the ENode discovers the FCFs (including FC switches) on the FCoE VLAN:

1. The ENode sends a fabric login (FLOGI) request message to the FCF.
2. The FCF replies to confirm the ENode login and provides the ENode a locally unique MAC address to use for FCoE frame transactions. The locally unique MAC address identifies the VN_Port interface of the ENode for the session the login establishes. (The ENode continues to use the globally unique ENode MAC address for FIP frame transactions.)

The locally unique ENode MAC address for FCoE operations depends on whether the ENode address mode is configured as a fabric-provided MAC address (FPMA) or as a server-provided MAC address (SPMA; the gateway does not support ENodes in SPMA mode and rejects login attempts from ENodes in SPMA mode):

- For FPMA mode, the FCF provides a MAC address to the ENode during the FIP FLOGI exchange. The FPMA MAC address is a 48-bit value that is unique to the local fabric and consists of a 24-bit FCoE mapped address prefix (FC-MAP) and a 24-bit FC identifier (FCID). You can configure the FC-MAP value on the FCF or use the default value of 0EFC00h. The FCoE device must use the same FC-MAP value as the FCF, or else discovery and login fail.
- For SPMA mode, the server provides its MAC address to the FCF. The FCF compares the server MAC address to a list of addresses approved for FCoE access. The gateway does not support ENodes in SPMA mode.

Successful login instantiates a secure virtual link between the ENode and the FCF and terminates the FIP virtual link instantiation phase. The initiating server behind the ENode can exchange FC payloads with storage devices in the FC SAN by sending FCoE frames over the virtual link.

FIP FDISC

After an ENode successfully logs in to an FCF and establishes a virtual link, the ENode can request more virtual links (sessions) over the same physical link by sending a FIP fabric discovery (FDISC) request. FDISC allows the creation of multiple separate secure VN_Port virtual links on one physical link. Each virtual link receives a locally unique identifier from the FCF to enable security and separation between the VN_Port virtual links sharing a physical ENode port. This is called N_Port ID virtualization (NPIV).

FDISC is similar to FLOGI in that it requests a login and a unique ID from the FCF. The difference is that FLOGI obtains the initial login and ID for the physical link, whereas FDISC obtains additional logins and IDs so that multiple virtual links can share one physical link securely.

After a VN_Port FDISC is complete, the application using that VN_Port can send FCoE frames over the virtual link.

FIP Maintenance (Keepalive Messages)

Although FCoE protocol handles the payload communication between the initiating ENode and the target FC device, FIP continues to run in the background. FIP constantly updates ENode FCF lists by listening to the periodic FCF multicast discovery advertisements, and it verifies the ability to reach the FCF by transmitting periodic FIP keepalive advertisements.

The ENode sends periodic ENode FIP keepalive advertisements to the FCF with the ENode MAC address as the identifier. The ENode also sends periodic VN_Port FIP keepalive advertisements on behalf of each VN_Port on the ENode, using the VN_Port MAC address as the source MAC. The VN_Port FIP keepalive advertisements occur every 90 seconds. The keepalive advertisements reset the session timer for the virtual link connection to the FCF. If the FCF does not receive a keepalive advertisement for a logged-in ENode or VN_Port before the session timer expires, the virtual link is terminated.

The periodic unsolicited multicast discovery advertisements the FCF sends to the *ALL-ENode-MACs* address continuously verify that the FCF is still reachable. The ENode and the FCF periodic unsolicited multicast discovery advertisements occur at the configured FIP keepalive advertisement period interval (FKA_ADV_PERIOD) plus or minus a random offset to prevent a flood of simultaneous keepalive advertisements.

If the FCF does not receive the ENode keepalive advertisements before the FCF's FIP keepalive timer expires, the FCF considers the virtual link to the ENode as "down" and terminates the virtual link to the ENode. The keepalive timer expires in 2.5 times the configured timer value. This also terminates any VN_Port virtual links instantiated by that ENode.

If the FCF does not receive a VN_Port keepalive advertisement before the FCF's FIP keepalive timer expires, the FCF considers the virtual link to the VN_Port as "down" and terminates the virtual link to that VN_Port. The VN_Port keepalive timer expires in 2.5 times the configured timer value.

If the ENode does not receive the FCF unsolicited multicast discovery advertisement before the ENode's FIP keepalive timer expires, the ENode considers the virtual link to the FCF as "down" and all of the VN_Port virtual links to that FCF on the ENode are terminated.

FIP LOGO

FIP handles ENode and VN_Port logout when a session is finished.

Related Documentation

- Overview of FIP on page 9
- Understanding FIP Implementation on page 42
- Understanding FIP Parameters on an FCoE-FC Gateway on page 46
- Understanding Fibre Channel Virtual Links on page 49
- Understanding FCoE on page 19

Understanding FIP Implementation

In a network that converges Fibre Channel (FC) and Ethernet traffic, when you configure the QFX Series as a Fibre Channel over Ethernet (FCoE)-FC gateway, it translates FCoE Initialization Protocol (FIP) frames from FCoE nodes (ENodes) into native FC frames for FC switches and translates native FC frames from FC switches into FIP frames for ENodes. To an FCoE device, the gateway appears to be an FCoE forwarder (FCF) and presents a fabric port (F_Port) interface to the FCoE device ENode. To an FC switch, the gateway appears to be an FC host capable of N_Port ID virtualization (NPIV) and presents a node port (N_Port) interface to the FC switch F_Port interface.

NOTE: The N_Ports that the gateway presents to the FC switch are called proxy N_Ports (NP_Ports). To the FC switch, the gateway NP_Ports appear to be native FC N_Ports that are capable of performing NPIV. The NP_Ports are proxies for the FCoE devices in the Ethernet network. The NP_Ports convert FCoE traffic from the FCoE devices into native FC traffic for the FC switch. The NP_Ports also convert native FC traffic from the FC switch into FCoE traffic for the FCoE devices on the Ethernet network.

FIP Basics

FIP is enabled by default on all VLAN interfaces that belong to each FC fabric configured on the gateway. You can configure FIP parameters at a global level or on an individual interface. When you configure a parameter on an interface, it overrides the global configuration only for that interface. If you do not explicitly configure a FIP parameter, the gateway uses the default value.

In order for the gateway to connect FCoE devices with FCFs, the FIP parameters you configure on the gateway must be compatible with the parameters configured on the FC switch (for example, the FC-MAP values of the FC switch and of the FC fabric FIP configuration on the gateway must match, or the FC switch drops the frames).

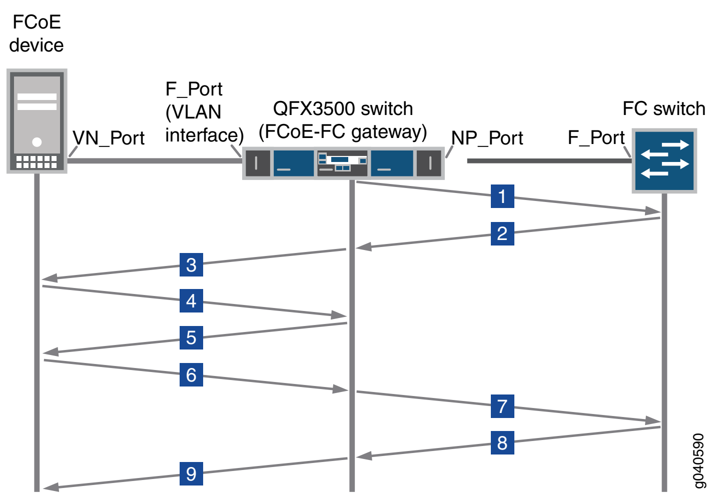
When the NP_Ports on the gateway come up, they perform an FC FLOGI to the connected FC switch. Successful login establishes communication between the gateway and the FC switch, and gateway NP_Ports are marked for sending FDISC messages. Successful login also creates a next-hop entry in the gateway for the FC switch. If the FC switch rejects the FLOGI request, no link is established. The gateway maintains a list of valid FCF-MACs with which ENodes can connect.

After establishing communication with an FC switch, the gateway can connect FCoE devices in the Ethernet network to the FC switch. All of the subsequent connections the gateway makes with FC switches as a proxy for ENodes (on behalf of ENodes) are virtualized (NPIV) connections.

Fabric Login and FIP Login Overview

Figure 5 on page 43 provides a brief overview of the FCoE-FC gateway fabric login to the FC switch and the FCoE device FIP login to the gateway.

Figure 5: FCoE-FC Gateway Fabric Login and FIP Login



The numbers in the following list correspond to the numbers in Figure 5 on page 43 and briefly describe each step of the login process:

1. The FCoE-FC gateway NP_Port sends an FC fabric login (FLOGI) request to the FC switch F_Port.
2. The FC switch accepts the gateway FLOGI.

3. The gateway sends FIP multicast discovery advertisements on the FCoE VLAN (the gateway F_Port interface) to all connected FCoE device ENodes.
4. The FCoE device ENode sends a discovery solicitation message to the gateway.
5. The gateway responds with a unicast discovery advertisement to the ENode.

NOTE: The gateway limits the number of discovery solicitations it accepts from FCoE devices to a maximum of 100 outstanding requests at any given time. If the gateway has 100 discovery solicitations outstanding, the gateway does not respond to new discovery solicitations. Instead, the gateway drops new discovery solicitations and reports the number of dropped discovery solicitations in the **Dropped** field of the **show fibre-channel fip statistics** command output. When there are fewer than 100 outstanding discovery solicitations, the system responds to new requests as usual with a discovery advertisement.

6. The FCoE device sends a FIP FLOGI or FIP FDISC message to the gateway.
7. The gateway converts the FIP FLOGI or FIP FDISC to an FC FDISC and forwards it to the FC switch to obtain a login for the FCoE device.
8. The FC switch responds to the FC FDISC by sending a new ID for the NPIV session to the gateway.
9. The gateway converts the FC FDISC response from the FC switch to a FIP FDISC response and forwards it to the FCoE device.

The following sections describe some of these steps in greater detail.

Proxy FIP Discovery

After the gateway establishes a connection with an FC switch:

1. The gateway sends periodic FIP multicast discovery advertisements on the FCoE VLAN so that ENodes can add the gateway to their FCF lists.
2. The ENode initializes and sends a multicast discovery solicitation message on the FCoE VLAN. If the ENode has already initialized and has a list of FCFs, it can send a unicast discovery solicitation message to a particular FCF such as the gateway.

NOTE: The gateway limits the number of discovery solicitations it accepts from FCoE devices to a maximum of 100 outstanding requests at any given time. If the gateway has 100 discovery solicitations outstanding, the gateway does not accept new discovery solicitations until there are fewer than 100 discovery solicitations outstanding.

3. When the gateway receives a multicast discovery solicitation from an ENode, it responds by sending a unicast discovery advertisement to that ENode.

When the gateway receives a unicast discovery solicitation from an ENode, it also responds with a unicast discovery advertisement to the ENode.

To the ENode, the gateway appears to be an FCF.

The FIP discovery process adds the ENode to the gateway ENode database.

Proxy FIP Initialization

1. If the ENode chooses to log in to the gateway, it responds to the gateway's unicast discovery advertisement by sending a login request in the form of a FIP FLOGI if it is the initial connection to the gateway. If the ENode already has an established session with the gateway and another application or virtual machine wants to connect to the gateway, the ENode sends a FIP FDISC to the gateway.
2. The gateway receives the FIP FLOGI or FIP FDISC from the ENode, converts it into an FC FDISC, and sends it through the least-loaded NP_Port to the FC switch on behalf of the ENode. The FC FDISC message requests an FCID for the new virtual link.

   NOTE: **The gateway converts both ENode FIP FLOGI and FIP FDISC messages into FC FDISC messages, because the gateway has already performed FC FLOGI with the FC switch, so all subsequent connection requests on the gateway NP_Port are FDISC requests for virtual (NPIV) connections. FDISC messages request a virtual N_Port connection over an existing physical N_Port connection.**

3. The FC switch processes the request, accepts it, assigns a unique FCID for the connection, and then sends the response to the gateway. If the FC switch rejects the FDISC request, no virtual link is established.
4. The gateway maps the FC switch response to the ENode VN_Port, converts the FC acceptance message to a FIP FLOGI or FIP FDISC response, and sends it to the ENode VN_Port.
5. The ENode VN_Port accepts the FCID, and the virtual link is established.

If an ENode sends an FDISC, the proxy gateway switch checks whether the ENode has already performed a FLOGI to create the initial connection. If the ENode has not performed a FLOGI, the FDISC request is dropped.

The FC protocol does not recognize multipoint-to-point connections. Although the gateway can aggregate traffic from multiple FCoE servers on one NP_Port, each virtual link appears to be an individual point-to-point link between an FCoE ENode VN_Port and the FC switch, not as an aggregated multipoint-to-point link. The gateway is essentially invisible to the FC protocol, so the virtual link looks and acts like a point-to-point link from the FCoE device to the FC switch.

Proxy FIP Maintenance

The gateway sends and receives periodic FIP keepalive messages to and from ENode VN_Ports to maintain the connection between the gateway and the ENodes.

Proxy FIP Logout

As with FIP discovery and FIP FLOGI, the gateway represents the FCoE device in transactions with the FC switch and represents the FC switch in transactions with the FCoE device:

1. An ENode VN_Port sends a FIP LOGO message to log off and terminate the virtual link connection.
2. The gateway converts the FIP LOGO to an FC LOGO and relays it to the FC switch.
3. The FC switch responds to the LOGO request.
4. The gateway converts the FC LOGO response to a FIP LOGO response and relays it to the VN_Port, completing the logout and terminating the virtual link.

Related Documentation

- Overview of FIP on page 9
- Understanding FIP Functions on page 38
- Understanding FIP Parameters on an FCoE-FC Gateway on page 46
- Understanding Fibre Channel Virtual Links on page 49
- Understanding FCoE on page 19
- Configuring FIP on an FCoE-FC Gateway on page 283

Understanding FIP Parameters on an FCoE-FC Gateway

By default, FIP is enabled, and the default FIP settings are valid on all FCoE interfaces that are part of the gateway FC fabric. You can configure some FIP parameters at a global level or on a specific interface. Some FIP parameters can be configured only at the global level or only at the individual interface level. When you configure a parameter at the interface level, the configuration overrides the global setting for that interface only.

- FIP Keepalive Advertisement Period on page 46
- Addressing Mode on page 47
- FC-MAP on page 48
- FCoE Trusted Fabric on page 48
- Maximum Number of FCoE Sessions Per ENode on page 48
- Priority on page 49

FIP Keepalive Advertisement Period

The FIP keepalive advertisement period (fka-adv-period) is the time interval between messages that verify the connection is still valid and the device at the other end of the virtual link is still reachable. The ENode sends an ENode FIP keepalive advertisement to the gateway with the ENode MAC address as the source address to verify its reachability. The ENode also sends VN_Port FIP keepalive messages for every VN_Port on the ENode that is logged in to the gateway, with the VN_Port MAC address as the source address.

The FIP keepalive advertisement period also determines the time interval between unsolicited multicast discovery advertisements from the gateway to the *ALL-ENode-MACs* multicast address. Unsolicited multicast discovery advertisements serve as keepalive messages from the gateway to the ENodes and also advertise the gateway's presence on the network.

The gateway sends the periodic unsolicited multicast discovery advertisements to the ENodes. On the gateway, you can configure a global FIP keepalive advertisement period for an FC fabric and you can configure a FIP keepalive advertisement period for individual interfaces to override the global setting.

Addressing Mode

For FIP transactions, the ENode identifies itself using the globally unique MAC address assigned to the CNA by the manufacturer. After FIP has established a virtual link between an ENode VN_Port and the gateway, for FCoE transactions, the VN_Port identifies itself using a locally unique MAC address. The format of the locally unique MAC address depends on the addressing mode the fabric supports and the addressing mode the ENode is programmed to use.

The addressing mode is not a configurable parameter on the gateway. FC fabrics on the gateway support only the fabric provided MAC address (FPMA) addressing mode for FCoE transactions. The gateway does not support the server provided MAC address (SPMA) addressing mode. ENodes that use SPMA cannot log in to the gateway.

The FC switch assigns a locally unique FPMA to an ENode MAC through the FLOGI or FDISC process:

1. During the FIP discovery process, the ENode compiles a list of compatible FCFs (including the gateway) in the fabric. A compatible addressing mode is one of the criteria an FCF must meet to be added to an ENode's compatible FCFs list.
2. The ENode MAC transmits a FLOGI or FDISC to the FCF that includes the addressing modes the ENode supports.
3. If the FCF supports an addressing mode the ENode uses, the FCF accepts the FLOGI or FDISC and assigns the FPMA in the accept message (FIP FLOGI LS_ACC or FIP NPIV FDISC LS_ACC). If the ENode uses an addressing mode that is incompatible with the FCF, the FLOGI or FDISC is rejected.

The FPMA uniquely identifies a single VN_Port at that ENode MAC in FCoE transactions with the FCF. Each VN_Port connection receives its own unique FPMA to identify its virtual link connection. When an ENode uses NPIV to create multiple VN_Ports, each VN_Port virtual link receives its own unique FPMA to identify its traffic.

An FPMA consists of two concatenated 24-bit values:

1. The upper 24 bits are the FCF's FC-MAP value, which is a MAC address prefix that is unique to the fabric.
2. The lower 24 bits are the locally unique FCID that the FCF (FC switch) assigns to the VN_Port.

The combination of these values guarantees that each FPMA is unique within a fabric.

FC-MAP

The FCoE mapped address prefix (FC-MAP) value is a MAC address prefix used by the FCF that is unique within a given fabric. The FCF uses the FC-MAP for FCoE traffic within that fabric. The FCF rejects FCoE traffic that uses an FC-MAP value that does not match the FCF's FC-MAP value. In most cases, the FCF uses the default FC-MAP value (0EFC00), but a pool of 256 values is available (0EFC00 through 0EFCFF).

The gateway learns FC switches in the fabric that match the gateway fabric's FC-MAP value. To learn and communicate with an FC switch, the FC-MAP value for a fabric (or for the fabric's FCoE VLAN) on the gateway must match the FC switch's FC-MAP value. If the FC-MAP values do not match, no connection is established.

NOTE: **Changing the FC-MAP value causes all logins to drop and forces the ENodes to log in again.**

FCoE Trusted Fabric

By default, all interfaces are untrusted interfaces. You can globally configure all of the ports in a specified gateway FC fabric to be FCoE trusted. This reduces system overhead by eliminating the need for filters. The total number of FCoE sessions (ENode to FCF sessions) the system can support is 2500 sessions. Sessions are defined as the combined number of VN_Port to VF_Port sessions and VN_Port to VN_Port sessions. (Although VN2VF and VN2VN sessions run in different FCoE VLANs, the session limit is a system limit, not a per-VLAN limit.)

NOTE: **A session is a FLOGI or FDISC login to the FC SAN fabric. Session does not refer to end-to-end server-to-storage sessions. There is no limit to the number of end-to-end server-to-storage sessions.**

NOTE: **Changing the fabric ports from untrusted to trusted removes any existing FIP snooping filters from the ports and terminates the existing sessions. Changing the fabric ports from trusted to untrusted forces all of the FCoE sessions on those ports to log out so that when the ENodes and VN_Ports log in again, the switch can build the appropriate FIP snooping filters.**

Maximum Number of FCoE Sessions Per ENode

You can configure the maximum number of FCoE session logins from each ENode that are permitted on the gateway FC fabric. The number of sessions is the ENode FLOGI session plus the VN_Port FDISC sessions on that ENode. Regardless of whether the fabric is trusted or untrusted, the maximum number of FCoE sessions per ENode is

2500 sessions. The total number of sessions cannot exceed the gateway fabric's maximum limit of 2500 sessions.

The maximum number of FCoE sessions per ENode is a global configuration for all members of the gateway FC fabric and cannot be configured on a per-interface basis.

NOTE: **Session does not refer to end-to-end server-to-storage sessions. There is no limit to the number of end-to-end server-to-storage sessions.**

Priority

When the FIP discovery process offers an ENode the choice of more than one FCF-MAC on a given FCF to use for login, the ENode chooses the FCF-MAC to which to send a login request based on the FCF-MAC priority. The lower the priority number, the higher the FCF-MAC's priority. The ENode selects the highest-priority (lowest priority number) FCF-MAC for the login request.

An ENode can receive multiple FCF-MAC advertisements from the same FCF in two ways:

- During the FIP discovery process, an FCF can receive an ENode MAC's multicast discovery solicitation on multiple FCF-MACs. Each FCF-MAC replies with a unicast discovery advertisement to the ENode. The ENode determines that the advertisements are from the same FCF, because the value in the Name_Identifier descriptor is the same in each advertisement.
- During the FIP discovery process, an ENode MAC can receive unsolicited multicast discovery advertisements from multiple FCF-MACs on the same FCF. The ENode determines that the advertisements are from the same FCF, because the value in the Name_Identifier descriptor is the same in each advertisement.

On the gateway, you can configure the priority value for an entire fabric or for an individual interface. The default value for both the fabric and the individual interfaces is 128 (the highest priority is 0; the lowest priority is 255).

Related Documentation

- Overview of FIP on page 9
- Understanding FIP Functions on page 38
- Understanding FIP Implementation on page 42
- Understanding Fibre Channel Virtual Links on page 49
- Understanding FCoE on page 19
- Configuring FIP on an FCoE-FC Gateway on page 283

Understanding Fibre Channel Virtual Links

A virtual link emulates a secure point-to-point connection between the virtual node port (VN_Port) of a Fibre Channel over Ethernet (FCoE) node (ENode) and the virtual fabric port (VF_Port) of an FCoE forwarder (FCF). The combination of the FCF media access control (MAC) address and the VN_Port MAC address uniquely identifies each virtual link. Uniquely identifying each virtual link enables the logical separation of traffic that belongs to each virtual link. A single physical link between an ENode and an FCF can carry multiple virtual links and maintain secure, separate transport of traffic on the different virtual links.

Virtual links are necessary because Fibre Channel protocol does not recognize multipoint-to-point connections. Even when multiple connections are aggregated on one physical port, FCoE Initialization Protocol (FIP) presents each virtual link as an individual point-to-point link between an ENode VN_Port and an FCF VF_Port.

Related Documentation

- Overview of FIP on page 9
- Understanding FIP Functions on page 38
- Understanding FIP Implementation on page 42
- Understanding FIP Parameters on an FCoE-FC Gateway on page 46
- Understanding FCoE on page 19

Understanding Interfaces on an FCoE-FC Gateway

When the QFX Series functions as an FCoE-FC gateway to connect FCoE devices on an Ethernet network to a Fibre Channel (FC) switch in a storage area network (SAN), it handles FCoE traffic from hosts and native FC traffic from the FC switch. To support this architecture, each local FC fabric configured on the gateway (in the **fc-fabrics** configuration hierarchy) must have:

- An Ethernet-network-facing F_Port interface for the FCoE VLAN to connect to FCoE device VN_Ports in the form of an FCoE VLAN interface. Multiple VF_Ports are initiated on the F_Port interface, one VF_Port for each ENode that logs in to the FC network.
- One or two blocks of six proxy N_Port (NP_Port) interfaces to connect to FC switch fabric ports (F_Ports).

Each FC fabric is local to the gateway on which you configure it. This means that both the FC switch and the FCoE devices must be connected to the same gateway (QFX3500 switch or QFabric system Node device), and that all of the interfaces configured for the local fabric also must be on that gateway. FC fabric traffic does not flow between different Node devices in a QFabric system.

This topic describes:

- Native FC Interfaces to the FC Switch on page 51
- FIP Login Session Limits on page 53
- Trusted and Untrusted Interfaces on page 56
- Buffer-to-Buffer Credit Recovery on page 57
- FCoE VLAN Interface to FCoE Devices on page 58

- Assigning Interfaces to a Fibre Channel Fabric on page 62
- Deleting a Fibre Channel Interface on page 62

Native FC Interfaces to the FC Switch

You must configure either 6 or 12 of the physical interfaces on the gateway as native FC NP_Port interfaces to connect to FC switch F_Port interfaces. By default, all of the gateway interfaces are Ethernet interfaces, so you must explicitly configure the interfaces that you want to use as FC interfaces.

You can configure the FC-capable ports xe-0/0/0 through xe-0/0/5 as fc-0/0/0 through fc-0/0/5, and ports xe-0/0/42 through xe-0/0/47 as fc-0/0/42 through fc-0/0/47 to create blocks of native FC interfaces. You cannot individually configure a single port as a native FC interface. Within these port blocks, you cannot mix FC interfaces with Ethernet interfaces. All of the ports in a block must be either native FC interfaces or Ethernet interfaces.

You cannot configure ports xe-0/0/6 through xe-0/0/41 and ports xe-0/1/1 through xe-0/1/15 as native FC ports; they can only be Ethernet ports. Native FC ports do not handle Ethernet traffic (including FCoE traffic); they handle only native FC traffic and must connect to native FC ports.

You can configure:

- Six native FC interfaces by configuring either ports xe-0/0/0 through xe-0/0/5 as fc-0/0/0 through fc-0/0/5 *or* ports xe-0/0/42 through xe-0/0/47 as fc-0/0/42 through fc-0/0/47.
- Twelve native FC interfaces by configuring ports xe-0/0/0 through xe-0/0/5 as fc-0/0/0 through fc-0/0/5 *and* ports xe-0/0/42 through xe-0/0/47 as fc-0/0/42 through fc-0/0/47.
- No native FC interfaces by leaving ports xe-0/0/0 through xe-0/0/5 and ports xe-0/0/42 through xe-0/0/47 in their default state as Ethernet interfaces.

Each native FC interface can belong to only one local FC fabric configured on the gateway. You can configure up to 12 FC fabrics on a gateway, but each FC fabric must use different native FC interfaces to connect to an FCF. (Although the native FC ports are configured in blocks, each individual port can belong to a different FC fabric.) Native FC interfaces can be configured as loopback interfaces.

- Port Mode on page 51
- NPIV on page 52
- Port Speed on page 52

Port Mode

The gateway presents a proxy N_Port (NP_Port) interface to the FC switch. An NP_Port connects to a single FC switch F_Port using a point-to-point link (in other architectures an N_Port can also connect in a point-to-point link to another N_Port, but that is not a valid configuration on the gateway).

You must explicitly configure each native FC interface connected to an FC switch as an NP_Port. The gateway NP_Ports act as a proxy for the FCoE device virtual N_Ports (VN_Ports) when the VN_Ports attempt to connect to the FC switch.

When the FC switch is a trusted switch, configure the fabric as **fcoe-trusted** to reduce overhead caused by the VN_Port to VF_Port (VN2VF_Port) FIP snooping filters that are automatically installed on untrusted ports.

NPIV

FC requires a unique point-to-point link between the FC switch and each host N_Port. The gateway creates an independent virtual link for each FCoE device session by mapping each FCoE device to a virtualized N_Port through the gateway's proxy function. This process is called N_Port ID virtualization (NPIV).

NPIV makes each virtual link look like a dedicated point-to-point link to the FC switch. In this way, multiple FCoE devices, multiple applications, and multiple virtual machines on an FCoE device can connect to an FC switch using one physical port instead of using a physical port for each host connection. The virtual link creates a secure boundary between traffic from different sources that are on a single physical port.

FCoE-FC gateway mode implements NPIV as follows:

1. An NP_Port on the gateway comes up and logs in to the attached F_Port on the FC switch. The FC switch sees the gateway port as a physical FC device N_Port and assigns it a unique FCID. This establishes the physical point-to-point link between the gateway and the FC switch.
2. The gateway receives a FIP discovery message from an FCoE device that seeks to log in to the FC network. To the FCoE device, the gateway presents a virtual F_Port (VF_Port) interface and appears to be an FCF.
3. The gateway converts the FCoE device's message into an FC fabric discovery (FDISC) message and sends it through the least-loaded physical NP_Port to the FC switch. The FDISC message requests an FCID for the new virtual link.
4. The FC switch processes the request, accepts it, assigns a unique FCID for the connection, and sends the response.
5. The gateway maps the FC switch response to the host FCoE device's VN_Port and sends a FIP acceptance advertisement to the FCoE device.
6. The FCoE device accepts the FCID.

If the FC switch rejects the FDISC, the gateway relays the rejection to the FCoE device VN_Port.

Port Speed

The gateway supports configuring FC port speeds of 2 Gbps, 4 Gbps, or 8 Gbps. FC ports can also autonegotiate the port speed to 2, 4, or 8 Gbps.

FIP Login Session Limits

A FIP login session is a fabric login (FLOGI) or fabric discovery (FDISC) login to the FC SAN fabric. (A session here does not refer to an end-to-end server-to-storage session; there is no limit to the number of end-to-end server-to-storage sessions.) You can limit the maximum number of FIP login sessions on each gateway Node device (QFX3500 switch or QFabric system Node device configured in FCoE-FC gateway mode), on each local gateway FC fabric, and on each individual NP_Port interface in a local FC fabric:

- Gateway Node devices and Node groups—The total number of FIP login sessions on the gateway Node or Node group (the sum of the sessions on all of the NP_Port interfaces in all of the local FC fabrics on the gateway Node or Nodes) cannot exceed the limit. When a gateway reaches the maximum session limit, the gateway sends subsequent multicast discovery advertisements (MDAs) with the availability bit set to 0 (zero) to prevent additional ENode login attempts. If the maximum number of sessions is running on the gateway, ENodes cannot use the gateway to log in new sessions to the FC switch. When the number of sessions falls below the maximum, the gateway sets the availability bit in MDAs to 1 so that ENodes can again log in new sessions. When a session slot becomes available, the system accepts the first session request to fill the slot.
- FC fabric—The total number of FIP login sessions on an FC fabric (the sum of the sessions on all of the NP_Port interfaces that belong to the fabric) cannot exceed the limit. When a fabric reaches the maximum session limit, the gateway sends MDAs associated with that fabric with the availability bit set to 0 to prevent additional ENode login attempts.

  NOTE: **Other FC fabrics on the same gateway can still accept ENode logins as long as the maximum session limit for those fabrics and the maximum session limit for the gateway (the Node device) have not been met.**

- NP_Port interfaces—The total number of FIP login sessions cannot exceed the interface's limit. When an interface reaches the maximum session limit, the gateway removes it from the load-balancing list for that FC fabric to prevent the gateway from attempting to assign new sessions to the interface. Other interfaces in the FC fabric can still accept logins until the FC fabric or gateway reaches its maximum session limit. However, the interface that reached the maximum session limit cannot be assigned new sessions until the number of sessions on the interface falls below the limit.

  BEST PRACTICE: **Configure a maximum session limit for each NP_Port interface that is less than or equal to the number of FIP sessions the directly connected FC switch port is configured to support. This prevents the** gateway from attempting to assign new login sessions to an interface when the connected FC switch port reaches its maximum number of sessions.

FCoE Trusted and Untrusted Interface Session Limits

The maximum number of VN2VF_Port FCoE login sessions that each gateway can support is 2500 sessions, regardless of whether interfaces are trusted or untrusted. (In software releases earlier than Junos OS Release 12.3, the session limit on untrusted interfaces and untrusted fabrics was 376 sessions.)

NOTE: If you configure an FCoE LAG on interfaces that are members of an FCoE-FC gateway fabric, the number of supported sessions depends on whether the FC fabric (fc-fabric) is an FCoE trusted fabric or an FCoE untrusted fabric. If the FC fabric is a trusted fabric, then 2,500 sessions are supported.

However, if the FC fabric is an untrusted fabric, you must disable FIP snooping session scaling on the gateway, which decreases the number of supported sessions to 376 sessions. (Disable FIP snooping scaling by including the **no-fip-snooping-scaling** option in the **[edit fc-options]** hierarchy.)

Configuring Consistent Session Limits

The system does not perform commit checks to enforce consistent session limit configuration. For example, the system does not prevent you from configuring a higher limit for ENode sessions than the total session limit for the gateway Node device, or from configuring a higher limit on an interface than on the fabric to which the interface belongs.

To prevent unexpected FIP login rejections, you should configure consistent Node device, fabric, and interface session limits. For example:

- The session limit of an interface should not exceed the session limit of the fabric to which it belongs.
- For interfaces that belong to the same fabric, the sum of the interface session limits should not exceed the fabric session limit.
- The fabric session limit should not exceed the session limit of the gateway Node device.
- For fabrics that belong to the same gateway Node device, the sum of the fabric session limits should not exceed the Node device session limit.

Session limit configuration considerations include:

- The fabric session limit restricts how many sessions can run on the NP_Port interfaces that belong to that fabric. If the combined session limits of the interfaces exceed the fabric session limit, the total number of sessions on the interfaces is the fabric limit.

  For example, if a fabric has three NP_Port interfaces, and each NP_Port interface has a limit of 500 sessions (total of 1500 sessions for the three interfaces), but the fabric has a limit of 1000 sessions, the combined number of sessions on the three interfaces is limited to 1000 sessions.

- The gateway Node device session limit restricts how many sessions can run on the fabrics that belong to that gateway. If the combined session limits of the fabrics exceed the gateway Node device session limit, the total number of sessions on the fabrics is the gateway Node device limit.

  For example, if a gateway has two fabrics, and each fabric has a limit of 1000 sessions (total of 2000 sessions for the two fabrics), but the gateway has a limit of 1500 sessions, the combined number of sessions on the two fabrics is limited to 1500 sessions.

Hierarchically, the gateway Node device session limit is the maximum limit for all sessions on the gateway, regardless of fabric and interface session limits. In the same way, the fabric session limit supersedes the interface session limit.

When session limits are exceeded, no new logins are accepted until a session slot becomes free.

Decreasing Session Limits

If you decrease the session limit, the currently logged in sessions are terminated as follows:

- Gateway Node devices and Node groups—Decreasing the session limit terminates all of the sessions on the Node device (all sessions on all interfaces on all fabrics). If the gateway Node device is part of a Node group, all sessions on all members of the Node group are terminated.
- Fabric—Decreasing the session limit terminates all of the sessions on all of the interfaces that belong to the fabric.
- NP_Port interfaces—Decreasing the session limit terminates all of the sessions on the interface and also terminates all of the sessions on any other interfaces that belong to the same fabric.

After you decrease a session limit, the sessions are terminated even if the new session limit is greater than the number of currently active sessions. For example:

- An interface has 300 active sessions.
- The current session limit is 1000 sessions.
- You decrease the session limit to 500 sessions and commit the new configuration.
- All 300 sessions are logged out, even though the new session limit is greater than the number of sessions running.

After the session limit change takes effect, the ENodes log in again and establish new sessions, up to the new session limits.

Increasing Session Limits

Increasing the session limits does not disrupt logged in sessions.

Effect of Deactivating and Then Reactivating the Configuration on Session Limits

If you decrease session limits, all ENodes are logged out. Deactivating and then reactivating the configuration can have the same effect as decreasing the session limit, which results in the ENodes being logged out.

The ENode logouts occur because when you deactivate the configuration, the system reverts to the default session limit of 2500 sessions (the maximum number of sessions). When you reactivate the configuration, the system uses the configured session limit. Unless the configured session limit is equal to the maximum session limit, reactivating the configuration decreases the session limit, which causes the ENodes to be logged out.

For example, if you:

1. Configure and commit a limit of 400 sessions.
2. Allow ENodes to log in and start sessions.
3. Deactivate the configuration.
4. Reactivate the configuration.
5. The ENode sessions are logged out because deactivating the session increased the session limit from 400 to 2500.

Because an increase in the session limit does not affect existing sessions, the running ENode sessions are not affected. However, reactivating the configuration decreased the session limit from 2500 back to 400. The session limit decrease causes the ENode sessions to be logged out.

Trusted and Untrusted Interfaces

By default, gateway fabric interfaces are untrusted interfaces. If you do not configure a gateway fabric as an FCoE trusted fabric to set all of the gateway fabric interfaces as trusted interfaces, the gateway installs VN2VF_Port FIP snooping filters on the fabric ports.

If you configure a gateway fabric as an FCoE trusted fabric, the gateway does not install VN2VF_Port FIP snooping filters on the fabric interfaces. This is usually done when the gateway is connected to an FCoE transit switch that has VN2VF_Port FIP snooping enabled.

Regardless of whether an interface is trusted or untrusted, the maximum session limit is 2500 sessions, unless the interface is a member of an FCoE LAG interface.

NOTE: If you configure an FCoE LAG on interfaces that are members of an FCoE-FC gateway fabric, the number of supported sessions depends on whether the FC fabric (fc-fabric) is an FCoE trusted fabric or an FCoE untrusted fabric. If the FC fabric is a trusted fabric, then 2,500 sessions are supported.

However, if the FC fabric is an untrusted fabric, you must disable FIP snooping session scaling on the gateway, which decreases the number of supported sessions to 376 sessions. (Disable FIP snooping scaling by including the **no-fip-snooping-scaling** option in the **[edit fc-options]** hierarchy.)

NOTE: The session limit for a Node group is the same as the session limit for an individual Node device, 2500 sessions. Even if more than one Node device in a Node group is acting as an FCoE-FC gateway, the total maximum number of sessions on all Node devices in the Node group is 2500 sessions.

The default maximum login session value for Node devices (on QFabric systems, the maximum applies to each Node device), FC fabrics, and interfaces in fabrics is 2500 sessions.

Buffer-to-Buffer Credit Recovery

Buffer-to-buffer credits represent the number of receive buffers an interface can use to store FC frames. Buffer-to-buffer credit determines buffer-to-buffer flow control. When an interface transmits a frame, it decrements its buffer-to-buffer credit count by one. When the destination interface forwards the frame and frees a buffer, it sends a receiver ready (R_RDY) primitive to the transmitting interface. Each R_RDY primitive the transmitting interface receives increments its buffer-to-buffer credit count by one.

Both interfaces on an FC link track buffer-to-buffer credits. As long as buffer-to-buffer credits are available, the transmitter continues to send frames. If the number of buffer-to-buffer credits reaches zero (0), transmission stops until buffer-to-buffer credits are available, as indicated by the reception of an R_RDY primitive. Buffer-to-buffer credits can compensate for long cable distances to limit throughput and prevent buffer overflow.

However, if frame corruption or errors transmitting R_RDY primitives occur, the buffer-to-buffer credit counters on the sending and receiving interfaces do not have the same values. This causes the permanent loss of buffer-to-buffer credits. When credits are lost, the buffer credit count can decrement to zero and indicate that there is no available buffer space even if buffer space is actually available. This can result in unnecessary link idle time.

To recover lost buffer-to-buffer credits, you can configure a buffer-to-buffer credit state change number (BB_SC_N). BB_SC_N must be configured on both ends of the connection. If only one end of the connection is configured for BB_SC_N, the feature is disabled. The two directly connected FC interfaces communicate the BB_SC_N value during fabric login (FLOGI).

When you enable BB_SC_N on the interfaces on both ends of an FC link, the interfaces exchange buffer-to-buffer state change send (BB_SCs) and buffer-to-buffer state change receive (BB_SCr) primitives to track the number of frames sent and the number of R_RDY primitives received. The state change number determines the number of frames and R_RDY primitives the interfaces exchange between consecutive BB_SCn primitives and between consecutive BB_SCr primitives. The state change primitives inform each interface of the other interface's frame count and R_RDY count states.

The state counters should match so that each interface knows and agrees with the other interface's state. If the interface at either end of the link detects a discrepancy, it knows that a frame or an R_RDY primitive was corrupted or dropped.

For example, if a receiving interface has sent two R_RDY primitives but the BB_SCr that the interface receives from the sending interface only counts one R_RDY primitive received, it reveals that one R_RDY primitive was not delivered successfully and that one buffer-to-buffer credit was lost. When one of the interfaces on the link detects a discrepancy, the interfaces can take corrective action and recover the lost buffer-to-buffer credits.

Enabling the buffer-to-buffer credit recovery feature does not impact buffer resources and has an insignificant impact on processing resources.

If buffer-to-buffer credit recovery is not used, then when there is no buffer credit on a port, a timeout and recovery mechanism prevents buffer overflow.

FCoE VLAN Interface to FCoE Devices

Each FC fabric configured on the gateway includes at least one FCoE VLAN interface to connect the FCoE devices on the FCoE VLAN to the FC switch. (Including the FCoE VLAN interface and the native FC interfaces in the FC fabric configuration connects them.) FCoE VLANs can include any Ethernet interface on the switch that is in tagged-access or trunk mode. The best practice is to configure Ethernet interfaces that belong to FCoE VLANs in **tagged-access** port mode.

NOTE: **The Ethernet interfaces that connect to FCoE devices must include a native VLAN to transport FIP traffic, because FIP VLAN discovery and notification frames are exchanged as untagged packets.**

FCoE VLANs should carry only FCoE traffic. You should not mix FCoE traffic and standard Ethernet traffic on the same VLAN.

NOTE: FCoE VLANs (any VLAN that carries FCoE traffic) support only Spanning Tree Protocol (STP) and link aggregation group (LAG) Layer 2 features.

FCoE traffic cannot use a standard LAG because traffic might be hashed to different physical LAG links on different transmissions. This breaks the (virtual) point-to-point link that Fibre Channel traffic requires. If you configure a standard LAG interface for FCoE traffic, FCoE traffic might be rejected by the FC SAN.

Beginning with Junos OS Release 13.2X52, QFabric systems support a special LAG called an FCoE LAG, which enables you to transport FCoE traffic and regular Ethernet traffic (traffic that is not FCoE traffic) across the same link aggregation bundle. An FCoE LAG ensures that FCoE traffic uses the same physical link in the LAG for requests and replies in order to preserve the virtual point-to-point link between the FCoE device converged network adapter (CNA) and the FC SAN switch across the QFabric system Node device. An FCoE LAG does not provide load balancing or link redundancy for FCoE traffic. However, regular Ethernet traffic uses the standard hashing algorithm and receives the usual LAG benefits of load balancing and link redundancy in an FCoE LAG.

On FCoE-FC gateway untrusted FC fabrics, you must disable FIP snooping session scaling on the gateway, which decreases the number of supported sessions from 2,500 to 376 sessions. (Disable FIP snooping scaling by including the **no-fip-snooping-scaling** option in the **[edit fc-options]** hierarchy.) On FCoE trusted FC fabrics, the session limit is 2,500 sessions.

Each FCoE VLAN interface can belong to only one FC fabric configured on the gateway. A gateway FC fabric can have more than one FCoE VLAN, but each FCoE VLAN in the FC fabric must belong only to that FC fabric. You can configure more than one FC fabric on a gateway; each FC fabric must use different FCoE VLAN interfaces to connect to FCoE devices.

NOTE: Storm control must be disabled on all Ethernet interfaces that belong to the FCoE VLAN to prevent FCoE traffic from being dropped.

- Port Mode on page 59
- Disabling Storm Control on FCoE Interfaces on page 61
- NPIV Support on page 62
- VN2VF_Port FIP Snooping on page 62

Port Mode

You must explicitly configure the FCoE VLAN interface in F_Port mode. All members of the FCoE VLAN use the FCoE VLAN interface as the connection to the gateway NP_Port interfaces and ultimately to the FC switch.

All of the 10-Gigabit Ethernet interfaces that are members of an FCoE VLAN should be configured as **tagged-access** port mode interfaces. However, the system also supports configuring these interfaces in **trunk** port mode.

BEST PRACTICE: Use **tagged-access** port mode for Ethernet interfaces that are connected to converged network adapters (CNAs) in FCoE access devices.

Use **trunk** port mode when an Ethernet interface is an interswitch link (ISL)—that is, when the port is connected to another switch. For example, if a port is connected to a transit switch that is performing VN2VF_Port FIP snooping, configure the port in **trunk** mode and as an FCoE trusted port.

The **tagged-access** port mode was not available in Junos OS Release 11.3 and earlier releases. In Release 11.3 and earlier, only **trunk** port mode was used for Ethernet interfaces that belong to an FCoE VLAN. Because **tagged-access** mode is now available, using **trunk** mode for interfaces connected to FCoE CNAs is not recommended.

If an existing configuration uses **trunk** mode for ports connected to FCoE CNAs, you can change the port mode to **tagged-access** without disrupting traffic. Although we recommend changing the port mode of these ports from **trunk** mode to **tagged-access** mode as a best practice, it is not mandatory. New configurations should use **tagged-access** mode for interfaces that connect to FCoE devices.

There are several advantages of configuring Ethernet ports connected to FCoE devices in **tagged-access** mode instead of in **trunk** mode:

- It is standard practice to configure ISL ports as trunk ports.
- It is standard practice not to configure ports that connect to servers as trunk ports.
- When an interface goes down, if that interface is in **trunk** mode, then the FCoE sessions on that interface are terminated only after the gateway stops receiving FIP keepalive messages from the ENode and exceeds 2.5 times the FIP keepalive timeout advertisement value. If the interface is in **tagged-access** mode and the interface goes down, the gateway sends a FIP message to terminate the sessions on the interface.
- Similarly, if an ENode session moves from one interface to another interface, if the original interface is in **trunk** mode, the session is not removed from the interface until the gateway stops receiving FIP keepalive messages and exceeds 2.5 times the FIP keepalive advertisement timeout value. But if the interface is in **tagged-access** mode, the gateway detects that the session is no longer on the interface, does not refresh the FIP keepalive timer, and thus ages out the session.

NOTE: FIP is enabled on the FCoE VLAN, which is a Layer 3 interface. As with other Layer 3 interfaces under Junos OS, when the last member (10-Gigabit Ethernet interface) of the FCoE VLAN is deleted, the FCoE VLAN interface is internally marked as "down." When the Layer 3 FCoE VLAN interface is marked as "down", FIP stops running on it. When the last member interface is deleted from an FCoE VLAN and FIP stops running, the result could be an immediate timeout for the VN_Ports that were connected on that interface, regardless of whether the port mode is **tagged-access** or **trunk**.

Disabling Storm Control on FCoE Interfaces

Storm control is not supported on the FCoE interfaces of an FCoE-FC gateway VLAN. Enabling storm control on an FCoE-FC gateway VLAN interface may cause FCoE packet loss. Storm control is disabled by default on all interfaces. However, if you enabled storm control globally on all switch interfaces or on any interfaces that are part of the FCoE VLAN interface, you must disable storm control on the Ethernet interfaces of the FCoE VLAN.

If storm control is enabled on only a few interfaces of the FCoE VLAN, you can disable storm control on individual interfaces by including the **delete ethernet-switching-options storm-control interface *interface-name*** statement in the configuration, where ***interface-name*** is the name of the interface on which you want to disable storm control.

If storm control is enabled globally on the switch when the switch is acting as an FCoE-FC gateway, it is often easiest to disable storm control on all interfaces, then enable storm control only on Ethernet interfaces that are not part of the FCoE VLAN interface.

If storm control is enabled globally, you can disable storm control in either of two ways:

- Disable storm control on all interfaces, then enable storm control on the interfaces you want to use storm control. (From the default configuration, you cannot disable storm control on individual interfaces because the default configuration enables storm control on **all** interfaces, not on individual interfaces.)

  For example, if you want interfaces xe-0/0/20, xe-0/0/21, and xe-0/0/22 to use storm control, disable storm control on all interfaces, then enable storm control on those three interfaces:

  1. Disable storm control on all interfaces:

     ```
     user@switch# delete ethernet-switching-options storm-control interface all
     ```

  2. Enable storm control on interfaces xe-0/0/20, xe-0/0/21, and xe-0/0/22:

     ```
     user@switch# set ethernet-switching-options storm-control interface xe-0/0/20
     user@switch# set ethernet-switching-options storm-control interface xe-0/0/21
     user@switch# set ethernet-switching-options storm-control interface xe-0/0/22
     ```

- Disable storm control for all unknown unicast traffic on all interfaces by including the following statement in your configuration:

  ```
  user@switch# set ethernet-switching-options storm-control interface all no-unknown-unicast
  ```


NPIV Support

The gateway supports FCoE device NPIV. For example, a single physical FCoE device can have multiple virtual machines running on it. Each virtual machine can instantiate a separate virtual connection to the gateway, which results in its own virtual link to the FC switch. In this way, an FCoE device can have multiple separate connections to the FC SAN on a single physical port.

This is similar to the NPIV function the gateway performs with the FC switch to support multiple virtual FCoE device connections on one physical NP_Port.

The gateway presents multiple VF_Port interfaces on each FCoE VLAN interface to support the requirement for unique, secure virtual links.

VN2VF_Port FIP Snooping

The FCoE-facing ports that belong to an FCoE VLAN on a gateway are enabled for VN2VF_Port FIP snooping automatically. You can disable VN2VF_Port FIP snooping on any individual interface by configuring it as a trusted interface.

Assigning Interfaces to a Fibre Channel Fabric

You assign at least one FCoE VLAN interface and at least one native FC interface to each FC fabric you configure on the gateway. All of the interfaces that belong to an FC fabric must reside on the same gateway device. Interfaces on different gateways cannot belong to the same FC fabric, because an FC fabric is local to a single gateway device.

Deleting a Fibre Channel Interface

To delete an FC interface or an FCoE VLAN interface, you must delete the interface from the fabric first and then delete the interface from the switch.

Related Documentation

- Overview of Fibre Channel on the QFX Series on page 4
- Understanding Fibre Channel on page 12
- Understanding Load Balancing in an FCoE-FC Gateway Proxy Fabric on page 63
- Understanding FCoE LAGs on page 158
- Configuring a Physical Fibre Channel Interface on page 268
- Configuring a Fibre Channel Interface on page 269
- Configuring an FCoE LAG on page 302
- Disabling VN2VF_Port FIP Snooping on an FCoE-FC Gateway Switch Interface on page 299
- Assigning Interfaces to a Fibre Channel Fabric on page 276
- Configuring an FCoE VLAN Interface on an FCoE-FC Gateway on page 272
- Disabling Storm Control on FCoE Interfaces on an FCoE-FC Gateway on page 278
- Disabling VN2VF_Port FIP Snooping on an FCoE-FC Gateway Switch Interface on page 299

- Deleting a Fibre Channel Interface on page 277
- Setting the Maximum Number of FIP Login Sessions per FC Interface on page 286
- Setting the Maximum Number of FIP Login Sessions per FC Fabric on page 287
- Setting the Maximum Number of FIP Login Sessions per Node Device on page 288
- Example: Setting Up Fibre Channel and FCoE VLAN Interfaces in an FCoE-FC Gateway Fabric on page 171
- Understanding Fibre Channel Terminology on page 131

Understanding Load Balancing in an FCoE-FC Gateway Proxy Fabric

You can balance or rebalance the load on the ports in an FCoE-FC gateway proxy fabric in order to avoid overutilizing or underutilizing the links. Load balancing is distributing sessions across the available native Fibre Channel (FC) interfaces (NP_Ports) that belong to a local gateway FC fabric to create a relatively equal load on all the fabric links. Load rebalancing is redistributing the existing sessions across the available NP_Port links on a local gateway FC fabric.

NOTE: A session is a fabric login (FLOGI) or fabric discovery (FDISC) login to the FC SAN fabric. Session does not refer to end-to-end server-to-storage sessions.

The fabric-facing NP_Port links of an FCoE-FC gateway use different load-balancing methods than the FCoE-network-facing Ethernet links.

Balancing the load on FCoE-FC gateway NP_Port links consists of two steps:

1. Choosing the algorithm used to balance and rebalance the link load
2. Choosing whether to rebalance link loads automatically or only when you explicitly request a rebalance (load-rebalancing method)

You can configure a different load-balancing algorithm and use a different rebalancing method for each local FC fabric on the FCoE-FC gateway. The load-balancing algorithm and automated rebalancing, if configured, apply to all NP_Port interfaces in the local FC fabric.

This topic describes:

- Load-Balancing Algorithms on page 64
- Load-Rebalancing Methods on page 68
- NP_Port Interface FIP Session Limit Effect on Load Balancing on page 69
- Load-Balancing Triggers and Timing on page 69
- Load Rebalancing Behavior When a Link Goes Down on page 71

- Interface Load Calculation Algorithm on page 72
- Load-Balancing Scenarios on page 73

Load-Balancing Algorithms

You can choose one of three load-balancing algorithms to configure the way the switch balances the link loads. The switch uses the configured algorithm to balance the link loads when NP_Ports are initialized and whenever link loads are rebalanced. Regardless of whether you configure automated load rebalancing or use on-demand load rebalancing, the switch uses the configured algorithm to balance the link load:

- Simple load balancing—The switch assigns each ENode FLOGI session and VN_Port FDISC session to the least-loaded link. The switch can place FDISC sessions on a different link than the parent FLOGI session (an ENode FLOGI session and its subsequent FDISC sessions can be placed on different links). Simple load balancing is the default load-balancing algorithm. Rebalancing the link load disrupts only selected sessions to minimize the impact (the switch uses an algorithm to log out only the sessions that need to be moved to other links to balance the load when those sessions log in again).
- ENode-based load balancing—When an ENode logs in to the fabric, the switch places all subsequent VN_Port FDISC sessions associated with that ENode on the same link as the ENode FLOGI session, regardless of the link load. New ENode FLOGIs are placed on the least-loaded link. The switch calculates the link load based on the combined total of FLOGIs and FDISCs on each NP_Port link. Rebalancing the link load disrupts all sessions (all sessions log out and then log in again).
- FLOGI-based load balancing—Similar to ENode-based load balancing; when an ENode logs in to the fabric, the switch places all subsequent VN_Port FDISC sessions associated with that ENode on the same link as the ENode FLOGI session, regardless of the link load. New ENode FLOGIs are placed on the least-loaded link.

  One difference between ENode-based load balancing and FLOGI-based load balancing is that the switch calculates the link load based only on the number of FLOGIs on each NP_Port link. The algorithm does not count FDISCs. Another difference is that instead of disrupting all sessions on a link load rebalance, the system disrupts only selected sessions to minimize the impact (the switch uses an algorithm to log out only the sessions that need to be moved to other links to balance the load when those sessions log in again).

NOTE: **Changing the load-balancing algorithm when FCoE sessions are running forces the FCoE sessions to log out and then log in again.**

If you do not explicitly configure the load-balancing algorithm, the switch uses simple load balancing by default on the all NP_Port interfaces that belong to a given local FC fabric.

The following sections describe how each algorithm works, its advantages and disadvantages, and what happens when NP_Port links come up for the first time, when an NP_Port link is added to existing links, and when you rebalance the link load:

Simple Load Balancing

Simple load balancing provides the most equal load balancing across links because each VN_Port FDISC session can be assigned to the least-loaded link, regardless of whether the parent ENode FLOGI session is on that link. (The parent ENode is the ENode that originates the logins to the fabric. After the parent ENode logs in, the VN_Ports on that ENode can log in to the fabric using FDISC.)

The FCoE-FC gateway performs simple load balancing by default on the NP_Ports that connect the gateway to the FC SAN. When an ENode sends a FLOGI request to the gateway, the gateway checks the NP_Ports that connect it to the FC SAN and assigns the new session to the least-loaded link.

Every time an ENode sends a FLOGI or an FDISC request, the gateway assigns the new session to the least-loaded NP_Port link. After the gateway assigns an ENode FLOGI session to an NP_Port, subsequent FDISC requests by the same ENode can result in sessions being assigned to different NP_Ports, because the gateway always assigns the new session to the least-loaded interface.

NOTE: **Because VN_Port sessions might be placed on a different link than their parent ENode, if the link that contains the ENode goes down, only the ENode session and any of its VN_Port sessions that are on that link go down. VN_Port sessions on other links remain active as long as the link is up and the VN_Port is not logged out.**

When a new link comes up, the switch logs out enough sessions so that when the sessions log in again, they are placed on the new link and the link loads are balanced. The switch uses an algorithm to log out sessions in the least disruptive manner by first logging out FDISCs whose FLOGI is not on the same link, then the least-loaded FLOGIs (loaded in terms of related FDISC logins).

Similarly, when you rebalance an existing link load, the switch logs out only enough sessions so that when the sessions log in again, they balance the load on the existing links. In this case (rebalance without a new link up), the switch takes into account the dependencies between FLOGIs and FDISCs when selecting sessions to log out.

The simple load-balancing algorithm uses the sum of the FLOGI and FDISC sessions to determine the session load on each link for both initial load balancing and load rebalancing.

ENode-Based Load Balancing

ENode-based load balancing can result in a less balanced load across the NP_Port links because the VN_Port FDISC sessions are assigned to the same link as the parent ENode FLOGI session, regardless of how many FDISC sessions are associated with the ENode. However, ENode-based load balancing has the advantage of keeping all of the sessions associated with a particular ENode on one link, which provides better control and predictability.

When you use the ENode-based load-balancing algorithm, the gateway assigns the ENode to an NP_Port link when the ENode sends its FLOGI message to the gateway. The gateway places the ENode session on the least-loaded link at that time. The VN_Port FDISC sessions associated with an ENode are placed on the same link as the ENode FLOGI session, regardless of the link load. Essentially, the ENode sessions are load-balanced, but the VN_Port sessions are not.

ENode-based load balancing ensures that each ENode and its associated VN_Port sessions are assigned to the same NP_Port link. ENode-based load balancing provides more control and predictability and ensures that if the link carrying an ENode goes down, all of the ENodes associated VN_Port sessions also go down.

The disadvantage of ENode-based load balancing is that if one ENode has a large number of sessions and the other ENodes do not, the link that carries the ENode with the large number of sessions might have a much larger load than the other NP_Port links in the gateway proxy fabric.

For example, if a gateway fabric has two NP_Ports connected to the FC fabric, and two ENodes log in to the fabric, one ENode session is placed on each link. If two VN_Port sessions are initiated on one of the ENodes, those sessions are placed on the same link as the parent ENode. If 1000 VN_Port sessions are initiated on the other ENode, all of the 1000 VN_Port sessions are placed on the same link as that ENode. In this case, one link has 3 sessions (1 ENode FLOGI session and 2 VN_Port FDISC sessions) and the other link has 1001 sessions (1 ENode FLOGI session and 1000 VN_Port FDISC sessions).

When a new link comes up or when you rebalance an existing load, the switch logs out all sessions (FLOGIs and FDISCs) in the fabric. As the sessions log in again, the switch assigns them to NP_Ports in a balanced manner, with all FDISCs assigned to the same link as the parent FLOGI. A new link coming up or a rebalance disrupts all of the existing sessions.

The ENode-based load-balancing algorithm uses the sum of the FLOGI and FDISC sessions to determine the session load on each link for both initial load balancing and load rebalancing.

FLOGI-Based Load Balancing

FLOGI-based load balancing is similar to ENode-based load balancing in most ways:

- It can result in a less balanced load across the NP_Port links because the VN_Port FDISC sessions are assigned to the same link as the parent ENode FLOGI session, regardless of how many FDISC sessions are associated with the ENode.
- When an ENode logs in with a FLOGI, the gateway places the session on the least-loaded link, and the FDISC logins associated with the FLOGI are placed on the same link, regardless of link load.
- Provides control and predictability because each ENode and its associated VN_Port (FDISC) sessions are assigned to the same link, so if the link an ENode is on goes down, all of its associated sessions also go down.
- If one ENode has a large number of sessions and the other ENodes do not, the link that carries the ENode with the large number of sessions might have a much larger load than the other NP_Port links in the gateway proxy fabric.

FLOGI-based load balancing differs from ENode-based load balancing in two important ways:

1. The switch uses the sum of the FLOGI sessions on a link to determine the link load. The switch does not use FDISC sessions when calculating the number of sessions on a link. (ENode-based load balancing uses the sum of the FLOGI and FDISC sessions to calculate the number of sessions on a link.)
2. When a new link comes up or when you rebalance an existing load, the switch logs out enough FLOGI (and FDISC) sessions so that when the FLOGI sessions log in again, the load is balanced. The switch balances the load based only on the number of FLOGI sessions, not the sum of FLOGI and FDISC sessions. However, the FDISC sessions associated with a FLOGI follow the FLOGI to the new link if the FLOGI session is part of the rebalancing.

The FLOGI-based load-balancing algorithm uses only the FLOGI sessions to determine the session load on each link for both initial load balancing and load rebalancing.

Load-Balancing Algorithm Comparison

Table 4 on page 67 compares the three load-balancing algorithms and summarizes their differences, advantages, and disadvantages.

Table 4: Load-Balancing Algorithm Comparison

Load-Balancing Algorithm	Session Assignment	Session Disruption on Rebalance	Session Count Method	Advantages	Disadvantages
Simple (default algorithm)	FDISC sessions can be placed on different links than the parent FLOGI session	Minimum number of selected sessions logged out (FDISC sessions can be logged out independent of the parent FLOGI session)	Sum of FLOGI and FDISC sessions	• Most equal session distribution across links • Minimum number of sessions logged out when rebalancing • Least disruptive algorithm	• Less session control and predictability

Table 4: Load-Balancing Algorithm Comparison *(continued)*

Load-Balancing Algorithm	Session Assignment	Session Disruption on Rebalance	Session Count Method	Advantages	Disadvantages
ENode-based	FDISC sessions are always placed on the same link as the parent FLOGI session	All sessions are logged out	Sum of FLOGI and FDISC sessions	• Better session control and predictability (on link down, all sessions associated with an ENode go down)	• Most disruptive algorithm; all sessions logged out on rebalance • Might result in less balanced link load because FDISCs are placed on the same link as parent FLOGI
FLOGI-based	FDISC sessions are always placed on the same link as the parent FLOGI session	Minimum number of selected sessions logged out (but FDISC sessions logged out when parent FLOGI session is logged out)	FLOGI sessions only (FDISC sessions not included in the session count)	• Better session control and predictability (on link down, all sessions associated with an ENode go down) • Minimum number of sessions logged out when rebalancing	• Might result in less balanced link load because FDISCs are placed on the same link as parent FLOGI

Load-Rebalancing Methods

The load-rebalancing method determines the way the system redistributes sessions to balance the load on the NP_Ports that belong to a local FC fabric on an FCoE-FC gateway.

You can rebalance the existing load on existing NP_Port links using either of two methods:

- Automated load rebalancing—When a load rebalancing trigger occurs, the switch automatically rebalances the link loads by redistributing the sessions across the active NP_Port links. There are three possible load rebalancing triggers:

  1. When you enable automated load rebalancing, the switch checks the load balance on the existing NP_Port links. If the links are already balanced, the switch does not rebalance the link load. If the links are not balanced, the switch rebalances the link loads using the configured load-balancing algorithm.

     Enabling automated load rebalancing causes sessions to be logged out in accordance with the configured load-balancing algorithm if the link load is unbalanced. If the link load is already balanced when you enable automated load rebalancing, the links are not rebalanced. (Disabling automated load rebalancing is not disruptive because the link load is already balanced.)

  2. When a new NP_Port link comes up on a local FCoE-FC gateway fabric, the switch rebalances the link load using the configured load-balancing algorithm if automated load balancing is enabled.

  3. When the port speed is changed (unless the port speed change does not change the actual port speed, for example, changing the port speed from auto to 8 Gbps).

Use automated load rebalancing if you want link loads to be rebalanced automatically when a load-balancing trigger occurs, instead of at times of your choosing. Keep in mind that load rebalancing is a disruptive event (sessions are logged out).

- On-demand load rebalancing—You choose when to rebalance the NP_Port links by explicitly requesting a load rebalance using an operational command. The system rebalances the link load only when you issue the rebalancing command.

  Use on-demand load rebalancing if you only want to rebalance the link load once or if you want to rebalance the link loads at controlled times instead of automatically.

  You can also request a load rebalancing *dry run*. A dry run simulates rebalancing and lists the sessions that might be affected if you choose to perform an actual load-rebalancing operation. The link loads are not rebalanced when you request a dry run.

NP_Port Interface FIP Session Limit Effect on Load Balancing

The maximum number of FIP login sessions configured for each NP_Port interface affects load balancing. When an interface reaches its maximum number of FIP login sessions, that interface is removed from the list of interfaces used for load balancing. The other interfaces in the gateway fabric continue to accept ENode login sessions until they reach their configured maximum session limit. Only interfaces that have not reached their maximum session limit are included in the load-balancing calculations.

NOTE: **If all NP_Port interfaces in a gateway fabric reach their FIP login session limits, the fabric sends subsequent multicast discovery advertisements (MDAs) with the availability bit set to 0 (zero) to prevent additional ENode login attempts. While the maximum number of sessions is running on the gateway fabric, ENodes cannot use that fabric to log in to the FC switch. When the number of sessions falls below the maximum, the gateway sets the availability bit in MDAs to 1 so that ENodes can log in to the fabric again.**

Load-Balancing Triggers and Timing

Several events trigger load balancing. Some of the events trigger load balancing only when automated load balancing is enabled. Other events trigger load rebalancing whether or not automated rebalancing is enabled.

This section describes the load-balancing triggers, what happens when the trigger action occurs, and how the switch determines if and when to balance the link load:

- Load-Balancing Triggers on page 69
- Load-Balancing Timer on page 70

Load-Balancing Triggers

Table 5 on page 70 describes the four different events can trigger load balancing or load rebalancing. In every case, link load rebalancing uses the configured load-balancing algorithm to determine the placement of sessions on links.

Table 5: Load-Balancing Triggers and Actions

Trigger Event	Action
New link comes up	Triggers a load-rebalancing operation regardless of whether or not automated load rebalancing is enabled. (The new link has no sessions, so the sessions on other links must be redistributed to balance the load.) The link load is not rebalanced if there are no sessions on the existing links or if there are so few sessions on the existing links that they cannot be redistributed.
On-demand load rebalancing request issued from CLI	The switch checks the NP_Port link load. If the load is not balanced across the links, the switch rebalances the link load. If the load is already balanced, nothing happens. NOTE: Requesting a dry run displays sessions that might be disrupted if you rebalance the link load, but does not rebalance the link load.
Automated load balancing configured for the first time	The switch checks the NP_Port link load. If the load is not balanced across the links, the switch rebalances the link load. If the load is already balanced, nothing happens.
NP_Port speed change	If automated rebalancing is enabled, changing the port speed brings the port up and down (flaps the port) and causes the switch to rebalance the link loads. If the port speed change does not change the actual port speed (for example, changing the port speed from *auto* to 8 Gbps), the link loads are not rebalanced. If automated rebalancing is not enabled, port speed changes do not cause link load rebalancing.

NOTE: When an NP_Port link goes down, it does not trigger load rebalancing. The loads on the remaining active links are already balanced, and as the sessions logged out from the down link log in again, they are they assigned to links in a balanced manner determined by the configured load-balancing algorithm.

Load-Balancing Timer

When you trigger load balancing from the CLI, the load-balancing action occurs immediately after you execute the command. However, when a load-balancing trigger occurs that is not a CLI command, the switch does not balance the link loads immediately. Instead, the switch follows an intelligent timer process:

1. The switch checks the current load balance on the NP_Port links in the local gateway FC fabric. If the load is already balanced, the switch does nothing, and there is no session disruption.
2. If the check shows that the link load is not balanced, the switch starts a 10-second timer. If no other load-balancing triggers occur during the 10-second interval, the switch rebalances the load.

If another load-balancing trigger occurs during the 10-second interval, the timer resets to 10 seconds. The 10-second timer prevents the switch from performing multiple disruptive load-rebalancing actions in a short period of time.

NOTE: The switch processes new sessions that log in after the timer starts in the normal manner. The new sessions are considered in the load-balancing evaluation and operation.

3. At a maximum of 30 seconds after the first load-balancing trigger occurs, the switch checks the link load balance again. If the links are already balanced, the switch cancels the load-rebalancing operation. If the links are not balanced, the switch rebalances the link loads.

NOTE: If the trigger event that started the load-rebalancing timer is no longer valid when the timer elapses, the switch cancels the rebalancing operation. For example, if a new NP_Port link comes up and triggers the timer, then goes down before the timer expires, the original link up event is no longer valid, and the switch cancels the rebalancing operation (unless another valid rebalancing trigger occurs in that time frame).

When a link load rebalancing operation is in progress, the switch defers any load-rebalancing triggers that occur until the load-rebalancing operation is complete. The new rebalancing operation begins after the current rebalancing operation finishes if a check shows that rebalancing is required.

If you explicitly request load rebalancing from the CLI using the **request fibre-channel proxy load-rebalance** operational command, the switch rejects the command and displays an error message stating that rebalancing is already in progress.

Load Rebalancing Behavior When a Link Goes Down

If an NP_Port link goes down, the ENode and VN_Port sessions on that link are logged out. The ENodes and VN_Port sessions log in again and are assigned to NP_Port links based on the link load and the load-balancing algorithm. If a link goes down, the switch does not rebalance the remaining load on the remaining links to avoid disrupting the existing ENode and VN_Port sessions. (Also, it is not necessary to rebalance the links in that manner because after a link goes down, the sessions on the remaining links are already balanced. As the logged out sessions log back in, the switch places them on the remaining active links in a balanced manner, according to the configured load-balancing algorithm.)

NOTE: When you use the simple load-balancing algorithm, an ENode and its associated VN_Port sessions might be on different links. In that case, if the NP_Port with the ENode goes down, only the VN_Ports on the same link are logged out. VN_Ports on other links remain up and running.

Interface Load Calculation Algorithm

A weighted round-robin (WRR) algorithm determines the interface load based on:

- The current number of sessions on the interface

NOTE: The configured load-balancing algorithm determines how the switch counts the number of sessions. For simple and ENode-based load balancing, the number of sessions is the sum of the FLOGI and FDISC sessions on each link. For FLOGI-based load balancing, the number of sessions is the sum of the FLOGI sessions on each link.

- The interface weight, which is the speed of the Fibre Channel link (2 Gbps, 4 Gbps, or 8 Gbps)

The interface load algorithm is:

(number-of-sessions * max-weight) / weight

where *max-weight* is an internal constant.

If the load on the FC interfaces is equal, the session is assigned to the interface with the highest link speed (the greatest weight).

For example, if the three FC interfaces have the characteristics shown in Table 6 on page 72, the loads of the interfaces are not equal:

Table 6: FC Interface Session-Based Load-Balancing Characteristics for Unequal Loads

Interface	Number of Sessions	Weight (Speed)
fc-0/0/0	4	4 Gbps
fc-0/0/1	1	2 Gbps
fc-0/0/2	8	8 Gbps

In this example, interfaces fc-0/0/0 and fc-0/0/2 have a greater load than fc-0/0/1. For simple load balancing, the gateway assigns the next new FLOGI or FDISC to fc-0/0/1 because it is the least-loaded interface. For both ENode-based and FLOGI-based load balancing, the gateway assigns the next new FLOGI to fc-0/0/1 because it is the least-loaded interface. Then all VN_Port FDISCs from that ENode follow the ENode FLOGI and are also assigned to fc-0/0/1 regardless of the link load.

For another example, if the three FC interfaces have the characteristics shown in Table 7 on page 73, the loads of the interfaces are equal:

Table 7: FC Interface Session-Based Load-Balancing Characteristics for Equal Loads

Interface	Number of Sessions	Weight (Speed)
fc-0/0/0	4	4 Gbps
fc-0/0/1	2	2 Gbps
fc-0/0/2	8	8 Gbps

In this case, all interfaces have the same relative load. For simple load balancing, the gateway assigns the next new FLOGI or FDISC to fc-0/0/2 because although the loads of the three interfaces are equal, fc-0/0/2 has the greatest weight. For both ENode-based and FLOGI-based load balancing, the gateway assigns the next new FLOGI to fc-0/0/2, and all VN_Port FDISCs from that ENode follow the ENode FLOGI and are also assigned to fc-0/0/2 regardless of the link load.

After the gateway establishes a session between an ENode or a VN_Port and an FC switch on an NP_Port, the session remains on that NP_Port until the ENode or VN_Port performs a LOGO.

If the physical FC interface link goes down, the FLOGI and FDISC sessions on the down link are logged out. The ENodes and VN_Ports log in again to start new sessions on other NP_Ports in the local gateway FC fabric in accordance with the configured load-balancing algorithm (assuming there is more than one NP_Port connected to the FC fabric).

Load-Balancing Scenarios

The configured load-balancing algorithm, the sequence in which ENodes log in to the FC network, the current session count (number of sessions per interface) and the interface speed determine the way the session load is balanced across the native FC interfaces (NP_Ports) in a gateway FC fabric. Whether you are balancing the link load for the first time or rebalancing an existing link load, the way the load is distributed across the active links is the same.

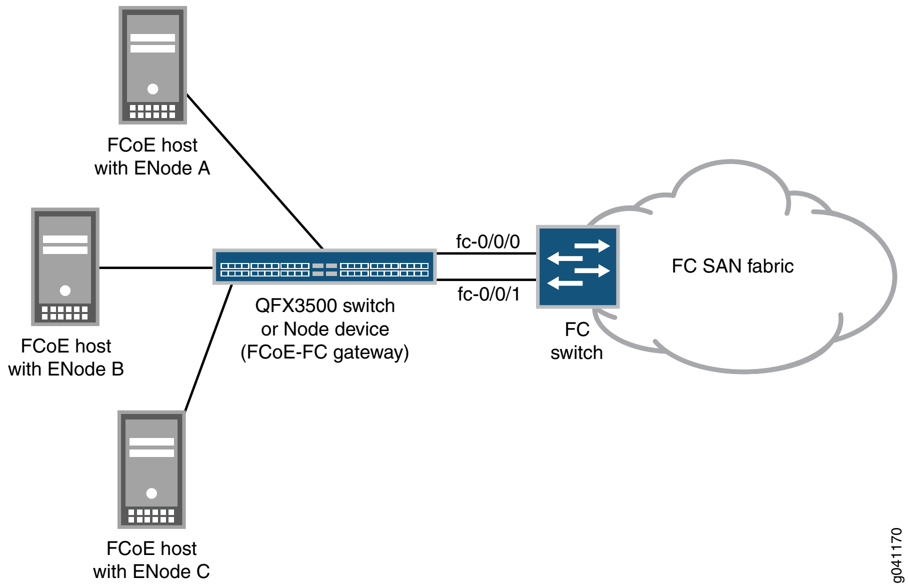
NOTE: **The way the switch counts the number of sessions on a port depends on the load-balancing algorithm. For simple and ENode-based load balancing, the sum of the FLOGI and FDISC sessions equals the session count. For FLOGI-based load balancing, only the FLOGI sessions are counted in the total session count.**

The following scenarios demonstrate how sessions are assigned to links for each load-balancing algorithm:

- Simple Load-Balancing Algorithm Scenario on page 74
- ENode-Based Load-Balancing Algorithm Scenarios on page 75
- FLOGI-Based Load-Balancing Algorithm Scenarios on page 77

All of the scenarios use the topology shown in Figure 6 on page 74.

Figure 6: Sample Load-Balancing Topology



Simple Load-Balancing Algorithm Scenario

Simple load balancing results in the most equal load distribution among the NP_Ports connected to an FC SAN fabric because VN_Port FDISC sessions do not need to “follow” the parent ENode FLOGI session on the same link between the gateway and the FC fabric. When a new FLOGI or FDISC session is initiated, it is assigned to the least-loaded link.

The simple load-balancing algorithm example uses the topology shown in Figure 6 on page 74 and has the following characteristics:

- QFX Series configured as an FCoE-FC gateway
- Two gateway NP_Ports, **fc-0/0/0** and **fc-0/0/1**, connected to an FC SAN fabric switch at a speed of 8 Gbps
- Three ENodes, ENode_A, ENode_B, and ENode_C connected to the gateway
- NP_Ports **fc-0/0/0** and **fc-0/0/1**, and ENode_A, ENode_B, and ENode_C, belong to the same local FC fabric on the gateway

When the NP_Ports initialize, they send FLOGI messages to the FC switch and log in to the FC SAN fabric. The gateway then advertises the fabric to the ENodes on the Ethernet side of the network. At this point, the load on both of the NP_Ports is equal. Now the ENodes and VN_Ports start to log in to the fabric:

1. ENode_A sends a FLOGI to log in to the fabric. Because the loads on the two NP_Ports are equal, the session for ENode_A is randomly placed on one of the links. In this example, the ENode_A FLOGI session is placed on port **fc-0/0/0**.
2. Enode_B logs in. Because the load is less on port **fc-0/0/1**, the Enode_B FLOGI session is placed on port **fc-0/0/1**.

3. ENode_C logs in. Because the link loads are equal, the ENode_C login session is randomly placed on one of the links. In this example, the ENode_C login session is placed on port **fc-0/0/0**.
4. A VN_Port on ENode_A sends an FDISC to log in to the fabric. Because port **fc-0/0/1** currently is the least-loaded link, the VN_Port session is placed on port **fc-0/0/1**, even though its parent ENode session is on port **fc-0/0/0**.
5. As each new VN_Port session comes up, it is placed on the least-loaded link, regardless of the link on which its parent ENode session is placed.

ENode-Based Load-Balancing Algorithm Scenarios

ENode-based load balancing ensures that VN_Port FDISC sessions are placed on the same link as their parent ENode FLOGI sessions, regardless of the link load. ENode-based load balancing can result in a less-balanced load among the NP_Port links, but it provides the control and predictability of keeping ENodes and their VN_Port sessions on the same link.

The examples in this section use the topology shown in Figure 6 on page 74.

- QFX Series configured as an FCoE-FC gateway
- Two gateway NP_Ports, **fc-0/0/0** and **fc-0/0/1**, connected to an FC SAN fabric switch at a speed of 8 Gbps
- Three ENodes connected to the gateway:
  - ENode_A, which has 2 VN_Port FDISC sessions
  - ENode_B, which has 20 VN_Port FDISC sessions
  - ENode_C, which has 100 VN_Port FDISC sessions
- NP_Ports **fc-0/0/0** and **fc-0/0/1**, and ENode_A, ENode_B, and ENode_C, belong to the same local FC fabric on the gateway

When the NP_Ports initialize, they send FLOGI messages to the FC switch and log in to the FC SAN fabric. The gateway then advertises the fabric to the ENodes on the Ethernet side of the network. At this point, the load on both of the NP_Ports is equal. Now the ENodes and VN_Ports start to log in to the fabric. As the following two scenarios show, how these sessions are placed on the links depends on the sequence in which they log in to the fabric.

Scenario 1:

1. ENode_A sends a FLOGI to log in to the fabric. Because the loads on the two NP_Ports are equal, the session for ENode_A is randomly placed on one of the links. In this example, the ENode_A FLOGI session is placed on port **fc-0/0/0**.
2. ENode_B logs in. Because the load is less on port **fc-0/0/1**, the ENode_B FLOGI session is placed on port **fc-0/0/1**.

3. The two VN_Ports on ENode_A log in to the fabric. Their sessions are placed on port **fc-0/0/0**, following ENode_A on the link. Now port **fc-0/0/0** has a greater load (one FLOGI session plus two FDISC sessions) than port **fc-0/0/1** (one FLOGI session).
4. The 20 VN_Ports on ENode_B log in to the fabric. Their sessions are placed on port **fc-0/0/1**, following ENode_B on the link. Now port **fc-0/0/0** has a lesser load (one FLOGI, two FDISC) than port **fc-0/0/1**.
5. ENode_C logs in. Because the load is less on port **fc-0/0/0**, the ENode_C FLOGI session is placed on port **fc-0/0/0**.
6. The 100 VN_Ports on ENode_C log in to the fabric. Their sessions follow the ENode_C session onto port **fc-0/0/0**.
7. If more VN_Ports come up, their FDISC sessions are placed on the same link as the corresponding parent ENode session.

Scenario 2:

1. ENode_A sends a FLOGI to log in to the fabric. Because the loads on the two NP_Ports are equal, the session for ENode_A is randomly placed on one of the links. In this example, the ENode_A FLOGI session is placed on port **fc-0/0/0**.
2. ENode_B logs in. Because the load is less on port **fc-0/0/1**, the ENode_B FLOGI session is placed on port **fc-0/0/1**.
3. The two VN_Ports on ENode_A log in to the fabric. Their sessions are placed on port **fc-0/0/0**, following ENode_A on the link. Now port **fc-0/0/0** has a greater load (one FLOGI session plus two FDISC sessions) than port **fc-0/0/1** (one FLOGI session).
4. In this step, the login sequence in Scenario 2 differs from the login sequence in Scenario 1, resulting in a different placement of sessions on the links, and therefore a different load on the links. ENode_C logs in before the ENode_B VN_Ports log in, which changes the session count on the links compared to the first scenario. Because the load in this scenario is less on port **fc-0/0/1**, the ENode_C FLOGI session is placed on port **fc-0/0/1** (instead of port **fc-0/0/0** as in the first scenario).
5. The 20 VN_Ports on ENode_B log in to the fabric. Their sessions are placed on port **fc-0/0/1**, following ENode_B on the link. Now port **fc-0/0/0** carries one FLOGI and two FDISC sessions, and port **fc-0/0/1** carries two FLOGI and 20 FDISC sessions.
6. The 100 VN_Ports on ENode_C log in to the fabric. Their sessions follow the ENode_C session onto port **fc-0/0/1**. Now port **fc-0/0/1** carries 2 FLOGI and 120 FDISC sessions, whereas port **fc-0/0/0** carries one FLOGI and two FDISC sessions.
7. If more VN_Ports come up, their FDISC sessions are placed on the same link as the corresponding parent ENode session.

Because of the sequence of ENode logins in Scenario 2, port **fc-0/0/1** carries a greater load than port **fc-0/0/0**. If the simple load-balancing algorithm had been used, the FLOGI and FDISC sessions would be allocated to the two links evenly. However, because the FDISC sessions are placed on the same link as their parent FLOGI sessions, this example demonstrates how using the ENode-based load-balancing algorithm can lead to scenarios in which the link loads are not equal.

FLOGI-Based Load-Balancing Algorithm Scenarios

FLOGI-based load balancing is similar in many ways to ENode-based load balancing. An important difference that affects how the switch places sessions on links is that for FLOGI-based load balancing, only the FLOGI sessions are counted when the link load is calculated. FDISC sessions are not counted to determine the link load. Because ENode-based load balancing uses the sum of the FLOGI and FDISC sessions to determine the link load, an interface with exactly the same combination of FLOGI and FDISC sessions can have a different session count depending on the algorithm used. A different session count can change the interface to which the switch assigns the next session.

As with ENode-based load balancing, FLOGI-based load balancing ensures that VN_Port FDISC sessions are placed on the same link as their parent ENode FLOGI sessions, regardless of the link load. FLOGI-based load balancing can result in a less-balanced load among the NP_Port links, but it provides the control and predictability of keeping ENodes and their VN_Port sessions on the same link.

The examples in this section use the topology shown in Figure 6 on page 74.

- QFX Series configured as an FCoE-FC gateway
- Two gateway NP_Ports, **fc-0/0/0** and **fc-0/0/1**, connected to an FC SAN fabric switch at a speed of 8 Gbps
- Three ENodes connected to the gateway:
  - ENode_A, which has 2 VN_Port FDISC sessions
  - ENode_B, which has 20 VN_Port FDISC sessions
  - ENode_C, which has 100 VN_Port FDISC sessions
- NP_Ports **fc-0/0/0** and **fc-0/0/1**, and ENode_A, ENode_B, and ENode_C, belong to the same local FC fabric on the gateway

When the NP_Ports initialize, they send FLOGI messages to the FC switch and log in to the FC SAN fabric. The gateway then advertises the fabric to the ENodes on the Ethernet side of the network. At this point, the load on both of the NP_Ports is equal. Now the ENodes and VN_Ports start to log in to the fabric.

Because FLOGI-based load balancing does not count FDISC sessions when calculating the link load, how the sessions are placed on the link depends only on the number of FLOGI sessions per interface, not on the number of FLOGI sessions plus FDISC sessions. This means that an ENode with a FLOGI session and many FDISC sessions is counted as having the same load as an ENode with a FLOGI session and no FDISC sessions.

Scenario 1:

1. ENode_A sends a FLOGI to log in to the fabric. Because the loads on the two NP_Ports are equal, the session for ENode_A is randomly placed on one of the links. In this example, the ENode_A FLOGI session is placed on port **fc-0/0/0**.
2. ENode_B logs in. Because the load is less on port **fc-0/0/1**, the ENode_B FLOGI session is placed on port **fc-0/0/1**.

3. The two VN_Ports on ENode_A log in to the fabric. Their sessions are placed on port **fc-0/0/0**, following ENode_A on the link. However, unlike simple load balancing or ENode-based load balancing, the session count of the two ports is still equal (one session each) because the FDISC sessions are not used in the session count.
4. The 20 VN_Ports on ENode_B log in to the fabric. Their sessions are placed on port **fc-0/0/1**, following ENode_B on the link. Again, unlike simple load balancing or ENode-based load balancing, the session count of the two ports is still equal (one session each) because the FDISC sessions are not used in the session count.
5. ENode_C logs in. Because the link loads are counted as equal, the ENode_C login session is randomly placed on one of the links. In this example, the ENode_C login session is placed on port **fc-0/0/0**.
6. The 100 VN_Ports on ENode_C log in to the fabric. Their sessions follow the ENode_C session onto port **fc-0/0/0**.
7. If more VN_Ports come up, their FDISC sessions are placed on the same link as the corresponding parent ENode session.

If a fourth ENode, ENode_D, sends a FLOGI to log in to the fabric, it is placed on port **fc-0/0/1** because port **fc-0/0/0** has a session count of two (two FLOGIs from ENode_A and ENode_C, FDISCs not counted) and port **fc-0/0/1** has a session count of one (one FLOGI from ENode_B, FDISCs not counted), so port **fc-0/0/1** is the least-loaded port.

With FLOGI-based load balancing, it is possible for ENodes with many FDISC sessions to be placed on the same link, whereas ENodes with few FDISC sessions are placed on different links because only FLOGIs are used in the session count.

Related Documentation

- Understanding an FCoE-FC Gateway on page 29
- Understanding FCoE-FC Gateway Functions on page 33
- Understanding Interfaces on an FCoE-FC Gateway on page 50
- Defining the Proxy Load-Balancing Algorithm on page 279
- Simulating On-Demand Fibre Channel Link Load Rebalancing (Dry Run Test) on page 280
- Example: Configuring Automated Fibre Channel Interface Load Rebalancing on page 250
- **show fibre-channel proxy fabric-state on page 479**
- **request fibre-channel proxy load-rebalance on page 401**
- Monitoring Fibre Channel Interface Load Balancing on page 383

Understanding OxID Hash Control for FCoE Traffic Load Balancing on QFX Series Standalone Switches

The originator exchange identifier (OxID) field is one of several fields that the switch can use in its hash function computation for FCoE traffic load balancing over multiple outgoing links in an Ethernet link aggregation group (LAG) on ports that face an FCoE forwarder (FCF). The originator of an exchange between a pair of Fibre Channel (FC) endpoints (such as an FCoE host and an FC storage device) uses the OxID field as an identifier for that exchange. The originator also uses the OxID field to track the progress of the series of sequences that comprise the exchange.

When FCoE traffic traverses a LAG that faces an FCF, it can take multiple different links between the source and destination endpoints. The idea is to distribute the FCoE traffic across the FCF-facing LAG links, thus balancing the link load. The switch creates a hash value from some of the packet header fields, and uses the hash value to assign each packet to one of the LAG links. The switch always uses five packet header fields to compute the hash value:

- Source ID (SID)
- Destination ID (DID)
- Fabric ID (FID)
- Source Port ID (SPID)
- Source Module ID (SMID)

In addition, the OxID field is included by default in the FCoE load-balancing hash computation. However, if you do not want to use the OxID field in the FCoE load-balancing hash computation, you can remove it from the computation by using the **set forwarding-options hash-key family fcoe oxid disable** command.

Including the OxID field in the load-balancing hash computation allows different exchanges between a pair of Fibre Channel (FC) endpoints (such as an FCoE host and an FC storage device) to take different paths across the network, thus improving the aggregate network throughput.

However, if the paths between different sets of FC endpoints have common links, congestion on one set of FC endpoints can affect the other set of endpoints. Such congestion can happen if the FCoE traffic on the two sets of endpoints uses the same priority (IEEE 802.1p code point). It is common for networks to use priority 3 (IEEE 802.1p code point 011) for FCoE traffic. However, on the QFX3500 switch, you can assign different IEEE priorities to different lossless FCoE flows as described in *Understanding CoS IEEE 802.1p Priorities for Lossless Traffic Flows* to further separate the traffic flows.

Related Documentation

- Understanding OxID Hash Control for FCoE Traffic Load Balancing on QFabric Systems on page 163
- Enabling and Disabling CoS OxID Hash Control on QFX Series Standalone Switches on page 281

Understanding VN_Port to VF_Port FIP Snooping on an FCoE Transit Switch

Fibre Channel over Ethernet (FCoE) Initialization Protocol (FIP) snooping is a security mechanism that is designed to prevent unauthorized access and data transmission to a Fibre Channel (FC) network. It works by filtering traffic to permit only servers that have logged in to an FC network to access that network.

You explicitly enable VN_Port to VF_Port (VN2VF_Port) FIP snooping (FC-BB-5) on FCoE VLANs when the QFX Series is an FCoE transit switch at the access edge that connects FCoE devices on the Ethernet network to FC switches or gateways at the FC storage area network (SAN) edge. The transit switch applies FIP snooping filters at the ports associated with the FCoE VLANs on which you enable VN2VF_Port FIP snooping. An FCoE transit switch is a data center bridging (DCB) switch with FIP snooping capability.

An FCoE device that has a converged network adapter (CNA) uses the FIP process to log in to the FC network as an FCoE Node (ENode). The login process establishes a dedicated virtual link between a virtual N_Port (VN_Port) on the ENode and a virtual F_Port (VF_Port) on the FC switch. This dedicated virtual link emulates a point-to-point connection. The emulated connection is called a virtual link.

Virtual links pass transparently through the transit switch. The ENode VN_Port and the FC switch VF_Port do not detect the transit switch, and virtual links appear to be direct point-to-point links.

The QFX Series applies VN2VF_Port FIP snooping firewall filters at the FCoE-network facing ports associated with the FCoE VLANs on which you enable VN2VF_Port FIP snooping. FIP snooping provides security for virtual links by creating firewall filters based on information gathered (snooped) about FC devices during FIP transactions.

The QFX Series also supports VN_Port to VN_Port (VN2VN_Port) FIP snooping (FC-BB-6) to allow FCoE initiators and targets to communicate directly through the switch without going through an FCoE forwarder (FCF) or an FC switch, as described in "Understanding VN_Port to VN_Port FIP Snooping on an FCoE Transit Switch" on page 87.

NOTE: **An FCoE VLAN can support either VN2VF_Port FIP snooping (FC-BB-5) or VN2VN_Port FIP snooping (FC-BB-6), but not both. The same QFX Series switch can have multiple FCoE VLANs configured, some for VN2VF_Port FIP snooping traffic and others for VN2VN_Port FIP snooping traffic. On FCoE VLANs that are configured as VN2VN_Port snooping VLANs, VN2VF_Port FIP snooping traffic is dropped.**

When you enable VN2VF_Port FIP snooping on an FCoE VLAN, the system snoops VN_Port to VF_Port packets and enforces security only on VN2VF_Port virtual links.

When you enable VN2VN_Port FIP snooping on an FCoE VLAN, the system snoops VN_Port to VN_Port packets and enforces security only on VN2VN_Port virtual links.

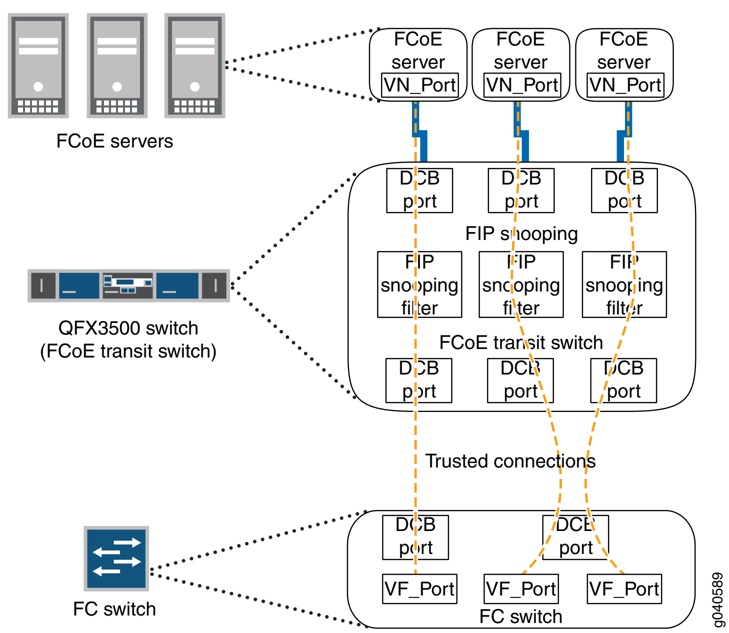
This topic describes:

FC Network Security

In traditional FC networks, the FC switch is usually a trusted entity, and server ENodes connect directly to its VF_Ports. After an ENode gains access to the network through the fabric login (FLOGI) process, the FC switch enforces zoning configurations, ensures that the ENode uses valid addresses, monitors the connection, and performs other security functions to prevent unauthorized access.

However, FCoE exposes FC frames to Ethernet networks, which do not have the same level of security as native FC networks. VN2VF_Port FIP snooping firewall filters emulate the native FC network security functions by preventing unauthorized access to the FC switch through the transit switch and by ensuring the security of the virtual link between each ENode and the FC switch, as shown in Figure 7 on page 81. VN2VF_Port FIP snooping also prevents man-in-the-middle attacks.

Figure 7: FCoE Transit Switch Performs VN2VF_Port FIP Snooping



The transit switch performs VN2VF_Port FIP snooping at the ports connected to the FCoE devices. At the SAN edge, the FC switch must be able to convert the FCoE traffic to native FC traffic.

VN2VF_Port FIP Snooping Functions

When VN2VF_Port FIP snooping is enabled, the QFX3500 transit switch sets and applies filters to block all FCoE traffic by default. The transit switch monitors FIP logins, solicitations, and advertisements that pass through it and gathers information about the ENode address and the address of the port on the FC switch. The transit switch uses the information to construct firewall filters that permit access only to logged-in ENodes. All other traffic on the VLAN is denied.

For example, when an ENode on an FCoE VLAN performs a successful login to an FC switch port, the transit switch snoops the FIP information and constructs a firewall filter that provides access for the ENode to that port on the FC switch.

The firewall filters enable FCoE frames to pass through the transit switch only on a virtual link established between an FCoE device ENode VN_Port and the FC switch VF_Port to which it has logged in. The firewall filters ensure that ENodes can only connect to the FC switches they have successfully logged in to and that only valid FCoE traffic along valid paths is transmitted. VN2VF_Port FIP snooping maintains the filters by tracking FCoE sessions (ENode to FCF sessions).

FIP Snooping Firewall Filters

The effect of the firewall filters is to protect the FCoE ports. VN2VF_Port FIP snooping performs the following actions and checks to ensure that FCoE traffic is valid:

- Denies ENodes that use the FC switch media access control (MAC) address as the source address.
- Enables ENodes to transmit FIP and FCoE frames to the FC switch address.
- Ensures that the FCoE source address the FC switch assigns or accepts is only used for FCoE traffic.
- Ensures that FCoE frames are only addressed to the accepting FC switch.

FIP Snooping Session Scalability

Enhanced FIP snooping session scaling, which supports up to 2,500 sessions, is enabled by default. On QFabric systems, if you want to disable enhanced FIP snooping scaling (which reduces the number of supported sessions to 376 sessions), you can do so as described in “Disabling Enhanced FIP Snooping Scaling” on page 298.

By default, up to 2500 total FIP snooping sessions are supported on an interface, an FCoE-FC gateway fabric, a QFX Series switch, a QFabric Node device, or a QFabric Node group. For example, you can:

- Place all 2500 sessions on one FCoE interface.
- Split the 2500 sessions among multiple FCoE interfaces on one FCoE VLAN.
- Split the 2500 sessions among multiple FCoE interfaces on multiple FCoE VLANs.

- Split the 2500 sessions among the FCoE interfaces on multiple gateway FC fabrics on a switch.
- Split the 2500 sessions among the FCoE interfaces on multiple gateway FC fabrics on multiple Node devices in a QFabric Node group.

Regardless of how you allocate the sessions among interfaces and local FC fabrics on a QFX Series switch or on a QFabric system Node device or Node group, the combined FIP session limit is a maximum of 2500 sessions.

NOTE: **The total number of sessions the system can support is the combined number of VN2VF_Port sessions and VN2VN_Port sessions. If VN2VN_Port sessions are active, the total number of available VN2VF_Port sessions is reduced.**

VN2VF_Port FIP Snooping Implementation

You enable VN2VF_Port FIP snooping on a per-VLAN basis on VLANs that carry FCoE traffic. The switch snoops FIP frames at the ports associated with FCoE VLANs enabled for VN2VF_Port FIP snooping. The switch then installs the resulting firewall filters on the ports to ensure that all VN2VF_Port FIP snooping occurs on the switch network edge.

VN2VF_Port FIP snooping FCoE VLANs must meet the following criteria:

- An FCoE VLAN should be dedicated to FCoE traffic only.
- An FCoE VLAN cannot support both VN2VF_Port FIP snooping and VN2VN_Port FIP snooping simultaneously. You must configure separate FCoE VLANs for VN2VF_Port FIP snooping traffic and for VN2VN_Port FIP snooping traffic.

  NOTE: **Changing an FCoE VLAN from VN2VF_Port FIP snooping mode to VN2VN_Port snooping mode terminates the existing virtual links on the VLAN. The transit switch removes the existing FIP snooping filters, creates the new FIP snooping filters, and applies them to the FIP snooping ports. If you downgrade the software to Junos OS Release 12.1 or earlier, VLANs configured for VN2VN_Port FIP snooping revert to VN2VF_Port FIP snooping VLANs.**

- For QFX3500 switches, QFX3600 switches, and QFabric system Node devices, configure all access ports that belong to an FCoE VLAN (ports connected to a converged network adapter [CNA] in an FCoE device) in **tagged-access** port mode. Access ports associated with an FCoE VLAN should not be configured as access ports or trunk ports on these platforms, although trunk port configuration is supported.

  However, on QFX5100 switches that run the Enhanced Layer 2 Software (ELS) CLI, configure access ports that belong to an FCoE VLAN in **trunk** interface mode.

- All ports connected to an FC switch (or FCoE forwarder) must be configured in **trunk** port mode. Ports connected to an FC switch must be configured as trusted ports.

- FIP traffic uses the native VLAN (FIP VLAN discovery and notification frames are exchanged as untagged packets).
- All FCoE VLAN traffic must be tagged and cannot belong to the native VLAN.
- FCoE VLAN traffic cannot be untagged or priority-tagged.

When you enable VN2VF_Port FIP snooping, the switch inspects FIP frames.

The VN2VF_Port FIP snooping implementation includes these considerations:

- ENode-Facing Interfaces on page 84
- Network-Facing Interfaces on page 85
- FC-MAP on page 85

ENode-Facing Interfaces

When the interfaces that belong to an FCoE VLAN connect directly to FCoE devices (there is no other transit switch between the FCoE devices and the QFX Series switch), we recommend that you enable VN2VF_Port FIP snooping on all FCoE VLANs that connect VN_Ports to VF_Ports. Enabling FIP snooping ensures secure connections between server ENodes and FC switches. (Enabling VN2VN_Port FIP snooping ensures secure connections on FCoE VLANs that connect VN_Ports to other VN_Ports). FIP snooping should always be enabled at the access edge.

Systems that run Enhanced Layer 2 Software (ELS) such as the QFX5100 switch support a slightly different configuration on ENode-facing interfaces than systems that do not run ELS. This section describes:

- Non-ELS Port Mode for FCoE Interfaces on page 84
- ELS Interface Mode for FCoE Interfaces on page 85
- Trusted and Untrusted FCoE Interfaces on page 85

Non-ELS Port Mode for FCoE Interfaces

The interfaces that belong to FCoE VLANs (interfaces that connect to CNAs in FCoE devices) on systems that do not support ELS should be configured in **tagged-access** port mode. After you enable VN2VF_Port FIP snooping on an FCoE VLAN, the transit switch denies FCoE traffic from any ENode on that VLAN until the ENode performs a valid fabric login with an FC switch.

The **tagged-access** port mode was not available in Junos OS Release 11.3 and prior releases. In Release 11.3 and earlier, **trunk** port mode was used for Ethernet interfaces that connected to FCoE access devices. Because **tagged-access** mode is now available, using **trunk** mode for interfaces connected to FCoE CNAs is not recommended.

If an existing configuration uses **trunk** mode for ports connected to FCoE CNAs, you can change the port mode to **tagged-access** without disrupting traffic. Although we recommend changing the port mode of these ports from **trunk** to **tagged-access** as a best practice, it is not mandatory. New configurations should use **tagged-access** mode for interfaces that connect to FCoE devices.

ELS Interface Mode for FCoE Interfaces

The interfaces that belong to FCoE VLANs (interfaces that connect to CNAs in FCoE devices) on systems that support ELS should be configured in **trunk** interface mode. After you enable VN2VF_Port FIP snooping on an FCoE VLAN, the transit switch denies FCoE traffic from any ENode on that VLAN until the ENode performs a valid fabric login with an FC switch.

Trusted and Untrusted FCoE Interfaces

Do not configure ENode-facing interfaces as FCoE trusted interfaces when VN2VF_Port FIP snooping is enabled on those interfaces. If you enable VN2VF_Port FIP snooping on an FCoE VLAN and you configure ENode-facing interfaces that are members of the FIP snooping VLAN as **fcoe-trusted**, then FCoE devices might not be able to log in to the FC network.

Changing ports from untrusted to trusted removes any existing VN2VF_Port FIP snooping filters from the ports and terminates the existing sessions. Changing the fabric ports from trusted to untrusted forces all of the FCoE sessions on those ports to log out so that when the ENodes and VN_Ports log in again, the switch can build the appropriate VN2VF_Port FIP snooping filters.

Network-Facing Interfaces

When the switch acts an FCoE transit switch, you must configure any interface that is connected to a switch as an FCoE trusted interface in **trunk** port mode and as a 10-Gigabit Ethernet interface.

Switch-facing Ethernet interfaces have the following requirements and behaviors:

- You must explicitly configure switch-facing trunk ports on an FCoE transit switch as FCoE trusted interfaces.
- After you configure an FC switch-facing trunk port as a trusted interface, the FCoE transit switch always processes FC switch frames because they come from a source on a trusted interface.
- All ports in an FCoE VLAN must be configured as tagged access or trunk ports.

FC-MAP

When the switch acts as an FCoE transit switch and you enable VN2VF_Port FIP snooping on an FCoE VLAN, you can optionally specify a 24-bit FCoE mapped address prefix (FC-MAP) value. On a given VLAN, the transit switch learns only those FC switches that have a matching FC-MAP value. If the transit switch FCoE VLAN FC-MAP value does not match the FC switch FC-MAP value, the transit switch does not discover the FC switch on that VLAN, and the ENodes on that VLAN cannot access the FC switch. An FCoE VLAN can have one and only one FC-MAP value.

The FC-MAP value is a MAC address prefix unique to an FC switch in the FC SAN fabric that the FC switch uses to identify FCoE traffic for a given FC fabric (traffic on a particular FCoE VLAN). The FC switch combines the FC-MAP value with a unique 24-bit FCID value for the ENode VN_Port during the login process. This creates a 48-bit identifier that is unique to the fabric. The FC switch assigns this 48-bit value to the ENode VN_Port as its MAC address and unique identifier for the session. Each VN_Port session the ENode establishes with the FC switch receives a unique FCID from the FC switch, so an FCoE device can host multiple virtual links (one for each VN_Port) to an FC switch, each with a 48-bit MAC address that is unique to the fabric.

The VN2VF_Port FIP snooping filter compares the configured FC-MAP value with the FC-MAP value in the header of frames coming from the ENode VN_Port. If the values do not match, the transit switch denies access.

NOTE: Changing the FC-MAP value causes all logins to be dropped and forces ENodes to log in again.

NOTE: Do not configure static MAC addresses with the FC-MAP value as a prefix (the first 24 bits of the MAC address). If you configure a static MAC address that uses the FC-MAP value as a prefix, the system deletes the static MAC address automatically after you enable FIP snooping. The static MAC address configuration is not restored even if you disable FIP snooping later. (The system considers a static MAC address with the FC-MAP value as the prefix to be a misconfiguration.) Do not use a MAC address with the FC-MAP value as the prefix for any traffic other than the FIP snooping traffic when the QFX Series is acting as a transit switch.

T11 VN2VF_Port FIP Snooping Specification

For more details about VN2VF_Port FIP snooping, see http://www.t11.org/ftp/t11/pub/fc/bb-5/08-264v3.pdf for the Technical Committee T11 organization document *Increasing FCoE Robustness using FIP Snooping*.

Related Documentation

- Overview of Fibre Channel on the QFX Series on page 4
- Understanding DCB Features and Requirements on page 16
- Understanding FCoE Transit Switch Functionality on page 25
- Understanding an FCoE-FC Gateway on page 29
- Overview of FIP on page 9
- Understanding VN_Port to VN_Port FIP Snooping on an FCoE Transit Switch on page 87
- Understanding FCoE LAGs on page 158
- Understanding FIP Snooping, FBF, and MVR Filter Scalability on page 95
- Configuring VN2VF_Port FIP Snooping and FCoE Trusted Interfaces on an FCoE Transit Switch on page 294
- Disabling VN2VF_Port FIP Snooping on an FCoE-FC Gateway Switch Interface on page 299
- Disabling Enhanced FIP Snooping Scaling on page 298

- Configuring VLANs for FCoE Traffic on an FCoE Transit Switch on page 289
- Configuring an FCoE LAG on page 302
- Understanding Fibre Channel Terminology on page 131

Understanding VN_Port to VN_Port FIP Snooping on an FCoE Transit Switch

VN_Port to VN_Port (VN2VN_Port) Fibre Channel over Ethernet (FCoE) Initialization Protocol (FIP) snooping (FC-BB-6) on an FCoE transit switch is conceptually similar to VN_Port to VF_Port (VN2VF_Port) FIP snooping (FC-BB-5) on an FCoE transit switch. An FCoE transit switch is a data center bridging (DCB) switch with FIP snooping capability. VN2VN_Port FIP snooping provides security in the form of filters. The filters help prevent unauthorized access and data transmission on a bridge that connects ENodes on the Ethernet network.

The main difference between VN2VN_Port FIP snooping and VN2VF_Port FIP snooping is that you use VN2VN_Port FIP snooping when the FCoE devices reside on the Ethernet network, so there is no need to forward traffic between FCoE devices to the Fibre Channel (FC) network, and you use VN2VF_Port FIP snooping when FCoE devices on the Ethernet network need to access targets on the FC network, so FCoE traffic must be forwarded to the FC network. See "Understanding VN_Port to VF_Port FIP Snooping on an FCoE Transit Switch" on page 80 for information about VN2VF_Port FIP snooping.

You enable VN2VN_Port FIP snooping on the FCoE VLAN that transports the VN2VN traffic. The transit switch applies VN2VN_Port FIP snooping filters at the ports associated with the FCoE VLANs on which you enable VN2VN FIP snooping.

A key benefit of VN2VN_Port FIP snooping is that it enables FCoE initiators and targets to communicate directly through the switch without going through an FCoE forwarder (FCF) or an FC switch. The transit switch does not differentiate between initiators and targets because the transit switch sees both VN_Ports as FIP virtual link end points. Direct VN2VN_Port communication requires secure access (FIP snooping filters) because ENodes are not trusted entities.

This topic describes:

VN2VN_Port FIP Snooping and FIP Snooping Virtual Links

FIP snooping under the T11 FC-BB-5 specification requires that an FC switch or an FCF be in the path between two VN_Ports when they communicate. Introduced in the T11 FC-BB-6 specification (see http://www.t11.org/ftp/t11/pub/fc/bb-6/10-019v3.pdf), VN2VN_Port FIP snooping allows the FCoE transit switch to connect two VN_Ports to each other directly, without going through an FC switch or an FCF, provided that the ENodes have logged in to the FC network.

In VN2VF_Port FIP snooping, when an ENode logs in to the FC network, the FCoE transit switch snoops the FIP communication between the ENode and the FC switch. In VN2VN_Port FIP snooping mode, the transit switch creates filters on the switch access ports to control VN_Port access to other VN_Ports on the Ethernet network. The VN2VN_Port FIP snooping filters allow the switch to establish a dedicated virtual link that emulates a point-to-point connection between two VN_Ports, through the switch.

Virtual links pass transparently through the transit switch. The VN_Ports do not detect the transit switch, and virtual links appear to be direct point-to-point links.

You explicitly enable VN2VN_Port FIP snooping on FCoE VLANs when the QFX Series switch or QFabric system is an FCoE transit switch connecting FCoE devices on the Ethernet network to each other and to FC switches or gateways at the FC storage area network (SAN) edge.

NOTE: **An FCoE VLAN can support either VN2VF_Port FIP snooping or VN2VN_Port FIP snooping, but not both. Configure separate FCoE VLANs for VN2VF_Port FIP snooping traffic and for VN2VN_Port FIP snooping traffic. On FCoE VLANs that are configured as VN2VN_Port FIP snooping VLANs, VN_Port to VF_Port traffic is dropped.**

When you enable FIP snooping, the system snoops VN2VF_Port packets and enforces security only on VN_Port to VF_Port virtual links. When you enable VN2VN_Port FIP snooping, the system snoops VN_Port to VN_Port FIP packets and enforces security only on VN_Port to VN_Port virtual links.

The transit switch applies VN2VN_Port FIP snooping filters at the ports associated with the FCoE VLANs on which you enable VN2VN_Port FIP snooping. VN2VN_Port FIP snooping provides security for virtual links by creating filters based on information gathered (snooped) about FCoE devices during FIP transactions.

VN2VN_Port Communication Modes

The transit switch supports two VN2VN_Port communication modes:

- Point-to-point mode
- Multipoint mode

In point-to-point mode, two ENodes are connected to the network and form a single VN_Port to VN_Port virtual link. This is analogous to the point-to-point FC link between an FC initiator and an FC target.

In multipoint mode, multiple ENodes are connected to the network and form multiple virtual links. Each virtual link is created between one pair of VN_Ports. This is analogous to loop mode in traditional FC networks.

The VN2VN_Port communication mode is not configured; it is determined by the number of ENodes connected to the network.

Network Security

In traditional FC networks, the FC switch is usually a trusted entity and the server ENodes are untrusted entities. The ENodes connect directly to the FC switch VF_Ports. After an ENode gains access to the network through the fabric login (FLOGI) process, the FC switch enforces zoning configurations, ensures that the ENode uses valid addresses, monitors the connection, and performs other security functions to prevent unauthorized access.

However, FCoE exposes FC frames to Ethernet networks, which do not have the same level of security as native FC networks. VN2VN_Port FIP snooping filters emulate the native FC network security functions by preventing unauthorized access and by ensuring the security of the virtual link between ENode VN_Ports. The transit switch performs VN2VN_Port FIP snooping at the ports connected to the FCoE VN_Port devices.

VN2VN_Port FIP Snooping Functions

When you enable VN2VN_Port FIP snooping, the transit switch sets and applies filters to block all FCoE traffic on the VLAN by default. The transit switch monitors FIP logins, solicitations, and advertisements that pass through it and gathers information about the ENode address. The transit switch uses the information to construct filters that permit access only to logged-in ENodes. All other traffic on the VLAN is denied.

The filters enable FCoE frames to pass through the transit switch only on a virtual link established between two VN_Ports. The filters ensure that ENodes can only connect to other ENodes if they have successfully logged in to each other, and that only valid FCoE traffic along valid paths is transmitted. VN2VN_Port FIP snooping maintains the filters by tracking VN_Port to VN_Port sessions.

Scalability

Because ENodes are untrusted and the system needs to apply filters to untrusted FIP snooping interfaces, the total number of combined VN2VN_Port FIP snooping sessions per switch is 376 sessions (ENode to ENode sessions) on untrusted interfaces. On interfaces that are configured as trusted interfaces, no FIP snooping filters are applied.

NOTE: **The total number of sessions the system can support is the combined number of VN2VF_Port sessions and VN2VN_Port sessions. If VN2VF_Port sessions are active, the total number of available VN2VN_Port sessions is reduced.**

VN2VN_Port FIP Snooping Implementation

You enable VN2VN_Port FIP snooping on a per-VLAN basis on VLANs that carry FCoE traffic. The switch snoops FIP frames at the ports associated with FCoE VLANs enabled for VN2VN_Port FIP snooping. The switch then installs the resulting filters on the ENode-facing ports to ensure that all FIP snooping occurs on the switch network edge.

VN2VN_Port FIP snooping FCoE VLANs must meet the following criteria:

- An FCoE VLAN should be dedicated to FCoE traffic only.
- An FCoE VLAN cannot support both VN2VF_Port FIP snooping (FC-BB-5) and VN2VN_Port FIP snooping (FC-BB-6) simultaneously. You must configure separate FCoE VLANs for FIP snooping traffic and for VN2VN_Port FIP snooping traffic.

> NOTE: Changing an FCoE VLAN from VN2VF_Port FIP snooping mode to VN2VN_Port FIP snooping mode terminates the existing virtual links on the VLAN. The transit switch removes the existing FIP snooping filters, creates the new FIP snooping filters, and applies them to the FIP snooping ports. If you downgrade the software to Junos OS Release 12.1 or earlier, VLANs configured for VN2VN_Port FIP snooping revert to VN2VF_Port FIP snooping VLANs.

- For QFX3500 switches, QFX3600 switches, and QFabric system Node devices, as a best practice, you should configure all access ports that belong to an FCoE VLAN (ports connected to a converged network adapter [CNA] in an FCoE device) in **tagged-access** port mode. However, access and trunk port modes are also supported. For QFX5100 switches, configure access ports that belong to an FCoE VLAN in **trunk** interface mode.
- Access ports should be configured as untrusted ports.
- All ports connected to another transit switch must be configured in **trunk** port mode.
- FIP traffic uses the native VLAN.
- You can enable VN2VN_Port FIP snooping on a native VLAN.

ENode-Facing Interfaces

When the interfaces that belong to an FCoE VLAN connect directly to FCoE devices (there is no other transit switch between the FCoE devices and the QFX Series switch), we recommend that you either enable VN2VN_Port FIP snooping on all FCoE VLANs to ensure secure connections between VN_Ports, or enable VN2VF_Port FIP snooping on FCoE VLANs that connect ENodes to an FC switch. FIP snooping should always be enabled at the access edge.

Systems that run Enhanced Layer 2 Software (ELS) such as the QFX5100 switch support a slightly different configuration on ENode-facing interfaces than systems that do not run ELS. This section describes:

- Non-ELS Port Mode for FCoE Interfaces on page 91
- ELS Interface Mode for FCoE Interfaces on page 91
- Trusted and Untrusted FCoE Interfaces on page 91

Non-ELS Port Mode for FCoE Interfaces

The interfaces that belong to FCoE VLANs (interfaces that connect to CNAs in FCoE devices) should be configured in **tagged-access** port mode, unless your CNA does not support tagged VN2VN traffic. After you enable VN2VN_Port FIP snooping on an FCoE VLAN, the transit switch denies FCoE traffic from any ENode on that VLAN until the ENode performs a valid fabric login (FIP FLOGI) with another ENode.

The **tagged-access** port mode was not available in Junos OS Release 11.3 and prior releases. In Release 11.3 and earlier, **trunk** port mode was used for Ethernet interfaces that connected to FCoE access devices. Because **tagged-access** mode is now available, using **trunk** mode for interfaces connected to FCoE CNAs is not recommended.

If an existing configuration uses **trunk** mode for ports connected to FCoE CNAs, you can change the port mode to **tagged-access** without disrupting traffic. Although we recommend changing the port mode of these ports from **trunk** to **tagged-access** as a best practice, it is not mandatory. New configurations should use **tagged-access** mode for interfaces that connect to FCoE devices.

ELS Interface Mode for FCoE Interfaces

The interfaces that belong to FCoE VLANs (interfaces that connect to CNAs in FCoE devices) on systems that support ELS should be configured in **trunk** interface mode. After you enable VN2VF_Port FIP snooping on an FCoE VLAN, the transit switch denies FCoE traffic from any ENode on that VLAN until the ENode performs a valid fabric login with an FC switch.

Trusted and Untrusted FCoE Interfaces

Do not configure ENode-facing interfaces as FCoE trusted interfaces when VN2VF_Port FIP snooping is enabled on those interfaces. If you enable VN2VF_Port FIP snooping on an FCoE VLAN and you configure ENode-facing interfaces that are members of the FIP snooping VLAN as **fcoe-trusted**, then FCoE devices might not be able to log in to the FC network.

Changing ports from untrusted to trusted removes any existing VN2VF_Port FIP snooping filters from the ports and terminates the existing sessions. Changing the fabric ports from trusted to untrusted forces all of the FCoE sessions on those ports to log out so that when the ENodes and VN_Ports log in again, the switch can build the appropriate VN2VF_Port FIP snooping filters.

Network-Facing Interfaces (Connecting to Another Transit Switch)

Configure any interface that is connected to another transit switch (not to an ENode) as an FCoE trusted interface, in **trunk** port mode, and as a 10-Gigabit Ethernet interface.

Network-facing Ethernet interfaces have the following requirements and behaviors:

- You must explicitly configure network-facing trunk ports on an FCoE transit switch as FCoE trusted interfaces.
- After you configure a network-facing trunk port as a trusted interface, the FCoE transit switch always processes frames from the connected switch because they come from a source on a trusted interface.
- As a best practice, configure ports in an FCoE VLAN as tagged access ports, but access and trunk port modes are also supported to accommodate whatever types of VN2VN traffic your CNA supports.

Beacon Period (VN2VN_Port FIP Snooping Link Maintenance)

The transit switch needs to maintain the virtual links between VN_Ports, and needs to know when sessions begin and end, and when to install and remove the FIP snooping filters. FIP snooping uses a FIP keepalive advertisement to accomplish this task. VN2VN_Port FIP snooping does not exchange FIP keepalive timer information. Instead, you configure a *beacon period*, which performs the same function as a keepalive timer.

The beacon period is the time interval between messages which verify that the connection is still valid and that the device at the other end of the virtual link is still reachable. You set the beacon period value for each FCoE VLAN that you configure to do VN2VN_Port FIP snooping.

NOTE: Explicitly set the beacon period when you configure VN2VN_Port FIP snooping. VN_Ports do not automatically send beacons.

ENodes transmit periodic multicast N_Port_ID beacons to the ALL-VN2VN-ENode-MACs address. The transmission period varies by a random delay of between 0 ms and 100 ms to avoid synchronized bursts of multicast traffic on the network.

If the transit switch does not receive a beacon message from an ENode within 2.5 times the configured beacon period, the transit switch considers the virtual link to be down and terminates the virtual link to that ENode.

QFabric System Differences in VN2VN_Port FIP Snooping Traffic Handling

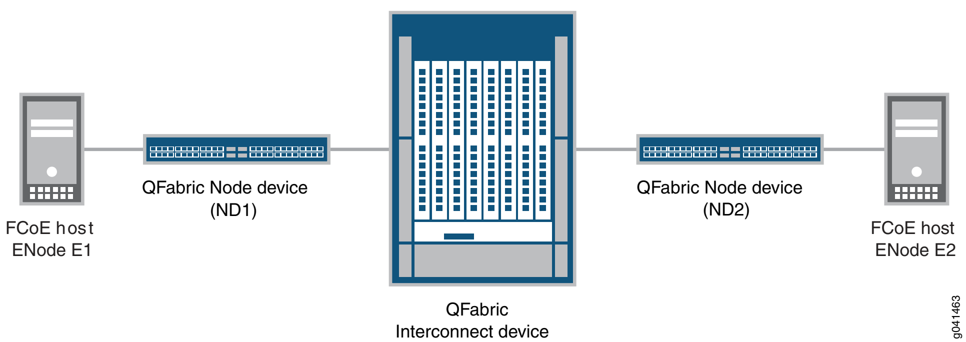
Configuring VN2VN_Port FIP snooping on a QFabric system is the same as configuring VN2VN_Port FIP snooping on the QFX Series. However, there are internal differences in the way a QFabric system handles VN2VN_Port FIP snooping traffic compared to the way a QFX Series switch handles VN2VN_Port FIP snooping traffic. The internal differences are transparent. Whether you configure VN2VN_Port FIP snooping on a QFabric system or on the QFX Series, the proper FIP snooping filters and forwarding information are installed on each device.

On the QFX Series, the VN2VN_Port FIP snooping traffic does not cross a fabric (Interconnect device). VN2VN_Port traffic enters and exits ports on a single switch, so the ingress port and the egress port have access to the same *local* forwarding and FIP snooping databases.

However, on a QFabric system, VN2VN_Port FIP snooping traffic might enter on the ingress port of one Node device, traverse the Interconnect device fabric, and exit on the egress port of a different Node device. In this case, the QFabric system must ensure that the FIP snooping database and forwarding information for the VN2VN_Port traffic is installed correctly on both of the Node devices so that traffic is correctly filtered and forwarded.

For example, Figure 8 on page 93 shows that VN2VN_Port traffic from FCoE host ENode E1 enters the QFabric system at Node device ND1, traverses the Interconnect device fabric, and then exits from Node device ND2 before arriving at FCoE host ENode E2. Similarly, VN2VN_Port traffic from FCoE host ENode E2 enters the QFabric system at Node device ND2, traverses the Interconnect device fabric, and then exits from Node device ND1 before arriving at FCoE host ENode E1.

Figure 8: VN2VN_Port Traffic Across a QFabric Interconnect Device



When the QFabric system receives a FLOGI ACC from either ENode E1 or ENode E2, the QFabric system creates and installs the correct VN2VN_Port FIP snooping filters on both Node devices, and updates the forwarding tables accordingly.

In addition, the QFabric system must also ensure that the VN2VN_Port FIP snooping session statistics are correctly counted. Even though a session is running on each of the two Node devices, the QFabric system counts the complete VN2VN_Port connection as one session because the two Node devices belong to the same session. This ensures that VN2VN_Port sessions that traverse the Interconnect device fabric are counted as one unique session, not as two separate sessions.

Related Documentation

- Understanding DCB Features and Requirements on page 16
- Understanding FCoE Transit Switch Functionality on page 25
- Understanding VN_Port to VF_Port FIP Snooping on an FCoE Transit Switch on page 80
- Overview of FIP on page 9

- Understanding Fibre Channel Terminology on page 131
- Understanding FIP Snooping, FBF, and MVR Filter Scalability on page 95
- Configuring VLANs for FCoE Traffic on an FCoE Transit Switch on page 289
- Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Directly Connected to the Same FCoE Transit Switch) on page 228
- Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Directly Connected to Different FCoE Transit Switches) on page 233
- Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Indirectly Connected Through an Aggregation Layer FCoE Transit Switch) on page 241
- Enabling VN2VN_Port FIP Snooping and Configuring the Beacon Period on an FCoE Transit Switch on page 300

Understanding FIP Snooping, FBF, and MVR Filter Scalability

The VLAN filter processor (VFP) ternary content addressable memory (TCAM) stores the VLAN filter configuration for three filter types:

- Fibre Channel over Ethernet (FCoE) Initialization Protocol (FIP) snooping—FIP snooping filters prevent an FCoE device from gaining unauthorized access to a Fibre Channel (FC) storage device or to another FCoE device. VN2VF_Port FIP snooping filters prevent an FCoE device from gaining unauthorized access to devices on an FC network. VN2VN_Port FIP snooping filters prevent an FCoE device from gaining unauthorized access to another FCoE device directly through the QFX Series or QFabric system, without traversing the FC network.

  The VFP TCAM stores the VN2VF_Port and VN2VN_Port FIP snooping filters that the switch automatically creates when you enable FIP snooping on a VLAN that carries FCoE traffic. See “Understanding VN_Port to VF_Port FIP Snooping on an FCoE Transit Switch” on page 80 and “Understanding VN_Port to VN_Port FIP Snooping on an FCoE Transit Switch” on page 87 for more information.
- Filter-based forwarding (FBF)—FBF enables you to use firewall filters to direct packets to virtual routing instances. The switch then forwards the matching packets based on the configuration of the routing instances. The VFP TCAM stores the terms you configure for FBF filters. See *Understanding Filter-Based Forwarding* for more information.
- Multicast VLAN registration (MVR)—MVR enables you to configure a multicast source VLAN (MVLAN) that is shared across a Layer 2 network. An MVLAN distributes IPTV multicast streams across different VLANs without having to create a separate multicast stream for each VLAN, and without compromising the security and separation of traffic in the different VLANs. The VFP TCAM stores the MVR rules you configure for MVLANs. See *Understanding Multicast VLAN Registration* for more information.

FIP snooping filters, FBF filters, and MVR rules share the VFP TCAM memory space. In most use cases, the VFP TCAM memory is sufficient to store filter terms and information for all three applications.

VFP TCAM Architecture and Allocation

When packets arrive at an ingress interface, the VFP TCAM is the first TCAM in the packet pipeline. The VFP TCAM stores a total of 1024 entries. The 1024 entries are partitioned into four equal *slices* of 256 entries.

The VFP TCAM allocates entries to three filter types (FIP snooping filters, FBF filter terms, and MVR rules) in 256-entry slices. The VFP TCAM dynamically allocates the minimum number of memory slices required to store the filters for a particular filter type, as needed.

The TCAM does not allocate partial slices to a filter type, and slices cannot be shared among filter types. At any given time, each slice contains entries for one and only one filter type.

For example, if you configure one MVR rule, the system allocates a whole slice to MVR rules, even if the MVR rule consumes only one TCAM entry. The remaining 256 entries in the slice allocated to MVR rules can store subsequently configured MVR rules, but not FIP snooping or FBF filters. Similarly, if FIP snooping filters consume 50 entries of a 256-entry slice, the remaining 206 entries in the FIP snooping slice are available only to store more FIP snooping filters, not to store FBF filter terms or MVR rules.

The VFP TCAM allocates slices to a filter type only if there is at least one configured filter or rule for that filter type. If no filters exist for a filter type, then the VFP TCAM does not allocate a slice to that filter type.

NOTE: **The VFP TCAM rejects partial filters. For example, if an FBF filter contains six terms, but there is only space in the TCAM for four of those terms, the whole filter is not committed.**

Each filter type can use from zero slices to all four slices of VFP TCAM space. However, if one filter type uses three slices, then only one slice remains, so only one other filter type can use the remaining slice. In that situation, if you configure filters for all three filter types, the last filter type that you configure receives no TCAM space for its filter entries. Filters that receive no TCAM entry space are not implemented.

VFP TCAM Entry Consumption

FIP snooping filters, FBF filters, and MVR rules consume VFP TCAM entry space in different ways:

- FIP Snooping Filter VFP TCAM Consumption on page 96
- FBF Filter VFP TCAM Consumption on page 97
- MVR Filter VFP TCAM Consumption on page 98
- VFP TCAM Consumption Summary Table on page 98

FIP Snooping Filter VFP TCAM Consumption

VN2VF_Port FIP snooping filters consume VFP TCAM entry space differently than VN2VN_Port FIP snooping filters:

- VN2VF_Port FIP Snooping Filter VFP TCAM Consumption on page 97
- VN2VN_Port FIP Snooping Filter VFP TCAM Consumption on page 97

NOTE: **One FCoE VLAN cannot support both VN2VF_Port traffic and VN2VN_Port traffic. Configure separate FCoE VLANs for VN2VF_Port traffic and for VN2VN_Port traffic.**

VN2VF_Port FIP Snooping Filter VFP TCAM Consumption

The switch uses an algorithm that allows one 256-entry slice of the VFP TCAM to store the maximum possible number of VN2VF_Port FIP snooping filters (2500 filters). VN2VF_Port FIP snooping filters never consume more than one slice of the VFP TCAM.

Regardless of whether there is one VN2VF_Port FIP snooping session or there are 2500 VN2VF_Port FIP snooping sessions, VN2VF_Port FIP snooping filters consume one slice of the VFP TCAM. (If there are no VN2VF_Port or VN2VN_Port FIP snooping sessions, the TCAM does not allocate a slice for FIP snooping filters.)

VN2VN_Port FIP Snooping Filter VFP TCAM Consumption

VN2VN_Port FIP snooping filters consume one VFP TCAM entry for each VN2VN_Port session. The maximum number of VN2VN_Port FIP snooping sessions is 376 sessions per switch. (If you configure an interface that carries VN2VN_Port FIP snooping traffic as a trusted interface, the switch does not apply filters on the trusted interface.)

Because the switch can have up to 376 VN2VN_Port sessions running simultaneously, with each session consuming one entry, VN2VN_Port FIP snooping filters consume VFP TCAM space as follows:

- 1–256 filters consume one slice
- 257–376 filters consume two slices

FBF Filter VFP TCAM Consumption

Each FBF filter term is double-wide, so each FBF filter term consumes two entries in the VFP TCAM. One 256-entry slice can contain up to 128 FBF filter terms. FBF filters consume VFP TCAM space as follows:

- 1–128 entries consume one slice
- 129–256 entries consume two slices
- 257–384 entries consume three slices
- 385–512 entries consume four slices

NOTE: **In practice, FBF filters can consume only three slices of the VFP TCAM because FBF filters are also stored simultaneously in the ingress filter processor (IFP) TCAM, and the IFP TCAM can store only 384 FBF filter terms (768 entries, or 3 TCAM slices).**

For example, if you configure FBF filters that contain 200 terms, then the FBF filters require 400 VFP TCAM entries and consume 2 slices.

FBF filter entries are simultaneously stored in the VFP TCAM and the IFP TCAM. The IFP TCAM can only contain up to 768 entries—256 fewer entries (1 slice) than the VFP TCAM. As with the VFP TCAM, FBF filters consume two IFP TCAM entries per filter term. In addition to FBF filter terms, the IFP TCAM stores filter entries for firewall filters.

CAUTION: There must be enough space in the VFP TCAM *and* the IFP TCAM for the FBF filter entries. If both TCAMs do not have enough space for the FBF filters, the switch rejects the portion of the configuration that it cannot store and sends a syslog message to notify you.

For example, if you configure FBF filters that have 400 terms, even though the VFP TCAM has enough space to store the resulting 800 entries, the switch rejects a portion of the configuration because the IFP TCAM can store a maximum of only 768 entries. If the IFP TCAM stores no other filter entries, the switch rejects 32 FBF filter entries.

In another example, if you configure firewall filters that have a total of 200 terms, which consume 200 entries in the IFP TCAM, and you then configure FBF filters that have a total of 300 terms, the switch rejects a portion of the configuration because the FBF filters require 600 entries. Combined with the 200 entries required for the firewall filters, the total number of 800 entries exceeds the maximum of 768 entries that the IFP TCAM can store. In this case, the switch accepts the first 768 entries and rejects the rest of the filter entries. The switch installs the filter entries in the order that they are committed; the rejected entries are the last entries the switch attempts to commit after the TCAM space is exhausted.

The IFP TCAM limit of 768 entries means that the true maximum number of FBF filter terms is 384 terms, even though the VFP TCAM can store up to 512 FBF terms.

MVR Filter VFP TCAM Consumption

Each MVR rule consumes one entry in the VFP TCAM, so MVR rules consume VFP TCAM space as follows:

- 1–256 rules consume one slice
- 257–512 rules consume two slices
- 513–758 rules consume three slices
- 759–1024 rules consume four slices

VFP TCAM Consumption Summary Table

Table 8 on page 99 summarizes VFP TCAM consumption.

NOTE: FBF filters are simultaneously stored in the VFP TCAM and in the IFP TCAM. Due to the IFP TCAM limit of 768 entries (384 FBF filters), which is 256 entries fewer than the VFP TCAM, the effective VFP TCAM consumption limit for FBF filters is lower than the total amount of VFP TCAM entry space, even when no other filters consume VFP TCAM space.

Table 8: VFP TCAM Entry Consumption Summary

Filter Type	VPF TCAM Entry Consumption	Maximum VFP TCAM Slices Consumed	Other Limitations
VN2VF_Port FIP snooping filters	Never consumes more than one slice	One slice (regardless of number of sessions)	2500 session maximum
VN2VN_Port FIP snooping filters	One entry per session	Two	376 session maximum
FBF filters	Two entries per filter	Three (due to IFP TCAM limitation)	384 filters (due to IFP TCAM limitation)
MVR rules	One entry per rule	Four	1024 rule maximum

Rejected Filter Configurations (No Available VFP TCAM Space)

If there is not enough space available in the VFP TCAM to store the FIP snooping filters, the configured FBF filters, and the MVR rules, the switch rejects only the portion of the configuration that it cannot store. Any portion of the filter configuration that the TCAM can store, is stored. In most cases, even if the switch rejects part of the configuration, part of the configuration is also stored.

If the switch rejects any portion of a configuration, the switch sends a syslog message to notify you of the failure. The switch does not generate a commit error, and the rejected portion of the configuration remains on the switch, even though the rejected configuration does not function. (The accepted portions of the configuration function as expected.) The syslog message shows you the filter configuration that the switch rejected.

We strongly recommend that you always delete rejected filter configurations from the switch. It is important to delete rejected filter configurations because:

- Even though the rejected configuration remains on the switch, it does not function.
- After a reboot, there is no guarantee that the same filters will be rejected. The previously rejected filters might be accepted, and other filters that had previously been accepted might be rejected. Therefore, the functioning filter configuration could be changed inadvertently and unexpectedly.
- Even if a VFP TCAM slice becomes available, the switch does not automatically allocate the available slice to the rejected configuration. To use the available slice, you must delete and reconfigure the rejected configuration.

  For example, you configure FBF filters and MVR rules on a switch, and that switch also transports FCoE traffic with VN2VF_Port FIP snooping (never consumes more than one slice) enabled on FCoE access interfaces. After you commit the configuration, you check the syslog. You find that the VN2VF_Port FIP snooping and FBF filters consume all four slices of the VFP TCAM, and the MVR configuration was rejected. Instead of deleting the MVR configuration, you leave it on the switch. Subsequently, all VN2VF_Port FIP snooping sessions end, the FIP snooping filters time out and are removed from the VFP TCAM, so the slice that was allocated to VN2VF_Port FIP snooping filters becomes free. However, the MVR rules do *not* automatically receive the free slice.

To force the switch to allocate the free slice to the MVR rules, you should delete the MVR rules from the configuration and then reconfigure the MVR rules. When you commit the new configuration, check the syslog messages to ensure that the MVR rule configuration was accepted.

In this example, you could also choose to free a VFP TCAM slice for MVR rule storage by deleting some of the FBF filters. To do this, you delete both the unneeded FBF filters and the MVR rule configuration. Then you reconfigure the MVR rules, and check the syslog to ensure that the configuration was successful.

VFP TCAM Allocation and Consumption (Scaling) Examples

The following examples illustrate how FIP snooping entries, FBF filter entries, and MVR rule entries consume VFP TCAM slices:

Example 1: Three Filter Types Consume Three Slices

Filters and rules are configured in the following sequence:

- 100 VN2VN_Port FIP snooping filters (1 slice)
- 2 MVR rules (1 slice, 2 entries)
- 60 FBF filter terms (1 slice, 120 entries)

One slice remains free. The slice allocated to VN2VN_Port FIP snooping filters can store 156 more filters before another slice is required. The slice allocated to MVR rules can store 254 more rules before another slice is required. The slice allocated to FBF filters can store 68 more filter terms (136 entries) before another slice is required. Providing that the IFP TCAM has space for the FBF filter terms, the switch accepts this configuration and rejects no filters.

Example 2: Three Filter Types Consume Four Slices

Filters and rules are configured in the following sequence:

- 2000 VN2VF_Port FIP snooping filters (always 1 slice)
- 18 MVR rules (1 slice, 18 entries)
- 150 FBF filter terms (2 slices, 300 entries)

All four slices are allocated to filter types. The slice allocated to MVR rules can store 238 more rules before it is full. The slice allocated to FBF filters can store 106 more filter terms (212 entries) before it is full. Providing that the IFP TCAM has space for the FBF filter terms, the switch accepts this configuration and rejects no filters.

NOTE: If you configure more MVR rules or FBF filters than entry space remaining in the slices, the switch rejects those rules and filters because no slice is available. The switch installs filters in the order that they were configured, so if filters are rejected, the filters configured last are rejected.

Example 3: Two Filter Types Consume Four Slices

Filters and rules are configured in the following sequence:

- 50 VN2VF_Port FIP snooping filters (always 1 slice)
- 300 FBF filter terms (3 slices, 600 entries)

All four slices are allocated to filter types. No slices are available for MVR rules. The third slice allocated to FBF filters can store 84 more filter terms (168 entries) before it consumes all of its entry space. Providing that the IFP TCAM has space for the FBF filter terms, the switch accepts this configuration and rejects no filters.

NOTE: If you configure MVR rules or if you configure more than 84 more FBF filters, the switch rejects those rules and filters because no slice is available for the MVR rules, and the FBF filter slice has entry space for only 84 more filter terms.

Example 4: Three Filter Types Oversubscribe the VFP TCAM

Filters and rules are configured in the following sequence:

- 1750 VN2VF_Port FIP snooping filters (always 1 slice)
- 10 MVR rules (1 slice, 10 entries)
- 275 FBF filter terms (2 slices, 512 accepted entries, 38 rejected entries)

All four slices are allocated to filter types. The slice allocated to MVR rules can store 246 more rules before it is full, but the number of FBF filter terms exceeds the amount of available VFP TCAM storage space. (The 275 FBF filter terms consume 550 VFP TCAM entries. However, there are only two available slices, for a total of 512 available entry spaces, so only 256 FBF filter terms can be stored, leaving 19 rejected FBF filter terms.)

The switch accepts the VN2VF_Port FIP snooping filters, the MVR rules, and 256 FBF filter terms. The switch retains the excess FBF filters in the configuration, but does not install those filters in the VFP TCAM. In this case, you delete the rejected FBF filter terms from the configuration. Alternatively, you could delete the MVR rules from the configuration to free a slice of the TCAM, and then delete and reconfigure the rejected FBF filters so that the system allocates the freed slice to the FBF filters.

NOTE: The sequence of configuration makes a difference; if there is not enough VFP TCAM space for a given filter type, the switch installs the filters that fit in the order they are configured. For example, if you configure the FBF filters before you configure the MVR rules, the VFP TCAM allocates one slice to FIP snooping filters, three slices to FBF filters (assuming the IFP TCAM has available space), and no slices to MVR rules, because all four slices are allocated before the switch attempts to install the MVR rules in the VFP TCAM.

Filter Configuration Recommendations

To utilize the VFP TCAM space most efficiently:

- Configure and Maintain the Fewest Number of Filters Needed on page 102
- Always Delete Rejected Filter Configurations on page 103

Configure and Maintain the Fewest Number of Filters Needed

To conserve VFP TCAM entry space, and because FBF filter storage also depends on the availability of IFP TCAM space, we recommend that you configure as few FBF filters and MVR rules as is practical to serve your network needs. The more filters you configure, the greater the possibility of exceeding TCAM storage capacity.

Several factors determine VFP TCAM consumption:

- Type of filters configured—Different filter types consume different amounts of VFP TCAM space. VN2VF_Port FIP snooping filters never consume more than one slice. MVR rules and VN2VN_Port FIP snooping filters consume entries in a slice at a rate of one entry per MVR rule or VN2VN_Port session. FBF filter terms consume entries in a slice at a rate of two entries per FBF filter term.
- Number of filters configured—Although the number of filters does not affect the number of slices allocated to the VN2VF_Port FIP snooping filter type (it is always one slice for one or more VN2VF_Port FIP snooping filters and no slice for no FIP snooping filters), the number of VN2VN_Port FIP snooping filters, MVR rules, and FBF filter terms that you configure determine how many VFP TCAM slices are required for each filter type.

  For example, if you configure 257 MVR rules, the MVR rule entries consume 2 slices. One slice stores 256 MVR rules (entries), and one slice stores 1 MVR rule (entry). In this case, if you can eliminate one MVR rule, you can free a slice to allocate to other filter types.
- Sequence of filter configuration—If you configure too many filters for the VFP TCAM to store, the last filters you configure are not stored in the TCAM.

  Always check the syslog after you configure FBF filters or MVR rules to ensure that the configuration was not rejected. If you enable FIP snooping on access ports, check the syslog to ensure that the configuration was not rejected due to lack of VFP TCAM space.

If you check the syslog and a filter configuration has been rejected, delete the filters that were rejected from the configuration.

TIP: If you no longer need an FBF filter or an MVR rule, delete it from the configuration to conserve VFP TCAM space. Enable VN2VF_Port or VN2VN_Port FIP snooping on access ports only if the switch port is directly connected to FCoE devices. (FIP snooping should be performed at the access edge. FIP snooping should not be performed on traffic that has already been snooped and filtered at the access edge. If another switch between the QFX Series or QFabric system and the FCoE devices already performs FIP snooping, do not enable FIP snooping on the QFX Series or QFabric system.)

Always Delete Rejected Filter Configurations

The switch does not return a commit error if it rejects any portion of a configuration. Instead, the switch sends a syslog message to report the rejected portion of the configuration. The rejected portion of the configuration remains on the switch, but does not function.

After you configure FBF filters or MVR rules, or enable FIP snooping, check the syslog messages to ensure that the switch accepted the configuration. If the switch rejected any portion of the configuration, delete that portion of the configuration. (You do not need to delete the portion of the configuration that was accepted, unless you want to reconfigure those filters or rules.)

CAUTION: If you do not delete rejected filter configurations, and if you reboot the system, you cannot predict which filters the system installs after the reboot. For example, a switch with the following configuration has more configured filters than the VFP TCAM can support:

- VN2VF_Port FIP snooping sessions (always consumes one slice)
- 20 MVR rules (consume one slice)
- 300 FBF filters (attempt to consume three slices, but because only two slices are available, 256 filters consume two slices, and the remaining 44 filters are rejected)

If you do not delete the 44 rejected FBF filters, then if the switch reboots, the 44 FBF filters that were rejected might be accepted, and 44 different FBF filters might be rejected. This unpredictable behavior is the reason that you should check the syslog messages after you configure filters, and if any filters were rejected, you should always delete the rejected filters from the configuration.

Related Documentation

- Understanding VN_Port to VF_Port FIP Snooping on an FCoE Transit Switch on page 80
- Understanding VN_Port to VN_Port FIP Snooping on an FCoE Transit Switch on page 87

- *Understanding Filter-Based Forwarding*
- *Understanding Multicast VLAN Registration*
- Configuring VN2VF_Port FIP Snooping and FCoE Trusted Interfaces on an FCoE Transit Switch on page 294
- Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Directly Connected to the Same FCoE Transit Switch) on page 228
- Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Directly Connected to Different FCoE Transit Switches) on page 233
- Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Indirectly Connected Through an Aggregation Layer FCoE Transit Switch) on page 241
- *Example: Using Filter-Based Forwarding to Route Application Traffic to a Security Device*
- *Configuring Multicast VLAN Registration (CLI Procedure)*

Understanding MC-LAGs on an FCoE Transit Switch

Multichassis link aggregation groups (MC-LAGs) provide redundancy and load balancing between two QFX Series switches, multihoming support for client devices such as servers, and a loop-free Layer 2 network without running Spanning Tree Protocol (STP).

You can use an MC-LAG to provide a redundant aggregation layer for Fibre Channel over Ethernet (FCoE) traffic. To support lossless transport of FCoE traffic across an MC-LAG, you must configure the appropriate class of service (CoS) on both of the QFX Series switches with MC-LAG port members. The CoS configuration must be the same on both of the MC-LAG switches because MC-LAGs do not carry forwarding class and IEEE 802.1p priority information.

Ports that are part of an FCoE-FC gateway configuration (a virtual FCoE-FC gateway fabric) do not support MC-LAGs. Ports that are members of an MC-LAG act as passthrough transit switch ports.

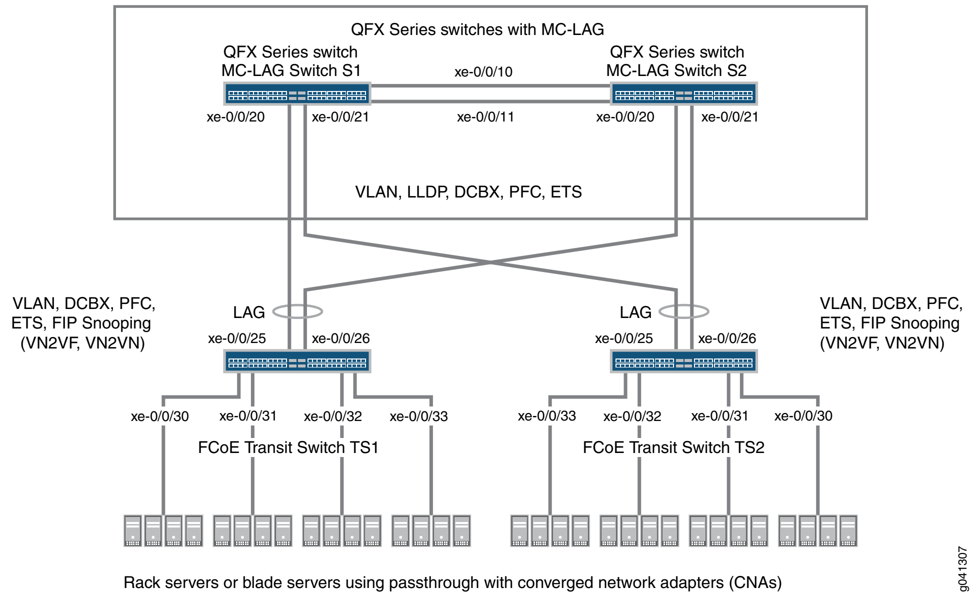
QFX Series switches support MC-LAGs. QFabric system Node devices do not support MC-LAGs, and QFX3500 and QFX3600 Virtual Chassis switches do not support FCoE.

This topic describes:

Supported Topology

QFX Series switches that are not directly connected to FCoE hosts and that act as passthrough transit switches support MC-LAGs for FCoE traffic in an *inverted-U* network topology. Figure 9 on page 105 shows an inverted-U topology using QFX3500 switches.

Figure 9: Supported Topology for an MC-LAG on an FCoE Transit Switch



The following rules and guidelines apply to MC-LAGs when used for FCoE traffic. The rules and guidelines help ensure the proper handling and lossless transport characteristics required for FCoE traffic:

- The two QFX Series switches that form the MC-LAG (Switches S1 and S2) cannot use ports that are part of an FCoE-FC gateway fabric. The MC-LAG switch ports must be passthrough transit switch ports (used as part of an intermediate transit switch that is not directly connected to FCoE hosts).
- MC-LAG Switches S1 and S2 cannot be not directly connected to the FCoE hosts.
- The two QFX Series switches that serve as access devices for FCoE hosts (FCoE Transit Switches TS1 and TS2) use standard LAGs to connect to MC-LAG Switches S1 and S2. FCoE Transit Switches TS1 and TS2 can be standalone QFX Series switches or they can be Node devices in a QFabric system.
- Transit Switches TS1 and TS2 must use transit switch ports for the FCoE hosts and for the standard LAGs to MC-LAG Switches S1 and S2.
- Enable FIP snooping on the FCoE VLAN on Transit Switches TS1 and TS2. You can configure either VN_Port to VF_Port (VN2VF_Port) FIP snooping or VN_Port to VN_Port (VN2VN_Port) FIP snooping, depending on whether the FCoE hosts need to access targets in the FC SAN (VN2VF_Port FIP snooping) or targets in the Ethernet network (VN2VN_Port FIP snooping).

  FIP snooping should be performed at the access edge and is not supported on MC-LAG switches. Do not enable FIP snooping on MC-LAG Switches S1 and S2. (Do not enable FIP snooping on the MC-LAG ports that connect Switches S1 and S2 to Switches TS1 and TS2 or on the LAG ports that connect Switch S1 to S2.)
- The CoS configuration must be consistent on the MC-LAG switches. Because MC-LAGs carry no forwarding class or priority information, each MC-LAG switch needs to have

the same CoS configuration to support lossless transport. (On each MC-LAG switch, the name, egress queue, and CoS provisioning of each forwarding class must be the same, and the priority-based flow control (PFC) configuration must be the same.)

Transit Switches (Server Access)

The role of FCoE Transit Switches TS1 and TS2 is to connect FCoE hosts in a multihomed fashion to the MC-LAG switches. In essence, Transit Switches TS1 and TS2 act as access switches for the FCoE hosts. (FCoE hosts are directly connected to Transit Switches TS1 and TS2.)

The transit switch configuration depends on whether you want to do VN2VF_Port FIP snooping or VN2VN_Port FIP snooping, and whether the QFX Series transit switches also have ports configured as part of an FCoE-FC gateway virtual fabric. Ports that the QFX Series switch uses in an FCoE-FC gateway virtual fabric cannot be included in the transit switch LAG connection to the MC-LAG switches. (Ports cannot belong to both a transit switch and an FCoE-FC gateway; you must use different ports for each mode of operation.)

MC-LAG Switches (FCoE Aggregation)

The role of MC-LAG Switches S1 and S2 is to provide redundant, load-balanced connections between FCoE transit switches. In essence, MC-LAG Switches S1 and S2 act as aggregation switches. FCoE hosts are not directly connected to the MC-LAG switches.

The MC-LAG switch configuration is the same regardless of which type of FIP snooping that FCoE Transit Switches TS1 and TS2 perform.

FIP Snooping and FCoE Trusted Ports

To maintain secure access, enable VN2VF_Port FIP snooping or VN2VN_Port FIP snooping at the transit switch access ports connected directly to the FCoE hosts. FIP snooping should be performed at the access edge of the network to prevent unauthorized access. For example, in Figure 9 on page 105, you enable FIP snooping on the FCoE VLANs on Transit Switches TS1 and TS2 that include the access ports connected to the FCoE hosts.

Do not enable FIP snooping on the switches used to create the MC-LAG. For example, in Figure 9 on page 105, you would not enable FIP snooping on the FCoE VLANs on Switches S1 and S2.

Configure links between switches as FCoE trusted ports to reduce FIP snooping overhead and ensure that the system performs FIP snooping only at the access edge. In the sample topology, configure the Transit Switch TS1 and TS2 LAG ports connected to the MC-LAG switches as FCoE trusted ports, configure the Switch S1 and S2 MC-LAG ports connected to Switches TS1 and TS2 as FCoE trusted ports, and configure the ports in the LAG that connects Switches S1 to S2 as FCoE trusted ports.

CoS and Data Center Bridging (DCB)

The MC-LAG links do not carry forwarding class or priority information. The following CoS properties must have the same configuration on each MC-LAG switch or on each MC-LAG interface to support lossless transport:

- FCoE forwarding class name—For example, the forwarding class for FCoE traffic could use the default **fcoe** forwarding class on both MC-LAG switches.
- FCoE output queue—For example, the **fcoe** forwarding class could be mapped to queue 3 on both MC-LAG switches (queue 3 is the default mapping for the **fcoe** forwarding class).
- Classifier—The forwarding class for FCoE traffic must be mapped to the same IEEE 802.1p code point on each member interface of the MC-LAG on both MC-LAG switches. For example, the FCoE forwarding class **fcoe** could be mapped to IEEE 802.1p code point **011** (code point **011** is the default mapping for the **fcoe** forwarding class).
- Priority-based flow control (PFC)—PFC must be enabled on the FCoE code point on each MC-LAG switch and applied to each MC-LAG interface using a congestion notification profile.

You must also configure enhanced transmission selection (ETS) on the MC-LAG interfaces to provide sufficient scheduling resources (bandwidth, priority) for lossless transport. The ETS configuration can be different on each MC-LAG switch, as long as enough resources are scheduled to support lossless transport for the expected FCoE traffic.

LLDP and DCBX must be enabled on each MC-LAG member interface (LLDP and DCBX are enabled by default on all interfaces).

NOTE: As with all other FCoE configurations, FCoE traffic requires a dedicated VLAN that carries only FCoE traffic, and IGMP snooping must be disabled on the FCoE VLAN.

Related Documentation

- *Understanding Multichassis Link Aggregation*
- Example: Configuring CoS for FCoE Transit Switch Traffic Across an MC-LAG on page 194
- *Example: Configuring Multichassis Link Aggregation*

Understanding DCBX

Data Center Bridging Capability Exchange protocol (DCBX) is an extension of Link Layer Data Protocol (LLDP). If you disable LLDP on an interface, that interface cannot run DCBX. If you attempt to enable DCBX on an interface on which LLDP is disabled, the configuration commit operation fails. Data center bridging (DCB) devices use DCBX to exchange configuration information with directly connected peers.

Video: **What is DCBX Protocol?**

This topic describes:

DCBX Basics

DCBX can:

- Discover the DCB capabilities of peers.
- Detect DCB feature misconfiguration or mismatches between peers.
- Configure DCB features on peers.

You can configure DCBX operation for priority-based flow control (PFC), Layer 2 and Layer 4 applications such as FCoE and iSCSI, and ETS. DCBX is enabled or disabled on a per-interface basis.

By default, for PFC and ETS, DCBX automatically negotiates administrative state and configuration with each interface's connected peer. To enable DCBX negotiation for applications, you must configure the applications, map them to IEEE 802.1p code points in an application map, and apply the application map to interfaces.

The FCoE application only needs to be included in an application map when you want an interface to exchange type, length, and values (TLVs) for other applications in addition to FCoE. If FCoE is the only application you want an interface to advertise, then you do not need to use an application map. For ETS, DCBX pushes the switch configuration to peers if they are set to learn the configuration from the switch (unless you disable sending the ETS recommendation TLV on interfaces in IEEE DCBX mode).

You can override the default behavior for PFC, for ETS, or for all applications mapped to an interface by turning off autonegotiation to force an interface to enable or disable that feature. You can also disable DCBX autonegotiation for applications on an interface by excluding those applications from the application map you apply to that interface or by deleting the application map from the interface.

The default autonegotiation behavior for applications that are mapped to an interface is:

- DCBX is enabled on the interface if the connected peer device also supports DCBX.
- DCBX is disabled on the interface if the connected peer device does not support DCBX.

During negotiation of capabilities, the switch can push the PFC configuration to an attached peer if the peer is configured as “willing” to learn the PFC configuration from other peers. The Juniper Networks switch does not support self autoprovisioning and does not change its configuration during autonegotiation to match the peer configuration. (The Juniper switch is not “willing” to learn the PFC configuration from peers.)

NOTE: When a port with DCBX enabled begins to exchange type, length, and value (TLV) entries, optional LLDP TLVs on that port are not advertised to neighbors, so that the switch can interoperate with a wider variety of converged network adapters (CNAs) and Layer 2 switches that support DCBX.

DCBX Modes and Support

This section describes DCBX support on the QFX Series:

DCBX Modes (Versions)

The QFX Series supports the two most common DCBX modes:

- IEEE DCBX—The newest DCBX version. Different TLVs have different subtypes (for example, the subtype for the ETS configuration TLV is 9); the IEEE DCBX Organizationally Unique Identifier (OUI) is 0x0080c2.
- DCBX version 1.01—The Converged Enhanced Ethernet (CEE) version of DCBX. It has a subtype of 2 and an OUI of 0x001b21.

IEEE DCBX and DCBX version 1.01 differ mainly in frame format. DCBX version 1.01 uses one TLV that includes all DCBX attribute information, which is sent as sub-TLVs. IEEE DCBX uses a unique TLV for each DCB attribute.

NOTE: The QFX Series does not support pre-CEE (pre-DCB) DCBX versions. Unsupported older versions of DCBX have a subtype of 1 and an OUI of 0x001b21. The QFX Series drops LLDP frames that contain pre-CEE DCBX TLVs.

Table 9 on page 110 summarizes the differences between IEEE DCBX and DCBX version 1.01, including show command output:

Table 9: Summary of Differences Between IEEE DCBX and DCBX Version 1.01

Characteristic	IEEE DCBX	DCBX Version 1.01
OUI	0x0080c2	0x001b21
Frame Format	Sends a separate, unique TLV for each DCBX attribute. For example, IEEE DCBX uses separate TLVs for ETS, PFC, and each application. Configuration and Recommendation information is sent in different TLVs	Sends one TLV that includes all DCBX attribute information organized in sub-TLVs. The “willing” bit determines whether or not an interface can change its configuration to match the connected peer.
Symmetric/asymmetric configuration with peer	Asymmetric or symmetric	Symmetric only
Differences in the **show dcbx interface** ***interface-name*** operational command	• Synchronization information is not shown because symmetric configuration is not required. • Operational state information is not shown because the operational states do not have to be symmetric. • TLV type is shown because unique TLVs are sent for each DCBX attribute. • ETS peer Configuration TLV and Recommendation TLV information is shown separately because they are different TLVs.	• Synchronization information is shown because symmetric configuration is required. • Operational state information is shown because the operational states do have to be symmetric. • TLV type is not shown because one TLV is used for all attribute information. • Recommendation TLV is not sent (DCBX Version 1.01 uses the “willing” bit to determine whether or not an interface uses the peer interface configuration).

For more information about how each DCBX mode exchanges TLVs, see the following specifications:

- For DCBX version 1.01—http://www.ieee802.org/1/files/public/docs2008/az-wadekar-dcbx-capability-exchange-discovery-protocol-1108-v1.01.pdf
- For IEEE DCBX—http://www.ieee802.org/1/files/private/az-drafts/d2/802-1az-d2-4.pdf

NOTE: As of Junos OS Release 12.2, this document is located in a private area of the IEEE website, and access requires a password from the IEEE organization. If you are not an IEEE member, you might not be able to access this document until it moves to the public area of the IEEE website.

You can configure interfaces to use the following DCBX modes:

- IEEE DCBX—The interface uses IEEE DCBX regardless of the configuration on the connected peer.
- DCBX version 1.01—The interface uses DCBX version 1.01 regardless of the configuration on the connected peer.
- Autonegotiation—The interface automatically negotiates with the connected peer to determine the DCBX version the peers use. Autonegotiation is the default DCBX mode.

If you configure a DCBX mode on an interface, the interface ignores DCBX protocol data units (PDUs) it receives from the connected peer if the PDUs do not match the DCBX version configured on the interface. For example, if you configure an interface to use IEEE DCBX and the connected peer sends DCBX version 1.01 LLDP PDUs, the interface ignores the version 1.01 PDUs. If you configure an interface to use DCBX version 1.01 and the peer sends IEEE DCBX LLDP PDUs, the interface ignores the IEEE DCBX PDUs.

NOTE: On interfaces that use the IEEE DCBX mode, the **show dcbx neighbors interface** ***interface-name*** operational command does not include application, PFC, or ETS operational state in the output.

Autonegotiation

Autonegotiation is the default DCBX mode. Each interface automatically negotiates with its connected peer to determine the DCBX version that both interfaces use to exchange DCBX information.

When an interface connects to its peer interface, the interface advertises IEEE DCBX TLVs to the peer. If the interface receives one IEEE DCBX PDU from the peer, the interface sets the DCBX mode as IEEE DCBX. If the interface receives three DCBX version 1.01 TLVs from the peer, the interface sets DCBX version 1.01 as the DCBX mode.

Autonegotiation works slightly differently on QFX3500 switches and QFabric systems:

- QFX3500 switch—When an interface connects to its peer interface, the interface advertises IEEE DCBX TLVs to the peer. If the interface receives an IEEE DCBX TLV from the peer, the interface sets IEEE DCBX as the DCBX mode. If the interface receives three consecutive DCBX version 1.01 TLVs from the peer, the interface sets DCBX version 1.01 as the DCBX mode.
- QFabric system—When an interface connects to its peer interface, the interface advertises DCBX version 1.01 TLVs to the peer. If the interface receives an IEEE DCBX TLVs from the peer, the interface sets IEEE DCBX as the DCBX mode. If the interface receives three consecutive DCBX version 1.01 TLVs from the peer, the interface retains DCBX version 1.01 as the DCBX mode.

NOTE: If the link flaps or the LLDP process restarts, the interface starts the autonegotiation process again. The interface does not use the last received DCBX communication mode.

CNA Support for DCBX Modes

Different CNA vendors support different versions and capabilities of DCBX. The DCBX configuration you use on QFX Series interfaces depends on the DCBX features that the CNAs in your network support.

Interface Support for DCBX

You can configure DCBX on 10-Gigabit Ethernet interfaces and on link aggregation group (LAG) interfaces whose member interfaces are all 10-Gigabit Ethernet interfaces.

DCBX Attribute Types

DCBX has three attribute types:

- Informational—These attributes are exchanged using LLDP, but do not affect DCBX state or operation; they only communicate information to the peer. For example, application priority TLVs are informational TLVs.
- Asymmetric—The values for these types of attributes do not have to be the same on the connected peer interfaces. Peers exchange asymmetric attributes when the attribute values can differ on each peer interface. The peer interface configurations might match or they might differ. For example, ETS Configuration and Recommendation TLVs are asymmetric TLVs.
- Symmetric—The intention is that the values for these types of attributes should be the same on both of the connected peer interfaces. Peer interfaces exchange symmetric attributes to ensure symmetric DCBX configuration for those attributes. For example, PFC Configuration TLVs are symmetric TLVs.

The following sections describe asymmetric and symmetric DCBX attributes:

Asymmetric Attributes

DCBX passes asymmetric attributes between connected peer interfaces to communicate parameter information about those attributes (features). The resulting configuration for an attribute might be different on each peer, so the parameters configured on one interface might not match the parameters on the connected peer interface.

There are two types of asymmetric attribute TLVs:

- Configuration TLV—Configuration TLVs communicate the current operational state and the state of the "willing" bit. The "willing" bit communicates whether or not the interface is willing to accept and use the configuration from the peer interface. If an interface is "willing," the interface uses the configuration it receives from the peer interface. (The peer interface configuration can override the configuration on the "willing" interface.) If an interface is "not willing", the configuration on the interface cannot be overridden by the peer interface configuration.

- Recommendation TLV—Recommendation TLVs communicate the parameters the interface recommends that the connected peer interface should use. When an interface sends a Recommendation TLV, if the connected peer is “willing,” the connected peer changes its configuration to match the parameters in the Recommendation TLV.

Symmetric Attributes

DCBX passes symmetric attributes between connected peer interfaces to communicate parameter information about those attributes (features), with the objective that both interfaces should use the same configuration. The intent is that the parameters configured on one interface should match the parameters on the connected peer interface.

There is one type of symmetric attribute TLV, the Configuration TLV. As with asymmetric attributes, symmetric attribute Configuration TLVs communicate the current operational state and the state of the “willing” bit. “Willing” interfaces use the peer interface parameter values for the attribute. (The attribute configuration of the peer overrides the configuration on the “willing” interface.)

DCBX Application Protocol TLV Exchange

DCBX advertises the switch’s capabilities for Layer 2 applications such as FCoE and Layer 4 applications such as iSCSI:

Application Protocol TLV Exchange

For all applications, DCBX advertises the application’s state and IEEE 802.1p code points on the interfaces to which the application is mapped. If an application is not mapped to an interface, that interface does not advertise the application’s TLVs. There is an exception for FCoE application protocol TLV exchange when FCoE is the only application you want DCBX to advertise on an interface.

FCoE Application Protocol TLV Exchange

Protocol TLV exchange for the FCoE application depends on whether FCoE is the only application you want the interface to advertise or whether you want the interface to exchange other application TLVs in addition to FCoE TLVs.

If FCoE is the only application you want DCBX to advertise on an interface, DCBX exchanges FCoE application protocol TLVs by default if the interface:

- Carries FCoE traffic (traffic mapped by CoS configuration to the FCoE forwarding class)
- Has a congestion notification profile with PFC enabled on the FCoE priority (IEEE 802.1p code point)
- Does *not* have an application map

NOTE: If no CoS configuration for FCoE is mapped to an interface, that interface does not exchange FCoE application protocol TLVs.

If you want DCBX to advertise FCoE and other applications on an interface, you must specify all of the applications, including FCoE, in an application map, and apply the application map to the desired interfaces.

NOTE: If an application map is applied to an interface, the FCoE application must be explicitly configured in the application map, or the interface does not exchange FCoE TLVs.

When DCBX advertises the FCoE application, it advertises the FCoE state and IEEE 802.1p code points. If a peer device connected to a switch interface does not support FCoE, DCBX uses autonegotiation to mark the interface as “FCoE down,” and FCoE is disabled on that interface.

Disabling Application Protocol TLV Exchange

To disable DCBX application protocol exchange for all applications on an interface, issue the **set protocols dcbx interface** ***interface-name*** **applications no-auto-negotiation** command.

You can also disable DCBX application protocol exchange for applications on an interface by deleting the application map from the interface, or by deleting a particular application from the application map. However, when you delete an application from an application map, the application protocol is no longer exchanged on any interface which uses that application map.

DCBX and PFC

After you enable PFC on a switch interface, DCBX uses autonegotiation to control the operational state of the PFC functionality.

If the peer device connected to the interface supports PFC and is provisioned compatibly with the switch, DCBX sets the PFC operational state to enabled. If the peer device connected to the interface does not support PFC or is not provisioned compatibly with the switch, DCBX sets the operational state to disabled. (PFC must be symmetrical.)

If the peer advertises that it is “willing” to learn its PFC configuration from the switch, DCBX pushes the switch’s PFC configuration to the peer and does not check the peer’s administrative state.

You can manually override DCBX control of the PFC operational state on a per-interface basis by disabling autonegotiation. If you disable autonegotiation on an interface on which you have configured PFC, then PFC is enabled on that interface regardless of the peer configuration. To disable PFC on an interface, do not configure PFC on that interface.

DCBX and ETS

This section describes:

Default DCBX ETS Advertisement

If you do not configure ETS on an interface, the switch automatically creates a default priority group that contains all of the priorities (forwarding classes, which represent output queues) and assigns 100 percent of the port output bandwidth to that priority group. The default priority group is transparent. It does not appear in the configuration and is used for DCBX advertisement. DCBX advertises the default priority group, its priorities, and the assigned bandwidth.

If you configure ETS on an interface, DCBX advertises:

- Each priority group on the interface
- The priorities in each priority group
- The bandwidth properties of each priority group and priority

Any priority on that interface that is not part of an explicitly configured priority group (forwarding class set) is assigned to the automatically generated default priority group and receives no bandwidth. If you configure ETS on an interface, every forwarding class (priority) on that interface for which you want to forward traffic must belong to a forwarding class set (priority group).

ETS Advertisement and Peer Configuration

DCBX does not control the switch's ETS (hierarchical scheduling) operational state. If the connected peer is configured as “willing,” DCBX pushes the switch's ETS configuration to the switch's peers if the ETS Recommendation TLV is enabled (it is enabled by default). If the peer does not support ETS or is not consistently provisioned with the switch, DCBX does not change the ETS operational state on the switch. The ETS operational state remains enabled or disabled based only on the switch hierarchical scheduling configuration and is enabled by default.

When ETS is configured, DCBX advertises the priority groups, the priorities in the priority groups, and the bandwidth configuration for the priority groups and priorities. Any priority (essentially a forwarding class or queue) that is not part of a priority group has no scheduling properties and receives no bandwidth.

You can manually override whether DCBX advertises the ETS state to the peer on a per-interface basis by disabling autonegotiation. This does not affect the ETS state on the switch or on the peer , but it does prevent the switch from sending the Recommendation TLV or the Configuration TLV to the connected peer. To disable ETS on an interface, do not configure priority groups (forwarding class sets) on the interface.

ETS Recommendation TLV

The ETS Recommendation TLV communicates the ETS settings that the switch wants the connected peer interface to use. If the peer interface is “willing,” it changes its configuration to match the configuration in the ETS Recommendation TLV. By default, the switch interfaces send the ETS Recommendation TLV to the peer. The settings communicated are the egress ETS settings defined by configuring hierarchical scheduling on the interface.

We recommend that you use the same ETS settings on the connected peer that you use on the switch interface and that you leave the ETS Recommendation TLV enabled. However, on interfaces that use IEEE DCBX as the DCBX mode, if you want an asymmetric configuration between the switch interface and the connected peer, you can disable the ETS Recommendation TLV by including the **no-recommendation-tlv** statement at the **[edit protocols dcbx interface *interface-name* enhanced-transmission-selection]** hierarchy level.

NOTE: **You can disable the ETS Recommendation TLV only when the DCBX mode on the interface is IEEE DCBX. Disabling the ETS Recommendation TLV has no effect if the DCBX mode on the interface is DCBX version 1.01. (IEEE DCBX uses separate application attribute TLVs, but DCBX version 1.01 sends all application attributes in the same TLV and uses sub-TLVs to separate the information.)**

If you disable the ETS Recommendation TLV, the switch still sends the ETS Configuration TLV to the connected peer. The result is that the connected peer is informed about the switch DCBX ETS configuration, but even if the peer is “willing,” the peer does not change its configuration to match the switch configuration. This is asymmetric configuration—the two interfaces can have different parameter values for the ETS attribute.

For example, if you want a CNA connected to a switch interface to have different bandwidth allocations than the switch ETS configuration, you can disable the ETS Recommendation TLV and configure the CNA for the desired bandwidth. The switch interface and the CNA exchange configuration parameters, but the CNA does not change its configuration to match the switch interface configuration.

Related Documentation

- Understanding DCBX Application Protocol TLV Exchange on page 117
- Understanding DCB Features and Requirements on page 16
- Understanding CoS Flow Control (Ethernet PAUSE and PFC) on page 121
- *Understanding CoS Hierarchical Port Scheduling (ETS)*
- Understanding FCoE on page 19
- Configuring the DCBX Mode on page 305
- Configuring DCBX Autonegotiation on page 306
- Disabling the ETS Recommendation TLV on page 309

• Example: Configuring DCBX Application Protocol TLV Exchange on page 217

Understanding DCBX Application Protocol TLV Exchange

Data Center Bridging Capability Exchange protocol (DCBX) discovers the data center bridging (DCB) capabilities of connected peers. DCBX also advertises the capabilities of applications on interfaces by exchanging application protocol information through application type, length, and value (TLV) elements. DCBX is an extension of Link Layer Discovery Protocol (LLDP). LLDP must remain enabled on every interface on which you want to use DCBX.

NOTE: LLDP and DCBX are enabled by default on all interfaces.

Setting up application protocol exchange consists of:

- Defining applications
- Mapping the applications to IEEE 802.1p code points in an *application map*
- Configuring classifiers to prioritize incoming traffic and map the incoming traffic to the application by the traffic code points
- Applying the application maps and classifiers to interfaces

You need to explicitly define the applications that you want an interface to advertise. The FCoE application is a special case (see "Applications" on page 117) and only needs to be defined on an interface if you want DCBX to exchange application protocol TLVs for other applications in addition to FCoE on that interface.

You also need to explicitly map all defined applications that you want an interface to advertise to IEEE 802.1p code points in an application map. The FCoE application is a special case (see "Application Maps" on page 118) and only requires inclusion in an application map when you want an interface to use DCBX for other applications in addition to FCoE, as described later in this topic.

This topic describes:

Applications

Before an interface can exchange application protocol information, you need to define the applications that you want to advertise, except FCoE if FCoE is the only application that you want the interface to advertise.

NOTE: If FCoE is the only application that you want DCBX to advertise on an interface, DCBX exchanges FCoE application protocol TLVs by default if the interface:

- Carries FCoE traffic (traffic mapped by CoS configuration to the FCoE forwarding class and applied to the interface)
- Has a congestion notification profile with PFC enabled on the FCoE priority (IEEE 802.1p code point)
- Does *not* have an application map

If you apply an application map to an interface, then all applications that you want DCBX to advertise must be defined and configured in the application map, including the FCoE application.

If no CoS configuration for FCoE is mapped to an interface, that interface does not exchange FCoE application protocol TLVs.

You can define:

- Layer 2 applications by EtherType
- Layer 4 applications by a combination of protocol (TCP or UDP) and destination port number

The EtherType is a two-octet field in the Ethernet frame that denotes the protocol encapsulated in the frame. For a list of common EtherTypes, see http://standards.ieee.org/develop/regauth/ethertype/eth.txt on the IEEE standards organization website. For a list of port numbers and protocols, see the *Service Name and Transport Protocol Port Number Registry* at http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml on the Internet Assigned Numbers Authority (IANA) website.

You must explicitly define each application that you want to advertise, except FCoE. The FCoE application is defined by default (EtherType 0x8906).

Application Maps

An application map maps defined applications to one or more IEEE 802.1p code points. Each application map contains one or more applications. DCBX includes the configured application code points in the protocol TLVs exchanged with the connected peer.

To exchange protocol TLVs for an application, you must include the application in an application map. The FCoE application is a special case:

- If you want DCBX to exchange application protocol TLVs for more than one application on a particular interface, you must configure the applications, define an application map to map the applications to code points, and apply the application map to the interface. In this case, you must also define the FCoE application and add it to the application map.

This is the same process and treatment required for all other applications. In addition, for DCBX to exchange FCoE application TLVs, you must enable priority-based flow control (PFC) on the FCoE priority (the FCoE IEEE 802.1p code point) on the interface.

- If FCoE is the only application that you want DCBX to advertise on an interface, then you do not need to configure an application map and apply it to the interface. By default, when an interface has no application map, and the interface carries traffic mapped to the FCoE forwarding class, and PFC is enabled on the FCoE priority, the interface advertises FCoE TLVs (autonegotiation mode). DCBX exchanges FCoE application protocol TLVs by default until you apply an application map to the interface, remove the FCoE traffic from the interface (you can do this by removing the or editing the classifier for FCoE traffic), or disable PFC on the FCoE priority.

  If you apply an application map to an interface that did not have an application map and was exchanging FCoE application TLVs, and you do not include the FCoE application in the application map, the interface stops exchanging FCoE TLVs. Every interface that has an application map must have FCoE included in the application map (and PFC enabled on the FCoE priority) in order for DCBX to exchange FCoE TLVs.

Mapping an application to code points does two things:

- Maps incoming traffic with the same code points to that application
- Allows you to configure classifiers that map incoming application traffic, by code point, to a forwarding class and a loss priority, in order to apply class of service (CoS) to application traffic and prioritize application traffic

You apply an application map to an interface to enable DCBX application protocol exchange on that interface for each application specified in the application map. All of the applications that you want an interface to advertise must be configured in the application map that you apply to the interface, with the previously noted exception for the FCoE application when FCoE is the only application for which you want DCBX to exchange protocol TLVs on an interface.

Classifying and Prioritizing Application Traffic

When traffic arrives at an interface, the interface classifies the incoming traffic based on its code points. Classifiers map code points to loss priorities and forwarding classes. The loss priority prioritizes the traffic. The forwarding class determines the traffic output queue and CoS service level.

When you map an application to an IEEE 802.1p code point in an application map and apply the application map to an interface, incoming traffic on the interface that matches the application code points is mapped to the appropriate application. The application receives the loss priority and the CoS associated with the forwarding class for those code points, and is placed in the output queue associated with the forwarding class.

You can use the default classifier or you can configure a classifier to map the application code points defined in the application map to forwarding classes and loss priorities.

Enabling Interfaces to Exchange Application Protocol Information

Each interface with the **fcoe** forwarding class and PFC enabled on the FCoE code point is enabled for FCoE application protocol exchange by default until you apply an application map to the interface. If you apply an application map to an interface and you want that interface to exchange FCoE application protocol TLVs, you must include the FCoE application in the application map. (In all cases, to achieve lossless transport, you must also enable PFC on the FCoE code point or code points.)

Except when FCoE is the only protocol you want DCBX to advertise on an interface, interfaces on which you want to exchange application protocol TLVs must include the following two items:

- The application map that contains the application(s)
- A classifier

NOTE: You must also enable PFC on the code point of any traffic for which you want to achieve lossless transport.

Disabling DCBX Application Protocol Exchange

To disable DCBX application protocol exchange for all applications on an interface, issue the **set protocols dcbx interface *interface-name* applications no-auto-negotiation** command.

You can also disable DCBX application protocol exchange for applications on an interface by deleting the application map from the interface, or by deleting a particular application from the application map. However, when you delete an application from an application map, the application protocol is no longer exchanged on any interface which uses that application map.

On interfaces that use IEEE DCBX mode to exchange DCBX parameters, you can disable sending the enhanced transmission selection (ETS) Recommendation TLV to the peer if you want an asymmetric ETS configuration between the peers.

Related Documentation

- Understanding DCBX on page 108
- *Understanding CoS Classifiers*
- Configuring DCBX Autonegotiation on page 306
- Disabling the ETS Recommendation TLV on page 309
- Defining an Application for DCBX Application Protocol TLV Exchange on page 309
- Configuring an Application Map for DCBX Application Protocol TLV Exchange on page 311
- Applying an Application Map to an Interface for DCBX Application Protocol TLV Exchange on page 312
- Example: Configuring DCBX Application Protocol TLV Exchange on page 217

- *Example: Configuring Unicast Classifiers*

Understanding CoS Flow Control (Ethernet PAUSE and PFC)

Flow control supports lossless transmission by regulating traffic flows to avoid dropping frames during periods of congestion. Flow control stops and resumes the transmission of network traffic between two connected peer nodes on a full-duplex Ethernet physical link. Controlling the flow by pausing and restarting it prevents buffers on the nodes from overflowing and dropping frames. You configure flow control on a per-interface basis.

The QFX Series supports two methods of flow control:

- IEEE 802.3X Ethernet PAUSE
- IEEE 802.1Qbb priority-based flow control (PFC)

Ethernet PAUSE and PFC are link-level flow control mechanisms.

Ethernet PAUSE pauses transmission of all traffic on a physical Ethernet link.

PFC decouples the pause function from the physical Ethernet link and enables you to divide traffic on one link into eight priorities. You can think of the eight priorities as eight "lanes" of traffic that are mapped to forwarding classes and output queues. Each priority is mapped to a 3-bit IEEE 802.1p CoS code point flag in the VLAN header. You can enable PFC on one or more priorities (IEEE 802.1p code points) on a link. When PFC-enabled traffic is paused on a link, traffic that is not PFC-enabled continues to flow (or is dropped if congestion is severe enough).

Video: Why Use PFC in a Data Center Network?

Use Ethernet PAUSE when you want to prevent packet loss on all of the traffic on a link. Use PFC to prevent traffic loss only on specified types of traffic (for example, Fibre Channel over Ethernet traffic).

NOTE: **Depending on the amount of traffic on a link or assigned to a priority, pausing traffic can cause ingress port congestion and spread congestion through the network.**

Attempting to configure both Ethernet PAUSE and PFC on a link causes a commit error. Ethernet PAUSE and PFC are mutually exclusive configurations on an interface.

By default, all forms of flow control are disabled. You must explicitly enable flow control on interfaces to pause traffic.

- Ethernet PAUSE on page 122
- PFC on page 126
- Lossless Transport Support Summary on page 129

Ethernet PAUSE

Ethernet PAUSE is a congestion relief feature that works by providing link-level flow control for all traffic on a full-duplex Ethernet link. Ethernet PAUSE works in both directions on the link. In one direction, an interface generates and sends Ethernet PAUSE messages to stop the connected peer from sending more traffic. In the other direction, the interface responds to Ethernet PAUSE messages it receives from the connected peer to stop sending traffic. Ethernet PAUSE also works on aggregated Ethernet interfaces. For example, if the connected peer interfaces are called Node A and Node B:

- When the receive buffers on interface Node A reach a certain level of fullness, the interface generates and sends an Ethernet PAUSE message to the connected peer (interface Node B) to tell the peer to stop sending frames. The Node B buffers store frames until the time period specified in the Ethernet PAUSE frame elapses; then Node B resumes sending frames to Node A.
- When interface Node A receives an Ethernet PAUSE message from interface Node B, interface Node A stops transmitting frames until the time period specified in the Ethernet PAUSE frame elapses; then Node A resumes transmission. (The Node A transmit buffers store frames until Node A resumes sending frames to Node B.)

  In this scenario, if Node B sends an Ethernet PAUSE frame with a time value of 0 to Node A, the 0 time value indicates to Node A that it can resume transmission. This happens when the Node B buffer empties to below a certain threshold and the buffer can once again accept traffic.

Symmetric flow control means an interface has the same Ethernet PAUSE configuration in both directions. The Ethernet PAUSE generation and Ethernet PAUSE response functions are both configured as enabled, or they are both disabled. You configure symmetric flow control by including the **flow-control** statement at the **[edit interfaces *interface-name* ether-options]** hierarchy level.

Asymmetric flow control allows you to configure the Ethernet PAUSE functionality in each direction independently on an interface. The configuration for generating Ethernet PAUSE messages and for responding to Ethernet PAUSE messages does not have to be the same. It can be enabled in both directions, disabled in both directions, or enabled in one direction and disabled in the other direction. You configure asymmetric flow control by including the **configured-flow-control** statement at the **[edit interfaces *interface-name* ether-options]** hierarchy level.

On any particular interface, symmetric and asymmetric flow control are mutually exclusive. Asymmetric flow control overrides and disables symmetric flow control. (If PFC is configured on an interface, the PFC configuration overrides Ethernet PAUSE flow control.) The QFX Series supports both symmetric and asymmetric flow control.

Symmetric Flow Control

Symmetric flow control configures both the receive and transmit buffers in the same state. The interface can both send Ethernet PAUSE messages and respond to them (flow control is enabled), or the interface cannot send Ethernet PAUSE messages or respond to them (flow control is disabled).

When you enable symmetric flow control on an interface, the Ethernet PAUSE behavior depends on the configuration of the connected peer. With symmetric flow control enabled, the interface can perform any Ethernet PAUSE functions that the connected peer can perform. (When symmetric flow control is disabled, the interface does not send or respond to Ethernet PAUSE messages.)

Asymmetric Flow Control

Asymmetric flow control enables you to specify independently whether or not the interface receive buffer generates and sends Ethernet PAUSE messages to stop the connected peer from transmitting traffic, and whether or not the interface transmit buffer responds to Ethernet PAUSE messages it receives from the connected peer and stops transmitting traffic. The receive buffer configuration determines if the interface transmits Ethernet PAUSE messages, and the transmit buffer configuration determines if the interface receives and responds to Ethernet PAUSE messages:

- Receive buffers on—Enable Ethernet PAUSE transmission (generate and send Ethernet PAUSE frames)
- Transmit buffers on—Enable Ethernet PAUSE reception (respond to received Ethernet PAUSE frames)

You must explicitly set the flow control for both the receive buffer and the transmit buffer (**on** or **off**) to configure asymmetric Ethernet PAUSE. Table 10 on page 123 describes the configured flow control state when you set the receive (Rx) and transmit (Tx) buffers on an interface:

Table 10: Asymmetric Ethernet PAUSE Flow Control Configuration

Receive (Rx) Buffer	Transmit (Tx) Buffer	Configured Flow Control State
On	Off	Interface generates and sends Ethernet PAUSE messages. Interface does not respond to Ethernet PAUSE messages (interface continues to transmit even if peer requests that the interface stop sending traffic).
Off	On	Interface responds to Ethernet PAUSE messages received from the connected peer, but does not generate or send Ethernet PAUSE messages. (The interface does not request that the connected peer stop sending traffic.)
On	On	Same functionality as symmetric Ethernet PAUSE. Interface generates and sends Ethernet PAUSE messages and responds to received Ethernet PAUSE messages.
Off	Off	Ethernet PAUSE flow control is disabled.

The configured flow control is the Ethernet PAUSE state configured on the interface.

On 1-Gigabit Ethernet interfaces, the QFX Series supports autonegotiation of Ethernet PAUSE with the connected peer. (The QFX Series does not support autonegotiation on 10-Gigabit Ethernet interfaces.) Autonegotiation enables the interface to exchange state advertisements with the connected peer so that the two devices can agree on the Ethernet PAUSE configuration. Each interface advertises its flow control state to the connected peer using a combination of the Ethernet PAUSE and ASM_DIR bits, as described in Table 11 on page 124:

Table 11: Flow Control State Advertised to the Connected Peer (Autonegotiation)

Rx Buffer State	Tx Buffer State	PAUSE Bit	ASM_DIR Bit	Description
Off	Off	0	0	The interface advertises no Ethernet PAUSE capability. This is equivalent to disabling flow control on an interface.
On	On	1	0	The interface advertises symmetric flow control (both the transmission of Ethernet PAUSE messages and the ability to receive and respond to Ethernet PAUSE messages).
On	Off	0	1	The interface advertises asymmetric flow control (the transmission of Ethernet PAUSE messages, but not the ability to receive and respond to Ethernet PAUSE messages).
Off	On	1	1	The interface advertises both symmetric and asymmetric flow control. Although the interface does not generate and send Ethernet PAUSE requests to the peer, the interface supports both symmetric and asymmetric Ethernet PAUSE configuration on the peer because the peer is not affected if the peer does not receive Ethernet PAUSE requests. (If the interface responds to the peer's Ethernet PAUSE requests, that is sufficient to support either symmetric or asymmetric flow control on the peer.)

The flow control configuration on each switch interface interacts with the flow control configuration of the connected peer. Each peer advertises its state to the other peer. The interaction of the flow control configuration of the peers determines the flow control behavior (resolution) between them, as shown in Table 12 on page 125. The first four columns show the Ethernet PAUSE configuration on the local QFX Series and on the connected peer (also known as the link partner). The last two columns show the Ethernet PAUSE resolution that results from the local and peer configurations on each interface. This illustrates how the Ethernet PAUSE configuration of each interface affects the Ethernet PAUSE behavior on the other interface.

NOTE: In the Resolution columns of the table, disabling Ethernet PAUSE transmit means that the interface receive buffers do not generate and send Ethernet PAUSE messages to the peer. Disabling Ethernet PAUSE receive means that the interface transmit buffers do not respond to Ethernet PAUSE messages received from the peer.

Table 12: Asymmetric Ethernet PAUSE Behavior on Local and Peer Interfaces

Local Interface (QFX Series)		Peer Interface			
PAUSE Bit	ASM_DIR Bit	PAUSE Bit	ASM_DIR Bit	Local Resolution	Peer Resolution
0	0	Don't care	Don't care	Disable Ethernet PAUSE transmit and receive	Disable Ethernet PAUSE transmit and receive
0	1	0	Don't care	Disable Ethernet PAUSE transmit and receive	Disable Ethernet PAUSE transmit and receive
0	1	1	0	Disable Ethernet PAUSE transmit and receive	Disable Ethernet PAUSE transmit and receive
0	1	1	1	Enable Ethernet PAUSE transmit and disable Ethernet PAUSE receive	Disable Ethernet PAUSE transmit and enable Ethernet PAUSE receive
1	0	0	Don't care	Disable Ethernet PAUSE transmit and receive	Disable Ethernet PAUSE transmit and receive
1	0	1	Don't care	Enable Ethernet PAUSE transmit and receive	Enable Ethernet PAUSE transmit and receive
1	1	0	0	Disable Ethernet PAUSE transmit and receive	Disable Ethernet PAUSE transmit and receive
1	1	0	1	Enable Ethernet PAUSE receive and disable Ethernet PAUSE transmit	Enable Ethernet PAUSE transmit and disable Ethernet PAUSE receive
1	1	Don't care	Don't care	Enable Ethernet PAUSE transmit and receive	Enable Ethernet PAUSE transmit and receive

NOTE: For your convenience, Table 12 on page 125 replicates Table 28B-3 of Section 2 of the IEEE 802.X specification.

PFC

PFC is a lossless transport and congestion relief feature that works by providing granular link-level flow control for each IEEE 802.1p code point (priority) on a full-duplex Ethernet link. When the receive buffer on a switch interface fills to a threshold, the switch transmits a pause frame to the sender (the connected peer) to temporarily stop the sender from transmitting more frames. The buffer threshold must be low enough so that the sender has time to stop transmitting frames and the receiver can accept the frames already on the wire before the buffer overflows. The switch automatically sets queue buffer thresholds to prevent frame loss.

When congestion forces one priority on a link to pause, all of the other priorities on the link continue to send frames. Only frames of the paused priority are not transmitted. When the receive buffer empties below another threshold, the switch sends a message that starts the flow again.

You configure PFC using a congestion notification profile (CNP). A CNP has two parts:

- Input—Specify the code point (or code points) on which to enable PFC, and optionally specify the maximum receive unit (MRU) and the cable length between the interface and the connected peer interface.
- Output—Specify the output queue or output queues that respond to pause messages from the connected peer.

You apply a PFC configuration by configuring a CNP on one or more interfaces. Each interface that uses a particular CNP is enabled to pause traffic with the priorities (code points) specified in that CNP. You can configure one CNP on an interface, and you can configure different CNPs on different interfaces. When you configure a CNP on an interface, ingress traffic that is mapped to a priority that the CNP enables for PFC is paused whenever the queue buffer fills to the pause threshold. (The pause threshold is not user-configurable.)

Configure PFC for a priority end to end along the entire data path to create a lossless lane of traffic on the network. You can selectively pause the traffic in any queue without pausing the traffic for other queues on the same link. You can create lossless lanes for traffic such as Fibre Channel over Ethernet (FCoE), LAN backup, or management, while using standard frame-drop congestion management for IP traffic on the same link.

Potential consequences of link-level flow control are:

- Ingress port congestion (configuring too many lossless flows can cause ingress port congestion)
- A paused priority that causes upstream devices to pause the same priority, thus spreading congestion back through the network

By definition, PFC supports symmetric pause only (as opposed to Ethernet PAUSE, which supports symmetric and asymmetric pause). With symmetric pause, a device can:

- Transmit pause frames to pause incoming traffic. (You configure this using the input stanza of a congestion notification profile.)

- Receive pause frames and stop sending traffic to a device whose buffer is too full to accept more frames. (You configure this using the output stanza of a congestion notification profile.)

Receiving a PFC frame from a connected peer pauses traffic on egress queues based on the IEEE 802.1p priorities that the PFC pause frame identifies. The priorities are 0 through 7. By default, the priorities map to queue numbers 0 through 7, respectively, and to specific forwarding classes, as shown in Table 13 on page 127:

Table 13: Default PFC Priority to Queue and Forwarding Class Mapping

IEEE 802.1p Priority (Code Point)	Queue	Forwarding Class
0 (000)	0	best-effort
1 (001)	1	best-effort
2 (010)	2	best-effort
3 (011)	3	fcoe
4 (100)	4	no-loss
5 (101)	5	best-effort
6 (110)	6	network-control
7 (111)	7	network-control

For example, a received PFC pause frame that pauses priority 3 pauses output queue 3. If you do not want to use the default configuration, you can configure customized mapping of priorities to queues and forwarding classes.

NOTE: By convention, deployments with converged server access typically use IEEE 802.1p priority 3 for FCoE traffic. The default forwarding class configuration sets the **fcoe** forwarding class as a lossless forwarding class that is mapped to queue 3. The default classifier maps incoming priority 3 traffic to the **fcoe** forwarding class. *However, you must apply PFC to the entire FCoE data path to configure the end-to-end lossless behavior that FCoE traffic requires.*

If your network uses priority 3 for FCoE traffic, we recommend that you use the default configuration. If your network uses a priority other than 3 for FCoE traffic, you can configure lossless FCoE transport on any IEEE 80.21p priority as described in *Understanding CoS IEEE 802.1p Priorities for Lossless Traffic Flows* and *Understanding CoS IEEE 802.1p Priority Remapping on an FCoE-FC Gateway*.

You enable PFC on a priority by:

1. Specifying the IEEE 802.1p code point to pause in the input stanza of a CNP
2. Applying the CNP to the ingress interfaces on which you want to pause the traffic

CAUTION: Any change to the PFC configuration on a port temporarily blocks the entire port (not just the priorities affected by the PFC change) so that the port can implement the change, then unblocks the port. Blocking the port stops ingress and egress traffic, and causes packet loss on all queues on the port until the port is unblocked.

A change to the PFC configuration means any change to a CNP, including changing the input portion of the CNP (enabling or disabling PFC on a priority, or changing the MRU or cable-length values) or changing the output portion the CNP that enables or disables output flow control on a queue. A PFC configuration change only affects ports that use the changed CNP.

The following actions change the PFC configuration:

- Deleting or disabling a PFC configuration (input or output) in a CNP that is in use on one or more interfaces. For example:
  1. An existing CNP with an input stanza that enables PFC on priorities 3, 5, and 6 is configured on interfaces xe-0/0/20 and xe-0/0/21.
  2. We disable the PFC configuration for priority 6 in the input CNP, and then commit the configuration.
  3. The PFC configuration change causes all traffic on interfaces xe-0/0/20 and xe-0/0/21 to stop until the PFC change has been implemented. When the PFC change has been implemented, traffic resumes.
- Configuring a CNP on an interface. (This changes the PFC state by enabling PFC on one or more priorities.)
- Deleting a CNP from an interface. (This changes the PFC state by disabling PFC on one or more priorities.)

When you associate the CNP with an interface, the interface uses PFC to send pause requests when the output queue buffer for the lossless traffic fills to the pause threshold.

Although unicast traffic and multidestination (multicast, broadcast, and destination lookup fail) traffic must use different classifiers, you can map a unicast queue (queue 0 through 7) and a multidestination queue (queue 8, 9, 10, or 11) to the same PFC priority so that both unicast and multicast traffic use that priority. Do not map multidestination traffic to lossless priorities. Starting with Junos OS Release 12.3, you can map one priority to multiple output queues.

NOTE: You can attach a maximum of one CNP to an interface, but you can create an unlimited number of CNPs that explicitly configure only the input stanza and use the default output stanza.

The output stanza of the CNP maps to a profile that interfaces use to respond to pause messages received from the connected peer. On standalone QFX3500 switches and QFX3600 switches, you can create two CNPs with an explicitly configured output stanza.

When a QFX3500 switch or a QFX3600 switch is a Node device in a QFabric system, you can create one CNP with an explicitly configured output stanza. (One fewer profile is available on QFabric systems because the system needs a default profile for fabric interfaces, which are not used as fabric interfaces when the switches are not part of a QFabric system. *Understanding CoS IEEE 802.1p Priorities for Lossless Traffic Flows* describes configuring output flow control.

Lossless Transport Support Summary

The QFX Series supports up to six lossless forwarding classes. For lossless transport, you must enable PFC on the IEEE 802.1p priorities (code points) mapped to lossless forwarding classes.

CAUTION: Any change to the PFC configuration on a port temporarily blocks the entire port (not just the priorities affected by the PFC change) so that the port can implement the change, then unblocks the port. Blocking the port stops ingress and egress traffic, and causes packet loss on all queues on the port until the port is unblocked.

The following limitation applies to support lossless transport on QFabric systems only:

- The internal fiber cable length from the QFabric system Node device to the QFabric system Interconnect device cannot exceed 150 meters.

The default CoS configuration provides two lossless forwarding classes, *fcoe* and *no-loss*. If you explicitly configure lossless forwarding classes, you must include the **no-loss** packet drop attribute to enable lossless behavior, or the traffic is not lossless. For both default and explicit lossless forwarding class configuration, you must configure CNP input stanzas to enable PFC on the priority of the lossless traffic and apply the CNPs to ingress interfaces.

NOTE: Junos OS Release 12.2 introduced changes to the way the QFX Series handles lossless forwarding classes (including the default **fcoe** and **no-loss** forwarding classes).

In Junos OS Release 12.1, either explicitly configuring the **fcoe** and **no-loss** forwarding classes or using the default configuration for these forwarding classes resulted in the same lossless behavior for traffic mapped to those forwarding classes.

However, in Junos OS Release 12.2, if you explicitly configure the **fcoe** or the **no-loss** forwarding class, that forwarding class is no longer treated as a lossless forwarding class. Traffic mapped to these forwarding classes is treated as lossy (best-effort) traffic. This is true even if the explicit configuration is exactly the same as the default configuration.

If your CoS configuration from Junos OS Release 12.1 or earlier includes the explicit configuration of the **fcoe** or the **no-loss** forwarding class, then when you upgrade to Junos OS Release 12.2, those forwarding classes are not lossless. To preserve the lossless treatment of these forwarding classes, delete the the explicit **fcoe** and **no-loss** forwarding class configuration before you upgrade to Junos OS Release 12.2.

See *Overview of CoS Changes Introduced in Junos OS Release 12.2* for detailed information about this change and how to delete an existing lossless configuration.

In Junos OS Release 12.3, the default behavior of the **fcoe** and **no-loss** forwarding classes is the same as in Junos OS Release 12.2. However, in Junos OS Release 12.3, you can configure up to six lossless forwarding classes. All explicitly configured lossless forwarding classes must include the new **no-loss** packet drop attribute or the forwarding class is lossy.

Understanding CoS IEEE 802.1p Priorities for Lossless Traffic Flows provides detailed information about the explicit configuration of lossless priorities and about the default configuration of lossless priorities, including the input and output stanzas of the CNP.

NOTE: PFC and Ethernet PAUSE are used only on Ethernet interfaces. Fabric (fte) ports on QFabric systems (Node device fabric ports and Interconnect device fabric ports) use link-layer flow control (LLFC) to ensure the appropriate treatment of lossless traffic.

Related Documentation

- *Overview of CoS Changes Introduced in Junos OS Release 12.2*
- *Understanding CoS IEEE 802.1p Priorities for Lossless Traffic Flows*
- *Understanding CoS IEEE 802.1p Priority Remapping on an FCoE-FC Gateway*
- Understanding DCB Features and Requirements on page 16

- *Configuring CoS PFC (Congestion Notification Profiles)*
- *Enabling and Disabling CoS Symmetric Ethernet PAUSE Flow Control*
- *Configuring CoS Asymmetric Ethernet PAUSE Flow Control*
- Example: Configuring CoS PFC for FCoE Traffic on page 186

Understanding Fibre Channel Terminology

To understand the Fibre Channel (FC) and Fibre Channel over Ethernet (FCoE) capabilities of the QFX Series, you should become familiar with the terms defined in Table 14 on page 131.

Table 14: Fibre Channel Terms

Term	Definition
addressing mode	Format for the locally unique MAC address the FC switch assigns to FCoE devices for FCoE transactions after FIP establishes a connection between an FCoE device and the FC switch. The two addressing modes are *fabric-provided MAC address (FPMA)* and *server-provided MAC address (SPMA)*. The QFX Series supports only FPMA. During FLOGI or FDISC, the ENode advertises the addressing modes it supports. If the FC switch supports an addressing mode that the ENode uses, the virtual link can be established, and the devices can communicate. See also *fabric-provided MAC address (FPMA)* and *server-provided MAC address (SPMA)*.
ALL-ENode-MACs	Well-known multicast MAC address to which all FCoE ENodes listen. FCFs send multicast *FIP discovery advertisement* messages and *FIP keepalive* messages to the ALL-ENode-MACs address so that ENodes can discover and maintain connections to FCFs. The hexadecimal format of the address is **01:10:18:01:00:01**. See also *well-known address (WKA)*.
ALL-FCF-MACs	Well-known multicast MAC address to which all FCFs listen. ENodes send multicast *FIP discovery solicitation* messages to the ALL-FCF-MACs address to find out which FCFs can accept a login. The hexadecimal format of the address is **01:10:18:01:00:02**. See also *well-known address (WKA)*.
congestion notification	See *quantized congestion notification (QCN)*.
converged network adapter (CNA)	Physical adapter that combines the functions of a Fibre Channel *host bus adapter (HBA)* to process FCoE frames and a *lossless Ethernet network interface card (NIC)* to process non-FCoE Ethernet frames. CNAs have one or more Ethernet ports. CNAs encapsulate Fibre Channel frames in Ethernet for FCoE transport and de-encapsulate Fibre Channel frames from FCoE to native Fibre Channel. See also *host bus adapter (HBA)*.

Table 14: Fibre Channel Terms *(continued)*

Term	Definition
data center bridging (DCB)	Set of IEEE specifications that enhance Ethernet to allow it to support converged Ethernet (LAN) and Fibre Channel (SAN) traffic on one Ethernet network. DCB features include *priority-based flow control (PFC)*, *enhanced transmission selection (ETS)*, *Data Center Bridging Capability Exchange protocol (DCBX)*, *quantized congestion notification (QCN)*, and full-duplex 10-Gigabit Ethernet ports. See also *priority-based flow control (PFC)*, *Ethernet PAUSE*, *enhanced transmission selection (ETS)*, *Data Center Bridging Capability Exchange protocol (DCBX)*, and *quantized congestion notification (QCN)*.
expansion port (E_Port)	An expansion port in an FC switch/FCF that connects the FC switch/FCF to the E_Port of another FC switch/FCF to form an *Interswitch Link (ISL)* in a common FC fabric.
Data Center Bridging Capability Exchange protocol (DCBX)	Discovery and exchange protocol for conveying configuration and capabilities among neighbors to ensure consistent configuration across the network. It is an extension of the Link Layer Data Protocol (LLDP, described in IEEE 802.1AB) See also *data center bridging (DCB)*.
enhanced transmission selection (ETS)	Mechanism that provides finer granularity of bandwidth management within a link. See also *data center bridging (DCB)*.
ENode	See *FCoE Node (ENode)*
ENode MAC	*Lossless Ethernet MAC* paired with an *FCoE controller* in an ENode. See also *FCoE node (ENode)*.
ENode MAC address	Globally unique address assigned to the CNA by the manufacturer and used to identify the node for FIP transactions.
Ethernet PAUSE	As defined in IEEE 802.3X, a flow control mechanism that temporarily stops the transmission of Ethernet frames on a link for a specified period. A receiving element sends an Ethernet PAUSE frame when a sender transmits data faster than the receiver can accept it. Ethernet PAUSE affects the entire link, not just an individual flow. An Ethernet PAUSE frame temporarily stops all traffic transmission on the link and allows the receiver's input buffer to empty sufficiently to restart traffic on the link. Ethernet PAUSE messages are sent to the previous hop and do not automatically propagate to the source of the congestion. See also *priority-based flow control (PFC)*.
fabric	Interconnection of network nodes using one or more network switches that function as a network single logical entity.

Table 14: Fibre Channel Terms *(continued)*

Term	Definition
fabric discovery (FDISC)	Subsequent logins from the same ENode for different users, applications, or virtual machines after an ENode performs an initial FLOGI to log in to a switch. FC and FIP FDISC messages serve the same function in FC and FCoE networks, respectively. N_Ports send FC FDISC messages to the FC switch and VN_Ports send FIP FDISC messages to the FCF. After an N_Port acquires its initial N_Port ID through the FC FLOGI process, it can acquire additional N_Port IDs by sending an FC FDISC with a new worldwide port name and a source ID of 0x000000. The new port name and blank source ID tell the FC switch to assign a new N_Port ID to the N_Port. The different N_Port IDs allow multiple virtual machines or users on the N_Port to have separate, secure virtual links on the same physical N_Port. These additional ports are also referred to as VN_Ports. FIP FDISC works the same way, except the VN_Port logs in using a FIP FLOGI message. See also *fabric login (FLOGI)* and *N_Port ID*.
fabric login (FLOGI)	Creation of a logical connection to the FC switch and establishment of a node's operating environment. For FC devices, an N_Port logs in to the FC network by sending an FC FLOGI message to the F_Port of an FC switch. For FCoE devices, a VN_Port logs in to the FC network by sending a FIP FLOGI message to the VF_Port of an FC switch.
fabric port (F_Port)	FC port on an FC switch or an FCF that connects point-to-point to an FC node port (N_Port) on an FC host (server or storage device). An F_Port provides access to fabric services for FC devices. F_Ports are intermediate ports in a connection between FC device end-point N_Ports. For example, a connection between an FC host server and an FC storage device through an FC switch looks like this: FC server N_Port to FC switch ingress F_Port to FC switch egress F_Port to FC storage device N_Port. See also *node port (N_Port)*.
fabric-provided MAC address (FPMA)	MAC address that an FCF assigns to a single ENode MAC through the FLOGI or FDISC process that is unique to the local fabric. The FPMA uniquely identifies a single VN_Port at that ENode MAC in FCoE transactions with the FCF. Because an ENode can have more than one ENode MAC, an FCF can assign multiple FPMAs to an ENode, one FPMA per ENode MAC. An FPMA is a 48-bit value that consists of two 24-bit values, the N_Port ID and the FC-MAP value. The N_Port ID uniquely identifies the VN_Port and the FC-MAP value identifies the FCF. See also *FCoE node (ENode)*, *N_Port ID*, and *FCoE mapped address prefix (FC-MAP)*.
FCF-MAC	Lossless Ethernet MAC paired with an FCoE controller in an FCF. The FCF-MAC enables the FCF to handle FCoE traffic.

Table 14: Fibre Channel Terms *(continued)*

Term	Definition
FCoE controller	Instantiates and terminates VN_Port and VF_Port instances on an ENode. An ENode can have more than one FCoE controller. Each FCoE controller is paired with a lossless Ethernet MAC on the ENode. See also *lossless Ethernet MAC*.
FC forwarder (FCF)	Alternative term and acronym to refer to an FC switch that has all physical Fibre Channel ports and the necessary set of services as defined in the T11 Organization *Fibre Channel Switched Fabric* (FC-SW) standards.
FCoE forwarder (FCF)	Defined by the *Fibre Channel Backbone - 5 (FC-BB-5) Rev 2.00* specification available at http://www.t11.org/ftp/t11/pub/fc/bb-5/09-056v5.pdf as a device that has the necessary set of services as defined in FC-SW and the FCoE capabilities to act as an FCoE-based FC switch.
FCoE Initialization Protocol (FIP)	Layer 2 protocol for endpoint discovery, fabric login, and fabric association. FIP enables FCoE devices and FC switches to discover one another. Through FIP, FCoE nodes can log in to an FC switch, access the SAN FC fabric, and communicate with target FC devices. FIP messages also maintain the connection between the FCoE initiator and the FCF. FIP has its own EtherType (0x8914) to distinguish its traffic from payload-carrying FCoE traffic and other Ethernet traffic.
FCoE link endpoint (LEP)	Virtual FC interface mapped onto a physical Ethernet interface to handle FC frame encapsulation and de-encapsulation and transmission and reception of FC frames encapsulated in Ethernet through a single virtual link.
FCoE mapped address prefix (FC-MAP)	24-bit value that identifies the FC switch and is half of the 48-bit FPMA MAC address. The FC-MAP value can be configured on the FC switch and has a default value of 0EFC00h. The FC-MAP value was originally called the Fibre Channel Organizationally Unique Identifier (FC-OUI). See also *fabric-provided MAC address (FPMA)*.
FCoE node (ENode)	Fibre Channel node that has one or more lossless Ethernet MACs, each paired with an *FCoE Controller* in order to transmit FCoE frames. An ENode combines FCoE termination functions and the FC stack on a CNA. ENodes present virtual FC interfaces to FC switches or FCFs in the form of VN_Ports, which can establish FCoE virtual links with FC switch/FCF VF_Ports. ENodes perform FCoE related functions in a *converged network adapter (CNA)*. See also *converged network adapter (CNA)*.
FCoE-FC gateway	A form of N_Port virtualizer in which the node-facing ports are FCoE ports and the FC switch-facing ports are FC ports.
FCoE-FCoE gateway	A form of N_Port virtualizer in which the node-facing ports are FCoE ports and the FC switch-facing ports are FCoE ports.
FC-FC gateway	A form of N_Port virtualizer in which the node-facing ports are FC ports and the FC switch-facing ports are FC ports.

Table 14: Fibre Channel Terms *(continued)*

Term	Definition
FCoE transit switch (also known as a FIP snooping bridge)	Switch with a minimum set of features designed to support FCoE Layer 2 forwarding and FCoE security. The switch can also have optional additional features. Minimum feature support is: • Priority-based flow control (PFC) • Enhanced transmission selection (ETS) • Data Center Bridging Capability Exchange Protocol (DCBX), including the FCoE application TLV • FIP snooping (minimum support is FIP automated filter programming at the ENode edge) Additional FIP snooping capabilities can include learning the virtual FC connection paths (VN2VF, VN2VN, or VE2VE) and monitoring the FIP keepalive mechanisms. Other optional capabilities can also enhance FCoE within the standards. FIP snooping is typically configurable on a per-VLAN basis. A transit switch has an FC stack even though it is not an FC switch or an FCF.
FCoE VLAN	VLAN dedicated to carrying only FCoE traffic. FCoE traffic must travel in a VLAN. Only FCoE interfaces should be members of an FCoE VLAN. Ethernet traffic that is not FCoE traffic must travel in a different VLAN.
Fibre Channel	High-speed network technology used for storage area networks (SANs).
Fibre Channel fabric	Network of Fibre Channel devices that allows communication among devices, device name lookup, security, and redundancy. Also a local fabric on a QFX3500 switch with FCoE interfaces connected to FCoE devices on the Ethernet network and native FC interfaces connected to an FC switch in a SAN.
Fibre Channel ID (FCID)	24-bit value the FC switch assigns to the N_Port or VN_Port as a unique identifier within the local FC network. The FCID consists of an 8-bit domain value, an 8-bit area value, and an 8-bit port value. The FCID is sometimes called an N_Port ID. See also *N_Port ID*.
Fibre Channel over Ethernet (FCoE)	Standard for transporting FC frames over Ethernet networks. FCoE encapsulates Fibre Channel frames in Ethernet so that the same high-speed Ethernet physical infrastructure can transport both data and storage traffic while preserving the lossless CoS that FC requires. FCoE has its own EtherType (0x8906) to differentiate it from other Ethernet traffic. FCoE runs on a DCB network. FCoE servers connect to a switch that supports both FCoE and native FC protocols. This allows FCoE servers on the Ethernet network to access FC storage devices in the SAN fabric on one converged network. See also *data center bridging (DCB)*.

Table 14: Fibre Channel Terms *(continued)*

Term	Definition
Fibre Channel services	Functions required for establishing FC network connectivity among devices and for managing devices on the FC network, such as login servers, domain managers, name servers, and zone servers.
FC stack	FC or FCoE protocol capability implemented on a device to support the FC or FCoE functionality. Having an FC stack does not imply consuming a domain ID. Each FC or FCoE enabled server or storage device has an FC stack. Similarly, an FC or FCoE switch, an FCF, an FCoE-FC gateway, and an FCoE transit switch have FC stacks.
Fibre Channel switch	Network switch that implements the Fibre Channel protocol.
FIP discovery advertisement	Multicast or unicast message that the FC switch (or FCF) transmits to ENodes to advertise the switch's presence on the network so that ENodes can discover the switch and request to log in to the FC fabric. The FC switch periodically sends multicast FIP discovery advertisements to the ALL-ENode-MACs address, a well-known address to which all ENodes listen. The multicast messages advertise the FC switch to all ENodes on the VLAN and serve as keepalive messages to maintain connectivity between the FC switch and ENodes. When an ENode sends a FIP discovery solicitation message to the FC switch, the FC switch responds with a unicast FIP discovery advertisement to that ENode.
FIP discovery solicitation	Multicast or unicast message that an ENode transmits to FC switches (or FCFs) to find compatible switches in the network. When an ENode initializes, it sends a multicast FIP discovery solicitation to the ALL-FCF-MACs address, a well-known address to which all FC switches and FCFs listen. Compatible switches reply with a unicast FIP discovery advertisement. The ENode compiles a list of compatible switches, selects a switch, and logs in to that switch.
FIP keepalive	Periodic multicast FIP discovery advertisement sent from the FC switch or FCF to all ENodes to maintain connectivity.

Table 14: Fibre Channel Terms *(continued)*

Term	Definition
FIP snooping	For VN_Port to VF_Port (VN2VF) paths (Technical Committee T11 BB-FC-5 specification), FIP snooping is a security feature enabled for FCoE VLANs on an Ethernet switch that connects ENodes to FC switches or FCFs. FIP snooping inspects data in FIP frames and uses that data to create firewall filters. The filters permit only traffic from sources that perform a successful FLOGI to the FC switch. All other traffic on the VLAN is denied. FIP snooping filters are installed on the ports in the FCoE VLAN. For VN_Port to VN_Port (VN2VN) paths (Technical Committee T11 BB-FC-6 specification), the FIP snooping security feature filters access between VN_Ports in a similar manner to VN2VF_Port FIP snooping. FIP snooping can also apply similarly to VE_Port to VE_Port (VE2VE) paths. FIP snooping can also snoop to provide additional visibility of FCoE Layer 2 operation. See also *FCoE node (ENode)*.
FIP snooping bridge	See *FCoE transit switch* and *FIP snooping*.
host bus adapter (HBA)	Physical mechanism that connects a host system to other FC network and storage devices. HBAs have a unique worldwide node name (WWNN) for the HBA node, which all of the ports on the HBA share, and each port on an HBA has a unique worldwide port name (WWPN).
initiator	System component that originates an I/O command over an I/O bus or network. An FCoE server sending a request to an FC storage device is an example of an initiator.
iSCSI transit switch	Layer 2 Ethernet switch with a minimum set of best-practice Ethernet features to support iSCSI, along with optional enhancements. Minimum feature support is: • IEEE 802.3X asymmetric and symmetric flow control on ports not running in DCB mode • Priority-based flow control (PFC) • Enhanced transmission selection (ETS) • Data Center Bridging Capability Exchange Protocol (DCBX), including the iSCSI application TLV Other capabilities such as Internet storage name service (iSNS) are optional.
interswitch link (ISL)	Link between the *E_Ports* of two FC switches in a common FC fabric. When two FCoE-based FC switches are connected together, there is a virtual ISL through Layer 2.
logout (LOGO)	For FC devices, an N_Port logs out from the FC network by sending an FC LOGO message to the F_Port of an FC switch. The switch can also send a LOGO message to an N_Port to terminate its connection. For FCoE devices, a VN_Port logs out from the FC network by sending a FIP LOGO message to the VF_Port of an FC switch. The switch can also send a LOGO message to a VN_Port to terminate its connection.

Table 14: Fibre Channel Terms *(continued)*

Term	Definition
lossless Ethernet MAC	Full-duplex Ethernet MAC that implements Ethernet extensions to avoid Ethernet frame loss due to congestion and supports at least 2.5-KB jumbo frames. Each lossless Ethernet MAC combines with an FCoE Controller to perform FCoE termination functions on an ENode. See also *priority-based flow control (PFC)*, *quantized congestion notification (QCN)*, *FCoE controller*, and *FCoE node (ENode)*.
lossless Ethernet network	Ethernet network composed of only full-duplex links and lossless Ethernet MACs and with CoS and flow control to prevent dropping of frames.
lossless transport	In DCB networks, the ability to switch FCoE frames over an Ethernet network without dropping any frames. Lossless transport uses mechanisms such as priority-based flow control and quantized congestion notification to control traffic flows and avoid congestion.
N_Port ID	See *Fibre Channel ID (FCID)*.
N_Port ID virtualizer	Presents itself as an FC or FCoE switch to external devices, but connects to an actual FC or FCoE switch in the other direction to provide the FC-SW services. An N_Port ID virtualizer logs in to the actual FC or FCoE switch in the same way as a normal node device and uses the NPIV mechanism to proxy incoming FLOGIs to FDISCs on the actual FC or FCoE switch. An N_Port ID virtualizes has an FC stack even though it is not an FC switch or an FCF. The acronym *NPV* is commonly used for N_Port ID virtualizer even though the acronym is not defined in the standards.
N_Port ID Virtualization (NPIV)	NPIV enables a physical N_Port to acquire multiple N_Port IDs. Each N_Port ID maps to a different application (such as a virtual machine) or to a different user. This allows you to associate one F_Port with many N_Port IDs and create multiple discrete, secure virtual links over one physical point-to-point connection. NPIV increases resource and bandwidth utilization and allows the implementation of access control, zoning, and port security on a per-application or per-user basis. After an N_Port performs a FLOGI and receives its first N_Port ID, it can request more N_Port IDs by sending FDISC messages. See also *fabric login (FLOGI)*, *fabric discovery (FDISC)*, and *virtual link*.

Table 14: Fibre Channel Terms *(continued)*

Term	Definition
node port (N_Port)	N_Ports can be in two modes: • Fabric N_Port—Node port that is an FC host or storage device end port in a point-to-point link between the device and the F_Port of an FC switch. The point-to-point link can be virtual or physical. • Point-to-point N_Port—Node port that connects to another N_Port. The QFX3500 switch does not support this configuration. N_Ports handle creation, detection, and flow of messages to and from the connected devices.
node worldwide name (NWWN)	WWN that is unique worldwide and is assigned to an FC node. An NWWN is valid for multiple ports that are on that node (this identifies the ports as network interfaces of a particular node).
port mode	Role that the port plays in the FC fabric (endpoint device, FC switch connection to endpoint devices, interswitch link). See also *node port (N_Port)*, *virtual node port (VN_Port)*, *proxy node port (NP_Port)*, *fabric port (F_Port)*, and *virtual fabric port (VF_Port)*.
port worldwide name (PWWN)	WWN that is unique worldwide and is assigned to an FC port.
priority-based flow control (PFC)	Link-level flow control mechanism defined by IEEE 802.1Qbb that allows independent flow control for each class of service (as defined in the 3-bit CoS field of the Ethernet header by IEEE 802.1Q tags) to ensure that no frame loss from congestion occurs in DCB networks. PFC is an enhancement of the Ethernet PAUSE mechanism, but PFC controls classes of flows, whereas Ethernet PAUSE indiscriminately pauses all of the traffic on a link. With PFC, a receiving device can signal a transmitting device to pause transmission based on traffic class. PFC provides application-specific bandwidth reservations so you can ensure that time-critical protocols and applications such as FCoE receive the priority necessary to prevent frame loss. PFC allows the same physical link to carry FCoE traffic and provide lossless service while also carrying loss-tolerant Ethernet traffic. See also *Ethernet PAUSE*.
proxy gateway mode	Connects FCoE initiators to FC switches in a converged Ethernet and Fibre Channel network and acts as an intermediary for these devices. The FCoE-FC gateway represents and acts for the FCoE initiators in transactions from the FCoE initiators destined for an FC switch, including converting FIP and FCoE frames to FC frames. The gateway represents and acts for an FC switch in transactions from the FC switch destined for an FCoE initiator, including converting FC frames to FIP frames and encapsulating FC frames in Ethernet.
proxy node port (NP_Port)	N_Port on the QFX Series that performs proxy functions when it is configured as an FCoE-FC gateway. The NP_Port acts as a proxy for the FCoE device VN_Ports in transactions with the FC switch.

Table 14: Fibre Channel Terms *(continued)*

Term	Definition
quantized congestion notification (QCN)	Mechanism defined by IEEE 802.1Qau that manages network congestion within a Layer 2 domain. When a queue reaches a configured threshold, QCN throttles traffic at the source of the congestion by transmitting messages that propagate back to the source and temporarily stop the source from transmitting. When the queue crosses the threshold that indicates the congestion has dissipated, QCN sends a message to allow the source to resume transmitting frames.
session	Fabric login (FLOGI) or fabric discovery (FDISC) login to the FC SAN fabric. Session does not refer to end-to-end server-to-storage sessions.
server-provided MAC address (SPMA)	MAC address that an ENode assigns to one of its ENode MACs and is not assigned to any other ENode MAC in the same FCoE VLAN. An SPMA can be associated with more than one VN_Port at that ENode MAC. The QFX Series does not support SPMA. See also *ENode MAC* and *fabric-provided MAC address (FPMA)*.
storage area network (SAN)	Network whose primary purpose is the transfer of data between computer systems and storage devices. This term is most commonly used in the context of any network that supports block storage, usually iSCSI, FC, and FCoE networks.
target	System component that receives an I/O command. An FC storage device that receives a request from a server is an example of a target.
VE_Port	Virtual ports created to form a connection (an *interswitch link*) between two FCoE-based FC switches as part of a common FC fabric.
VE2VE (VE_Port to VE_Port)	The *Fibre Channel Backbone - 5 (FC-BB-5) Rev 2.00* specification capability of FCFs to connect to each other as a single FCoE FC SAN.
VN2VF (VN_Port to VF_Port)	The *Fibre Channel Backbone - 5 (FC-BB-5) Rev 2.00* specification capability of an ENode to connect to an FCF or to an FCoE-enabled FC SAN.
VN2VN (VN_Port to VN_Port)	The *Fibre Channel Backbone - 6 (FC-BB-6)* specification capability of an ENode to connect directly over Layer 2 to another ENode without the need of any FC-related services. This capability is most often used in small-scale FCoE SANs.
virtual fabric port (VF_Port)	Data-forwarding component that emulates an F_Port. A VF_Port is dynamically instantiated on successful completion of a FIP FLOGI exchange and connects to one or more VN_Ports. The term *virtual* indicates the use of a non-FC link such as an FCoE link. See also *fabric port (F_Port)*.
virtual link	Logical link connecting two FCoE Link End Points (LEPs) over a lossless Ethernet network, for example, the link between a VF_Port and a VN_Port. The MAC addresses of the two LEPs identifies a virtual link. See also *FCoE link end point (LEP)* and *lossless Ethernet network*.

Table 14: Fibre Channel Terms *(continued)*

Term	Definition
virtual node port (VN_Port)	Data-forwarding component that emulates an N_Port. With FCoE, a VN_Port is dynamically instantiated on successful completion of a FIP FLOGI exchange and connects to one or more VF_Ports. The term *virtual* indicates the use of a non-FC link such as an FCoE link. VN_Port is also used for the virtual N_Ports created in both FC and FCoE when additional NPIV-based logins occur over a previously created N_Port-to-VN_Port or N_Port-to-VF_Port connection. See also *node port (N_Port)*.
well-known address (WKA)	Address identifier used to access a service provided by an FC fabric. The service can be distributed in many elements throughout a fabric, or it can be centralized in one element. A WKA is always accessible, regardless of zoning. An example of a WKA is the *ALL-FCF-MACs* address to which all FCFs listen.
worldwide name (WWN)	64-bit identifier that is similar to a MAC address except that it is not used for forwarding. It uniquely identifies an FC device. The WWN is derived from the IEEE organizationally unique identifier (OUI) and vendor-supplied information. A WWN is unique worldwide.
worldwide node name (WWNN)	See *node worldwide name (NWWN)*.
worldwide port name (WWPN)	See *port worldwide name (PWWN)*.

Related Documentation

- Overview of Fibre Channel on the QFX Series on page 4
- *Understanding QFabric System Terminology*

CHAPTER 3

QFabric Specific

Understanding Fibre Channel Fabrics on the QFabric System

A Fibre Channel (FC) fabric on a QFabric system is a construct that you configure on a QFX3500 Node device when the Node device is in FCoE-FC gateway mode. The FC fabric on a QFabric Node device is not the same as an FC fabric on a storage area network (SAN). The FC fabric on a QFabric Node device is local to that particular node device. We call the FC fabric on a QFabric Node device a *local FC fabric* to differentiate it from an FC fabric on the SAN.

NOTE: **The QFX3600 Node device does not support FC or FCoE features.**

A local FC fabric does not span Node devices and does not span the fabric Interconnect device. Local FC fabrics are entirely contained on a single Node device. A local FC fabric creates associations that connect FCoE devices that have converged network adapters (CNAs) on the Ethernet network to an FC switch or FCoE forwarder (FCF) on the FC network. A local FC fabric consists of:

- A unique fabric name.
- A unique fabric ID.
- One or more FCoE VLAN interfaces that include one or more 10-Gigabit Ethernet interfaces connected to FCoE devices. The FCoE VLANs transport traffic between the FCoE servers and the FCoE-FC gateway. Each FCoE VLAN must carry only FCoE traffic. You cannot mix FCoE traffic and standard Ethernet traffic on the same VLAN.

  The 10-Gigabit Ethernet interfaces that connect to FCoE devices must include a native VLAN to transport FIP traffic because FIP VLAN discovery and notification frames are exchanged as untagged packets.

Each FCoE VLAN interface can present multiple VF_Port interfaces to the FCoE network.

- One or more native FC interfaces. The native FC interfaces transport traffic between the gateway and the FC switch or FCF.

TIP: If the network does not use a dual-rail architecture for redundancy, configure more than one native FC interface for each local FC fabric to create redundant connections between the FCoE devices and the FC network. If one physical link goes down, any sessions it carried can log in again and connect to the FC network on a different interface.

All of the FC and FCoE traffic that belongs to a local FC fabric on a Node device must enter and exit that Node device. This means that the FC switch or FCF and the FCoE devices in the Ethernet network must be connected to the same Node device. The interfaces that connect to the FC switch and the interfaces that connect to the FCoE devices must be included in the local FC fabric. You cannot configure a local FC fabric that spans more than one Node device.

Traffic flows from FC and FCoE devices that are not in the same local FC fabric remain separate and cannot communicate with each other through the FCoE-FC gateway.

NOTE: The QFabric system enforces commit checks to ensure that local FC fabrics and FCoE VLANs on FCoE-FC gateways do not span more than one Node device.

Related Documentation

- Overview of Fibre Channel on the QFX Series on page 4
- Understanding an FCoE-FC Gateway on page 29
- Understanding FCoE-FC Gateway Functions on page 33
- Understanding Interfaces on an FCoE-FC Gateway on page 50

Understanding CoS Fabric Forwarding Class Sets

Fabric forwarding class sets (fabric fc-sets) are similar to the fc-sets (priority groups) you configure on Node devices. The major differences are:

1. Fabric fc-sets group traffic for transport across the QFX3008-I or QFX3600-I Interconnect device (the fabric). Node device fc-sets group traffic on a Node device for transport across that Node device.
2. Fabric fc-sets are global. They apply to the entire fabric. Node device fc-sets apply only to the Node device on which they are configured.
3. You can configure class of service (CoS) scheduling for Node device fc-sets, but you cannot configure CoS for fabric fc-sets.
4. Fabric fc-sets map to Interconnect device fabric output queues statically—you cannot configure the mapping of fabric fc-sets to fabric output queues. All traffic in a fabric fc-set maps to the same output queue.

   Node device fc-sets include forwarding classes that map to Node device output queues, and you can configure the mapping of forwarding classes to output queues (or you can use the default mapping). Because output queues are mapped to forwarding classes, different classes of traffic in a Node device fc-set can be mapped to different output queues.

Node device fc-sets consist of forwarding classes containing traffic that requires similar CoS treatment. (Forwarding classes are default forwarding classes or user-defined forwarding classes.) You can configure CoS for each fc-set to determine how the traffic of its forwarding classes is scheduled on a Node device.

When traffic exits a Node device interface and enters an Interconnect device fabric interface, the Interconnect device uses the same forwarding classes to group traffic. The forwarding classes are mapped to global fabric fc-sets for transport across the fabric. Like fc-sets on a Node device, fabric fc-sets also contain traffic that requires similar CoS treatment. Unlike fc-sets on a Node device, you cannot configure CoS on fabric fc-sets.

Fabric fc-sets reside on the Interconnect device and are global to the QFabric system. Fabric fc-sets apply to all traffic that traverses the fabric. The mapping of forwarding classes to fabric fc-sets is global and applies to all forwarding classes with traffic that traverses the fabric from all connected Node devices. You can change the mapping of forwarding classes to fabric fc-sets. All mapping changes you make are global. For example, if you change the fabric fc-set to forwarding class mapping of the default best-effort forwarding class, then every Node device's best-effort forwarding class traffic that traverses the fabric is mapped to that fabric fc-set.

This topic describes:

- Default Fabric Forwarding Class Sets on page 146
- Fabric Forwarding Class Set Configuration and Implementation on page 149
- Fabric Forwarding Class Set Scheduling (CoS) on page 151
- Support for Flow Control and Lossless Transport Across the Fabric on page 153

- Viewing Fabric Forwarding Class Set Information on page 155
- Summary of Fabric Forwarding Class Set and Node Device Forwarding Class Set Differences on page 157

Default Fabric Forwarding Class Sets

Interconnect devices have 12 default fabric fc-sets, including five visible default fabric fc-sets, four for unicast traffic and one for multidestination (multicast, broadcast, and destination lookup failure) traffic.

There are also seven hidden default fabric fc-sets. There are three hidden default fabric fc-sets for multidestination traffic that you can use if you want to map different multidestination forwarding classes to different multidestination fabric fc-sets. There are four hidden default fabric fc-sets for lossless traffic that you can use to map different lossless forwarding classes (priorities) to different lossless fabric fc-sets. Table 15 on page 146 shows the default fabric fc-sets:

Table 15: Default Fabric Forwarding Class Sets

Fabric Forwarding Class Set Name	Characteristics
fabric_fcset_be	Transports best-effort unicast traffic across the fabric.
fabric_fcset_strict_high	Transports unicast traffic that has been configured with **strict-high** priority and in the **network-control** forwarding class across the fabric. This fabric fc-set receives as much bandwidth across the fabric as it needs to service the traffic in the group up to the entire fabric interface bandwidth. For this reason, exercise caution when mapping traffic to this fabric fc-set to avoid starving other traffic.
fabric_fcset_noloss1	Transports unicast traffic in the default **fcoe** forwarding class across the fabric.
fabric_fcset_noloss2	Transports unicast traffic in the default **no-loss** forwarding class across the fabric.
fabric_fcset_noloss3	(Hidden) No traffic is assigned by default to this fabric fc-set. Unless traffic is mapped to this fabric fc-set, this fabric fc-set remains hidden. This fabric fc-set is valid only for lossless forwarding classes.
fabric_fcset_noloss4	(Hidden) No traffic is assigned by default to this fabric fc-set. Unless traffic is mapped to this fabric fc-set, this fabric fc-set remains hidden. This fabric fc-set is valid only for lossless forwarding classes.
fabric_fcset_noloss5	(Hidden) No traffic is assigned by default to this fabric fc-set. Unless traffic is mapped to this fabric fc-set, this fabric fc-set remains hidden. This fabric fc-set is valid only for lossless forwarding classes.

Table 15: Default Fabric Forwarding Class Sets *(continued)*

Fabric Forwarding Class Set Name	Characteristics
fabric_fcset_noloss6	(Hidden) No traffic is assigned by default to this fabric fc-set. Unless traffic is mapped to this fabric fc-set, this fabric fc-set remains hidden. This fabric fc-set is valid only for lossless forwarding classes.
fabric_fcset_multicast1	Transports multidestination traffic in the **mcast** forwarding class across the fabric. This fabric fc-set is valid only for multidestination forwarding classes.
fabric_fcset_multicast2	(Hidden) No traffic is assigned by default to this fabric fc-set. Unless traffic is mapped to this fabric fc-set, this fabric fc-set remains hidden. This fabric fc-set is valid only for multidestination forwarding classes.
fabric_fcset_multicast3	(Hidden) No traffic is assigned by default to this fabric fc-set. Unless traffic is mapped to this fabric fc-set, this fabric fc-set remains hidden. This fabric fc-set is valid only for multidestination forwarding classes.
fabric_fcset_multicast4	(Hidden) No traffic is assigned by default to this fabric fc-set. Unless traffic is mapped to this fabric fc-set, this fabric fc-set remains hidden. This fabric fc-set is valid only for multidestination forwarding classes.

The five default forwarding classes (**best-effort**, **fcoe**, **no-loss**, **network-control**, and **mcast**) are mapped to the fabric fc-sets by default as shown in Table 16 on page 147.

Table 16: Default Forwarding Class to Fabric Forwarding Class Set Mapping

Forwarding Class	Fabric Forwarding Class Set	Fabric Output Queue	Maximum MTU Supported for Lossless Operation
best-effort	fabric_fcset_be	0	NA
network-control	fabric_fcset_strict_high	7	NA
fcoe	fabric_fcset_noloss1	1	9K
no-loss	fabric_fcset_noloss2	2	9K
mcast	fabric_fcset_multicast1	8	NA
No forwarding classes are mapped by default to this hidden fabric fc-set.	fabric_fcset_noloss3	3	9k

Table 16: Default Forwarding Class to Fabric Forwarding Class Set Mapping *(continued)*

Forwarding Class	Fabric Forwarding Class Set	Fabric Output Queue	Maximum MTU Supported for Lossless Operation
No forwarding classes are mapped by default to this hidden fabric fc-set.	fabric_fcset_noloss4	4	9k
No forwarding classes are mapped by default to this hidden fabric fc-set.	fabric_fcset_noloss5	5	9k
No forwarding classes are mapped by default to this hidden fabric fc-set.	fabric_fcset_noloss6	6	9k
No forwarding classes are mapped by default to this hidden fabric fc-set.	fabric_fcset_multicast2	9	NA
No forwarding classes are mapped by default to this hidden fabric fc-set.	fabric_fcset_multicast3	10	NA
No forwarding classes are mapped by default to this hidden fabric fc-set.	fabric_fcset_multicast4	11	NA

The maximum fiber cable length between the QFabric system Node device and the QFabric system Interconnect device is 150 meters.

TIP: If you explicitly configure lossless forwarding classes, we recommend that you map each user-configured lossless forwarding class to an unused fabric fc-set (fabric_fcset_noloss3 through fabric_fcset_noloss6) on a one-to-one basis: one lossless forwarding class mapped to one lossless fabric fc-set.

The reason for one-to-one mapping is to avoid fate sharing of lossless flows. Because each fabric fc-set is mapped statically to an output queue, when you map more than one forwarding class to a fabric fc-set, all of the traffic in all of the forwarding classes that belong to the fabric fc-set uses the same output queue. If that output queue becomes congested due to congestion caused by one of the flows, the other flows are also affected. (They share fate because the flow that congests the output queue affects flows that are not experiencing congestion.)

However, it is important to understand that fabric_fcset_noloss1 and fabric_fcset_noloss2 have a scheduling weight of 35, while the other fabric fc-sets have a scheduling weight of 1. The scheduling weights mean that fabric_fcset_noloss1 and fabric_fcset_noloss2 receive most of the bandwidth available to lossless fabric fc-sets if the amount of traffic on fabric_fcset_noloss1 and fabric_fcset_noloss2 requires the bandwidth.

If you believe that the traffic on fabric_fcset_noloss1 and fabric_fcset_noloss2 will consume most of that bandwidth, then you should place all lossless traffic on fabric_fcset_noloss1 and fabric_fcset_noloss2. If you believe that the traffic on fabric_fcset_noloss1 and fabric_fcset_noloss2 will <emphasis>not</emphasis> consume most of that bandwidth, then you can map lossless forwarding classes in a one-to-one manner to lossless fabric fc-sets to avoid fate sharing.

If you want to map different multidestination forwarding classes to different multidestination fabric fc-sets, use one or more of the hidden multidestination fabric fc-sets.

NOTE: The global mapping of forwarding classes to fabric fc-sets is independent of the mapping of forwarding classes to Node device fc-sets. Global mapping of forwarding classes to fabric fc-sets occurs only on the Interconnect device. The Node device mapping of forwarding classes to fc-sets does not affect the global mapping of forwarding classes to fabric fc-sets on the Interconnect device, and vice versa.

When you define new forwarding classes on a Node device, you explicitly map those forwarding classes to Node device fc-sets. However, new (user-created) forwarding classes are mapped by default to fabric fc-sets. (You can override the default mapping if you want to configure the forwarding class to fabric fc-set mapping explicitly, as described in the next section.)

By default:

- All best-effort traffic forwarding classes that you create are mapped to the **fabric_fcset_be** fabric fc-set.
- All lossless traffic forwarding classes that you create are mapped to the **fabric_fcset_noloss1** or **fabric_fcset_noloss2** fabric fc-set.
- All multidestination traffic forwarding classes that you create are mapped to the **fabric_fcset_multicast1** fabric fc-set.
- All **strict-high** priority traffic and **network-control** forwarding classes that you create are mapped to the **fabric_fcset_strict_high** fabric fc-set.

Fabric Forwarding Class Set Configuration and Implementation

You can map forwarding classes to fabric fc-sets, but no other attributes of fabric fc-sets are user-configurable, including CoS. This section describes:

- Mapping Forwarding Classes to Fabric Forwarding Class Sets on page 150
- Fabric Forwarding Class Set Implementation on page 150

Mapping Forwarding Classes to Fabric Forwarding Class Sets

If you do not want to use the default mapping of forwarding classes to fabric fc-sets, you can map forwarding classes to fabric fc-sets the same way as you map forwarding classes to Node device fc-sets. To do this, use exactly the same statement that you use to map forwarding classes to fc-sets, but instead of specifying a Node device fc-set name, specify a fabric fc-set name.

NOTE: The global mapping of forwarding classes to fabric fc-sets does not affect the mapping of forwarding classes to Node device fc-sets. The global forwarding class mapping to fabric fc-sets pertains to the traffic only when it enters, traverses, and exits the fabric. The forwarding class mapping to fc-sets on a Node device is valid within that Node device.

Mapping forwarding classes to fabric fc-sets does not affect the scheduling configuration of the forwarding classes or fc-sets on Node devices. Fabric fc-set scheduling (which is not user-configurable) pertains to traffic only when it enters, traverses, and exits the Interconnect device fabric.

If you change the mapping of a forwarding class to a fabric fc-set, the new mapping is global and applies to all traffic in that forwarding class, regardless of which Node device forwards the traffic to the Interconnect device.

- To assign one or more forwarding classes to a fabric fc-set:

  ```
  [edit class-of-service]
  user@switch# set forwarding-class-sets fabric-forwarding-class-set-name class
  forwarding-class-name
  ```

  For example, to map a user-defined forwarding class named **best-effort-2** to the fabric fc-set **fabric_fcset_be**:

  ```
  [edit class-of-service]
  user@switch# set forwarding-class-sets fabric_fcset_be class best-effort-2
  ```

NOTE: Because fabric fc-set configuration is global, in this example all forwarding classes with the name **best-effort-2** on all of the Node devices attached to the fabric use the **fabric_fcset_be** fabric fc-set to transport traffic across the fabric.

Fabric Forwarding Class Set Implementation

The following rules apply to fabric fc-sets:

- You cannot create new fabric fc-sets. Only the twelve default fabric fc-sets are available.
- You cannot delete a default fabric fc-set.
- You cannot attach a fabric fc-set to a Node device interface. Fabric fc-sets are used only on the Interconnect device fabric, not on Node devices.

- You can map only multidestination forwarding classes to multidestination fabric fc-sets.
- You cannot map multidestination forwarding classes to unicast fabric fc-sets.
- You cannot map unicast forwarding classes to multidestination fabric fc-sets.
- You cannot configure CoS for fabric fc-sets. (However, default CoS scheduling properties are applied to traffic on the fabric, and the fabric interfaces use link layer flow control (LLFC) for flow control.)

Fabric Forwarding Class Set Scheduling (CoS)

Although fabric fc-set CoS is not user-configurable, CoS is applied to traffic on the fabric. (In addition, fabric interfaces use LLFC to ensure lossless transport for lossless traffic flows.) This section describes how the fabric applies CoS scheduling to traffic:

Class Groups for Fabric Forwarding Class Sets

To transport traffic across the fabric, the QFabric system organizes the fabric fc-sets into three classes called *class groups*. The three class groups are:

- **Strict-high priority**—All traffic in the fabric fc-set **fabric_fcset_strict_high**. This class group includes the traffic in **strict-high** priority and **network-control** forwarding classes and in any forwarding classes you create on a Node device that consist of **strict-high** priority or **network-control** forwarding class traffic.
- **Unicast**—All traffic in the fabric fc-sets **fabric_fcset_be**, **fabric_fcset_noloss1**, and **fabric_fcset_noloss2**. This class group includes the traffic in the **best-effort**, **fcoe**, and **no-loss** forwarding classes and in any forwarding classes you create on a Node device that consist of best-effort or lossless traffic. If you use any of the hidden no loss fabric fc-sets (**fabric_fcset_noloss3**, **fabric_fcset_noloss4**, **fabric_fcset_noloss5**, or **fabric_fcset_noloss6**), that traffic is part of this class group.
- **Multidestination**—All traffic in the fabric fc-set **fabric_fcset_multicast1**. This class group includes the traffic in the **mcast** forwarding class and in any forwarding classes you create on a Node device that consist of multidestination traffic. If you use any of the hidden multidestination fabric fc-sets (**fabric_fcset_multicast2**, **fabric_fcset_multicast3**, or **fabric_fcset_multicast4**), that traffic is also classified as part of this class group.

Class Group Scheduling

You cannot configure CoS for class groups or for fabric fc-sets (that is, you cannot attach a traffic control profile to a fabric fc-set—you attach traffic control profiles to Node device fc-sets to apply scheduling to the traffic that belongs to the Node device fc-set). By default, the fabric uses weighted round-robin (WRR) scheduling in which each class group receives a portion of the total available fabric bandwidth based on its type of traffic, as shown in Table 17 on page 152:

Table 17: Class Group Scheduling Properties and Membership

Class Group	Fabric fc-sets	Forwarding Classes (Default Mapping)	Class Group Scheduling Properties (Weight)
Strict-high priority	**fabric_fcset_strict_high**	• All **strict-high** priority forwarding classes • **network-control**	Traffic in the strict-high priority class group is served first. This class group receives all of the bandwidth it needs to empty its queues and therefore can starve other types of traffic during periods of high-volume strict-high priority traffic. Plan carefully and use caution when determining how much traffic to configure as strict-high priority traffic.
Unicast	• **fabric_fcset_be** • **fabric_fcset_noloss1** • **fabric_fcset_noloss2** Includes the hidden lossless fabric fc-sets if used: • **fabric_fcset_noloss3** • **fabric_fcset_noloss4** • **fabric_fcset_noloss5** • **fabric_fcset_noloss6**	• **best-effort** • **fcoe** • **no-loss** NOTE: No forwarding classes are mapped to the hidden lossless fabric_fcsets by default.	Traffic in the unicast class group receives an 80% weight in the WRR calculations. After the strict-high priority class group has been served, the unicast class group receives 80% of the remaining fabric bandwidth. (If more bandwidth is available, the unicast class group can use more bandwidth.)
Multidestination	**fabric_fcset_multicast1** Includes the hidden multidestination fabric fc-sets if used: • **fabric_fcset_multicast2** • **fabric_fcset_multicast3** • **fabric_fcset_multicast4**	• **mcast** NOTE: No forwarding classes are mapped to the hidden multidestination fabric_fcsets by default.	Traffic in the multidestination class group receives a 20% weight in the WRR calculations. After the strict-high priority class group has been served, the multidestination class group receives 20% of the remaining fabric bandwidth. (If more bandwidth is available, the multidestination class group can use more bandwidth.)

The fabric fc-sets within each class group are weighted equally and receive bandwidth using round-robin scheduling. For example:

- If the unicast class group has three member fabric fc-sets, **fabric_fcset_be**, **fabric_fcset_noloss1**, and **fabric_fcset_noloss2**, then each of the three fabric fc-sets receives one-third of the bandwidth available to the unicast class group.
- If the multidestination class group has one member fc-set, **fabric_fcset_multicast1**, then that fc-set receives all of the bandwidth available to the multidestination class group.
- If the multidestination class group has two member fc-sets, **fabric_fcset_multicast1** and **fabric_fcset_multicast2**, then each of the two fabric fc-sets receives one-half of the bandwidth available to the multidestination class group.

QFabric System CoS

When traffic enters and exits the same Node device, CoS works the same as it works on a standalone QFX3500 switch.

However, when traffic enters a Node device, crosses the Interconnect device, and then exits a different Node device, CoS is applied differently:

1. Traffic entering the ingress Node device receives the CoS configured at the Node ingress (packet classification, congestion notification profile for PFC).
2. When traffic goes from the ingress Node device to the Interconnect device, the fabric fc-set CoS is applied as described in the discussion of fabric forwarding class set scheduling.
3. When traffic goes from the Interconnect device to the egress Node device, the egress Node device applies CoS at the egress port (egress queue scheduling, WRED, IEEE 802.1p or DSCP code-point rewrite).

Support for Flow Control and Lossless Transport Across the Fabric

The Interconnect device incorporates flow control mechanisms to support lossless transport during periods of congestion on the fabric. To support the priority-based flow control (PFC) feature on the Node devices, the fabric interfaces use LLFC to support lossless transport for up to six IEEE 802.1p priorities when the following two configuration constraints are met:

1. The IEEE 802.1p priority used for the traffic that requires lossless transport is mapped to a lossless forwarding class on the Node devices.
2. The lossless forwarding class must be mapped to a lossless fabric fc-set on the Interconnect device (**fabric_fcset_noloss1**, **fabric_fcset_noloss2**, **fabric_fcset_noloss3**, **fabric_fcset_noloss4**, **fabric_fcset_noloss5**, or **fabric_fcset_noloss6**).

When traffic meets the two configuration constraints, the fabric propagates the back pressure from the egress Node device across the fabric to the ingress Node device during periods of congestion. However, to achieve end-to-end lossless transport across the switch, you must also configure a congestion notification profile to enable PFC on the Node device ingress ports.

For all other combinations of IEEE 802.1p priority to forwarding class mapping and all other combinations of forwarding class to fabric fc-set mapping, the congestion control mechanism is normal packet drop. For example:

- **Case 1**—If the IEEE 802.1p priority **5** is mapped to the lossless **fcoe** forwarding class, and the **fcoe** forwarding class is mapped to the **fabric_fcset_noloss1** fabric fc-set, then the congestion control mechanism is PFC.
- **Case 2**—If the IEEE 802.1p priority **5** is mapped to the lossless **fcoe** forwarding class, and the **fcoe** forwarding class is mapped to the **fabric_fcset_be** fabric fc-set, then the congestion control mechanism is packet drop.

- **Case 3**—If the IEEE 802.1p priority **5** is mapped to the lossless **no-loss** forwarding class, and the **no-loss** forwarding class is mapped to the **fabric_fcset_noloss2** fabric fc-set, then the congestion control mechanism is PFC.
- **Case 4**—If the IEEE 802.1p priority **5** is mapped to the lossless **no-loss** forwarding class, and the **no-loss** forwarding class is mapped to the **fabric_fcset_be** fabric fc-set, then the congestion control mechanism is packet drop.
- **Case 5**—If the IEEE 802.1p priority **5** is mapped to the **best-effort** forwarding class, and the **best-effort** forwarding class is mapped to the **fabric_fcset_be** fabric fc-set, then the congestion control mechanism is packet drop.
- **Case 6**—If the IEEE 802.1p priority **5** is mapped to the **best-effort** forwarding class, and the **best-effort** forwarding class is mapped to the **fabric_fcset_noloss1** fabric fc-set, then the congestion control mechanism is packet drop.

NOTE: Lossless transport across the fabric also must meet the following two conditions:

1. The maximum cable length between the Node device and the Interconnect device is a 150 meters of fiber cable.
2. The maximum frame size is 9216 bytes.

If the MTU is 9216 KB, in some cases the QFabric system supports only five lossless forwarding classes instead of six lossless forwarding classes because of headroom buffer limitations.

The number of IEEE 802.1p priorities (forwarding classes) the QFabric system can support for lossless transport across the Interconnect device fabric depends on several factors:

- Approximate fiber cable length—The longer the fiber cable that connects Node device fabric (FTE) ports to the Interconnect device fabric ports, the more data the connected ports need to buffer when a pause is asserted. (The longer the fiber cable, the more frames are traversing the cable when a pause is asserted. Each port must be able to store all of the “in transit” frames in the buffer to preserve lossless behavior and avoid dropping frames.)
- MTU size—The larger the maximum frame sizes the buffer must hold, the fewer frames the buffer can hold. The larger the MTU size, the more buffer space each frame consumes.
- Total number of Node device fabric ports connected to the Interconnect device—The higher the number of connected fabric ports, the more headroom buffer space the Node device needs on those fabric ports to support the lossless flows that traverse the Interconnect device. Because more buffer space is used on the Node device fabric ports, less buffer space is available for the Node device access ports, and a lower total number of lossless flows are supported.

The QFabric system supports six lossless priorities (forwarding classes) under most conditions. The priority group headroom that remains after allocating headroom to lossless flows is sufficient to support best-effort and multidestination traffic.

Table 18 on page 155 shows how many lossless priorities the QFabric system supports under different conditions (fiber cable lengths and MTUs) in cases when the QFabric system supports fewer than six lossless priorities. The number of lossless priorities is the same regardless of how many Node device FTE ports are connected to the Interconnect device. However, the higher the number of FTE ports connected to the Interconnect device, the lower the number of total lossless flows supported. In all cases that are not shown in Table 18 on page 155, the QFabric system supports six lossless priorities.

NOTE: The system does not perform a configuration commit check that compares available system resources with the number of lossless forwarding classes configured. If you commit a configuration with more lossless forwarding classes than the system resources can support, frames in lossless forwarding classes might be dropped.

Table 18: Lossless Priority (Forwarding Class) Support for QFX3500 and QFX3600 Node Devices When Fewer than Six Lossless Priorities Are Supported

MTU in Bytes	Fiber Cable Length in Meters (Approximate)	Maximum Number of Lossless Priorities (Forwarding Classes) on the Node Device
9216 (9K)	100	5
9216 (9K)	150	5

NOTE: The total number of lossless flows decreases as resource consumption increases. For a Node device, the higher the number of FTE ports connected to the Interconnect device, the larger the MTU, and the longer the fiber cable length, the fewer total lossless flows the QFabric system can support.

Viewing Fabric Forwarding Class Set Information

You can display information about fabric fc-sets using the same CLI command you use to display information about Node device fc-sets:

```
user@switch> show class-of-service forwarding-class-set
Forwarding class set: fabric_fcset_be, Type: fabric-type, Forwarding class set
index: 1
  Forwarding class                       Index
  best-effort                              0

Forwarding class set: fabric_fcset_mcast1, Type: fabric-type, Forwarding class
set index: 5
  Forwarding class                       Index
```

```
  mcast                                   8

Forwarding class set: fabric_fcset_mcast2, Type: fabric-type, Forwarding class
set index: 6

Forwarding class set: fabric_fcset_mcast3, Type: fabric-type, Forwarding class
set index: 7

Forwarding class set: fabric_fcset_mcast4, Type: fabric-type, Forwarding class
set index: 8

Forwarding class set: fabric_fcset_noloss1, Type: fabric-type, Forwarding class
set index: 2
  Forwarding class                      Index
  fcoe                                    1

Forwarding class set: fabric_fcset_noloss2, Type: fabric-type, Forwarding class
set index: 3
  Forwarding class                      Index
  no-loss                                 2

Forwarding class set: fabric_fcset_noloss3, Type: fabric-type, Forwarding class
set index: 9

Forwarding class set: fabric_fcset_noloss4, Type: fabric-type, Forwarding class
set index: 10

Forwarding class set: fabric_fcset_noloss5, Type: fabric-type, Forwarding class
set index: 11

Forwarding class set: fabric_fcset_noloss6, Type: fabric-type, Forwarding class
set index: 12

Forwarding class set: fabric_fcset_strict_high, Type: fabric-type, Forwarding
class set index: 4
  Forwarding class                      Index
  network-control                         3
```

Table 19 on page 156 describes the meaning of the **show class-of-service forwarding-class-set** output fields when you display fabric fc-set information.

Table 19: show class-of-service forwarding-class-set Command Output Fields

Field Name	Field Description
Forwarding class set	Name of the fabric forwarding class set.
Type	Type of forwarding class set: • Fabric-type—Fabric fc-set • Normal-type—Node device fc-set
Forwarding class set index	Index of this forwarding class set.
Forwarding class	Name of a forwarding class.
Index	Index of the forwarding class.

Summary of Fabric Forwarding Class Set and Node Device Forwarding Class Set Differences

Table 20 on page 157 summarizes the differences between fabric fc-sets and Node device fc-sets:

Table 20: Summary of Differences Between Fabric fc-sets and Node Device fc-sets

Characteristic	Fabric fc-set	Node device fc-set
Location	QFX3008-I or QFX3600-I Interconnect device (the fabric).	QFabric system Node device.
Global or Node-device specific	Global, valid for the entire fabric.	Local to the Node device on which the fc-set is configured.
Ability to create (define) a new fc-set	No. Use the 12 default fabric fc-sets provided.	Yes.
Ability to configure CoS	Default CoS settings only. CoS is not user-configurable.	User-configurable using traffic control profiles.
Ability to map forwarding classes to an fc-set	Yes. Mapping is global and applies to all forwarding classes across Interconnect device fabric (traffic from all connected Node devices).	Yes. Mapping is local to a Node device and applies only to the forwarding classes on the Node device.

Related Documentation

- *Understanding CoS Forwarding Class Sets (Priority Groups)*
- *Understanding CoS Scheduling on QFabric System Node Device Fabric (fte) Ports*
- *Understanding Default CoS Scheduling on QFabric System Interconnect Devices (Junos OS Release 13.1 and Later Releases)*
- *Defining CoS Forwarding Class Sets*
- *Example: Configuring Forwarding Class Sets*
- *show class-of-service forwarding-class-set*

Understanding FCoE LAGs

A Fibre Channel over Ethernet (FCoE) link aggregation group (LAG) is a special LAG that enables you to transport FCoE traffic and regular Ethernet traffic across the same link aggregation bundle. Fibre Channel (FC) storage area network (SAN) switches require a point-to-point connection (or a virtual point-to-point connection) to FCoE devices. This requirement means that communication between an FCoE device and a QFabric system Node device must use the same physical link in a LAG to maintain the virtual point-to-point connection.

However, a standard LAG can use any member link for any particular transmission, so a standard LAG cannot guarantee that the same link is used for requests and responses between an FCoE device and a QFabric system Node device. Using different LAG member links for communication breaks the virtual point-to-point connection, which results in dropped FCoE traffic.

Unlike a standard LAG, an FCoE LAG always uses the same member link to transmit FCoE traffic between an FCoE device and a QFabric system Node device. However, regular Ethernet traffic (traffic is that is not FCoE traffic) on the LAG is distributed across member interfaces in the same way as on a standard LAG. The special treatment of FCoE traffic does not affect the way regular Ethernet traffic behaves on the LAG. FCoE traffic is treated properly in terms of maintaining a virtual point-to-point link, and regular Ethernet traffic enjoys the usual LAG benefits of load balancing and link redundancy.

NOTE: Configuring a LAG as an FCoE LAG does not provide link redundancy for FCoE traffic, and does not load balance FCoE traffic.

A LAG interface can be a member of both an FCoE VLAN and a regular Ethernet VLAN. An FCoE LAG allows FCoE and standard Ethernet traffic to coexist on the same LAG, and treats both types of traffic properly.

On QFabric systems, all of the member links of an FCoE LAG must belong to one Node group. The member links of an FCoE LAG cannot belong to different Node groups.

Like a standard LAG, an FCoE LAG can have up to 32 member interfaces. FCoE devices are usually servers with CNAs connected to a switch that performs FIP snooping, such as an FCoE transit switch or an FCoE-FC gateway switch that performs FIP snooping.

Why a Standard LAG Does Not Work for FCoE Traffic

Each physical link that carries FCoE traffic connects to a CNA port on an FCoE device. The connection that the FIP process creates between the CNA and the FC SAN switch emulates a point-to-point connection between that CNA and the SAN switch through the QFabric system Node device. If a connection to an FCoE device is not on a point-to-point link, communication from the FC SAN switch to the FCoE device CNA might not reach the CNA.

In a LAG, two (or more) physical links connect to the same device. Standard LAGs use a hashing algorithm to determine which physical LAG link to use for each transmission. Because the hashing algorithm might choose any LAG link for a given transmission, there is no way a standard LAG can guarantee that a response from the FC SAN will use the same LAG link on a Node device as the request from the CNA.

To ensure that communication between the CNA and the FC SAN is successful, communication from the SAN to the CNA must use the same physical link. If the FCoE CNA sends a request to the FC SAN, the response from the FC SAN must come on the same link the FCoE device CNA used to send the request. For example, if a if a request from the CNA goes out on Node device LAG member interface RSNG1:xe-0/0/20, then the response from the FC SAN must be received on interface RSNG1:xe-0/0/20.

If the FC SAN switch response to the FCoE CNA uses a different physical link on the Node device LAG, the response arrives at a different CNA port than the CNA port on which the request was sent. This breaks the virtual point-to-point link and the SAN switch response does not reach the correct requestor, so the response is lost. This is why a standard LAG does not work for FCoE traffic.

How an FCoE LAG Works

For FIP and FCoE transactions with the FC SAN to work properly, a LAG for FIP and FCoE traffic must allow the FC SAN switch to respond to the FCoE CNA device on the same link that the CNA used to communicate with the FC SAN switch.

To accomplish this, an FCoE LAG selects the member interface that the CNA used to communicate with the FC SAN switch as the link for the SAN switch response to the CNA. This preserves the virtual point-to-point link across the LAG and ensures that traffic from the FC SAN reaches the correct CNA port.

In a standard LAG, other devices learn the MAC address of the LAG interface, not the MAC address of the physical member interface that actually carries the traffic. However, for FCoE communication, other devices need to learn and use the VN_Port MAC address that the SAN switch assigns to the virtual node port (VN_Port) on the FCoE device's CNA. The VN_Port MAC address uniquely identifies the CNA port used for FCoE transmission. (The VN_Port MAC address is based on the Fibre Channel ID and the FC-MAP value, which the FC SAN switch provides to the FCoE CNA as a unique port identifier.)

In an FCoE LAG, the Node device performs FIP snooping to learn the VN_Port MAC address of the CNA (in addition to other information). The Node device assigns the VN_Port MAC address to the particular interface that was used to connect to the CNA. For FCoE traffic, this replaces the normal LAG hashing logic, so instead of using an arbitrary LAG interface on the Node device for FCoE communication between the SAN switch and the CNA, an FCoE LAG uses the same physical LAG link for all FCoE transactions based on the VN_Port MAC address.

VLAN discovery traffic is untagged, so it must use a native VLAN. When you configure an FCoE LAG, VLAN discovery traffic on a native VLAN in the LAG also automatically uses the same physical link, preserving the virtual point-to-point link.

For multicast packets such as multicast discovery advertisements (MDAs), the advertisement is forwarded on all member links of the FCoE LAG. This ensures that multicast advertisements reach all of the FCoE devices attached to FCoE LAG member interfaces.

Behavior on FCoE LAG Link Failure

If an FCoE LAG link goes down, FCoE traffic and regular Ethernet traffic are treated differently.

If an FCoE LAG link goes down, the FCoE sessions on that link also go down. The Node device cannot simply move a session to another LAG link because that breaks the virtual point-to-point link. FCoE LAGs do not provide link redundancy for FCoE traffic.

As on a normal LAG, an FCoE LAG provides link redundancy for regular Ethernet traffic. Regular Ethernet sessions on the down FCoE LAG link are moved to other member links of the FCoE LAG (assuming that other member links are up).

FIP Snooping Session Scaling on QFabric System Node Devices

When the switch is on the FCoE access edge, you must enable FIP snooping on the FCoE VLAN to provide secure access when connecting to the FC SAN. (You can also enable FIP snooping on FCoE VLANs on switches that are not at the access edge if you want to collect FIP snooping statistics on the switch or if you are not confident that the edge switch is properly snooping traffic.)

FIP snooping VLANs support scaling up to 2,500 sessions by default, which is called enhanced FIP snooping scaling mode. Software releases before Junos OS Release 12.3 limited VN2VF_Port FIP snooping session scaling to 376 sessions on untrusted interfaces and untrusted FC fabrics, but scaled to 2,500 sessions on trusted interfaces and trusted FC fabrics. Starting with Junos OS Release 12.3, by default, all VN2VF_Port FIP snooping VLANs used enhanced FIP snooping scaling (2,500 sessions) for both trusted and untrusted interfaces and FC fabrics. The old limit of 376 sessions for untrusted interfaces and untrusted FC fabrics was deprecated and could not be configured.

The FCoE LAG feature introduces the ability to disable FIP snooping session scaling so that only 376 sessions are supported instead of the default 2,500 sessions. The reason for reintroducing FIP snooping session scaling limits is that when a Node device is configured as an FCoE-FC gateway that has one or more untrusted gateway Fibre Channel fabric (fc-fabric), placing FCoE traffic in a LAG forces the TCAM to store additional session data to ensure that the virtual point-to-point link between the FCoE device and the FC SAN is maintained. This case is described later in this document.

FCoE LAG Configuration on an FCoE Transit Switch

To create an FCoE LAG on an FCoE transit switch, you include the **fcoe-lag** option in the **[edit interfaces *interface-name* aggregated-ether-options]** hierarchy.

In addition to creating the FCoE LAG, you also need to:

- Add interfaces to the FCoE LAG.
- Configure at least one dedicated VLAN for FCoE traffic (an FCoE VLAN).
- Configure a native VLAN to carry untagged FIP traffic.
- Configure the FCoE LAG interfaces as a member of both the FCoE VLAN and the native VLAN.
- Enable FIP snooping on the FCoE VLAN.

FCoE LAG Configuration and FIP Snooping Scaling on an FCoE-FC Gateway

There are differences in the way you configure an FCoE LAG on an FCoE-FC gateway compared to configuring an FCoE LAG on an FCoE transit switch.

Configuring an FCoE LAG on an FCoE-FC Gateway

To create an FCoE LAG on an FCoE-FC gateway, you include the **fcoe-lag** option in the **[edit interfaces *interface-name* aggregated-ether-options]** hierarchy.

In addition to creating the FCoE LAG, you also need to:

- Add interfaces to the FCoE LAG.
- Configure at least one dedicated VLAN for FCoE traffic (an FCoE VLAN).
- Configure a native VLAN to carry untagged FIP traffic.
- Configure the FCoE LAG interfaces as a member of both the FCoE VLAN and the native VLAN.
- Configure an FCoE VLAN interface (a Layer 3 routed VLAN interface that is configured as a virtual F_Port) for the FCoE traffic. This enables the FCoE VLAN (and the member FCoE LAG interfaces) to interface with the with the native Fibre Channel ports in the FCoE-FC gateway switch Fibre Channel fabric (fc-fabric).
- Add the FCoE VLAN interface to the fc-fabric.
- Enable FIP snooping on the FCoE VLAN.
- Configure FIP snooping session scaling as described in the next section. The FIP snooping scaling mode depends on whether the fc-fabric is trusted or untrusted.

FIP Snooping Session Scaling on an FCoE-FC Gateway

FIP snooping session scaling on an FCoE-FC gateway depends on whether or not the gateway has an untrusted fc-fabric:

- If the FCoE-FC gateway fc-fabric is FCoE trusted, then you can use enhanced FIP snooping scaling (2,500 sessions), and you do not have to do any additional configuration even if two or more FCFs in an FCoE VLAN have the same FC-MAP value.
- If the FCoE-FC gateway fc-fabric is FCoE untrusted, then you must disable enhanced FIP snooping scaling (reduce the number of supported sessions to 376 sessions) by including the **no-fip-snooping-scaling** statement in the **[edit fc-options]** hierarchy.

NOTE: On an FCoE-FC gateway, disabling enhanced FIP snooping scaling is global.

Gateway fc-fabrics are untrusted by default. FCoE-FC gateways do not support FCoE LAGs on untrusted fc-fabrics when enhanced FIP snooping scaling is enabled.

Summary of FCoE LAG and FIP Snooping Scaling on an FCoE-FC Gateway

Table 21 on page 162 summarizes FCoE LAG and FIP snooping scaling on an FCoE-FC gateway.

Table 21: Summary of FCoE LAG and FIP Snooping Scaling (FCoE-FC Gateway)

FCoE Fabric Trusted or Untrusted	FCoE LAG Configured	FIP Snooping Session Scaling	Configuration Notes
Trusted	Yes (**fcoe-lag** option included in the **[edit interfaces *interface-name* aggregated-ether-options]** hierarchy)	2,500 sessions (enhanced FIP snooping scaling)	Configure the fc-fabric as an FCoE trusted fabric by including the **fcoe-trusted** option in the **[edit fc-fabrics *fc-fabric-name* protocols fip fcoe-trusted]** hierarchy.
Untrusted	Yes (**fcoe-lag** option included in the **[edit interfaces *interface-name* aggregated-ether-options]** hierarchy)	376 sessions (no FIP snooping scaling)	Disable FIP snooping scaling by including the **no-fip-snooping-scaling** option in the **[edit fc-options]** hierarchy. This disables FIP snooping scaling globally on the gateway.
Untrusted	No (**fcoe-lag** option not included in LAG configuration)	2,500 sessions (enhanced FIP snooping scaling)	FCoE LAGs with enhanced FIP snooping scaling enabled are not supported on untrusted FCoE-FC gateway fc-fabrics. To configure an FCoE LAG on an untrusted fc-fabric, FIP snooping scaling must be disabled.

FCoE Blade Switches

If you are using an FCoE blade switch, you need to configure an FCoE LAG only if the blade switch uses a passthrough module instead of an integrated switch.

Limitations

There are several limitations to configuring FCoE LAGs:

1. All FCoE LAG member links must belong to the same QFabric system Node group.
2. On an FCoE-FC gateway, you must disable FIP snooping scaling on untrusted fc-fabrics. Disabling FIP snooping scaling is global to the gateway Node device. If all of the fc-fabrics on an FCoE-FC gateway are trusted fabrics, you do not need to disable FIP snooping scaling.
3. FCoE LAGs with enhanced FIP snooping scaling enabled are not supported on untrusted FCoE-FC gateway fc-fabrics.

Related Documentation

- *Understanding Aggregated Ethernet Interfaces and LACP*
- Understanding Interfaces on an FCoE-FC Gateway on page 50
- Configuring an FCoE LAG on page 302
- Configuring VLANs for FCoE Traffic on an FCoE Transit Switch on page 289
- Configuring VN2VF_Port FIP Snooping and FCoE Trusted Interfaces on an FCoE Transit Switch on page 294
- Example: Configuring an FCoE LAG on a Redundant Server Node Group on page 253

Understanding OxID Hash Control for FCoE Traffic Load Balancing on QFabric Systems

The originator exchange identifier (OxID) field is one of several fields used in the hash function computation for FCoE traffic load balancing over multiple outgoing links in an Ethernet link aggregation group (LAG) on ports that face an FCoE forwarder (FCF). The QFabric system Node device ports can be 10-Gigabit Ethernet ports or 40-Gigabit fabric ports. (The 40-Gigabit fabric ports that connect a QFabric system Node device to QFabric system Interconnect devices function as a LAG even though they are not explicitly configured as a LAG.)

The OxID field is a unique identifier used to identify an exchange between a target and an initiator. The OxID value can be different for different exchanges between the same target and initiator.

- OxID Hash Control on page 164
- Advantages and Disadvantages of OxID Hash Control on page 164
- Disabling OxID Hash Control on page 165

OxID Hash Control

When FCoE traffic has multiple paths to an FCF (crosses a LAG that faces an FCF), packets can take different links between the source and destination endpoints. For each packet, the network bases the LAG link selection on the cost of the path (for example, link bandwidth or the number of hops). Using multiple paths distributes the FCoE traffic across the FCF-facing links, thus balancing the link load. The switch creates a hash value from some of the packet header fields, and uses the hash value to assign each packet to one of the LAG links. The switch always uses the following five packet header fields to compute the hash value:

- Source ID (SID)
- Destination ID (DID)
- Fabric ID (FID)
- Source Port ID (SPID)
- Source Module ID (SMID)

In addition, the QFabric system includes the OxID field by default in the FCoE load-balancing hash computation. However, if you do not want to use the OxID field in the FCoE load-balancing hash computation, you can remove it from the computation.

Advantages and Disadvantages of OxID Hash Control

The advantage of including the OxID field in the load-balancing hash computation is that OxID hash control allows different exchanges between a pair of Fibre Channel (FC) endpoints (such as an FCoE host and an FC storage device) to take different paths across the network, thus improving the aggregate network throughput and balancing the link load.

However, if communication between two FC endpoints uses different links, frames might not be delivered in the order that they are sent because of variance in the time each path takes to process and transmit frames. If your network is not experiencing out-of-order delivery of FCoE frames, then you can leave OxID hash control enabled and enjoy the benefits of load balancing. However, if your network experiences out-of-order delivery of FCoE frames, you can disable OxID hash control to force FCoE traffic to use the same path to the FCF and ensure in-order delivery of FCoE frames.

For example, when OxID hash control is enabled on a QFabric system, a Node device that is connected by 40-Gigabit fabric ports to four QFabric system Interconnect devices can send FCoE traffic across any of the four Interconnect devices to the FCF. (The connections to the four Interconnect devices function as a fabric LAG, even though they are not explicitly configured as a LAG.) Different Interconnect devices might not forward the FCoE frames at the same rate, so the frames might not be delivered in the order they were sent.

If FCoE frames are delivered out-of-order, you can disable OxID hash control to prevent the FCoE traffic from using different fabric links that connect to different Interconnect devices. Because disabling OxID hash control forces the frames to be delivered over the same link, the frames traverse the same Interconnect device and are delivered in order.

The same scenario is true when FCoE traffic traverses an FCF-facing LAG composed of 10-Gigabit interfaces. When OxID hash control is enabled, FCoE traffic can use any LAG link, which could result in out-of-order frame delivery. If your network experiences out-of-order FCoE frame delivery, disabling OxID hash control ensures that the FCoE traffic uses the same LAG link for every transaction, so the FCoE frames are delivered in order.

Disabling OxID Hash Control

You can disable OxID hash control on the 40-Gigabit fabric interfaces and on the 10-Gigabit Ethernet interfaces of a QFabric system Node group. Disabling OxID hash control affects all of the fabric or Ethernet interfaces of a Node group. For example, you cannot disable OxID hash control on some fabric interfaces in a Node group and leave OxID hash control enabled on other fabric interfaces of the same Node group.

Related Documentation

- Enabling and Disabling CoS OxID Hash Control on QFabric Systems on page 282
- Enabling and Disabling CoS OxID Hash Control on QFX Series Standalone Switches on page 281
- Understanding OxID Hash Control for FCoE Traffic Load Balancing on QFX Series Standalone Switches on page 79
- *Understanding CoS IEEE 802.1p Priorities for Lossless Traffic Flows*

CHAPTER 4

Learn About Technology

Data Center Technology Overview Videos

Juniper Information Experience (iX) videos provide brief, high-level overviews of data center technologies and concepts. Each video runs approximately one-and-a-half to two minutes in length. This document contains SDN-related videos and links to conceptual documents that contain other data center technology videos:

Learn About Video: Why Do We Need an IP Fabric?

The video *Why Do We Need an IP Fabric?* presents a brief overview of IP Fabric use cases.

Video: Why Do We Need an IP Fabric?

Learn About Video: What is the Best Control Plane Protocol to Use in a Data Center IP Fabric?

The video *What is the Best Control Plane Protocol to Use in a Data Center IP Fabric?* presents a brief overview of the arguments for using Border Gateway Protocol (BGP) as the data center IP fabric control plane protocol.

Video: What is the Best Control Plane Protocol to Use in a Data Center IP Fabric?

Learn About Video: Why Use an Overlay Network in a Data Center?

The video *Why Use an Overlay Network in a Data Center?* presents a brief overview of the advantages of data center overlay networks.

Video: Why Use an Overlay Network in a Data Center?

Conceptual Documents That Contain Technology Overview Videos

The following conceptual documents include brief video overviews of the technology:

- Understanding DCB Features and Requirements on page 16
- *Understanding CoS Hierarchical Port Scheduling (ETS)*
- Understanding CoS Flow Control (Ethernet PAUSE and PFC) on page 121
- Understanding DCBX on page 108
- *Understanding PFC Functionality Across Layer 3 Interfaces*
- *Virtual Chassis Fabric Overview*
- *Understanding In-Service Software Upgrade (ISSU)* and *In-Service Software Upgrade (ISSU) System Requirements* (same video)

PART 2

Configuration

CHAPTER 5

Configuration Examples

Example: Setting Up Fibre Channel and FCoE VLAN Interfaces in an FCoE-FC Gateway Fabric

To transmit Fibre Channel (FC) traffic between FCoE devices and a storage area network (SAN) FC switch, you configure a local FC fabric on the gateway. The gateway FC fabric includes FCoE and native FC interfaces, and a VLAN to carry FCoE traffic from FCoE-capable devices. The gateway FC fabric creates the path between the FCoE devices and the SAN.

This example describes how to configure the interfaces, VLAN, and FC fabric to connect FCoE devices to the FC switch and route traffic between the VLAN and FC interfaces:

Requirements

This example uses the following hardware and software components:

- A configured and provisioned Juniper Networks QFX3500 Switch to act as an FCoE-FC gateway
- FCoE-capable devices in an Ethernet network equipped with converged network adapters (CNAs)
- An FC switch to transmit and receive native FC traffic
- FC storage devices in the SAN
- Junos OS Release 11.1 or later for the QFX Series

Overview

No interfaces are configured for FC network connectivity by default. You need to configure the FC fabric and its interfaces explicitly. Each FC fabric consists of a combination of at least one FCoE VLAN interface between the FCoE-FC gateway and the FCoE devices, and one or more native FC interfaces between the FCoE-FC gateway and the FC switch.

An FCoE VLAN interface connects the FCoE-FC gateway to FCoE devices. FCoE traffic between the devices and the FCoE-FC gateway requires a dedicated VLAN used only for FCoE traffic. You cannot mix standard Ethernet traffic and FCoE traffic on the FCoE VLAN.

NOTE: IGMP snooping is not supported on FCoE VLANs. IGMP snooping is enabled by default on all VLANs in all software versions before Junos OS Release 13.2. Disable IGMP snooping on FCoE VLANs if you are using software that is older than 13.2.

Storm control is not supported on Ethernet interfaces that belong to the FCoE VLAN. Ensure that storm control is disabled on all Ethernet interfaces that belong to the FCoE VLAN to prevent FCoE traffic from being dropped.

When FCoE frames enter the FCoE-FC gateway, the gateway:

1. Strips the Ethernet encapsulation from the FCoE frames.
2. Sends the resulting native FC frames to the FC switch through the gateway's native FC interfaces.

Each FC interface and FCoE VLAN interface can belong to only one FC fabric. Different FC fabrics must use different native FC interfaces and different FCoE VLAN interfaces. Multiple FC fabrics on the FCoE-FC gateway can connect to the same FC switch, but they must use different FC interfaces and different FCoE VLAN interfaces.

The Ethernet interfaces that belong to the FCoE VLAN should be configured in tagged-access port mode and must include the native VLAN because FIP VLAN discovery and notification frames are exchanged as untagged packets. These Ethernet interfaces require a maximum transmission unit (MTU) size of at least 2180 bytes to accommodate the FC payload and FCoE encapsulation. (Sometimes the MTU is rounded up to 2500 bytes. If larger frames are expected on the interface, set the MTU size accordingly.)

This example shows a simple configuration to illustrate the basic steps for creating:

- The FCoE-device-facing VLAN and its 10-Gigabit Ethernet interfaces
- The VLAN interface
- The FC-switch-facing native FC interfaces
- One FC fabric on the FCoE-FC gateway

Configuring these elements results in traffic being routed between the VLAN and FC interfaces, thus connecting the FCoE devices to the FC switch through the FCoE-FC gateway.

A VLAN called **blue** transports FCoE traffic between FCoE devices and the FCoE-FC gateway using an FCoE VLAN interface called **vlan.100**. The FCoE-FC gateway's **vlan.100** interface presents an F_Port interface to the FCoE devices on the VLAN. For each FCoE device ENode that logs in to the FCoE-FC gateway, the gateway instantiates a virtual F_Port (VF_Port) interface. This creates a virtual link between the ENode VN_Port and the FCoE-FC gateway. The FCoE-FC gateway's native FC interfaces transport FC traffic between the gateway and the FC switch.

Configuring both the FCoE VLAN interface and the native FC interfaces as part of a gateway fabric associates them in the switch and makes the connection between the FCoE servers and the FC switch.

Topology

The topology for this example consists of one QFX3500 switch with FC-capable ports to connect to the FC switch and with Ethernet ports in tagged-access mode to connect to the FCoE devices. Table 22 on page 173 and Figure 10 on page 175 show the configuration components of this example.

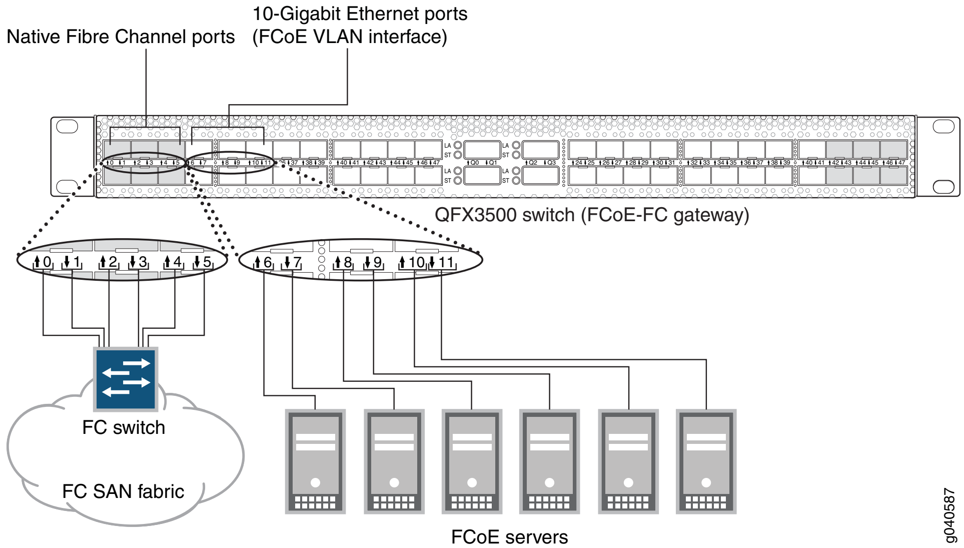
Table 22: Components of the Fibre Channel Interface Configuration Topology

Property	Settings
Switch hardware	QFX3500 switch in gateway mode
FCoE VLAN name and tag ID	**blue**, tag **100** IGMP snooping disabled on the FCoE VLAN.

Table 22: Components of the Fibre Channel Interface Configuration Topology *(continued)*

Property	Settings
Interfaces in VLAN **blue**	Interfaces: **xe-0/0/6**, **xe-0/0/7**, **xe-0/0/8**, **xe-0/0/9**, **xe-0/0/10**, **xe-0/0/11** Port mode: **tagged-access** MTU: **2180** Native VLAN: **1** NOTE: FCoE VLANs (any VLAN that carries FCoE traffic) support only Spanning Tree Protocol (STP) and link aggregation group (LAG) Layer 2 features. FCoE traffic cannot use a standard LAG because traffic might be hashed to different physical LAG links on different transmissions. This breaks the (virtual) point-to-point link that Fibre Channel traffic requires. If you configure a standard LAG interface for FCoE traffic, FCoE traffic might be rejected by the FC SAN. QFabric systems support a special LAG called an FCoE LAG, which enables you to transport FCoE traffic and regular Ethernet traffic across the same link aggregation bundle. An FCoE LAG ensures that FCoE traffic uses the same physical link in the LAG for requests and replies in order to preserve the virtual point-to-point link between the FCoE device converged network adapter (CNA) and the FC SAN switch across the QFabric system Node device. An FCoE LAG does not provide load balancing or link redundancy for FCoE traffic. However, regular Ethernet traffic receives the usual LAG benefits of load balancing and link redundancy in an FCoE LAG.
FCoE VLAN interface	**vlan.100** Port mode: **f-port**
Native Fibre Channel interfaces	Interfaces: **fc-0/0/0**, **fc-0/0/1**, **fc-0/0/2**, **fc-0/0/3**, **fc-0/0/4**, **fc-0/0/5** Port mode: **np-port** Speed: **4 Gbps**
Fibre Channel fabric **fcproxy1**	Fabric type: **proxy** Fabric ID: **1** FC interfaces: **fc-0/0/0**, **fc-0/0/1**, **fc-0/0/2**, **fc-0/0/3**, **fc-0/0/4**, **fc-0/0/5**

Figure 10: Fibre Channel Interface Configuration Topology



This configuration example creates a VLAN for FCoE traffic and routes its traffic to an FCoE VLAN interface that is part of the FC fabric. It also creates the FC interfaces needed to connect to the FC switch.

To set up FC interfaces and FCoE VLAN interfaces:

- Configure a VLAN to use as a dedicated FCoE VLAN:
  - Configure the interfaces the FCoE VLAN uses as Ethernet switching interfaces in tagged-access port mode.
  - If storm control is enabled, disable it on the interfaces.
  - Configure the interfaces the FCoE VLAN uses with the native VLAN.
  - Configure the FCoE VLAN to use the desired Ethernet interfaces.
  - Disable IGMP snooping on the FCoE VLAN. (Before Junos OS Release 13.2, IGMP snooping was enabled by default on all VLANs, but is not supported on FCoE VLANs. Starting with Junos OS Release 13.2, IGMP snooping is enabled by default only on the default VLAN.)
- Configure the FCoE VLAN interface.
- Define the interface for the FCoE VLAN (associate the VLAN with the FCoE VLAN interface).
- Configure the physical FC interfaces (either one or two 6-port blocks) that connect to the FC switch.
- Configure the logical FC interfaces that connect to the FC switch.
- Configure the FCoE-FC gateway fabric:
  - Configure the fabric ID.
  - Configure the fabric as a proxy fabric.

- Add the FCoE VLAN interface and the native FC interfaces to the fabric.

To keep the example simple, the configuration steps show six Ethernet interfaces in the FCoE VLAN and six native FC interfaces in the FC fabric. Use the same configuration procedure to add more interfaces to the FCoE VLAN or to the FC fabric.

Configuration

CLI Quick Configuration

To quickly configure FCoE and native FC interfaces on an FCoE-FC gateway and route traffic between the FCoE VLAN and FC interfaces, copy the following commands and paste them into the switch terminal window:

```
[edit]
set vlans blue vlan-id 100
set vlans native vlan-id 1
set interfaces xe-0/0/6 unit 0 family ethernet-switching port-mode tagged-access vlan members blue
set interfaces xe-0/0/7 unit 0 family ethernet-switching port-mode tagged-access vlan members blue
set interfaces xe-0/0/8 unit 0 family ethernet-switching port-mode tagged-access vlan members blue
set interfaces xe-0/0/9 unit 0 family ethernet-switching port-mode tagged-access vlan members blue
set interfaces xe-0/0/10 unit 0 family ethernet-switching port-mode tagged-access vlan members blue
set interfaces xe-0/0/11 unit 0 family ethernet-switching port-mode tagged-access vlan members blue
set interfaces xe-0/0/6 unit 0 family ethernet-switching native-vlan-id 1
set interfaces xe-0/0/7 unit 0 family ethernet-switching native-vlan-id 1
set interfaces xe-0/0/8 unit 0 family ethernet-switching native-vlan-id 1
set interfaces xe-0/0/9 unit 0 family ethernet-switching native-vlan-id 1
set interfaces xe-0/0/10 unit 0 family ethernet-switching native-vlan-id 1
set interfaces xe-0/0/11 unit 0 family ethernet-switching native-vlan-id 1
set interfaces xe-0/0/6 mtu 2180
set interfaces xe-0/0/7 mtu 2180
set interfaces xe-0/0/8 mtu 2180
set interfaces xe-0/0/9 mtu 2180
set interfaces xe-0/0/10 mtu 2180
set interfaces xe-0/0/11 mtu 2180
set vlans blue interface xe-0/0/6.0
set vlans blue interface xe-0/0/7.0
set vlans blue interface xe-0/0/8.0
set vlans blue interface xe-0/0/9.0
set vlans blue interface xe-0/0/10.0
set vlans blue interface xe-0/0/11.0
set protocols igmp-snooping vlan blue disable
set interfaces vlan unit 100 family fibre-channel port-mode f-port
set vlans blue l3-interface vlan.100
set chassis fpc 0 pic 0 fibre-channel port-range 0 5
set interfaces fc-0/0/0 unit 0 family fibre-channel port-mode np-port
set interfaces fc-0/0/1 unit 0 family fibre-channel port-mode np-port
set interfaces fc-0/0/2 unit 0 family fibre-channel port-mode np-port
set interfaces fc-0/0/3 unit 0 family fibre-channel port-mode np-port
set interfaces fc-0/0/4 unit 0 family fibre-channel port-mode np-port
set interfaces fc-0/0/5 unit 0 family fibre-channel port-mode np-port
set interfaces fc-0/0/0 fibrechannel-options speed 4g
set interfaces fc-0/0/1 fibrechannel-options speed 4g
set interfaces fc-0/0/2 fibrechannel-options speed 4g
```


set interfaces fc-0/0/3 fibrechannel-options speed 4g
set interfaces fc-0/0/4 fibrechannel-options speed 4g
set interfaces fc-0/0/5 fibrechannel-options speed 4g
set fc-fabrics fcproxy1 fabric-id 1
set fc-fabrics fcproxy1 fabric-type proxy
set fc-fabrics fcproxy1 interface vlan.100
set fc-fabrics fcproxy1 interface fc-0/0/0.0
set fc-fabrics fcproxy1 interface fc-0/0/1.0
set fc-fabrics fcproxy1 interface fc-0/0/2.0
set fc-fabrics fcproxy1 interface fc-0/0/3.0
set fc-fabrics fcproxy1 interface fc-0/0/4.0
set fc-fabrics fcproxy1 interface fc-0/0/5.0

Step-by-Step Procedure

Configure FCoE and FC interfaces in an FCoE-FC gateway FC fabric and set up traffic routing between the FCoE VLAN and FC interfaces:

1. Configure the VLAN for FCoE traffic:

```
[edit vlans]
user@switch# set blue vlan-id 100
```

2. Configure the native VLAN:

```
[edit vlans]
user@switch# set native vlan-id 1
```

3. Configure the Ethernet interfaces for the FCoE VLAN in tagged-access mode and as members of the FCoE VLAN (VLAN **blue**):

```
[edit interfaces]
user@switch# set xe-0/0/6 unit 0 family ethernet-switching port-mode tagged-access
vlan members blue
user@switch# set xe-0/0/7 unit 0 family ethernet-switching port-mode tagged-access
vlan members blue
user@switch# set xe-0/0/8 unit 0 family ethernet-switching port-mode tagged-access
vlan members blue
user@switch# set xe-0/0/9 unit 0 family ethernet-switching port-mode tagged-access
vlan members blue
user@switch# set xe-0/0/10 unit 0 family ethernet-switching port-mode tagged-access
vlan members blue
user@switch# set xe-0/0/11 unit 0 family ethernet-switching port-mode tagged-access
vlan members blue
```

4. Configure the native VLAN on the Ethernet interfaces in the FCoE VLAN:

```
[edit interfaces]
user@switch# set xe-0/0/6 unit 0 family ethernet-switching native-vlan-id 1
user@switch# set xe-0/0/7 unit 0 family ethernet-switching native-vlan-id 1
user@switch# set xe-0/0/8 unit 0 family ethernet-switching native-vlan-id 1
user@switch# set xe-0/0/9 unit 0 family ethernet-switching native-vlan-id 1
user@switch# set xe-0/0/10 unit 0 family ethernet-switching native-vlan-id 1
user@switch# set xe-0/0/11 unit 0 family ethernet-switching native-vlan-id 1
```

5. Set the MTU to 2180 for each Ethernet interface:

```
[edit interfaces]
user@switch# set xe-0/0/6 mtu 2180
user@switch# set xe-0/0/7 mtu 2180
user@switch# set xe-0/0/8 mtu 2180
user@switch# set xe-0/0/9 mtu 2180
user@switch# set xe-0/0/10 mtu 2180
user@switch# set xe-0/0/11 mtu 2180
```

6. Assign the Ethernet interfaces to the FCoE VLAN:

```
[edit vlans blue interface]
user@switch# set xe-0/0/6.0
user@switch# set xe-0/0/7.0
user@switch# set xe-0/0/8.0
user@switch# set xe-0/0/9.0
user@switch# set xe-0/0/10.0
user@switch# set xe-0/0/11.0
```

7. Disable IGMP snooping on the FCoE VLAN:

```
[edit protocols]
user@switch# set igmp-snooping vlan blue disable
```

8. Configure the FCoE VLAN interface and port mode for the FCoE traffic:

```
[edit interfaces]
user@switch# set vlan unit 100 family fibre-channel port-mode f-port
```

9. Define the FCoE VLAN interface as the interface for the FCoE VLAN:

```
[edit vlans]
user@switch# set blue l3-interface vlan.100
```

10. Configure the physical FC interfaces the fabric uses to connect to the FC switch:

```
[edit chassis fpc 0 pic 0]
user@switch# set fibre-channel port-range 0 5
```

NOTE: When you configure ports as FC ports, the port designation changes from xe-n/n/n.n format to fc-n/n/n.n format to indicate that the interface is an FC interface. FC interfaces do not support 10-Gbps interface speed but instead conform to FC interface speeds of 2 Gbps, 4 Gbps, or 8 Gbps.

11. Configure the native FC interfaces and port mode:

```
[edit interfaces]
user@switch# set fc-0/0/0 unit 0 family fibre-channel port-mode np-port
user@switch# set fc-0/0/1 unit 0 family fibre-channel port-mode np-port
user@switch# set fc-0/0/2 unit 0 family fibre-channel port-mode np-port
user@switch# set fc-0/0/3 unit 0 family fibre-channel port-mode np-port
user@switch# set fc-0/0/4 unit 0 family fibre-channel port-mode np-port
user@switch# set fc-0/0/5 unit 0 family fibre-channel port-mode np-port
```

12. Configure the native FC interface port speed:

```
[edit interfaces]
user@switch# set fc-0/0/0 fibrechannel-options speed 4g
user@switch# set fc-0/0/1 fibrechannel-options speed 4g
user@switch# set fc-0/0/2 fibrechannel-options speed 4g
user@switch# set fc-0/0/3 fibrechannel-options speed 4g
user@switch# set fc-0/0/4 fibrechannel-options speed 4g
user@switch# set fc-0/0/5 fibrechannel-options speed 4g
```

13. Configure the FC fabric name and unique ID:

```
[edit fc-fabrics]
user@switch# set fcproxy1 fabric-id 1
```


14. Define the FC fabric as an FCoE-FC gateway:

```
[edit fc-fabrics]
user@switch# set fcproxy1 fabric-type proxy
```

15. Assign the FCoE VLAN interface to the fabric:

```
[edit fc-fabrics]
user@switch# set fcproxy1 interface vlan.100
```

16. Assign the native FC interfaces to the fabric:

```
[edit fc-fabrics]
user@switch# set fcproxy1 interface fc-0/0/0.0
user@switch# set fcproxy1 interface fc-0/0/1.0
user@switch# set fcproxy1 interface fc-0/0/2.0
user@switch# set fcproxy1 interface fc-0/0/3.0
user@switch# set fcproxy1 interface fc-0/0/4.0
user@switch# set fcproxy1 interface fc-0/0/5.0
```

Results Display the results of the configuration:

```
user@switch> show configuration
fc-0/0/0 {
   fibrechannel-options {
      speed 4g;
   }
   unit 0 {
      family fibre-channel {
         port-mode np-port;
      }
   }
}
fc-0/0/1 {
   fibrechannel-options {
      speed 4g;
   }
   unit 0 {
      family fibre-channel {
         port-mode np-port;
      }
   }
}
fc-0/0/2 {
   fibrechannel-options {
      speed 4g;
   }
   unit 0 {
      family fibre-channel {
         port-mode np-port;
      }
   }
}
fc-0/0/3 {
   fibrechannel-options {
      speed 4g;
   }
   unit 0 {
```

```
        family fibre-channel {
            port-mode np-port;
        }
    }
}
fc-0/0/4 {
    fibrechannel-options {
        speed 4g;
    }
    unit 0 {
        family fibre-channel {
            port-mode np-port;
        }
    }
}
fc-0/0/5 {
    fibrechannel-options {
        speed 4g;
    }
    unit 0 {
        family fibre-channel {
            port-mode np-port;
        }
    }
}
xe-0/0/6 {
    mtu 2180;
    unit 0 {
        family ethernet-switching {
            port-mode tagged-access;
            vlan {
                members blue;
            }
            native-vlan-id 1;
        }
    }
}
xe-0/0/7 {
    mtu 2180;
    unit 0 {
        family ethernet-switching {
            port-mode tagged-access;
            vlan {
                members blue;
            }
            native-vlan-id 1;
        }
    }
}
xe-0/0/8 {
    mtu 2180;
    unit 0 {
        family ethernet-switching {
            port-mode tagged-access;
            vlan {
                members blue;
```

```
            }
            native-vlan-id 1;
        }
    }
}
xe-0/0/9 {
    mtu 2180;
    unit 0 {
        family ethernet-switching {
            port-mode tagged-access;
            vlan {
                members blue;
            }
            native-vlan-id 1;
        }
    }
}
xe-0/0/10 {
    mtu 2180;
    unit 0 {
        family ethernet-switching {
            port-mode tagged-access;
            vlan {
                members blue;
            }
            native-vlan-id 1;
        }
    }
}
xe-0/0/11 {
    mtu 2180;
    unit 0 {
        family ethernet-switching {
            port-mode tagged-access;
            vlan {
                members blue;
            }
            native-vlan-id 1;
        }
    }
}
vlan {
    unit 100 {
        family fibre-channel {
            port-mode f-port;
        }
    }
}
fc-fabrics {
    fcproxy1 {
        fabric-id 1
        fabric-type proxy
        interface {
            vlan.100
            fc-0/0/0.0;
            fc-0/0/1.0;
```

```
            fc-0/0/2.0;
            fc-0/0/3.0;
            fc-0/0/4.0;
            fc-0/0/5.0;
        }
    }
}
protocols {
    igmp-snooping {
        vlan blue {
            disable;
        }
    }
}
vlans {
    blue {
        vlan-id 100
        interface {
            xe-0/0/6.0;
            xe-0/0/7.0;
            xe-0/0/8.0;
            xe-0/0/9.0;
            xe-0/0/10.0;
            xe-0/0/11.0;
        }
        l3-interface vlan.100
    }
    native {
        vlan-id 1;
    }
}
```

TIP: To quickly configure the interfaces, issue the **load merge terminal** command and then copy the hierarchy and paste it into the switch terminal window.

Verification

To verify that the native FC interfaces and FCoE VLAN interface have been created, added to the FC fabric, and are operating properly, perform these tasks:

- Verifying That the Native FC Interfaces and the FCoE VLAN Interface Have Been Created on page 183
- Verifying That the FCoE VLAN Includes the Correct Ethernet Interfaces on page 183
- Verifying That the FC Fabric Includes the Correct Interfaces on page 184
- Verifying Native FC Interface Operation on page 184
- Verifying That IGMP Snooping Has Been Disabled on the FCoE VLAN on page 185

Verifying That the Native FC Interfaces and the FCoE VLAN Interface Have Been Created

Purpose Verify that the six native FC interfaces and the FCoE VLAN interface have been created on the switch and are configured in the correct mode.

Action List all of the FC interfaces configured on the switch using the **show fibre-channel interfaces** command:

```
user@switch> show fibre-channel interfaces
                               Native              Config  Oper
Interface       Idx    Type    Fabric-id     NPIV  Mode    Mode    State
fc-0/0/0.0      70     FC      1             YES   NP      NP      up
fc-0/0/1.0      71     FC      1             YES   NP      NP      up
fc-0/0/2.0      72     FC      1             YES   NP      NP      up
fc-0/0/3.0      73     FC      1             YES   NP      NP      up
fc-0/0/4.0      74     FC      1             YES   NP      NP      up
fc-0/0/5.0      75     FC      1             YES   NP      NP      up
vlan.100        67     FCOE    1             YES   F       F       up
```

Meaning The **show fibre-channel interfaces** command lists all native FC interfaces and FCoE VLAN interfaces configured on the switch. The command output shows that the FC interfaces **fc-0/0/0.0**, **fc-0/0/1.0**, **fc-0/0/2.0**, **fc-0/0/3.0**, **fc-0/0/4.0**, and **fc-0/0/5.0** have been created and that those six interfaces:

- Are native Fibre Channel interfaces (type **FC**).
- Belong to the FC fabric with a configured fabric ID of **1**.
- Are capable of N_Port ID virtualization (NPIV).
- Have a configured mode and an operational mode of proxy N_Port (**NP**), which means that they should be connected to an FCF or an FC switch, not to an FCoE device, and that they carry native FC traffic.
- Show an operational state of **up**.

The command output also shows that the FCoE VLAN interface **vlan.100** has been created and that interface:

- Is an FCoE VLAN interface (type **FCOE)**.
- Belongs to the FC fabric with a configured fabric ID of **1**.
- Is capable of N_Port ID virtualization (NPIV).
- Has a configured mode and an operational mode of F_Port (**F**), which means that its interfaces connect to FCoE devices and carry FCoE traffic.
- Shows an operational state of **up**.

Verifying That the FCoE VLAN Includes the Correct Ethernet Interfaces

Purpose Verify that the FCoE VLAN **blue** has been created with the correct VLAN tag (**100**) and with the correct Ethernet interfaces.

Action List all of the interfaces configured on the switch in VLAN **blue** using the **show vlans** command:

```
user@switch> show vlans blue
Name           Tag     Interfaces
blue           100
                     xe-0/0/6.0, xe-0/0/7.0, xe-0/0/8.0, xe-0/0/9.0, xe-0/0/10.0

                       xe-0/0/11.0
```

Meaning The **show vlans blue** command lists the interfaces that are members of the FCoE VLAN **blue**. The command output shows that the **blue** VLAN has a tag ID of 100 and includes the interfaces **xe-0/0/6.0**, **xe-0/0/7.0**, **xe-0/0/8.0**, **xe-0/0/9.0**, **xe-0/0/10.0**, and **xe-0/0/11.0**.

Verifying That the FC Fabric Includes the Correct Interfaces

Purpose Verify that the FC fabric configuration is configured on the switch with the correct native FC and FCoE VLAN interfaces.

Action List all of the interfaces configured on FC fabrics on the switch using the **show fibre-channel fabric** command:

```
user@switch> show fibre-channel fabric
Name           Fabric-id     Type     Interfaces
fcproxy1       1             PROXY
                                      fc-0/0/0.0
                                      fc-0/0/1.0
                                      fc-0/0/2.0
                                      fc-0/0/3.0
                                      fc-0/0/4.0
                                      fc-0/0/5.0
                                      vlan.100
```

Meaning The **show fibre-channel fabric** command lists the interfaces that are members of each FC fabric. The command output shows that the only fabric configured on the switch is named **fcproxy1**, has a fabric-id of **1**, and is a **proxy** fabric in an FCoE-FC gateway. The command output also shows that the native FC interfaces **fc-0/0/0.0**, **fc-0/0/1.0**, **fc-0/0/2.0**, **fc-0/0/3.0**, **fc-0/0/4.0**, and **fc-0/0/5.0**, and the FCoE VLAN interface **vlan.100** belong to **fcproxy1**.

Verifying Native FC Interface Operation

Purpose Verify that the native FC interfaces are online and display the number of FC sessions on each interface.

Action List all of the native FC NP_Port interface states and sessions by FC fabric using the **show fibre-channel proxy np-port** command:

```
user@switch> show fibre-channel proxy np-port
Fabric: fcproxy1, Fabric-id: 1
NP-Port     State              Sessions       LB state    LB weight
fc-0/0/0.0  online             3              ON          4
fc-0/0/1.0  online             3              ON          4
```



```
fc-0/0/2.0  online        2          ON        4
fc-0/0/3.0  online        2          ON        4
fc-0/0/4.0  online        2          ON        4
fc-0/0/5.0  online        2          ON        4
```

Meaning The **show fibre-channel proxy np-port** command lists the interfaces that are configured as native FC proxy N_Port interfaces. The command output shows:

- The fabric name is **fcproxy1** and its fabric ID is **1**.
- The interfaces are **online**.
- The number of FC sessions (virtual links) running on each interface.
- The load-balancing (LB) state is **ON** for all of the interfaces.
- The LB weight reflects the port speed of each interface, which is **4** Gbps.

Verifying That IGMP Snooping Has Been Disabled on the FCoE VLAN

Purpose Verify that IGMP snooping is disabled on the FCoE VLAN.

Action List the IGMP snooping protocol information for the FCoE VLAN using the **show configuration protocols igmp-snooping vlan blue** command:

```
user@switch> show configuration protocols igmp-snooping vlan blue
    disable;
```

Meaning The **show configuration protocols igmp-snooping vlan blue** command lists the IGMP snooping configuration for the FCoE VLAN. The command output shows that IGMP snooping is disabled on the FCoE VLAN.

Related Documentation

- Example: Configuring an FCoE LAG on a Redundant Server Node Group on page 253
- Configuring a Fibre Channel Interface on page 269
- Configuring an FCoE VLAN Interface on an FCoE-FC Gateway on page 272
- Disabling Storm Control on FCoE Interfaces on an FCoE-FC Gateway on page 278
- Assigning Interfaces to a Fibre Channel Fabric on page 276
- Configuring an FCoE-FC Gateway Fibre Channel Fabric on page 266
- Configuring FIP on an FCoE-FC Gateway on page 283
- Disabling VN2VF_Port FIP Snooping on an FCoE-FC Gateway Switch Interface on page 299
- Configuring an FCoE LAG on page 302
- Understanding Interfaces on an FCoE-FC Gateway on page 50

Example: Configuring CoS PFC for FCoE Traffic

Priority-based flow control (PFC, described in IEEE 802.1Qbb) is a link-level flow control mechanism that you apply at ingress interfaces. PFC enables you to divide traffic on one physical link into eight priorities. You can think of the eight priorities as eight "lanes" of traffic that correspond to queues (forwarding classes). Each priority is mapped to a 3-bit IEEE 802.1p CoS flag in the VLAN header.

You can selectively apply PFC to the traffic in any queue without pausing the traffic in other queues on the same link. You must apply PFC to FCoE traffic to ensure lossless transport.

To configure PFC on FCoE traffic, use the default FCoE forwarding-class-to-queue mapping and:

- Configure a classifier that associates the FCoE forwarding class with FCoE traffic.
- Configure a congestion notification profile to apply PFC to the FCoE traffic.
- Apply the classifier and the PFC configuration to ingress interfaces.
- Configure the bandwidth scheduling for the FCoE forwarding class output queue.
- Create a forwarding class set (priority group) that includes the FCoE forwarding class; this is required to configure enhanced transmission selection (ETS) and support data center bridging (DCB).
- Configure the bandwidth scheduling for the FCoE priority group.
- Apply the scheduling to the egress interfaces.

NOTE: If you are using Junos OS Release 12.2 or later, use the default forwarding classes for the lossless **fcoe** forwarding class. If you explicitly configure default lossless forwarding classes, the traffic mapped to those forwarding classes is treated as lossy (**best-effort**) traffic and does *not* receive lossless treatment.

In Junos OS Release 12.3 and later, you can include the *no-loss* packet drop attribute in explicit forwarding class configurations to configure a lossless forwarding class.

This example describes how to configure PFC for FCoE traffic:

- Requirements on page 187
- Overview on page 187
- Configuration on page 188
- Verification on page 192

Requirements

This example uses the following hardware and software components:

- A Juniper Networks QFX3500 Switch
- Junos OS Release 11.1 or later for the QFX Series

Overview

FCoE traffic requires PFC to ensure lossless packet transport. This example shows you how to:

- Assign FCoE traffic to the FCoE priority at the ingress.
- Create and apply CoS for the FCoE traffic using ETS (hierarchical port scheduling).
- Apply PFC to the FCoE traffic.
- Apply the configuration to ingress and egress interfaces.

NOTE: **Configuring or changing PFC on an interface blocks the entire port until the PFC change is completed. After a PFC change is completed, the port is unblocked and traffic resumes. Blocking the port stops ingress and egress traffic, and causes packet loss on all queues on the port until the port is unblocked.**

Each interface in this example is configured as both an ingress interface and an egress interface, so the classifier, congestion notification profile, and port scheduling are applied to all of the interfaces.

Topology

Table 23 on page 187 shows the configuration components for this example.

Table 23: Components of the PFC for FCoE Traffic Configuration Topology

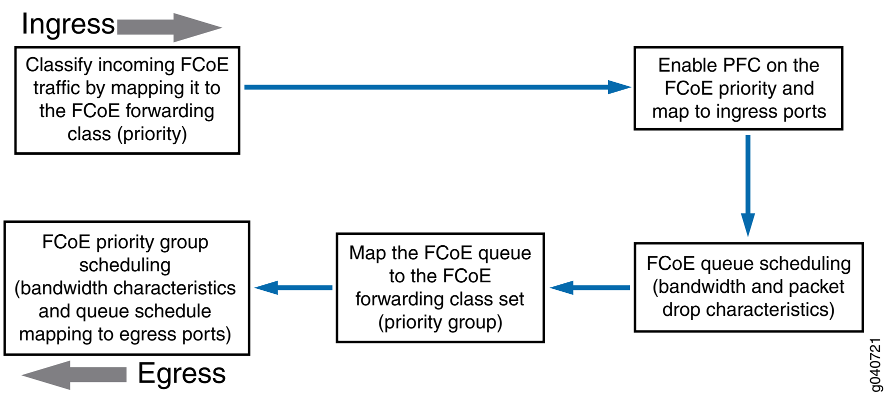
Component	Settings
Hardware	QFX3500 switch
Behavior aggregate classifier (maps the FCoE forwarding class to incoming packets by IEEE 802.1 code point)	Code point **011** to forwarding class **fcoe** and loss priority **low** Ingress interfaces: **xe-0/0/31**, **xe-0/0/32**, **xe-0/0/33**, **xe-0/0/34**
PFC congestion notification profile	**fcoe-cnp**: Code point **011** Ingress interfaces: **xe-0/0/31**, **xe-0/0/32**, **xe-0/0/33**, **xe-0/0/34**
FCoE queue scheduler	**fcoe-sched**: Minimum bandwidth **3g** Maximum bandwidth **100%** Priority **low**

Table 23: Components of the PFC for FCoE Traffic Configuration Topology *(continued)*

Component	Settings
Forwarding class-to-scheduler mapping	Scheduler map **fcoe-map**: Forwarding class **fcoe** Scheduler **fcoe-sched**
Forwarding class set (FCoE priority group)	**fcoe-pg**: Forwarding class **fcoe** Egress interfaces: **xe-0/0/31**, **xe-0/0/32**, **xe-0/0/33**, **xe-0/0/34**
Traffic control profile	**fcoe-tcp**: Scheduler map **fcoe-map** Minimum bandwidth **3g** Maximum bandwidth **100%**

Figure 11 on page 188 shows a block diagram of the configuration components and the configuration flow of the CLI statements used in the example.

Figure 11: PFC for FCoE Traffic Configuration Components Block Diagram



Configuration

CLI Quick Configuration

To quickly configure PFC for FCoE traffic, copy the following commands, paste them in a text file, remove line breaks, change variables and details to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level:

```
[edit class-of-service]
set classifiers ieee-802.1 fcoe-classifier forwarding-class fcoe loss-priority low code-points 011
set congestion-notification-profile fcoe-cnp input ieee-802.1 code-point 011 pfc
set interfaces xe-0/0/31 unit 0 classifiers ieee-802.1 fcoe-classifier
set interfaces xe-0/0/32 unit 0 classifiers ieee-802.1 fcoe-classifier
set interfaces xe-0/0/33 unit 0 classifiers ieee-802.1 fcoe-classifier
set interfaces xe-0/0/34 unit 0 classifiers ieee-802.1 fcoe-classifier
set interfaces xe-0/0/31 congestion-notification-profile fcoe-cnp
set interfaces xe-0/0/32 congestion-notification-profile fcoe-cnp
set interfaces xe-0/0/33 congestion-notification-profile fcoe-cnp
set interfaces xe-0/0/34 congestion-notification-profile fcoe-cnp
set schedulers fcoe-sched priority low transmit-rate 3g
set schedulers fcoe-sched shaping-rate percent 100
```

set scheduler-maps fcoe-map forwarding-class fcoe scheduler fcoe-sched
set forwarding-class-sets fcoe-pg class fcoe
set traffic-control-profiles fcoe-tcp scheduler-map fcoe-map guaranteed-rate 3g
set traffic-control-profiles fcoe-tcp shaping-rate percent 100
set interfaces xe-0/0/31 forwarding-class-set fcoe-pg output-traffic-control-profile fcoe-tcp
set interfaces xe-0/0/32 forwarding-class-set fcoe-pg output-traffic-control-profile fcoe-tcp
set interfaces xe-0/0/33 forwarding-class-set fcoe-pg output-traffic-control-profile fcoe-tcp
set interfaces xe-0/0/34 forwarding-class-set fcoe-pg output-traffic-control-profile fcoe-tcp

Step-by-Step Procedure

To configure the FCoE forwarding class (priority), ingress classifier, output queue scheduling, forwarding class set (priority group) and its output port scheduling, PFC application, and interfaces to set up PFC for FCoE traffic:

1. Configure a classifier to set the loss priority and IEEE 802.1 code point assigned to the FCoE forwarding class at the ingress:

   ```
   [edit class-of-service]
   user@switch# set classifiers ieee-802.1 fcoe-classifier forwarding-class fcoe loss-priority low code-points 011
   ```

2. Configure PFC on the FCoE queue by applying FCoE to the IEEE 802.1 code point **011**:

   ```
   [edit class-of-service]
   user@switch# set congestion-notification-profile fcoe-cnp input ieee-802.1 code-point 011 pfc
   ```

3. Apply the PFC configuration to the ingress interfaces:

   ```
   [edit class-of-service]
   user@switch# set interfaces xe-0/0/31 congestion-notification-profile fcoe-cnp
   user@switch# set interfaces xe-0/0/32 congestion-notification-profile fcoe-cnp
   user@switch# set interfaces xe-0/0/33 congestion-notification-profile fcoe-cnp
   user@switch# set interfaces xe-0/0/34 congestion-notification-profile fcoe-cnp
   ```

4. Assign the classifier to the ingress interfaces:

   ```
   [edit class-of-service]
   user@switch# set interfaces xe-0/0/31 unit 0 classifiers ieee-802.1 fcoe-classifier
   user@switch# set interfaces xe-0/0/32 unit 0 classifiers ieee-802.1 fcoe-classifier
   user@switch# set interfaces xe-0/0/33 unit 0 classifiers ieee-802.1 fcoe-classifier
   user@switch# set interfaces xe-0/0/34 unit 0 classifiers ieee-802.1 fcoe-classifier
   ```

5. Configure output scheduling for the FCoE queue:

   ```
   [edit class-of-service]
   user@switch# set schedulers fcoe-sched priority low transmit-rate 3g
   user@switch# set schedulers fcoe-sched shaping-rate percent 100
   ```

6. Map the FCoE forwarding class to the FCoE scheduler:

   ```
   [edit class-of-service]
   user@switch# set scheduler-maps fcoe-map forwarding-class fcoe scheduler fcoe-sched
   ```

7. Configure the forwarding class set for the FCoE traffic:

   ```
   [edit class-of-service]
   user@switch# set forwarding-class-sets fcoe-pg class fcoe
   ```

8. Define the traffic control profile for the FCoE forwarding class set:

   ```
   [edit class-of-service]
   ```

```
user@switch# set traffic-control-profiles fcoe-tcp scheduler-map fcoe-map
guaranteed-rate 3g
user@switch# set traffic-control-profiles fcoe-tcp shaping-rate percent 100
```

9. Apply the FCoE forwarding class set and traffic control profile to the egress ports:

```
[edit class-of-service]
user@switch# set interfaces xe-0/0/31 forwarding-class-set fcoe-pg
output-traffic-control-profile fcoe-tcp
user@switch# set interfaces xe-0/0/32 forwarding-class-set fcoe-pg
output-traffic-control-profile fcoe-tcp
user@switch# set interfaces xe-0/0/33 forwarding-class-set fcoe-pg
output-traffic-control-profile fcoe-tcp
user@switch# set interfaces xe-0/0/34 forwarding-class-set fcoe-pg
output-traffic-control-profile fcoe-tcp
```

Results

Display the results of the configuration (the system shows only the explicitly configured parameters; it does not show default parameters such as the **fcoe** lossless forwarding class):

```
user@switch> show configuration class-of-service
classifiers {
  ieee-802.1 fcoe-classifier {
    forwarding-class fcoe {
      loss-priority low code-points 011;
    }
  }
  traffic-control-profiles {
    fcoe-tcp {
      scheduler-map fcoe-map;
      shaping-rate percent 100;
      guaranteed-rate 3000000000;
    }
  }
  forwarding-class-sets {
    fcoe-pg {
      class fcoe;
    }
  }
  congestion-notification-profile {
    fcoe-cnp {
      input {
        ieee-802.1 {
          code-point 011 {
            pfc;
          }
        }
      }
    }
  }
  interfaces {
    xe-0/0/31 {
      congestion-notification-profile fcoe-cnp;
      forwarding-class-set {
```

```
        fcoe-pg {
            output-traffic-control-profile fcoe-tcp;
        }
    }
    unit 0 {
        classifiers {
            ieee-802.1 fcoe-classifier;
        }
    }
}
xe-0/0/32 {
    congestion-notification-profile fcoe-cnp;
    forwarding-class-set {
        fcoe-pg {
            output-traffic-control-profile fcoe-tcp;
        }
    }
    unit 0 {
        classifiers {
            ieee-802.1 fcoe-classifier;
        }
    }
}
xe-0/0/33 {
    congestion-notification-profile fcoe-cnp;
    forwarding-class-set {
        fcoe-pg {
            output-traffic-control-profile fcoe-tcp;
        }
    }
    unit 0 {
        classifiers {
            ieee-802.1 fcoe-classifier;
        }
    }
}
xe-0/0/34 {
    congestion-notification-profile fcoe-cnp;
    forwarding-class-set {
        fcoe-pg {
            output-traffic-control-profile fcoe-tcp;
        }
    }
    unit 0 {
        classifiers {
            ieee-802.1 fcoe-classifier;
        }
    }
}
}
scheduler-maps {
    fcoe-map {
        forwarding-class fcoe scheduler fcoe-sched;
    }
}
schedulers {
```

```
    fcoe-sched {
      transmit-rate 3000000000;
      shaping-rate percent 100;
      priority low;
    }
  }
```

TIP: To quickly configure the interfaces, issue the **load merge terminal** command and then copy the hierarchy and paste it into the switch terminal window.

Verification

To verify that the PFC configuration for FCoE traffic components has been created and is operating properly, perform these tasks:

- Verifying That Priority-Based Flow Control Has Been Enabled on page 192
- Verifying the Ingress Interface PFC Configuration on page 193

Verifying That Priority-Based Flow Control Has Been Enabled

Purpose Verify that PFC is enabled on the FCoE queue to enable lossless transport.

Action List the congestion notification profiles using the operational mode command **show class-of-service congestion-notification**:

```
user@switch> show class-of-service congestion-notification
Type: Input, Name: fcoe-cnp, Index: 51697
Cable Length: 100 m
  Priority      PFC          MRU
  000           Disabled
  001           Disabled
  010           Disabled
  011           Enabled      2500
  100           Disabled
  101           Disabled
  110           Disabled
  111           Disabled
Type: Output
  Priority      Flow-Control-Queues
  000
                0
  001
                1
  010
                2
  011
                3
  100
                4
  101
                5
  110
                6
```



```
111
            7
```

Meaning The **show class-of-service congestion-notification** operational command lists all of the congestion notification profiles and which IEEE 802.1p code points have PFC enabled. The command output shows that PFC is enabled on code point **011** for the **fcoe-cnp** congestion notification profile.

The command also shows the default cable length (100 meters), the default maximum receive unit (2500 bytes), and the default mapping of priorities to output queues because this example does not include configuring these options.

Verifying the Ingress Interface PFC Configuration

Purpose Verify that the classifier **fcoe-classifier** and the congestion notification profile **fcoe-cnp** are configured on ingress interfaces **xe-0/0/31**, **xe-0/0/32**, **xe-0/0/33**, and **xe-0/0/34**.

Action List the ingress interfaces using the operational mode command **show configuration class-of-service interfaces**:

```
user@switch> show configuration class-of-service interfaces xe-0/0/31
congestion-notification-profile fcoe-cnp;
unit 0 {
    classifiers {
        ieee-802.1 fcoe-classifier;
    }
}

user@switch> show configuration class-of-service interfaces xe-0/0/32
congestion-notification-profile fcoe-cnp;
unit 0 {
    classifiers {
        ieee-802.1 fcoe-classifier;
    }
}

user@switch> show configuration class-of-service interfaces xe-0/0/33
congestion-notification-profile fcoe-cnp;
unit 0 {
    classifiers {
        ieee-802.1 fcoe-classifier;
    }
}

user@switch> show configuration class-of-service interfaces xe-0/0/34
congestion-notification-profile fcoe-cnp;
unit 0 {
    classifiers {
        ieee-802.1 fcoe-classifier;
    }
}
```

Meaning The **show configuration class-of-service interfaces** commands list the congestion notification profile that is mapped to the interface (**fcoe-cnp**) and the IEEE 802.1p classifier associated with the interface (**fcoe-classifier**).

Related Documentation

- *Example: Configuring CoS Hierarchical Port Scheduling (ETS)*
- *Configuring CoS PFC (Congestion Notification Profiles)*
- *Overview of CoS Changes Introduced in Junos OS Release 12.2*
- Understanding CoS Flow Control (Ethernet PAUSE and PFC) on page 121

Example: Configuring CoS for FCoE Transit Switch Traffic Across an MC-LAG

Multichassis link aggregation groups (MC-LAGs) provide redundancy and load balancing between two QFX Series switches, multihoming support for client devices such as servers, and a loop-free Layer 2 network without running Spanning Tree Protocol (STP).

NOTE: This example uses Junos OS without support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see *Example: Configuring CoS for FCoE Transit Switch Traffic Across an MC-LAG*.

You can use an MC-LAG to provide a redundant aggregation layer for Fiber Channel over Ethernet (FCoE) traffic in an *inverted-U* topology. To support lossless transport of FCoE traffic across an MC-LAG, you must configure the appropriate class of service (CoS) on both of the QFX Series switches with MC-LAG port members. The CoS configuration must be the same on both of the MC-LAG switches because an MC-LAG does not carry forwarding class and IEEE 802.1p priority information.

NOTE: This example describes how to configure CoS to provide lossless transport for FCoE traffic across an MC-LAG that connects two QFX Series switches. It also describes how to configure CoS on the FCoE transit switches that connect FCoE hosts to the QFX Series switches that form the MC-LAG.

This example does *not* describe how to configure the MC-LAG itself. For a detailed example of MC-LAG configuration, see *Example: Configuring Multichassis Link Aggregation*. However, this example includes a subset of MC-LAG configuration that only shows how to configure interface membership in the MC-LAG.

Ports that are part of an FCoE-FC gateway configuration (a virtual FCoE-FC gateway fabric) do not support MC-LAGs. Ports that are members of an MC-LAG act as FCoE passthrough transit switch ports.

QFX Series switches support MC-LAGs. QFabric system Node devices do not support MC-LAGs, and QFX3500 and QFX3600 Virtual Chassis switches do not support FCoE.

This topic describes:

- Requirements on page 195
- Overview on page 195

- Configuration on page 200
- Verification on page 208

Requirements

This example uses the following hardware and software components:

- Two Juniper Networks QFX3500 Switches that form an MC-LAG for FCoE traffic.
- Two Juniper Networks QFX3500 Switches that provide FCoE server access in transit switch mode and that connect to the MC-LAG switches. These switches can be standalone QFX3500 switches or they can be Node devices in a QFabric system.
- FCoE servers (or other FCoE hosts) connected to the transit switches.
- Junos OS Release 12.2 or later for the QFX Series.

Overview

FCoE traffic requires lossless transport. This example shows you how to:

- Configure CoS for FCoE traffic on the two QFX3500 switches that form the MC-LAG, including priority-based flow control (PFC) and enhanced transmission selection (ETS; hierarchical scheduling of resources for the FCoE forwarding class priority and for the forwarding class set priority group).

  NOTE: **Configuring or changing PFC on an interface blocks the entire port until the PFC change is completed. After a PFC change is completed, the port is unblocked and traffic resumes. Blocking the port stops ingress and egress traffic, and causes packet loss on all queues on the port until the port is unblocked.**

- Configure CoS for FCoE on the two FCoE transit switches that connect FCoE hosts to the MC-LAG switches and enable FIP snooping on the FCoE VLAN at the FCoE transit switch access ports.
- Disable IGMP snooping on the FCoE VLAN.

  NOTE: **This is only necessary if IGMP snooping is enabled on the VLAN. Before Junos OS Release 13.2, IGMP snooping was enabled by default on VLANs. Beginning with Junos OS Release 13.2, IGMP snooping is enabled by default only on the default VLAN.**

- Configure the appropriate port mode, MTU, and FCoE trusted or untrusted state for each interface to support lossless FCoE transport.

Topology

QFX3500 switches that act as transit switches support MC-LAGs for FCoE traffic in an inverted-U network topology, as shown in Figure 12 on page 196.

Figure 12: Supported Topology for an MC-LAG on an FCoE Transit Switch

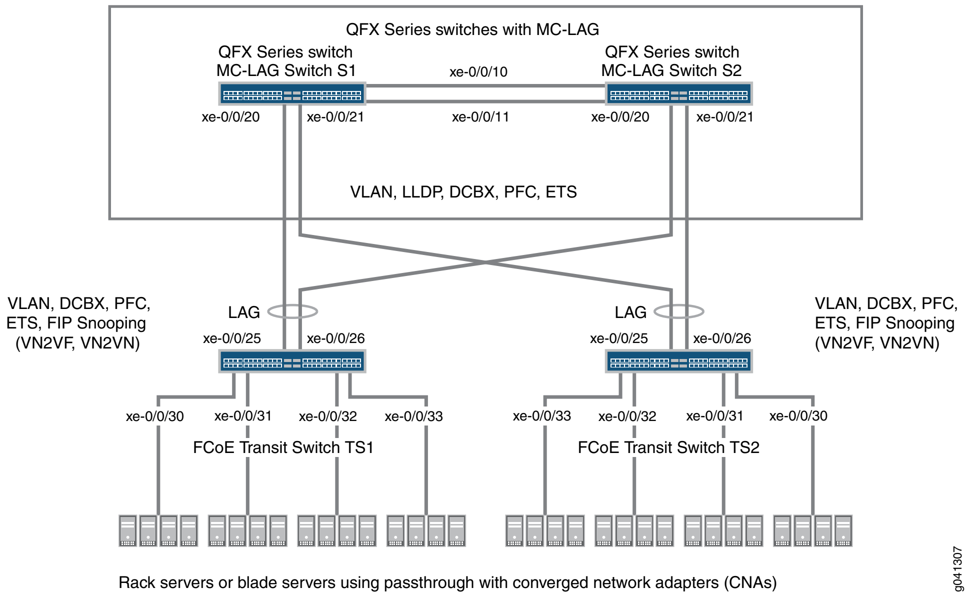


Table 24 on page 196 shows the configuration components for this example.

Table 24: Components of the CoS for FCoE Traffic Across an MC-LAG Configuration Topology

Component	Settings
Hardware	Four QFX3500 switches (two to form the MC-LAG as passthrough transit switches and two transit switches for FCoE access).
Forwarding class (all switches)	Default **fcoe** forwarding class.
Classifier (forwarding class mapping of incoming traffic to IEEE priority)	Default IEEE 802.1p trusted classifier on all FCoE interfaces.

Table 24: Components of the CoS for FCoE Traffic Across an MC-LAG Configuration Topology *(continued)*

Component	Settings
LAGs and MC-LAG	S1—Ports xe-0/0/10 and x-0/0/11 are members of LAG **ae0**, which connects Switch S1 to Switch S2. Ports xe-0/0/20 and xe-0/0/21 are members of MC-LAG **ae1**. All ports are configured in **trunk** port mode, as **fcoe-trusted**, and with an MTU of **2180**. S2—Ports xe-0/0/10 and x-0/0/11 are members of LAG **ae0**, which connects Switch S2 to Switch S1. Ports xe-0/0/20 and xe-0/0/21 are members of MC-LAG **ae1**. All ports are configured in **trunk** port mode, as **fcoe-trusted**, and with an MTU of **2180**. NOTE: Ports xe-0/0/20 and xe-0/0/21 on Switches S1 and S2 are the members of the MC-LAG. TS1—Ports xe-0/0/25 and x-0/0/26 are members of LAG **ae1**, configured in **trunk** port mode, as **fcoe-trusted**, and with an MTU of **2180**. Ports xe-0/0/30, xe-0/0/31, xe-0/0/32, and xe-0/0/33 are configured in **tagged-access** port mode, with an MTU of **2180**. TS2—Ports xe-0/0/25 and x-0/0/26 are members of LAG **ae1**, configured in **trunk** port mode, as **fcoe-trusted**, and with an MTU of **2180**. Ports xe-0/0/30, xe-0/0/31, xe-0/0/32, and xe-0/0/33 are configured in **tagged-access** port mode, with an MTU of **2180**.
FCoE queue scheduler (all switches)	**fcoe-sched**: Minimum bandwidth **3g** Maximum bandwidth **100%** Priority **low**
Forwarding class-to-scheduler mapping (all switches)	Scheduler map **fcoe-map**: Forwarding class **fcoe** Scheduler **fcoe-sched**
Forwarding class set (FCoE priority group, all switches)	**fcoe-pg**: Forwarding class **fcoe** Egress interfaces: • S1—LAG **ae0** and MC-LAG **ae1** • S2—LAG **ae0** and MC-LAG **ae1** • TS1—LAG **ae1**, interfaces **xe-0/0/30**, **xe-0/0/31**, **xe-0/0/32**, and **xe-0/0/33** • TS2—LAG **ae1**, interfaces **xe-0/0/30**, **xe-0/0/31**, **xe-0/0/32**, and **xe-0/0/33**
Traffic control profile (all switches)	**fcoe-tcp**: Scheduler map **fcoe-map** Minimum bandwidth **3g** Maximum bandwidth **100%**

Table 24: Components of the CoS for FCoE Traffic Across an MC-LAG Configuration Topology *(continued)*

Component	Settings
PFC congestion notification profile (all switches)	**fcoe-cnp**: Code point **011** Ingress interfaces: • S1—LAG **ae0** and MC-LAG **ae1** • S2—LAG **ae0** and MC-LAG **ae1** • TS1—LAG **ae1**, interfaces **xe-0/0/30**, **xe-0/0/31**, **xe-0/0/32**, and **xe-0/0/33** • TS2—LAG **ae1**, interfaces **xe-0/0/30**, **xe-0/0/31**, **xe-0/0/32**, and **xe-0/0/33**
FCoE VLAN name and tag ID	Name—**fcoe_vlan** ID—**100** Include the FCoE VLAN on the interfaces that carry FCoE traffic on all four switches. Disable IGMP snooping on the interfaces that belong to the FCoE VLAN on all four switches.
FIP snooping	Enable FIP snooping on Transit Switches TS1 and TS2 on the FCoE VLAN. Configure the LAG interfaces that connect to the MC-LAG switches as FCoE trusted interfaces so that they do not perform FIP snooping. This example enables VN2VN_Port FIP snooping on the FCoE transit switch interfaces connected to the FCoE servers. The example is equally valid with VN2VF_Port FIP snooping enabled on the transit switch access ports. The method of FIP snooping you enable depends on your network configuration.

NOTE: This example uses the default IEEE 802.1p trusted BA classifier, which is automatically applied to trunk mode and tagged access mode ports if you do not apply an explicitly configured classifier.

To configure CoS for FCoE traffic across an MC-LAG:

- Use the default FCoE forwarding class and forwarding-class-to-queue mapping (do not explicitly configure the FCoE forwarding class or output queue). The default FCoE forwarding class is **fcoe**, and the default output queue is queue **3**.

NOTE: In Junos OS Release 12.2, traffic mapped to explicitly configured forwarding classes, even lossless forwarding classes such as **fcoe**, is treated as lossy (**best-effort**) traffic and does *not* receive lossless treatment. To receive lossless treatment in Release 12.2, traffic must use one of the default lossless forwarding classes (**fcoe** or **no-loss**).

In Junos OS Release 12.3 and later, you can include the *no-loss* packet drop attribute in the explicit forwarding class configuration to configure a lossless forwarding class.

- Use the default trusted BA classifier, which maps incoming packets to forwarding classes by the IEEE 802.1p code point (CoS priority) of the packet. The trusted classifier is the default classifier for interfaces in trunk and tagged-access port modes. The default trusted classifier maps incoming packets with the IEEE 802.1p code point 3 (**011**) to the FCoE forwarding class. If you choose to configure the BA classifier instead of using the default classifier, you must ensure that FCoE traffic is classified into forwarding classes in exactly the same way on both MC-LAG switches. Using the default classifier ensures consistent classifier configuration on the MC-LAG ports.
- Configure a congestion notification profile that enables PFC on the FCoE code point (code point **011** in this example). The congestion notification profile configuration must be the same on both MC-LAG switches.
- Apply the congestion notification profile to the interfaces.
- Configure enhanced transmission selection (ETS, also known as hierarchical scheduling) on the interfaces to provide the bandwidth required for lossless FCoE transport. Configuring ETS includes configuring bandwidth scheduling for the FCoE forwarding class, a forwarding class set (priority group) that includes the FCoE forwarding class, and a traffic control profile to assign bandwidth to the forwarding class set that includes FCoE traffic.
- Apply the ETS scheduling to the interfaces.
- Configure the port mode, MTU, and FCoE trusted or untrusted state for each interface to support lossless FCoE transport.

In addition, this example describes how to enable FIP snooping on the Transit Switch TS1 and TS2 ports that are connected to the FCoE servers and how to disable IGMP snooping on the FCoE VLAN. To provide secure access, FIP snooping must be enabled on the FCoE access ports.

This example focuses on the CoS configuration to support lossless FCoE transport across an MC-LAG. This example does not describe how to configure the properties of MC-LAGs and LAGs, although it does show you how to configure the port characteristics required to support lossless transport and how to assign interfaces to the MC-LAG and to the LAGs.

Before you configure CoS, configure:

- The MC-LAGs that connect Switches S1 and S2 to Switches TS1 and TS2. (*Example: Configuring Multichassis Link Aggregation* describes how to configure MC-LAGs.)
- The LAGs that connect the Transit Switches TS1 and TS2 to MC-LAG Switches S1 and S2. (*Configuring Link Aggregation* describes how to configure LAGs.)
- The LAG that connects Switch S1 to Switch S2.

Configuration

To configure CoS for lossless FCoE transport across an MC-LAG, perform these tasks:

- Configuring MC-LAG Switches S1 and S2 on page 202
- Configuring FCoE Transit Switches TS1 and TS2 on page 203
- Results on page 205

CLI Quick Configuration

To quickly configure CoS for lossless FCoE transport across an MC-LAG, copy the following commands, paste them in a text file, remove line breaks, change variables and details to match your network configuration, and then copy and paste the commands into the CLI for MC-LAG Switch S1 and MC-LAG Switch S2 at the **[edit]** hierarchy level. The configurations on Switches S1 and S2 are identical because the CoS configuration must be identical, and because this example uses the same ports on both switches.

Switch S1 and Switch S2

```
set class-of-service schedulers fcoe-sched priority low transmit-rate 3g
set class-of-service schedulers fcoe-sched shaping-rate percent 100
set class-of-service scheduler-maps fcoe-map forwarding-class fcoe scheduler fcoe-sched
set class-of-service forwarding-class-sets fcoe-pg class fcoe
set class-of-service traffic-control-profiles fcoe-tcp scheduler-map fcoe-map guaranteed-rate
3g
set class-of-service traffic-control-profiles fcoe-tcp shaping-rate percent 100
set class-of-service interfaces ae0 forwarding-class-set fcoe-pg output-traffic-control-profile
fcoe-tcp
set class-of-service interfaces ae1 forwarding-class-set fcoe-pg output-traffic-control-profile
fcoe-tcp
set class-of-service congestion-notification-profile fcoe-cnp input ieee-802.1 code-point 011 pfc
set class-of-service interfaces ae0 congestion-notification-profile fcoe-cnp
set class-of-service interfaces ae1 congestion-notification-profile fcoe-cnp
set vlans fcoe_vlan vlan-id 100
set protocols igmp-snooping vlan fcoe_vlan disable
set interfaces xe-0/0/10 ether-options 802.3ad ae0
set interfaces xe-0/0/11 ether-options 802.3ad ae0
set interfaces xe-0/0/20 ether-options 802.3ad ae1
set interfaces xe-0/0/21 ether-options 802.3ad ae1
set interfaces ae0 unit 0 family ethernet-switching port-mode trunk vlan members fcoe_vlan
set interfaces ae1 unit 0 family ethernet-switching port-mode trunk vlan members fcoe_vlan
set interfaces ae0 mtu 2180
set interfaces ae1 mtu 2180
set ethernet-switching-options secure-access-port interface ae0 fcoe-trusted
```


set ethernet-switching-options secure-access-port interface ae1 fcoe-trusted

To quickly configure CoS for lossless FCoE transport across an MC-LAG, copy the following commands, paste them in a text file, remove line breaks, change variables and details to match your network configuration, and then copy and paste the commands into the CLI for Transit Switch TS1 and Transit Switch TS2 at the **[edit]** hierarchy level. The configurations on Switches TS1 and TS2 are identical because the CoS configuration must be identical, and because this example uses the same ports on both switches.

Switch TS1 and Switch TS2

```
set class-of-service schedulers fcoe-sched priority low transmit-rate 3g
set class-of-service schedulers fcoe-sched shaping-rate percent 100
set class-of-service scheduler-maps fcoe-map forwarding-class fcoe scheduler fcoe-sched
set class-of-service forwarding-class-sets fcoe-pg class fcoe
set class-of-service traffic-control-profiles fcoe-tcp scheduler-map fcoe-map guaranteed-rate
3g
set class-of-service traffic-control-profiles fcoe-tcp shaping-rate percent 100
set class-of-service interfaces ae1 forwarding-class-set fcoe-pg output-traffic-control-profile
fcoe-tcp
set class-of-service interfaces xe-0/0/30 forwarding-class-set fcoe-pg
output-traffic-control-profile fcoe-tcp
set class-of-service interfaces xe-0/0/31 forwarding-class-set fcoe-pg
output-traffic-control-profile fcoe-tcp
set class-of-service interfaces xe-0/0/32 forwarding-class-set fcoe-pg
output-traffic-control-profile fcoe-tcp
set class-of-service interfaces xe-0/0/33 forwarding-class-set fcoe-pg
output-traffic-control-profile fcoe-tcp
set class-of-service congestion-notification-profile fcoe-cnp input ieee-802.1 code-point 011 pfc
set class-of-service interfaces ae1 congestion-notification-profile fcoe-cnp
set class-of-service interfaces xe-0/0/30 congestion-notification-profile fcoe-cnp
set class-of-service interfaces xe-0/0/31 congestion-notification-profile fcoe-cnp
set class-of-service interfaces xe-0/0/32 congestion-notification-profile fcoe-cnp
set class-of-service interfaces xe-0/0/33 congestion-notification-profile fcoe-cnp
set vlans fcoe_vlan vlan-id 100
set protocols igmp-snooping vlan fcoe_vlan disable
set interfaces xe-0/0/25 ether-options 802.3ad ae1
set interfaces xe-0/0/26 ether-options 802.3ad ae1
set interfaces ae1 unit 0 family ethernet-switching port-mode trunk vlan members fcoe_vlan
set interfaces xe-0/0/30 unit 0 family ethernet-switching port-mode tagged-access vlan members
fcoe_vlan
set interfaces xe-0/0/31 unit 0 family ethernet-switching port-mode tagged-access vlan members
fcoe_vlan
set interfaces xe-0/0/32 unit 0 family ethernet-switching port-mode tagged-access vlan members
fcoe_vlan
set interfaces xe-0/0/33 unit 0 family ethernet-switching port-mode tagged-access vlan members
fcoe_vlan
set interfaces ae1 mtu 2180
set interfaces xe-0/0/30 mtu 2180
set interfaces xe-0/0/31 mtu 2180
set interfaces xe-0/0/32 mtu 2180
set interfaces xe-0/0/33 mtu 2180
set ethernet-switching-options secure-access-port interface ae1 fcoe-trusted
set ethernet-switching-options secure-access-port vlan fcoe_vlan examine-fip examine-vn2v2
beacon-period 90000
```

Configuring MC-LAG Switches S1 and S2

Step-by-Step Procedure

To configure CoS resource scheduling (ETS), PFC, the FCoE VLAN, and the LAG and MC-LAG interface membership and characteristics to support lossless FCoE transport across an MC-LAG (this example uses the default **fcoe** forwarding class and the default classifier to map incoming FCoE traffic to the FCoE IEEE 802.1p code point **011**, so you do not configure them):

1. Configure output scheduling for the FCoE queue:

   ```
   [edit class-of-service]
   user@switch# set schedulers fcoe-sched priority low transmit-rate 3g
   user@switch# set schedulers fcoe-sched shaping-rate percent 100
   ```

2. Map the FCoE forwarding class to the FCoE scheduler (**fcoe-sched**):

   ```
   [edit class-of-service]
   user@switch# set scheduler-maps fcoe-map forwarding-class fcoe scheduler fcoe-sched
   ```

3. Configure the forwarding class set (**fcoe-pg**) for the FCoE traffic:

   ```
   [edit class-of-service]
   user@switch# set forwarding-class-sets fcoe-pg class fcoe
   ```

4. Define the traffic control profile (**fcoe-tcp**) to use on the FCoE forwarding class set:

   ```
   [edit class-of-service]
   user@switch# set traffic-control-profiles fcoe-tcp scheduler-map fcoe-map
   guaranteed-rate 3g
   user@switch# set traffic-control-profiles fcoe-tcp shaping-rate percent 100
   ```

5. Apply the FCoE forwarding class set and traffic control profile to the LAG and MC-LAG interfaces:

   ```
   [edit class-of-service]
   user@switch# set interfaces ae0 forwarding-class-set fcoe-pg output-traffic-control-profile
   fcoe-tcp
   user@switch# set interfaces ae1 forwarding-class-set fcoe-pg output-traffic-control-profile
   fcoe-tcp
   ```

6. Enable PFC on the FCoE priority by creating a congestion notification profile (**fcoe-cnp**) that applies FCoE to the IEEE 802.1 code point **011**:

   ```
   [edit class-of-service]
   user@switch# set congestion-notification-profile fcoe-cnp input ieee-802.1 code-point
   011 pfc
   ```

7. Apply the PFC configuration to the LAG and MC-LAG interfaces:

   ```
   [edit class-of-service]
   user@switch# set interfaces ae0 congestion-notification-profile fcoe-cnp
   user@switch# set interfaces ae1 congestion-notification-profile fcoe-cnp
   ```

8. Configure the VLAN for FCoE traffic (**fcoe_vlan**):

   ```
   [edit vlans]
   user@switch# set fcoe_vlan vlan-id 100
   ```

9. Disable IGMP snooping on the FCoE VLAN:

   ```
   [edit protocols]
   user@switch# set igmp-snooping vlan fcoe_vlan disable
   ```


10. Add the member interfaces to the LAG between the two MC-LAG switches:

```
[edit interfaces]
user@switch# set xe-0/0/10 ether-options 802.3ad ae0
user@switch# set xe-0/0/11 ether-options 802.3ad ae0
```

11. Add the member interfaces to the MC-LAG:

```
[edit interfaces]
user@switch# set xe-0/0/20 ether-options 802.3ad ae1
user@switch# set xe-0/0/21 ether-options 802.3ad ae1
```

12. Configure the port mode as **trunk** and membership in the FCoE VLAN (**fcoe_vlan**) for the LAG (**ae0**) and for the MC-LAG (**ae1**):

```
[edit interfaces]
user@switch# set interfaces ae0 unit 0 family ethernet-switching port-mode trunk vlan
members fcoe_vlan
user@switch# set interfaces ae1 unit 0 family ethernet-switching port-mode trunk vlan
members fcoe_vlan
```

13. Set the MTU to **2180** for the LAG and MC-LAG interfaces. 2180 bytes is the minimum size required to handle FCoE packets because of the payload and header sizes; you can configure the MTU to a higher number of bytes if desired, but not less than 2180 bytes:

```
[edit interfaces]
user@switch# set ae0 mtu 2180
user@switch# set ae1 mtu 2180
```

14. Set the LAG and MC-LAG interfaces as FCoE trusted ports. Ports that connect to other switches should be trusted and should not perform FIP snooping:

```
[edit]
user@switch# set ethernet-switching-options secure-access-port interface ae0 fcoe-trusted
user@switch# set ethernet-switching-options secure-access-port interface ae1 fcoe-trusted
```

Configuring FCoE Transit Switches TS1 and TS2

Step-by-Step Procedure

The CoS configuration on FCoE Transit Switches TS1 and TS2 is similar to the CoS configuration on MC-LAG Switches S1 and S2. However, the port configurations differ, and you must enable FIP snooping on the Switch TS1 and Switch TS2 FCoE access ports.

To configure resource scheduling (ETS), PFC, the FCoE VLAN, and the LAG interface membership and characteristics to support lossless FCoE transport across the MC-LAG (this example uses the default **fcoe** forwarding class and the default classifier to map incoming FCoE traffic to the FCoE IEEE 802.1p code point **011**, so you do not configure them):

1. Configure output scheduling for the FCoE queue:

```
[edit class-of-service]
user@switch# set schedulers fcoe-sched priority low transmit-rate 3g
user@switch# set schedulers fcoe-sched shaping-rate percent 100
```

2. Map the FCoE forwarding class to the FCoE scheduler (**fcoe-sched**):

```
[edit class-of-service]
user@switch# set scheduler-maps fcoe-map forwarding-class fcoe scheduler fcoe-sched
```

3. Configure the forwarding class set (**fcoe-pg**) for the FCoE traffic:

```
[edit class-of-service]
user@switch# set forwarding-class-sets fcoe-pg class fcoe
```

4. Define the traffic control profile (**fcoe-tcp**) to use on the FCoE forwarding class set:

```
[edit class-of-service]
user@switch# set traffic-control-profiles fcoe-tcp scheduler-map fcoe-map
guaranteed-rate 3g
user@switch# set traffic-control-profiles fcoe-tcp shaping-rate percent 100
```

5. Apply the FCoE forwarding class set and traffic control profile to the LAG interface and to the FCoE access interfaces:

```
[edit class-of-service]
user@switch# set interfaces ae1 forwarding-class-set fcoe-pg output-traffic-control-profile
fcoe-tcp
user@switch# set class-of-service interfaces xe-0/0/30 forwarding-class-set fcoe-pg
output-traffic-control-profile fcoe-tcp
user@switch# set class-of-service interfaces xe-0/0/31 forwarding-class-set fcoe-pg
output-traffic-control-profile fcoe-tcp
user@switch# set class-of-service interfaces xe-0/0/32 forwarding-class-set fcoe-pg
output-traffic-control-profile fcoe-tcp
user@switch# set class-of-service interfaces xe-0/0/33 forwarding-class-set fcoe-pg
output-traffic-control-profile fcoe-tcp
```

6. Enable PFC on the FCoE priority by creating a congestion notification profile (**fcoe-cnp**) that applies FCoE to the IEEE 802.1 code point **011**:

```
[edit class-of-service]
user@switch# set congestion-notification-profile fcoe-cnp input ieee-802.1 code-point
011 pfc
```

7. Apply the PFC configuration to the LAG interface and to the FCoE access interfaces:

```
[edit class-of-service]
user@switch# set interfaces ae1 congestion-notification-profile fcoe-cnp
user@switch# set class-of-service interfaces xe-0/0/30 congestion-notification-profile
fcoe-cnp
user@switch# set class-of-service interfaces xe-0/0/31 congestion-notification-profile
fcoe-cnp
user@switch# set class-of-service interfaces xe-0/0/32 congestion-notification-profile
fcoe-cnp
user@switch# set class-of-service interfaces xe-0/0/33 congestion-notification-profile
fcoe-cnp
```

8. Configure the VLAN for FCoE traffic (**fcoe_vlan**):

```
[edit vlans]
user@switch# set fcoe_vlan vlan-id 100
```

9. Disable IGMP snooping on the FCoE VLAN:

```
[edit protocols]
user@switch# set igmp-snooping vlan fcoe_vlan disable
```

10. Add the member interfaces to the LAG:

```
[edit interfaces]
user@switch# set xe-0/0/25 ether-options 802.3ad ae1
user@switch# set xe-0/0/26 ether-options 802.3ad ae1
```

11. On the LAG (**ae1**), configure the port mode as **trunk** and membership in the FCoE VLAN (**fcoe_vlan**):


```
[edit interfaces]
user@switch# set interfaces ae1 unit 0 family ethernet-switching port-mode trunk vlan
members fcoe_vlan
```

12. On the FCoE access interfaces (**xe-0/0/30**, **xe-0/0/31**, **xe-0/0/32**, **xe-0/0/33**), configure the port mode as **tagged-access** and membership in the FCoE VLAN (**fcoe_vlan**):

```
[edit interfaces]
user@switch# set interfaces xe-0/0/30 unit 0 family ethernet-switching port-mode
tagged-access vlan members fcoe_vlan
user@switch# set interfaces xe-0/0/31 unit 0 family ethernet-switching port-mode
tagged-access vlan members fcoe_vlan
user@switch# set interfaces xe-0/0/32 unit 0 family ethernet-switching port-mode
tagged-access vlan members fcoe_vlan
user@switch# set interfaces xe-0/0/33 unit 0 family ethernet-switching port-mode
tagged-access vlan members fcoe_vlan
```

13. Set the MTU to **2180** for the LAG and FCoE access interfaces. 2180 bytes is the minimum size required to handle FCoE packets because of the payload and header sizes; you can configure the MTU to a higher number of bytes if desired, but not less than 2180 bytes:

```
[edit interfaces]
user@switch# set ae1 mtu 2180
user@switch# set xe-0/0/30 mtu 2180
user@switch# set xe-0/0/31 mtu 2180
user@switch# set xe-0/0/32 mtu 2180
user@switch# set xe-0/0/33 mtu 2180
```

14. Set the LAG interface as an FCoE trusted port. Ports that connect to other switches should be trusted and should not perform FIP snooping:

```
[edit]
user@switch# set ethernet-switching-options secure-access-port interface ae1 fcoe-trusted
```

NOTE: Access ports xe-0/0/30, xe-0/0/31, xe-0/0/32, and xe-0/0/33 are not configured as FCoE trusted ports. The access ports remain in the default state as untrusted ports because they connect directly to FCoE devices and must perform FIP snooping to ensure network security.

15. Enable FIP snooping on the FCoE VLAN to prevent unauthorized FCoE network access (this example uses VN2VN_Port FIP snooping; the example is equally valid if you use VN2VF_Port FIP snooping):

```
[edit]
user@switch# set ethernet-switching-options secure-access-port vlan fcoe_vlan
examine-fip examine-vn2vn beacon-period 90000
```

Results

Display the results of the CoS configuration on MC-LAG Switch S1 and on MC-LAG Switch S2 (the results on both switches are the same):

```
user@switch> show configuration class-of-service
```

```
traffic-control-profiles {
    fcoe-tcp {
        scheduler-map fcoe-map;
        shaping-rate percent 100;
        guaranteed-rate 3000000000;
    }
}
forwarding-class-sets {
    fcoe-pg {
        class fcoe;
    }
}
congestion-notification-profile {
    fcoe-cnp {
        input {
            ieee-802.1 {
                code-point 011 {
                    pfc;
                }
            }
        }
    }
}
interfaces {
    ae0 {
        forwarding-class-set {
            fcoe-pg {
                output-traffic-control-profile fcoe-tcp;
            }
        }
        congestion-notification-profile fcoe-cnp;
    }
    ae1 {
        forwarding-class-set {
            fcoe-pg {
                output-traffic-control-profile fcoe-tcp;
            }
        }
        congestion-notification-profile fcoe-cnp;
    }
}
scheduler-maps {
    fcoe-map {
        forwarding-class fcoe scheduler fcoe-sched;
    }
}
schedulers {
    fcoe-sched {
        transmit-rate 3000000000;
        shaping-rate percent 100;
        priority low;
    }
}
```

NOTE: The forwarding class and classifier configurations are not shown because the show command does not display default portions of the configuration.

For MC-LAG verification commands, see *Example: Configuring Multichassis Link Aggregation*.

Display the results of the CoS configuration on FCoE Transit Switch TS1 and on FCoE Transit Switch TS2 (the results on both transit switches are the same):

```
user@switch> show configuration class-of-service
traffic-control-profiles {
  fcoe-tcp {
    scheduler-map fcoe-map;
    shaping-rate percent 100;
    guaranteed-rate 3000000000;
  }
}
forwarding-class-sets {
  fcoe-pg {
    class fcoe;
  }
}
congestion-notification-profile {
  fcoe-cnp {
    input {
      ieee-802.1 {
        code-point 011 {
          pfc;
        }
      }
    }
  }
}
interfaces {
  xe-0/0/30 {
    forwarding-class-set {
      fcoe-pg {
        output-traffic-control-profile fcoe-tcp;
      }
    }
    congestion-notification-profile fcoe-cnp;
  }
  xe-0/0/31 {
    forwarding-class-set {
      fcoe-pg {
        output-traffic-control-profile fcoe-tcp;
      }
    }
    congestion-notification-profile fcoe-cnp;
  }
  xe-0/0/32 {
```

```
        forwarding-class-set {
          fcoe-pg {
            output-traffic-control-profile fcoe-tcp;
          }
        }
        congestion-notification-profile fcoe-cnp;
      }
      xe-0/0/33 {
        forwarding-class-set {
          fcoe-pg {
            output-traffic-control-profile fcoe-tcp;
          }
        }
        congestion-notification-profile fcoe-cnp;
      }
      ae1 {
        forwarding-class-set {
          fcoe-pg {
            output-traffic-control-profile fcoe-tcp;
          }
        }
        congestion-notification-profile fcoe-cnp;
      }
    }
    scheduler-maps {
      fcoe-map {
        forwarding-class fcoe scheduler fcoe-sched;
      }
    }
    schedulers {
      fcoe-sched {
        transmit-rate 3000000000;
        shaping-rate percent 100;
        priority low;
      }
    }
```

NOTE: The forwarding class and classifier configurations are not shown because the show command does not display default portions of the configuration.

Verification

To verify that the CoS components and FIP snooping have been configured and are operating properly, perform these tasks. Because this example uses the default **fcoe** forwarding class and the default IEEE 802.1p trusted classifier, the verification of those configurations is not shown:

- Verifying That the Output Queue Schedulers Have Been Created on page 209
- Verifying That the Priority Group Output Scheduler (Traffic Control Profile) Has Been Created on page 210

- Verifying That the Forwarding Class Set (Priority Group) Has Been Created on page 210
- Verifying That Priority-Based Flow Control Has Been Enabled on page 211
- Verifying That the Interface Class of Service Configuration Has Been Created on page 211
- Verifying That the Interfaces Are Correctly Configured on page 213
- Verifying That FIP Snooping Is Enabled on the FCoE VLAN on FCoE Transit Switches TS1 and TS2 Access Interfaces on page 216
- Verifying That the FIP Snooping Mode Is Correct on FCoE Transit Switches TS1 and TS2 on page 216
- Verifying That IGMP Snooping Is Disabled on the FCoE VLAN on page 217

Verifying That the Output Queue Schedulers Have Been Created

Purpose Verify that the output queue scheduler for FCoE traffic has the correct bandwidth parameters and priorities, and is mapped to the correct forwarding class (output queue). Queue scheduler verification is the same on each of the four switches.

Action List the scheduler map using the operational mode command **show class-of-service scheduler-map fcoe-map**:

```
user@switch> show class-of-service scheduler-map fcoe-map
Scheduler map: fcoe-map, Index: 9023

  Scheduler: fcoe-sched, Forwarding class: fcoe, Index: 37289
    Transmit rate: 3000000000 bps, Rate Limit: none, Buffer size: remainder,
    Buffer Limit: none, Priority: low
    Excess Priority: unspecified
    Shaping rate: 100 percent,
    drop-profile-map-set-type: mark
    Drop profiles:
      Loss priority   Protocol    Index    Name
      Low             any             1    <default-drop-profile>
      Medium high     any             1    <default-drop-profile>
      High            any             1    <default-drop-profile>
```

Meaning The **show class-of-service scheduler-map fcoe-map** command lists the properties of the scheduler map **fcoe-map**. The command output includes:

- The name of the scheduler map (**fcoe-map**)
- The name of the scheduler (**fcoe-sched**)
- The forwarding classes mapped to the scheduler (**fcoe**)
- The minimum guaranteed queue bandwidth (transmit rate **3000000000 bps**)
- The scheduling priority (**low**)
- The maximum bandwidth in the priority group the queue can consume (shaping rate **100 percent**)
- The drop profile loss priority for each drop profile name. This example does not include drop profiles because you do not apply drop profiles to FCoE traffic.

Verifying That the Priority Group Output Scheduler (Traffic Control Profile) Has Been Created

Purpose Verify that the traffic control profile **fcoe-tcp** has been created with the correct bandwidth parameters and scheduler mapping. Priority group scheduler verification is the same on each of the four switches.

Action List the FCoE traffic control profile properties using the operational mode command **show class-of-service traffic-control-profile fcoe-tcp**:

```
user@switch> show class-of-service traffic-control-profile fcoe-tcp
Traffic control profile: fcoe-tcp, Index: 18303
  Shaping rate: 100 percent
  Scheduler map: fcoe-map
  Guaranteed rate: 3000000000
```

Meaning The **show class-of-service traffic-control-profile fcoe-tcp** command lists all of the configured traffic control profiles. For each traffic control profile, the command output includes:

- The name of the traffic control profile (**fcoe-tcp**)
- The maximum port bandwidth the priority group can consume (shaping rate **100 percent**)
- The scheduler map associated with the traffic control profile (**fcoe-map**)
- The minimum guaranteed priority group port bandwidth (guaranteed rate **3000000000** in bps)

Verifying That the Forwarding Class Set (Priority Group) Has Been Created

Purpose Verify that the FCoE priority group has been created and that the **fcoe** priority (forwarding class) belongs to the FCoE priority group. Forwarding class set verification is the same on each of the four switches.

Action List the forwarding class sets using the operational mode command **show class-of-service forwarding-class-set fcoe-pg**:

```
user@switch> show class-of-service forwarding-class-set fcoe-pg
Forwarding class set: fcoe-pg, Type: normal-type, Forwarding class set index:
31420
  Forwarding class                       Index
  fcoe                                     1
```

Meaning The **show class-of-service forwarding-class-set fcoe-pg** command lists all of the forwarding classes (priorities) that belong to the **fcoe-pg** priority group, and the internal index number of the priority group. The command output shows that the forwarding class set **fcoe-pg** includes the forwarding class **fcoe**.

Verifying That Priority-Based Flow Control Has Been Enabled

Purpose Verify that PFC is enabled on the FCoE code point. PFC verification is the same on each of the four switches.

Action List the FCoE congestion notification profile using the operational mode command **show class-of-service congestion-notification fcoe-cnp**:

```
user@switch> show class-of-service congestion-notification fcoe-cnp
Type: Input, Name: fcoe-cnp, Index: 6879
Cable Length: 100 m
  Priority      PFC           MRU
  000           Disabled
  001           Disabled
  010           Disabled
  011           Enabled       2500
  100           Disabled
  101           Disabled
  110           Disabled
  111           Disabled
Type: Output
  Priority      Flow-Control-Queues
  000
                0
  001
                1
  010
                2
  011
                3
  100
                4
  101
                5
  110
                6
  111
                7
```

Meaning The **show class-of-service congestion-notification fcoe-cnp** command lists all of the IEEE 802.1p code points in the congestion notification profile that have PFC enabled. The command output shows that PFC is enabled on code point **011** (**fcoe** queue) for the **fcoe-cnp** congestion notification profile.

The command also shows the default cable length (100 meters), the default maximum receive unit (2500 bytes), and the default mapping of priorities to output queues because this example does not include configuring these options.

Verifying That the Interface Class of Service Configuration Has Been Created

Purpose Verify that the CoS properties of the interfaces are correct. The verification output on MC-LAG Switches S1 and S2 differs from the output on FCoE Transit Switches TS1 and TS2.

Action List the interface CoS configuration on MC-LAG Switches S1 and S2 using the operational mode command **show configuration class-of-service interfaces**:

```
user@switch> show configuration class-of-service interfaces
ae0 {
    forwarding-class-set {
        fcoe-pg {
            output-traffic-control-profile fcoe-tcp;
        }
    }
    congestion-notification-profile fcoe-cnp;
}

ae1 {
    forwarding-class-set {
        fcoe-pg {
            output-traffic-control-profile fcoe-tcp;
        }
    }
    congestion-notification-profile fcoe-cnp;
}
```

List the interface CoS configuration on FCoE Transit Switches TS1 and TS2 using the operational mode command **show configuration class-of-service interfaces**:

```
user@switch> show configuration class-of-service interfaces
xe-0/0/30 {
    forwarding-class-set {
        fcoe-pg {
            output-traffic-control-profile fcoe-tcp;
        }
    }
    congestion-notification-profile fcoe-cnp;
}
xe-0/0/31 {
    forwarding-class-set {
        fcoe-pg {
            output-traffic-control-profile fcoe-tcp;
        }
    }
    congestion-notification-profile fcoe-cnp;
}
xe-0/0/32 {
    forwarding-class-set {
        fcoe-pg {
            output-traffic-control-profile fcoe-tcp;
        }
    }
    congestion-notification-profile fcoe-cnp;
}
xe-0/0/33 {
    forwarding-class-set {
        fcoe-pg {
            output-traffic-control-profile fcoe-tcp;
        }
    }
    congestion-notification-profile fcoe-cnp;
}
ae1 {
    forwarding-class-set {
```



```
        fcoe-pg {
            output-traffic-control-profile fcoe-tcp;
        }
    }
    congestion-notification-profile fcoe-cnp;
}
```

Meaning The **show configuration class-of-service interfaces** command lists the class of service configuration for all interfaces. For each interface, the command output includes:

- The name of the interface (for example, **ae0** or **xe-0/0/30**)
- The name of the forwarding class set associated with the interface (**fcoe-pg**)
- The name of the traffic control profile associated with the interface (output traffic control profile, **fcoe-tcp**)
- The name of the congestion notification profile associated with the interface (**fcoe-cnp**)

NOTE: Interfaces that are members of a LAG are not shown individually. The LAG or MC-LAG CoS configuration is applied to all interfaces that are members of the LAG or MC-LAG. For example, the interface CoS configuration output on MC-LAG Switches S1 and S2 shows the LAG CoS configuration but does not show the CoS configuration of the member interfaces separately. The interface CoS configuration output on FCoE Transit Switches TS1 and TS2 shows the LAG CoS configuration but also shows the configuration for interfaces xe-0/0/30, xe-0/0/31, xe-0/0/32, and xe-0/0/33, which are not members of a LAG.

Verifying That the Interfaces Are Correctly Configured

Purpose Verify that the LAG membership, MTU, VLAN membership, and port mode of the interfaces are correct. The verification output on MC-LAG Switches S1 and S2 differs from the output on FCoE Transit Switches T1 and T2.

Action List the interface configuration on MC-LAG Switches S1 and S2 using the operational mode command **show configuration interfaces**:

```
user@switch> show configuration interfaces
xe-0/0/10 {
    ether-options {
        802.3ad ae0;
    }
}
xe-0/0/11 {
    ether-options {
        802.3ad ae0;
    }
}
xe-0/0/20 {
    ether-options {
```

```
        802.3ad ae1;
    }
}
xe-0/0/21 {
    ether-options {
        802.3ad ae1;
    }
}
ae0 {
    mtu 2180;
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members fcoe_vlan;
            }
        }
    }
}
ae1 {
    mtu 2180;
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members fcoe_vlan;
            }
        }
    }
}
```

List the interface configuration on FCoE Transit Switches TS1 and TS2 using the operational mode command **show configuration interfaces**:

```
user@switch> show configuration interfaces
xe-0/0/25 {
    ether-options {
        802.3ad ae1;
    }
}
xe-0/0/26 {
    ether-options {
        802.3ad ae1;
    }
}
xe-0/0/30 {
    mtu 2180;
    unit 0 {
        family ethernet-switching {
            port-mode tagged-access;
            vlan {
                members fcoe_vlan;
            }
        }
    }
}
xe-0/0/31 {
    mtu 2180;
    unit 0 {
        family ethernet-switching {
```



```
                port-mode tagged-access;
                vlan {
                    members fcoe_vlan;
                }
            }
        }
    }
    xe-0/0/32 {
        mtu 2180;
        unit 0 {
            family ethernet-switching {
                port-mode tagged-access;
                vlan {
                    members fcoe_vlan;
                }
            }
        }
    }
    xe-0/0/33 {
        mtu 2180;
        unit 0 {
            family ethernet-switching {
                port-mode tagged-access;
                vlan {
                    members fcoe_vlan;
                }
            }
        }
    }

    ae1 {
        mtu 2180;
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members fcoe_vlan;
                }
            }
        }
    }
```

Meaning The **show configuration interfaces** command lists the configuration of each interface by interface name.

For each interface that is a member of a LAG, the command lists only the name of the LAG to which the interface belongs.

For each LAG interface and for each interface that is not a member of a LAG, the command output includes:

- The MTU (**2180**)
- The unit number of the interface (**0**)
- The port mode (**trunk** mode for interfaces that connect two switches, **tagged-access** mode for interfaces that connect to FCoE hosts)
- The name of the VLAN in which the interface is a member (**fcoe_vlan**)

Verifying That FIP Snooping Is Enabled on the FCoE VLAN on FCoE Transit Switches TS1 and TS2 Access Interfaces

Purpose Verify that FIP snooping is enabled on the FCoE VLAN access interfaces. FIP snooping is enabled only on the FCoE access interfaces, so it is enabled only on FCoE Transit Switches TS1 and TS2. FIP snooping is not enabled on MC-LAG Switches S1 and S2 because FIP snooping is done at the Transit Switch TS1 and TS2 FCoE access ports.

Action List the port security configuration on FCoE Transit Switches TS1 and TS2 using the operational mode command **show configuration ethernet-switching-options secure-access-port**:

```
user@switch> show configuration ethernet-switching-options secure-access-port
interface ae1.0 {
    fcoe-trusted;
}
vlan fcoe_vlan {
    examine-fip {
        examine-vn2vn {
            beacon-period 90000;
        }
    }
}
```

Meaning The **show configuration ethernet-switching-options secure-access-port** command lists port security information, including whether a port is trusted. The command output shows that:

- LAG port **ae1.0**, which connects the FCoE transit switch to the MC-LAG switches, is configured as an FCoE trusted interface. FIP snooping is not performed on the member interfaces of the LAG (**xe-0/0/25** and **xe-0/0/26**).
- FIP snooping is enabled (**examine-fip**) on the FCoE VLAN (**fcoe_vlan**), the type of FIP snooping is VN2VN_Port FIP snooping (**examine-vn2vn**) and the beacon period is set to 90000 milliseconds. On Transit Switches TS1 and TS2, all interface members of the FCoE VLAN perform FIP snooping unless the interface is configured as FCoE trusted. On Transit Switches TS1 and TS2, interfaces **xe-0/0/30**, **xe-0/0/31**, **xe-0/0/32**, and **xe-0/0/33** perform FIP snooping because they are not configured as FCoE trusted. The interface members of LAG **ae1** (**xe-0/0/25** and **xe-0/0/26**) do not perform FIP snooping because the LAG is configured as FCoE trusted.

Verifying That the FIP Snooping Mode Is Correct on FCoE Transit Switches TS1 and TS2

Purpose Verify that the FIP snooping mode is correct on the FCoE VLAN. FIP snooping is enabled only on the FCoE access interfaces, so it is enabled only on FCoE Transit Switches TS1 and TS2. FIP snooping is not enabled on MC-LAG Switches S1 and S2 because FIP snooping is done at the Transit Switch TS1 and TS2 FCoE access ports.

Action List the FIP snooping configuration on FCoE Transit Switches TS1 and TS2 using the operational mode command **show fip snooping brief**:

```
user@switch> show fip snooping brief
VLAN: fcoe_vlan,     Mode: VN2VN Snooping
  FC-MAP: 0e:fd:00
…
```

NOTE: The output has been truncated to show only the relevant information.

Meaning The **show fip snooping brief** command lists FIP snooping information, including the FIP snooping VLAN and the FIP snooping mode. The command output shows that:

- The VLAN on which FIP snooping is enabled is **fcoe_vlan**
- The FIP snooping mode is VN2VN_Port FIP snooping (**VN2VN Snooping**)

Verifying That IGMP Snooping Is Disabled on the FCoE VLAN

Purpose Verify that IGMP snooping is disabled on the FCoE VLAN on all four switches.

Action List the IGMP snooping protocol information on each of the four switches using the **show configuration protocols igmp-snooping** command:

```
user@switch> show configuration protocols igmp-snooping
vlan fcoe_vlan {
    disable;
}
```

Meaning The **show configuration protocols igmp-snooping** command lists the IGMP snooping configuration for the VLANs configured on the switch. The command output shows that IGMP snooping is disabled on the FCoE VLAN (**fcoe_vlan**).

Related Documentation

- *Example: Configuring Multichassis Link Aggregation*
- *Configuring Link Aggregation*
- Example: Configuring CoS PFC for FCoE Traffic on page 186
- *Example: Configuring CoS Hierarchical Port Scheduling (ETS)*
- *Understanding Multichassis Link Aggregation*
- Understanding MC-LAGs on an FCoE Transit Switch on page 104

Example: Configuring DCBX Application Protocol TLV Exchange

Data Center Bridging Capability Exchange protocol (DCBX) discovers the data center bridging (DCB) capabilities of connected peers by exchanging application configuration information. DCBX detects feature misconfiguration and mismatches and can configure

DCB on peers. DCBX is an extension of the Link Layer Discovery Protocol (LLDP). LLDP must remain enabled on every interface on which you want to use DCBX.

NOTE: **LLDP and DCBX are enabled by default on all interfaces.**

The switch supports DCBX application protocol exchange for Layer 2 and Layer 4 applications such as the Internet Small Computer System Interface (iSCSI). You specify applications by EtherType (for Layer 2 applications) or by the destination port and protocol (for Layer 4 applications; the protocol can be either TCP or UDP).

The QFX Series handles Fibre Channel over Ethernet (FCoE) application protocol exchange differently than other protocols in some cases:

- If FCoE is the only application for which you want to enable DCBX application protocol TLV exchange on an interface, you do not have to explicitly configure the FCoE application or an application map. By default, the QFX Series exchanges FCoE application protocol TLVs on all interfaces that carry FCoE traffic (traffic mapped to the **fcoe** forwarding class) and have priority-based flow control (PFC) enabled on the FCoE priority (the FCoE IEEE 802.1p code point). The default priority mapping for the FCoE application is IEEE 802.1p code point 011 (the default **fcoe** forwarding class code point).
- If you want an interface to use DCBX to exchange application protocol TLVs for any other applications in addition to FCoE, you must configure the applications (including FCoE), define an application map (including FCoE), and apply the application map to the interface. If you apply an application map to an interface, you must explicitly configure the FCoE application, or the interface does not exchange FCoE application protocol TLVs.

This example shows how to configure interfaces to exchange both Layer 2 and Layer 4 applications by configuring one interface to exchange iSCSI and FCoE application protocol information and configuring another interface to exchange iSCSI and Precision Time Protocol (PTP) application protocol information.

Requirements

This example uses the following hardware and software components:

- Juniper Networks QFX Series device
- Junos OS Release 12.1 or later for the QFX Series

Overview

The switch supports DCBX application protocol exchange for:

- Layer 2 applications, defined by EtherType
- Layer 4 applications, defined by destination port and protocol

NOTE: DCBX also advertises PFC and enhanced transmission selection (ETS) information. See "Configuring DCBX Autonegotiation" on page 306 for how DCBX negotiates and advertises configuration information for these features and for the applications.

DCBX is configured on a per-interface basis for each supported feature or application. For applications that you want to enable for DCBX application protocol exchange, you must:

- Define the application name and configure the EtherType or the destination port and protocol (TCP or UDP) of the application. Use the EtherType for Layer 2 applications, and use the destination port and protocol for Layer 4 protocols.
- Map the application to an IEEE 802.1p code point in an application map.
- Add the application map to DCBX interface.

In addition, for all applications (including FCoE, even when you do not use an application map), you either must create an IEEE 802.1p classifier and apply it to the appropriate ingress interfaces or use the default classifier. A classifier maps the code points of incoming traffic to a forwarding class and a loss priority so that ingress traffic is assigned to the correct class of service (CoS). The forwarding class determines the output queue on the egress interface.

If you do not create classifiers, trunk and tagged-access ports use the unicast IEEE 802.1 default trusted classifier. Table 25 on page 219 shows the default mapping of IEEE 802.1 code-point values to unicast forwarding classes and loss priorities for ports in trunk mode or tagged-access mode. Table 26 on page 220 shows the default untrusted classifier IEEE 802.1 code-point values to unicast forwarding class mapping for ports in access mode.

Table 25: Default IEEE 802.1 Classifiers for Trunk Ports and Tagged-Access Ports (Default Trusted Classifier)

Code Point	Forwarding Class	Loss Priority
be (000)	best-effort	low
be1 (001)	best-effort	low
ef (010)	best-effort	low
ef1 (011)	fcoe	low

Table 25: Default IEEE 802.1 Classifiers for Trunk Ports and Tagged-Access Ports (Default Trusted Classifier) *(continued)*

Code Point	Forwarding Class	Loss Priority
af11 (100)	no-loss	low
af12 (101)	best-effort	low
nc1 (110)	network-control	low
nc2 (111)	network-control	low

Table 26: Default IEEE 802.1 Unicast Classifiers for Access Ports (Default Untrusted Classifier)

Code Point	Forwarding Class	Loss Priority
000	best-effort	low
001	best-effort	low
010	best-effort	low
011	best-effort	low
100	best-effort	low
101	best-effort	low
110	best-effort	low
111	best-effort	low

Topology

This example shows how to configure DCBX application protocol exchange for three protocols (iSCSI, PTP, and FCoE) on two interfaces. One interface exchanges iSCSI and FCoE application protocol information, and the other interface exchanges iSCSI and PTP application protocol information.

NOTE: You must map FCoE traffic to the interfaces on which you want to forward FCoE traffic. You must also enable PFC on the FCoE interfaces and create an ingress classifier for FCoE traffic, or else use the default classifier.

Table 27 on page 221 shows the configuration components for this example.

Table 27: Components of DCBX Application Protocol Exchange Configuration Topology

Component	Settings
Hardware	QFX Series device
LLDP	Enabled by default on Ethernet interfaces
DCBX	Enabled by default on Ethernet interfaces
iSCSI application (Layer 4)	Application name—**iscsi** protocol—**TCP** destination-port—**3260** code-points—**111**
PTP application (Layer 2)	Application name—**ptp** ether-type—**0x88F7** code-points—**001, 101**
FCoE application (Layer 2)	Application name—**fcoe** ether-type—**0x8906** code-points—**011** NOTE: You explicitly configure the FCoE application because you are applying an application map to the interface. When you apply an application map to an interface, all applications must be explicitly configured and included in the application map.
Application maps	**dcbx-iscsi-fcoe-app-map**—Maps the iSCSI and FCoE applications to IEEE 802.1p code points **dcbx-iscsi-ptp-app-map**—Maps iSCSI and PTP applications to IEEE 802.1p code points
Interfaces	**xe-0/0/10**—Configured to exchange FCoE and iSCSI application TLVs (uses application map **dcbx-iscsi-fcoe-app-map**, carries FCoE traffic, and has PFC enabled on the FCoE priority) **xe-0/0/11**—Configured to exchange iSCSI and PTP application TLVs (uses application map **dcbx-iscsi-ptp-app-map**)
PFC congestion notification profile for FCoE application exchange	**fcoe-cnp**: • Code point—**011** • Interface—**xe-0/0/10**

Table 27: Components of DCBX Application Protocol Exchange Configuration Topology *(continued)*

Component	Settings
Behavior aggregate classifiers (map forwarding classes to incoming packets by the packet's IEEE 802.1 code point)	**fcoe-iscsi-cl1**: • Maps the **fcoe** forwarding class to the IEEE 802.1p code point used for the FCoE application (**011**) and a loss priority of **high** • Maps the **network-control** forwarding class to the IEEE 802.1p code point used for the iSCSI application (**111**) and a loss priority of **high** • Applied to interface **xe-0/0/10** **iscsi-ptp-cl2**: • Maps the **network-control** forwarding class to the IEEE 802.1p code point used for the iSCSI application (**111**) and a loss priority of **low** • Maps the **best-effort** forwarding class to the IEEE 802.1p code points used for the PTP application (**001** and **101**) and a loss priority of **low** • Applied to interface **xe-0/0/11**

NOTE: This example does not include scheduling (bandwidth allocation) configuration or lossless configuration for the iSCSI forwarding class.

Configuration

CLI Quick Configuration

To quickly configure DCBX application protocol exchange, copy the following commands, paste them in a text file, remove line breaks, change variables and details to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set applications application iSCSI protocol tcp destination-port 3260
set applications application FCoE ether-type 0x8906
set applications application PTP ether-type 0x88F7
set policy-options application-maps dcbx-iscsi-fcoe-app-map application iSCSI code-points 111
set policy-options application-maps dcbx-iscsi-fcoe-app-map application FCoE code-points 011
set policy-options application-maps dcbx-iscsi-ptp-app-map application iSCSI code-points 111
set policy-options application-maps dcbx-iscsi-ptp-app-map application PTP code-points [001
101]
set protocols dcbx interface xe-0/0/10 application-map dcbx-iscsi-fcoe-app-map
set protocols dcbx interface xe-0/0/11 application-map dcbx-iscsi-ptp-app-map
set class-of-service congestion-notification-profile fcoe-cnp input ieee-802.1 code-point 011 pfc
set class-of-service interfaces xe-0/0/10 congestion-notification-profile fcoe-cnp
set class-of-service classifiers ieee-802.1 fcoe-iscsi-cl1 import default forwarding-class fcoe
loss-priority high code-points 011
set class-of-service classifiers ieee-802.1 fcoe-iscsi-cl1 import default forwarding-class
network-control loss-priority high code-points 111
set class-of-service classifiers ieee-802.1 iscsi-ptp-cl2 import default forwarding-class
network-control loss-priority low code-points 111
set class-of-service classifiers ieee-802.1 iscsi-ptp-cl2 import default forwarding-class best-effort
loss-priority low code-points [001 101]
set class-of-service interfaces xe-0/0/10 unit 0 classifiers ieee-802.1 fcoe-iscsi-cl1
```

set class-of-service interfaces xe-0/0/11 unit 0 classifiers ieee-802.1 iscsi-ptp-cl2

Configuring DCBX Application Protocol TLV Exchange

Step-by-Step Procedure

To define the applications, map the applications to IEEE 802.1p code points, apply the applications to interfaces, and create classifiers for DCBX application protocol exchange:

1. Define the iSCSI application by specifying its protocol and destination port, and define the FCoE and PTP applications by specifying their EtherTypes.

```
[edit applications]
user@switch# set application iSCSI protocol tcp destination-port 3260
user@switch# set application FCoE ether-type 0x8906
user@switch# set application PTP ether-type 0x88F7
```

2. Define an application map that maps the iSCSI and FCoE applications to IEEE 802.1p code points.

```
[edit policy-options]
user@switch# set application-maps dcbx-iscsi-fcoe-app-map application iSCSI code-points 111
user@switch# set application-maps dcbx-iscsi-fcoe-app-map application FCoE code-points 011
```

3. Define the application map that maps the iSCSI and PTP applications to IEEE 802.1p code points.

```
[edit policy-options]
user@switch# set application-maps dcbx-iscsi-ptp-app-map application iSCSI code-points 111
user@switch# set application-maps dcbx-iscsi-ptp-app-map application PTP code-points [001 101]
```

4. Apply the iSCSI and FCoE application map to interface **xe-0/0/10**, and apply the iSCSI and PTP application map to interface **xe-0/0/11**.

```
[edit protocols dcbx]
user@switch# set interface xe-0/0/10 application-map dcbx-iscsi-fcoe-app-map
user@switch# set interface xe-0/0/11 application-map dcbx-iscsi-ptp-app-map
```

5. Create the congestion notification profile to enable PFC on the FCoE code point (**011**), and apply the congestion notification profile to interface **xe-0/0/10**.

```
[edit class-of-service]
user@switch# set congestion-notification-profile fcoe-cnp input ieee-802.1 code-point 011 pfc
user@switch# set interfaces xe-0/0/10 congestion-notification-profile fcoe-cnp
```

6. Configure the classifier to apply to the interface that exchanges iSCSI and FCoE application information.

```
[edit class-of-service classifiers]
user@switch# set ieee-802.1 fcoe-iscsi-cl1 import default forwarding-class fcoe loss-priority high code-points 011
user@switch# set ieee-802.1 fcoe-iscsi-cl1 import default forwarding-class network-control loss-priority high code-points 111
```

7. Configure the classifier to apply to the interface that exchanges iSCSI and PTP application information.

```
[edit class-of-service classifiers]
```

user@switch# **set ieee-802.1 iscsi-ptp-cl2 import default forwarding-class network-control loss-priority low code-points 111**
user@switch# **set ieee-802.1 iscsi-ptp-cl2 import default forwarding-class best-effort loss-priority low code-points [001 101]**

8. Apply the classifiers to the appropriate interfaces.

```
[edit class-of-service]
user@switch# set interfaces xe-0/0/10 unit 0 classifiers ieee-802.1 fcoe-iscsi-cl1
user@switch# set interfaces xe-0/0/11 unit 0 classifiers ieee-802.1 iscsi-ptp-cl2
```

Verification

To verify that DCBX application protocol exchange configuration has been created and is operating properly, perform these tasks:

Verifying the Application Configuration

Purpose Verify that DCBX applications have been configured.

Action List the applications by using the configuration mode command **show applications**:

```
user@switch# show applications
application iSCSI {
   protocol tcp;
   destination-port 3260;
}

application fcoe {
   ether-type 0x8906;
}

application ptp {
   ether-type 0x88F7;
}
```

Meaning The **show applications** configuration mode command lists all of the configured applications and either their protocol and destination port (Layer 4 applications) or their EtherType (Layer 2 applications). The command output shows that the iSCSI application is configured with the **tcp** protocol and destination port **3260**, the FCoE application is configured with the EtherType **0x8906**, and that the PTP application is configured with the EtherType **0x88F7**.

Verifying the Application Map Configuration

Purpose Verify that the application maps have been configured.

Action List the application maps by using the configuration mode command **show policy-options application-maps**:

```
user@switch# show policy-options application-maps
dcbx-iscsi-fcoe-app-map {
   application iSCSI code-points 111;
   application FCoE code-points 011;
}

dcbx-iscsi-ptp-app-map {
   application iSCSI code-points 111;
   application PTP code-points [001 101];
}
```

Meaning The **show policy-options application-maps** configuration mode command lists all of the configured application maps and the applications that belong to each application map. The command output shows that there are two application maps, **dcbx-iscsi-fcoe-app-map** and **dcbx-iscsi-ptp-app-map**.

The application map **dcbx-iscsi-fcoe-app-map** consists of the iSCSI application, which is mapped to IEEE 802.1p code point **111**, and the FCoE application, which is mapped to IEEE 802.1p code point **011**.

The application map **dcbx-iscsi-ptp-app-map** consists of the iSCSI application, which is mapped to IEEE 802.1p code point **111**, and the PTP application, which is mapped to IEEE 802.1p code points **001** and **101**.

Verifying DCBX Application Protocol Exchange Interface Configuration

Purpose Verify that the application maps have been applied to the correct interfaces.

Action List the application maps by using the configuration mode command **show protocols dcbx**:

```
user@switch# show protocols dcbx
interface xe-0/0/10.0 {
   application-map dcbx-iscsi-fcoe-app-map;
}

interface xe-0/0/11.0 {
   application-map dcbx-iscsi-ptp-app-map;
}
```

Meaning The **show protocols dcbx** configuration mode command lists whether the interfaces are enabled for DCBX and lists the application map applied to each interface. The command output shows that interfaces **xe-0/0/10.0** and **xe-0/0/11.0** are enabled for DCBX, and that interface **xe-0/0/10.0** uses application map **dcbx-iscsi-fcoe-app-map**, and interface **xe-0/0/11.0** uses application map **dcbx-iscsi-ptp-app-map**.

Verifying the PFC Configuration

Purpose Verify that PFC has been enabled on the FCoE code point and applied to the correct interface.

Action Display the PFC configuration to verify that PFC is enabled on the FCoE code point (**011**) in the congestion notification profile **fcoe-cnp** by using the configuration mode command **show class-of-service congestion-notification-profile**:

```
user@switch# show class-of-service congestion-notification-profile
fcoe-cnp {
    input {
        ieee-802.1 {
            code-point 011 {
                pfc;
            }
        }
    }
}
```

Display the class-of-service (CoS) interface information to verify that the correct interface has PFC enabled for the FCoE application by using the configuration mode command **show class-of-service interfaces**:

```
user@switch# show class-of-service interfaces
xe-0/0/10 {
    congestion-notification-profile fcoe-cnp;
}
```

NOTE: The sample output does not include all of the information this command can show. The output is abbreviated to focus on verifying the PFC configuration.

Meaning The **show class-of-service congestion-notification-profile** configuration mode command lists the configured congestion notification profiles. The command output shows that the congestion notification profile **fcoe-cnp** has been configured and has enabled PFC on the IEEE 802.1p code point **011** (the default FCoE code point).

The **show class-of-service interfaces** configuration mode command shows the interface CoS configuration. The command output shows that the congestion notification profile **fcoe-cnp**, which enables PFC on the FCoE code point, is applied to interface **xe-0/0/10**.

Verifying the Classifier Configuration

Purpose Verify that the classifiers have been configured and applied to the correct interfaces.

Action Display the classifier configuration by using the configuration mode command **show class-of-service**:

```
user@switch# show class-of-service
classifiers {
    ieee-802.1 fcoe-iscsi-cl1 {
        import default;
        forwarding-class network-control {
            loss-priority high code-points 111;
        }
        forwarding-class fcoe {
```



```
            loss-priority high code-points 011;
        }
    }
    ieee-802.1 iscsi-ptp-cl2 {
        import default;
        forwarding-class network-control {
            loss-priority low code-points 111;
        }
        forwarding-class best-effort {
            loss-priority low code-points [ 001 101 ];
        }
    }
}
interfaces {
    xe-0/0/10 {
        congestion-notification-profile fcoe-cnp;
        unit 0 {
            classifiers {
                ieee-802.1 fcoe-iscsi-cl1;
            }
        }
    }
    xe-0/0/11 {
        unit 0 {
            classifiers {
                ieee-802.1 iscsi-ptp-cl2;
            }
        }
    }
}
```

NOTE: The sample output does not include all of the information this command can show. The output is abbreviated to focus on verifying the classifier configuration.

Meaning The **show class-of-service** configuration mode command lists the classifier and CoS interface configuration, as well as other information not shown in this example. The command output shows that there are two classifiers configured, **fcoe-iscsi-cl1** and **iscsi-ptp-cl2**.

Classifier **fcoe-iscsi-cl1** uses the **default** classifier as a template and edits the template as follows:

- The forwarding class **network-control** is set to a loss priority of **high** and is mapped to code point **111** (the code point mapped to the iSCSI application).
- The forwarding class **fcoe** is set to a loss priority of **high** and is mapped to code point **011** (the code point mapped by default to the FCoE application).

Classifier **iscsi-ptp-cl2** uses the **default** classifier as a template and edits the template as follows:

- The forwarding class **network-control** is set to a loss priority of **low** and is mapped to IEEE 802.1p code point **111** (the code point mapped to the iSCSI application).
- The forwarding class **best-effort** is set to a loss priority of **low** and is mapped to IEEE 802.1p code points **001** and **101** (the code points mapped by default to the PTP application).

The command output also shows that classifier **fcoe-iscsi-cl1** is mapped to interface **xe-0/0/10.0** and that classifier **iscsi-ptp-cl2** is mapped to interface **xe-0/0/11.0**.

Related Documentation

- *Example: Configuring Unicast Classifiers*
- Defining an Application for DCBX Application Protocol TLV Exchange on page 309
- Configuring an Application Map for DCBX Application Protocol TLV Exchange on page 311
- Applying an Application Map to an Interface for DCBX Application Protocol TLV Exchange on page 312
- Configuring DCBX Autonegotiation on page 306
- **show dcbx on page 414**
- **show dcbx neighbors on page 415**
- Understanding DCBX Application Protocol TLV Exchange on page 117
- *Using DCBX Protocol to Lower Costs*

Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Directly Connected to the Same FCoE Transit Switch)

This example shows how to configure VN_Port to VN_Port (VN2VN_Port) FIP snooping when the hosts are directly connected to the same FCoE transit switch.

NOTE: **This example uses Junos OS without support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see** ***Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Directly Connected to the Same FCoE Transit Switch)*****.**

VN2VN_Port FIP snooping on an FCoE transit switch provides security to help prevent unauthorized access and data transmission on a bridge that connects ENodes in the Ethernet network. VN2VN_Port FIP snooping provides security for virtual links by creating filters based on information gathered (snooped) about FCoE devices during FIP transactions.

VN2VN_Port FIP snooping is conceptually similar to VN2VN_Port FIP snooping between VN_Ports and VF_Ports, but VN2VN_Port FIP snooping does not require traffic between VN_Ports to traverse the Fibre Channel (FC) switch or FCoE forwarder (FCF). Instead, a

VN_Port communicates transparently through the transit switch on a virtual link that emulates a direct connection to the VN_Port at the other end of the virtual link.

To configure VN2VN_Port FIP snooping when the hosts are directly connected to the same FCoE transit switch, you must follow these configuration rules:

- VN2VN_Port traffic must use a dedicated FCoE VLAN, and all ENodes that communicate using VN2VN_Port FIP snooping must use that FCoE VLAN. You cannot mix VN2VN_Port FIP snooping traffic with VN2VF_Port FIP snooping traffic in the same FCoE VLAN.

NOTE: **An FCoE VLAN can support either VN2VF_Port FIP snooping or VN2VN_Port FIP snooping, but not both. Configure separate FCoE VLANs for VN2VF_Port FIP snooping traffic and for VN2VN_Port FIP snooping traffic. On FCoE VLANs that are configured as VN2VN_Port FIP snooping VLANs, VN_Port to VF_Port (FIP snooping) traffic is dropped.**

- ENode-facing ports must be set in **tagged-access** port mode.
- ENode-facing ports must be untrusted ports.
- Network-facing (switch-facing) ports must be set in **trunk** port mode.
- Network-facing ports must be FCoE trusted ports.
- Explicitly configure the beacon period. The beacon period is essentially a keepalive timer for virtual link maintenance.

When you enable VN2VF_Port FIP snooping, the system snoops VN_Port to VF_Port packets and enforces security only on VN_Port to VF_Port virtual links. When you enable VN2VN_Port FIP snooping, the system snoops VN_Port to VN_Port packets and enforces security only on VN_Port to VN_Port virtual links.

The transit switch applies VN2VN_Port FIP snooping filters at the ports associated with the FCoE VLANs on which you enable VN2VN FIP snooping.

This example describes how to configure VN2VN_Port FIP snooping when the FCoE hosts are directly connected to the same transit switch:

- Requirements on page 229
- Overview on page 230
- Configuration on page 230
- Verification on page 231

Requirements

This example uses the following hardware and software components:

- One Juniper Networks QFX3500 Switch used as a transit switch
- Junos OS Release 12.2 or later for the QFX Series
- Two FCoE hosts that have ENodes

Overview

This example shows you how to:

- Set the correct interface port modes on the transit switch.
- Configure the interfaces to use the dedicated FCoE VLAN for VN2VN_Port FIP snooping.
- Configure the dedicated FCoE VLAN for VN2VN_Port FIP snooping traffic.
- Enable VN2VN_Port FIP snooping on the FCoE VLAN and configure the beacon period.

Topology

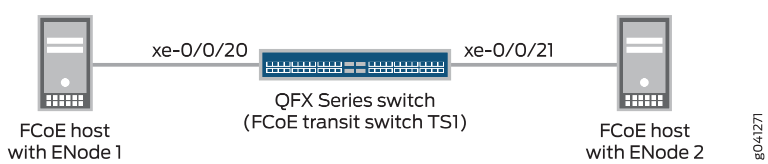
Table 28 on page 230 shows the configuration components for this example.

Table 28: Components of the VN2VN_Port FIP Snooping Configuration Topology (FCoE Hosts Directly Connected to the Same FCoE Transit Switch)

Component	Settings
Hardware	QFX3500 switch (FCoE transit switch TS1) Two FCoE hosts that have ENodes (ENode1 and ENode2, respectively)
Interfaces and port modes	• Interface **xe-0/0/20**, port mode **tagged-access**, connects directly to the FCoE host with ENode1. • Interface **xe-0/0/21**, port mode **tagged-access**, connects directly to the FCoE host with ENode2.
Interface VLAN membership	Both interfaces use VLAN **vlan200**.
VN2VN_Port FIP snooping VLAN	VLAN name—**vlan200** VLAN ID—200
FIP snooping mode and beacon period	Set **examine-vn2vn** (VN2VN_Port FIP snooping) Beacon period—90000 ms

Figure 13 on page 230 shows the network topology for this example.

Figure 13: VN2VN_Port FIP Snooping (FCoE Hosts Connected to Same Transit Switch) Topology



Configuration

CLI Quick Configuration

To quickly configure VN2VN_Port FIP snooping for FCoE hosts connected directly to the same transit switch, copy the following commands, paste them in a text file, remove line breaks, change variables and details to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level:

set interfaces xe-0/0/20 unit 0 family ethernet-switching port-mode tagged-access
set interfaces xe-0/0/21 unit 0 family ethernet-switching port-mode tagged-access
set interfaces xe-0/0/20 unit 0 family ethernet-switching vlan members vlan200
set interfaces xe-0/0/21 unit 0 family ethernet-switching vlan members vlan200
set vlans vlan200 vlan-id 200
set ethernet-switching-options secure-access-port vlan vlan200 examine-fip examine-vn2v2 beacon-period 90000

Configuring VN2VN_Port FIP Snooping (FCoE Hosts Directly Connected to the Same FCoE Transit Switch)

Step-by-Step Procedure

To configure interface port modes, configure interface VLAN membership in the FCoE VLAN dedicated to VN2VN_Port traffic, configure the VLAN, set the beacon period, and enable VN2VN_Port FIP snooping:

1. Configure the port modes of the interfaces that connect directly to the FCoE host ENodes:

   `user@switch#` **set interfaces xe-0/0/20 unit 0 family ethernet-switching port-mode tagged-access**
   set interfaces xe-0/0/21 unit 0 family ethernet-switching port-mode tagged-access

2. Configure the interface VLAN membership so that the interfaces connected to theENodes are members of the dedicated VN2VN_Port VLAN (**vlan200**):

   `user@switch#` **set interfaces xe-0/0/20 unit 0 family ethernet-switching vlan members vlan200**
   set interfaces xe-0/0/21 unit 0 family ethernet-switching vlan members vlan200

3. Configure the FCoE VLAN dedicated to VN2VN_Port FIP snooping:

   `user@switch#` **set vlans vlan200 vlan-id 200**

4. Enable VN2VN_Port FIP snooping on the VLAN and configure the beacon period:

   `user@switch#` **set ethernet-switching-options secure-access-port vlan vlan200 examine-fip examine-vn2v2 beacon-period 90000**

Verification

To verify that the VN2VN_Port FIP snooping configuration has been created and is operating properly, perform these tasks:

- Verifying That VN2VN_Port FIP Snooping is Enabled on the FCoE VLAN on page 231
- Verifying the Interface Port Mode on page 232

Verifying That VN2VN_Port FIP Snooping is Enabled on the FCoE VLAN

Purpose

Verify that VN2VN_Port FIP snooping is enabled on the correct VLAN (**vlan200**), the beacon period is set to **90000** milliseconds, and the correct interfaces (**xe-0/0/20** and **xe-0/0/21**) are members of the VLAN.

Action List the FIP snooping information using the operational mode command **show fip snooping detail**.

```
user@switch> show fip snooping detail
VLAN: vlan200,  Mode: VN2VN Snooping
  FC-MAP: 0e:fd:00
  Beacon_Period:  90000
  VN2VN Mode: Point-to-Point
    Enode Information
    Enode-MAC: 10:10:94:01:00:02,       Interface: xe-0/0/20
      Active VN_Ports : 1
      VN_Port Information
      VN-Port MAC: 0e:fd:00:00:0a:01
         Active Sessions    : 1
         Session Information
     Vlink far-end VN-Port-MAC: 0e:fd:00:00:0b:01
    Enode-MAC: 10:10:94:01:00:02,       Interface: xe-0/0/21
      Active VN_Ports : 1
      VN_Port Information
      VN-Port MAC: 0e:fd:00:00:0b:01
         Active Sessions    : 1
         Session Information
     Vlink far-end VN-Port-MAC: 0e:fd:00:00:0a:01
```

Meaning The **show fip snooping detail** command lists all of the transit switch information about VN2VN_Port FIP snooping and VN2VF_Port FIP snooping. The command shows that:

- The VLAN is **vlan200**.
- The mode is FIP snooping mode **VN2VN**, for VN2VN_Port FIP snooping. (If the Mode field shows **VN2VF**, then the FIP snooping mode is VN2VF_Port FIP snooping.)
- The beacon period is **90000**.
- The interfaces for the ENodes are **xe-0/0/20** and **xe-0/0/21**.

In addition, this useful command shows information about the ENodes and the VN2VN_Port sessions.

Verifying the Interface Port Mode

Purpose Verify that the interface port modes are **tagged-access**.

Action List the Ethernet switching interfaces to confirm the port mode using the **show ethernet-switching interfaces detail** operational command.

Use the operational mode commands **show ethernet-switching interfaces xe-0/0/20.0 detail** and **show ethernet-switching interfaces xe-0/0/21.0 detail** to list the Ethernet switching interface information. The output is truncated to show only the relevant portions:

```
user@switch> show ethernet-switching interfaces xe-0/0/20.0 detail
Interface: xe-0/0/20.0, Index: 75, State: up, Port mode: Tagged-Access
.
.
.
```



```
user@switch> show ethernet-switching interfaces xe-0/0/21.0 detail
Interface: xe-0/0/21.0, Index: 83, State: up, Port mode: Tagged-Access
.
.
.
```

Meaning The **show ethernet-switching interfaces detail** command lists the port mode as **tagged-access** for both interfaces.

Related Documentation

- Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Directly Connected to Different FCoE Transit Switches) on page 233
- Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Indirectly Connected Through an Aggregation Layer FCoE Transit Switch) on page 241
- Enabling VN2VN_Port FIP Snooping and Configuring the Beacon Period on an FCoE Transit Switch on page 300
- Understanding VN_Port to VN_Port FIP Snooping on an FCoE Transit Switch on page 87

Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Directly Connected to Different FCoE Transit Switches)

This example shows how to configure VN_Port to VN_Port (VN2VN_Port) FIP snooping when the hosts are directly connected to different FCoE transit switches, and the transit switches are directly connected to each other.

NOTE: **This example uses Junos OS without support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see *Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Directly Connected to Different FCoE Transit Switches)*.**

VN2VN_Port FIP snooping on an FCoE transit switch provides security to help prevent unauthorized access and data transmission on a bridge that connects ENodes in the Ethernet network. VN2VN_Port FIP snooping provides security for virtual links by creating filters based on information gathered (snooped) about FCoE devices during FIP transactions.

VN2VN_Port FIP snooping is conceptually similar to VN2VF_Port FIP snooping between VN_Ports and VF_Ports, but VN2VN_Port FIP snooping does not require traffic between VN_Ports to traverse the Fibre Channel (FC) switch or FCoE forwarder (FCF). Instead, a VN_Port communicates transparently through one or more transit switches on a virtual link that emulates a direct connection to the VN_Port at the other end of the virtual link.

To configure VN2VN_Port FIP snooping when the hosts are directly connected to different FCoE transit switches, and the transit switches are directly connected to each other, you must follow these configuration rules:

- VN2VN_Port traffic must use a dedicated FCoE VLAN, and all ENodes that communicate using VN2VN_Port FIP snooping must use that FCoE VLAN. The FCoE VLAN must be configured on each transit switch. You cannot mix VN2VN_Port FIP snooping traffic with VN2VF_Port FIP snooping traffic in the same FCoE VLAN.

  NOTE: **An FCoE VLAN can support either VN2VF_Port FIP snooping or VN2VN_Port FIP snooping, but not both. Configure separate FCoE VLANs for VN2VF_Port FIP snooping traffic and for VN2VN_Port FIP snooping traffic. On FCoE VLANs that are configured as VN2VN_Port FIP snooping VLANs, VN2VF_Port traffic is dropped.**

- ENode-facing ports must be set in **tagged-access** port mode.
- ENode-facing ports must be untrusted ports.
- Network-facing (switch-facing) ports must be set in **trunk** port mode.
- Network-facing ports must be FCoE trusted ports.
- Explicitly configure the beacon period. The beacon period is essentially a keepalive timer for virtual link maintenance.

When you enable VN2VF_Port FIP snooping, the system snoops VN_Port to VF_Port packets and enforces security only on VN_Port to VF_Port virtual links. When you enable VN2VN_Port FIP snooping, the system snoops VN_Port to VN_Port packets and enforces security only on VN_Port to VN_Port virtual links.

The transit switch applies VN2VN_Port FIP snooping filters at the ports associated with the FCoE VLANs on which you enable VN2VN FIP snooping.

This example describes how to configure VN2VN_Port FIP snooping when the FCoE hosts are directly connected to different transit switches, and the transit switches are directly connected to each other:

- Requirements on page 234
- Overview on page 235
- Configuration on page 236
- Verification on page 238

Requirements

This example uses the following hardware and software components:

- Two Juniper Networks QFX3500 Switches used as transit switches
- Junos OS Release 12.2 or later for the QFX Series
- Two FCoE hosts that have ENodes

Overview

This example shows you how to:

- Set the correct interface port modes on the transit switch.
- Configure the interfaces to use the dedicated FCoE VLAN for VN2VN_Port FIP snooping.
- Configure the network-facing interfaces as FCoE trusted interfaces.
- Configure the dedicated FCoE VLAN for VN2VN_Port FIP snooping traffic.
- Enable VN2VN_Port FIP snooping on the FCoE VLAN and configure the beacon period.

Topology

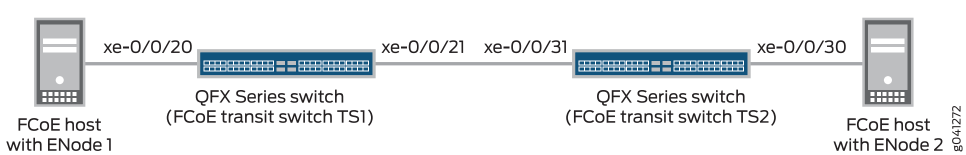
Table 29 on page 235 shows the configuration components for this example.

Table 29: Components of the VN2VN_Port FIP Snooping Configuration Topology (FCoE Hosts Directly Connected to Different FCoE Transit Switches)

Component	Settings
Hardware	Two QFX3500 switches (FCoE transit switch TS1 and FCoE transit switch TS2) Two FCoE hosts that have ENodes (ENode1 and ENode2, respectively)
Interfaces and port modes	• Interface **xe-0/0/20**, port mode **tagged-access**, connects directly from transit switch TS1 to the FCoE host with ENode1. • Interface **xe-0/0/21**, port mode **trunk**, connects directly from transit switch TS1 to transit switch TS2. • Interface **xe-0/0/31**, port mode **trunk**, connects directly from transit switch TS2 to transit switch TS1. • Interface **xe-0/0/30**, port mode **tagged-access**, connects directly from transit switch TS2 to the FCoE host with ENode2.
Interface VLAN membership	The interfaces on both transit switches use VLAN **vlan200**.
VN2VN_Port FIP snooping VLAN	VLAN name (both transit switches)—**vlan200** VLAN ID—200
FIP snooping mode and beacon period	Set **examine-vn2vn** (VN2VN_Port FIP snooping) Beacon period—90000 ms

Figure 14 on page 235 shows the network topology for this example.

Figure 14: VN2VN_Port FIP Snooping (FCoE Hosts Connected to Different Transit Switches) Topology



Configuration

To configure VN2VN_Port FIP snooping for VN_Ports that are directly connected to different transit switches (and the transit switches are directly connected to each other), perform these tasks:

- Configuring VN2VN_Port FIP Snooping on FCoE Transit Switch TS1 on page 236
- Configuring VN2VN_Port FIP Snooping on FCoE Transit Switch TS2 on page 237

CLI Quick Configuration

The configuration for each FCoE transit switch is shown separately.

To quickly configure VN2VN_Port FIP snooping for FCoE hosts connected directly to different transit switches, copy the following commands, paste them in a text file, remove line breaks, change variables and details to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level. To configure FCoE transit switch TS1:

```
set interfaces xe-0/0/20 unit 0 family ethernet-switching port-mode tagged-access
set interfaces xe-0/0/21 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/20 unit 0 family ethernet-switching vlan members vlan200
set interfaces xe-0/0/21 unit 0 family ethernet-switching vlan members vlan200
set ethernet-switching-options secure-access-port interface xe-0/0/21 fcoe-trusted
set vlans vlan200 vlan-id 200
set ethernet-switching-options secure-access-port vlan vlan200 examine-fip examine-vn2v2
beacon-period 90000
```

To quickly configure VN2VN_Port FIP snooping for FCoE hosts connected directly to different transit switches, copy the following commands, paste them in a text file, remove line breaks, change variables and details to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level. To configure FCoE transit switch TS2:

```
set interfaces xe-0/0/30 unit 0 family ethernet-switching port-mode tagged-access
set interfaces xe-0/0/31 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/30 unit 0 family ethernet-switching vlan members vlan200
set interfaces xe-0/0/31 unit 0 family ethernet-switching vlan members vlan200
set vlans vlan200 vlan-id 200
set ethernet-switching-options secure-access-port interface xe-0/0/31 fcoe-trusted
set ethernet-switching-options secure-access-port vlan vlan200 examine-fip examine-vn2v2
beacon-period 90000
```

Configuring VN2VN_Port FIP Snooping on FCoE Transit Switch TS1

Step-by-Step Procedure

To configure interface port modes, configure interface VLAN membership in the FCoE VLAN dedicated to VN2VN_Port traffic, set the network-facing port as FCoE trusted, configure the VLAN, set the beacon period, and enable VN2VN_Port FIP snooping:

1. Configure the port modes of the interfaces that connect directly to the FCoE host with ENode1 (**xe-0/0/20**) and to FCoE transit switch TS2 (**xe-0/0/21**):

   ```
   user@switch# set interfaces xe-0/0/20 unit 0 family ethernet-switching port-mode
   tagged-access
   user@switch# set interfaces xe-0/0/21 unit 0 family ethernet-switching port-mode trunk
   ```

2. Configure the interface VLAN membership so that the interfaces are members of the dedicated VN2VN_Port VLAN (**vlan200**):

`user@switch#` **set interfaces xe-0/0/20 unit 0 family ethernet-switching vlan members vlan200**
`user@switch#` **set interfaces xe-0/0/21 unit 0 family ethernet-switching vlan members vlan200**

3. Configure the network-facing port (**xe-0/0/21**) as an FCoE trusted port:

   `user@switch#` **set ethernet-switching-options secure-access-port interface xe-0/0/21 fcoe-trusted**

4. Configure the FCoE VLAN dedicated to VN2VN_Port FIP snooping:

   `user@switch#` **set vlans vlan200 vlan-id 200**

5. Enable VN2VN_Port FIP snooping on the VLAN and configure the beacon period:

   `user@switch#` **set ethernet-switching-options secure-access-port vlan vlan200 examine-fip examine-vn2v2 beacon-period 90000**

Configuring VN2VN_Port FIP Snooping on FCoE Transit Switch TS2

Step-by-Step Procedure

To configure interface port modes, configure interface VLAN membership in the FCoE VLAN dedicated to VN2VN_Port traffic, set the network-facing port as FCoE trusted, configure the VLAN, set the beacon period, and enable VN2VN_Port FIP snooping:

1. Configure the port modes of the interfaces that connect directly to the FCoE host with ENode2 (**xe-0/0/30**) and to FCoE transit switch TS1 (**xe-0/0/31**):

   `user@switch#` **set interfaces xe-0/0/30 unit 0 family ethernet-switching port-mode tagged-access**
   `user@switch#` **set interfaces xe-0/0/31 unit 0 family ethernet-switching port-mode trunk**

2. Configure the interface VLAN membership so that the interfaces are members of the dedicated VN2VN_Port VLAN (**vlan200**):

   `user@switch#` **set interfaces xe-0/0/30 unit 0 family ethernet-switching vlan members vlan200**
   `user@switch#` **set interfaces xe-0/0/31 unit 0 family ethernet-switching vlan members vlan200**

3. Configure the network-facing port (**xe-0/0/31**) as an FCoE trusted port:

   `user@switch#` **set ethernet-switching-options secure-access-port interface xe-0/0/31 fcoe-trusted**

4. Configure the FCoE VLAN dedicated to VN2VN_Port FIP snooping:

   `user@switch#` **set vlans vlan200 vlan-id 200**

5. Enable VN2VN_Port FIP snooping on the VLAN and configure the beacon period:

   `user@switch#` **set ethernet-switching-options secure-access-port vlan vlan200 examine-fip examine-vn2v2 beacon-period 90000**

Verification

To verify that the VN2VN_Port FIP snooping configuration has been created and is operating properly on both switches, perform these tasks:

- Verifying That VN2VN_Port FIP Snooping is Enabled on the FCoE VLAN (Transit Switches TS1 and TS2) on page 238
- Verifying the Interface Port Mode on page 240

Verifying That VN2VN_Port FIP Snooping is Enabled on the FCoE VLAN (Transit Switches TS1 and TS2)

Purpose Verify that VN2VN_Port FIP snooping is enabled on the correct VLAN (**vlan200**), the beacon period is set to **90000** milliseconds, and that the correct interfaces (**xe-0/0/20** and **xe-0/0/21** on TS1, and **xe-0/0/30**and **xe-0/0/31** on TS2) are members of the VLAN.

Action List the FIP snooping information on transit switch TS1 using the operational mode command **show fip snooping detail**

```
user@switch> show fip snooping detail
VLAN: vlan200,  Mode: VN2VN Snooping
  FC-MAP: 0e:fd:00
  Beacon_Period:  90000
  VN2VN Mode: Point-to-Point
    Enode Information
    Enode-MAC: 10:10:94:01:00:02,       Interface: xe-0/0/20
      Active VN_Ports : 1
      VN_Port Information
      VN-Port MAC: 0e:fd:00:00:0a:01
         Active Sessions    : 1
         Session Information
     Vlink far-end VN-Port-MAC: 0e:fd:00:00:0b:01
    Enode-MAC: 10:10:94:01:00:02,       Interface: xe-0/0/21
      Active VN_Ports : 1
      VN_Port Information
      VN-Port MAC: 0e:fd:00:00:0b:01
         Active Sessions    : 1
         Session Information
     Vlink far-end VN-Port-MAC: 0e:fd:00:00:0a:01
```

List the FIP snooping information on transit switch TS2 using the operational mode command **show fip snooping detail**

```
user@switch> show fip snooping detail
VLAN: vlan200,  Mode: VN2VN Snooping
  FC-MAP: 0e:fd:00
  Beacon_Period:  90000
  VN2VN Mode: Point-to-Point
    Enode Information
    Enode-MAC: 10:10:94:01:00:02,       Interface: xe-0/0/30
      Active VN_Ports : 1
      VN_Port Information
      VN-Port MAC: 0e:fd:00:00:0b:01
         Active Sessions    : 1
         Session Information
     Vlink far-end VN-Port-MAC: 0e:fd:00:00:0a:01
    Enode-MAC: 10:10:94:01:00:02,       Interface: xe-0/0/31
      Active VN_Ports : 1
      VN_Port Information
      VN-Port MAC: 0e:fd:00:00:0a:01
         Active Sessions    : 1
         Session Information
     Vlink far-end VN-Port-MAC: 0e:fd:00:00:0b:01
```

Meaning The **show fip snooping detail** command lists all of the transit switch information about VN2VN_Port FIP snooping and VN2VF_Port FIP snooping on each transit switch. The command shows that:

- The VLAN is **vlan200**.
- The mode is FIP snooping mode **VN2VN**, for VN2VN_Port FIP snooping. (If the Mode field shows **VN2VF**, then the FIP snooping mode is VN2VF_Port FIP snooping.)

- The beacon period is **90000**.
- The interfaces connected to the ENodes are **xe-0/0/20** and **xe-0/0/21** on transit switch TS1, and **xe-0/0/30** and **xe-0/0/31** on transit switch TS2. Because the transit switches are transparent passthrough switches, the network-facing trunk ports “see” the FCoE host ENodes at the far end of the VN2VN_Port virtual link.

In addition, this useful command shows information about the ENodes and the VN2VN_Port sessions.

Verifying the Interface Port Mode

Purpose Verify that the interface port modes are **tagged-access** for ENode-facing ports and **trunk** for network-facing ports on each transit switch.

Action List the Ethernet switching interfaces to confirm the port mode using the **show ethernet-switching interfaces detail** operational command.

Use the operational mode commands **show ethernet-switching interfaces xe-0/0/20.0 detail** and **show ethernet-switching interfaces xe-0/0/21.0 detail** to list the Ethernet switching interface information on FCoE transit switch TS1. The output is truncated to show only the relevant portions:

```
user@switch> show ethernet-switching interfaces xe-0/0/20.0 detail
Interface: xe-0/0/20.0, Index: 75, State: up, Port mode: Tagged-Access
.
.
.

user@switch> show ethernet-switching interfaces xe-0/0/21.0 detail
Interface: xe-0/0/21.0, Index: 83, State: up, Port mode: Trunk
.
.
.
```

List the Ethernet switching interface information on FCoE transit switch TS2 using the operational mode commands **show ethernet-switching interfaces xe-0/0/30.0 detail** and **show ethernet-switching interfaces xe-0/0/31.0 detail**:

```
user@switch> show ethernet-switching interfaces xe-0/0/30.0 detail
Interface: xe-0/0/30.0, Index: 56, State: up, Port mode: Tagged-Access
.
.
.

user@switch> show ethernet-switching interfaces xe-0/0/31.0 detail
Interface: xe-0/0/31.0, Index: 59, State: up, Port mode: Trunk
.
.
.
```

Related Documentation

- Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Directly Connected to the Same FCoE Transit Switch) on page 228
- Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Indirectly Connected Through an Aggregation Layer FCoE Transit Switch) on page 241

- Enabling VN2VN_Port FIP Snooping and Configuring the Beacon Period on an FCoE Transit Switch on page 300
- Understanding VN_Port to VN_Port FIP Snooping on an FCoE Transit Switch on page 87

Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Indirectly Connected Through an Aggregation Layer FCoE Transit Switch)

This example shows how to configure VN_Port to VN_Port (VN2VN_Port) FIP snooping when the hosts are indirectly connected through an aggregation layer FCoE transit switch. Each FCoE host ENode is directly connected to an FCoE transit switch, but the FCoE transit switches are not directly connected to each other. The FCoE transit switches are both connected to a third FCoE transit switch that acts as an aggregation layer switch.

NOTE: **This example uses Junos OS without support for the Enhanced Layer 2 Software (ELS) configuration style. If your switch runs software that supports ELS, see** ***Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Indirectly Connected Through an Aggregation Layer FCoE Transit Switch)*****.**

VN2VN_Port FIP snooping on an FCoE transit switch provides security to help prevent unauthorized access and data transmission on a bridge that connects ENodes in the Ethernet network. VN2VN_Port FIP snooping provides security for virtual links by creating filters based on information gathered (snooped) about FCoE devices during FIP transactions.

VN2VN_Port FIP snooping is conceptually similar to VN2VN_Port FIP snooping between VN_Ports and VF_Ports, but VN2VN_Port FIP snooping does not require traffic between VN_Ports to traverse the Fibre Channel (FC) switch or FCoE forwarder (FCF). Instead, a VN_Port communicates transparently through one or more transit switches on a virtual link that emulates a direct connection to the VN_Port at the other end of the virtual link.

To configure VN2VN_Port FIP snooping when the hosts are indirectly connected, you must follow these configuration rules:

- VN2VN_Port traffic must use a dedicated FCoE VLAN, and all ENodes that communicate using VN2VN_Port FIP snooping must use that FCoE VLAN. The FCoE VLAN must be configured on each transit switch. You cannot mix VN2VN_Port FIP snooping traffic with VN2VF_Port FIP snooping traffic in the same FCoE VLAN.

  NOTE: **An FCoE VLAN can support either VN2VF_Port FIP snooping or VN2VN_Port FIP snooping, but not both. Configure separate FCoE VLANs for VN2VF_Port FIP snooping traffic and for VN2VN_Port FIP snooping traffic. On FCoE VLANs that are configured as VN2VN_Port FIP snooping VLANs, VN_Port to VF_Port traffic is dropped.**

- ENode-facing ports must be set in **tagged-access** port mode.

- ENode-facing ports must be untrusted ports.
- Network-facing (switch-facing) ports must be set in **trunk** port mode.
- Network-facing ports must be FCoE trusted ports.
- Explicitly configure the beacon period. The beacon period is essentially a keepalive timer for virtual link maintenance.

When you enable FIP snooping, the system snoops VN_Port to VF_Port packets and enforces security only on VN_Port to VF_Port virtual links. When you enable VN2VN_Port FIP snooping, the system snoops VN_Port to VN_Port packets and enforces security only on VN_Port to VN_Port virtual links.

The transit switch applies VN2VN_Port FIP snooping filters at the ports associated with the FCoE VLANs on which you enable VN2VN FIP snooping.

This example describes how to configure VN2VN_Port FIP snooping when the FCoE hosts are indirectly connected across an aggregation layer FCoE transit switch:

Requirements

This example uses the following hardware and software components:

- Three Juniper Networks QFX3500 Switches used as transit switches
- Junos OS Release 12.2 or later for the QFX Series
- Two FCoE hosts that have ENodes

Overview

This example shows you how to:

- Set the correct interface port modes on the transit switch.
- Configure the interfaces to use the dedicated FCoE VLAN for VN2VN_Port FIP snooping.
- Configure the network-facing interfaces as FCoE trusted interfaces.
- Configure the dedicated FCoE VLAN for VN2VN_Port FIP snooping traffic.
- Enable VN2VN_Port FIP snooping on the FCoE VLAN and configure the beacon period.

Topology

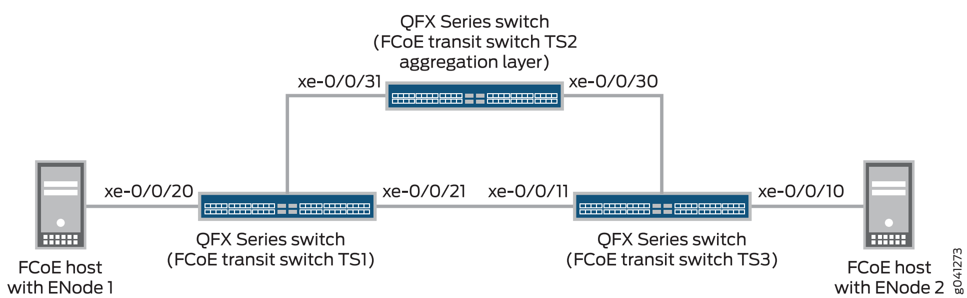
Table 30 on page 243 shows the configuration components for this example.

Table 30: Components of the VN2VN_Port FIP Snooping Configuration Topology (FCoE Hosts Indirectly Connected Across an Aggregation Layer FCoE Transit Switch)

Component	Settings
Hardware	Three QFX3500 switches, two of which are FCoE transit switches that are directly attached to the FCoE hosts (transit switches TS1 and TS2) and one of which is an aggregation layer FCoE transit switch (TS3) Two FCoE hosts that have ENodes (ENode1 and ENode2, respectively)
Interfaces and port modes	• Interface **xe-0/0/20**, port mode **tagged-access**, connects directly from transit switch TS1 to the FCoE host with ENode1. • Interface **xe-0/0/21**, port mode **trunk**, connects directly from transit switch TS1 to aggregation layer transit switch TS2. • Interface **xe-0/0/31**, port mode **trunk**, connects directly from aggregation layer transit switch TS2 to transit switch TS1. • Interface **xe-0/0/30**, port mode **trunk**, connects directly from aggregation layer transit switch TS2 to transit switch TS3. • Interface **xe-0/0/11**, port mode **trunk**, connects directly from transit switch TS3 to aggregation layer transit switch TS2. • Interface **xe-0/0/10**, port mode **tagged-access**, connects directly from transit switch TS3 to the FCoE host with ENode2.
Interface VLAN membership	The interfaces on all three switches use VLAN **vlan200**.
VN2VN_Port FIP snooping VLAN	VLAN name (all three switches)—**vlan200** VLAN ID—200
FIP snooping mode and beacon period	Set **examine-vn2vn** (VN2VN_Port FIP snooping) Beacon period—90000 ms

Figure 15 on page 243 shows the network topology for this example.

Figure 15: VN2VN_Port FIP Snooping (FCoE Hosts Indirectly Connected) Topology



Configuration

To configure VN2VN_Port FIP snooping for VN_Ports that are indirectly connected across an aggregation layer FCoE transit switch, perform these tasks:

- Configuring VN2VN_Port FIP Snooping on FCoE Transit Switch TS1 on page 245
- Configuring VN2VN_Port FIP Snooping on Aggregation Layer FCoE Transit Switch TS2 on page 245
- Configuring VN2VN_Port FIP Snooping on FCoE Transit Switch TS3 on page 246

CLI Quick Configuration

The configuration for each FCoE transit switch is shown separately.

To quickly configure VN2VN_Port FIP snooping for FCoE hosts that are indirectly connected across an aggregation layer FCoE transit switch, copy the following commands, paste them in a text file, remove line breaks, change variables and details to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level. To configure FCoE transit switch TS1:

```
set interfaces xe-0/0/20 unit 0 family ethernet-switching port-mode tagged-access
set interfaces xe-0/0/21 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/20 unit 0 family ethernet-switching vlan members vlan200
set interfaces xe-0/0/21 unit 0 family ethernet-switching vlan members vlan200
set ethernet-switching-options secure-access-port interface xe-0/0/21 fcoe-trusted
set vlans vlan200 vlan-id 200
set ethernet-switching-options secure-access-port vlan vlan200 examine-fip examine-vn2v2
beacon-period 90000
```

To quickly configure VN2VN_Port FIP snooping for FCoE hosts that are indirectly connected across an aggregation layer FCoE transit switch, copy the following commands, paste them in a text file, remove line breaks, change variables and details to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level. To configure FCoE transit switch TS2:

```
set interfaces xe-0/0/30 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/31 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/30 unit 0 family ethernet-switching vlan members vlan200
set interfaces xe-0/0/31 unit 0 family ethernet-switching vlan members vlan200
set vlans vlan200 vlan-id 200
set ethernet-switching-options secure-access-port interface xe-0/0/30 fcoe-trusted
set ethernet-switching-options secure-access-port interface xe-0/0/31 fcoe-trusted
set ethernet-switching-options secure-access-port vlan vlan200 examine-fip examine-vn2v2
beacon-period 90000
```

To quickly configure VN2VN_Port FIP snooping for FCoE hosts that are indirectly connected across an aggregation layer FCoE transit switch, copy the following commands, paste them in a text file, remove line breaks, change variables and details to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level. To configure FCoE transit switch TS3:

```
set interfaces xe-0/0/10 unit 0 family ethernet-switching port-mode tagged-access
set interfaces xe-0/0/11 unit 0 family ethernet-switching port-mode trunk
set interfaces xe-0/0/10 unit 0 family ethernet-switching vlan members vlan200
set interfaces xe-0/0/11 unit 0 family ethernet-switching vlan members vlan200
set vlans vlan200 vlan-id 200
set ethernet-switching-options secure-access-port interface xe-0/0/11 fcoe-trusted
set ethernet-switching-options secure-access-port vlan vlan200 examine-fip examine-vn2v2
beacon-period 90000
```

Configuring VN2VN_Port FIP Snooping on FCoE Transit Switch TS1

Step-by-Step Procedure

To configure interface port modes, configure interface VLAN membership in the FCoE VLAN dedicated to VN2VN_Port traffic, set the network-facing port as FCoE trusted, configure the VLAN, set the beacon period, and enable VN2VN_Port FIP snooping:

1. Configure the port modes of the interfaces that connect directly to the FCoE host with ENode1 (**xe-0/0/20**) and to aggregation layer FCoE transit switch TS2 (**xe-0/0/21**):

   ```
   user@switch# set interfaces xe-0/0/20 unit 0 family ethernet-switching port-mode tagged-access
   set interfaces xe-0/0/21 unit 0 family ethernet-switching port-mode trunk
   ```

2. Configure the interface VLAN membership so that the interfaces are members of the dedicated VN2VN_Port VLAN (**vlan200**):

   ```
   user@switch# set interfaces xe-0/0/20 unit 0 family ethernet-switching vlan members vlan200
   set interfaces xe-0/0/21 unit 0 family ethernet-switching vlan members vlan200
   ```

3. Configure the network-facing port (**xe-0/0/21**) as an FCoE trusted port:

   ```
   user@switch# set ethernet-switching-options secure-access-port interface xe-0/0/21 fcoe-trusted
   ```

4. Configure the FCoE VLAN dedicated to VN2VN_Port FIP snooping:

   ```
   user@switch# set vlans vlan200 vlan-id 200
   ```

5. Enable VN2VN_Port FIP snooping on the VLAN and configure the beacon period:

   ```
   user@switch# set ethernet-switching-options secure-access-port vlan vlan200 examine-fip examine-vn2v2 beacon-period 90000
   ```

Configuring VN2VN_Port FIP Snooping on Aggregation Layer FCoE Transit Switch TS2

Step-by-Step Procedure

To configure interface port modes, configure interface VLAN membership in the FCoE VLAN dedicated to VN2VN_Port traffic, set the network-facing ports as FCoE trusted, configure the VLAN, set the beacon period, and enable VN2VN_Port FIP snooping:

1. Configure the port modes of the interfaces that connect directly to FCoE transit switches TS1 (**xe-0/0/31**) and TS3 (**xe-0/0/30**). Both interfaces are network-facing and must be configured as trunk interfaces:

   ```
   user@switch# set interfaces xe-0/0/30 unit 0 family ethernet-switching port-mode trunk
   set interfaces xe-0/0/31 unit 0 family ethernet-switching port-mode trunk
   ```

2. Configure the interface VLAN membership so that the interfaces are members of the dedicated VN2VN_Port VLAN (**vlan200**):

   ```
   user@switch# set interfaces xe-0/0/30 unit 0 family ethernet-switching vlan members vlan200
   set interfaces xe-0/0/31 unit 0 family ethernet-switching vlan members vlan200
   ```

3. Configure the network-facing ports (**xe-0/0/30** and **xe-0/0/31**) as FCoE trusted ports:

```
user@switch# set ethernet-switching-options secure-access-port interface xe-0/0/30
fcoe-trusted
user@switch# set ethernet-switching-options secure-access-port interface xe-0/0/31
fcoe-trusted
```

4. Configure the FCoE VLAN dedicated to VN2VN_Port FIP snooping:

```
user@switch# set vlans vlan200 vlan-id 200
```

5. Enable VN2VN_Port FIP snooping on the VLAN and configure the beacon period:

```
user@switch# set ethernet-switching-options secure-access-port vlan vlan200 examine-fip
examine-vn2v2 beacon-period 90000
```

Configuring VN2VN_Port FIP Snooping on FCoE Transit Switch TS3

Step-by-Step Procedure

To configure interface port modes, configure interface VLAN membership in the FCoE VLAN dedicated to VN2VN_Port traffic, set the network-facing port as FCoE trusted, configure the VLAN, set the beacon period, and enable VN2VN_Port FIP snooping:

1. Configure the port modes of the interfaces that connect directly to the FCoE host with ENode2 (**xe-0/0/10**) and to aggregation layer FCoE transit switch TS2 (**xe-0/0/11**):

```
user@switch# set interfaces xe-0/0/10 unit 0 family ethernet-switching port-mode
tagged-access
set interfaces xe-0/0/11 unit 0 family ethernet-switching port-mode trunk
```

2. Configure the interface VLAN membership so that the interfaces are members of the dedicated VN2VN_Port VLAN (**vlan200**):

```
user@switch# set interfaces xe-0/0/10 unit 0 family ethernet-switching vlan members
vlan200
set interfaces xe-0/0/11 unit 0 family ethernet-switching vlan members vlan200
```

3. Configure the network-facing port (**xe-0/0/11**) as an FCoE trusted port:

```
user@switch# set ethernet-switching-options secure-access-port interface xe-0/0/11
fcoe-trusted
```

4. Configure the FCoE VLAN dedicated to VN2VN_Port FIP snooping:

```
user@switch# set vlans vlan200 vlan-id 200
```

5. Enable VN2VN_Port FIP snooping on the VLAN and configure the beacon period:

```
user@switch# set ethernet-switching-options secure-access-port vlan vlan200 examine-fip
examine-vn2v2 beacon-period 90000
```

Verification

To verify that the VN2VN_Port FIP snooping configuration has been created and is operating properly on all three switches, perform these tasks:

- Verifying That VN2VN_Port FIP Snooping Is Enabled on the FCoE VLAN (All Three Transit Switches) on page 247
- Verifying the Interface Port Mode on page 249

Verifying That VN2VN_Port FIP Snooping Is Enabled on the FCoE VLAN (All Three Transit Switches)

Purpose Verify that VN2VN_Port FIP snooping is enabled on the correct VLAN (**vlan200**), the beacon period is set to **90000** milliseconds, and that the correct interfaces (**xe-0/0/20** and **xe-0/0/21** on TS1, **xe-0/0/30** and **xe-0/0/31** aggregation layer TS2, and **xe-0/0/10** and **xe-0/0/11** on TS3) are members of the VLAN.

Action List the FIP snooping information on transit switch TS1 using the operational mode command **show fip snooping detail**

```
user@switch> show fip snooping detail
VLAN: vlan200,  Mode: VN2VN Snooping
  FC-MAP: 0e:fd:00
  Beacon_Period:  90000
  VN2VN Mode: Point-to-Point
    Enode Information
    Enode-MAC: 10:10:94:01:00:02,       Interface: xe-0/0/20
      Active VN_Ports : 1
      VN_Port Information
      VN-Port MAC: 0e:fd:00:00:0a:01
         Active Sessions    : 1
         Session Information
     Vlink far-end VN-Port-MAC: 0e:fd:00:00:0b:01
    Enode-MAC: 10:10:94:01:00:02,       Interface: xe-0/0/21
      Active VN_Ports : 1
      VN_Port Information
      VN-Port MAC: 0e:fd:00:00:0b:01
         Active Sessions    : 1
         Session Information
     Vlink far-end VN-Port-MAC: 0e:fd:00:00:0a:01
```

List the FIP snooping information on aggregation layer transit switch TS2 using the operational mode command **show fip snooping detail**

```
user@switch> show fip snooping detail
VLAN: vlan200,  Mode: VN2VN Snooping
  FC-MAP: 0e:fd:00
  Beacon_Period:  90000
  VN2VN Mode: Point-to-Point
    Enode Information
    Enode-MAC: 10:10:94:01:00:02,       Interface: xe-0/0/30
      Active VN_Ports : 1
      VN_Port Information
      VN-Port MAC: 0e:fd:00:00:0b:01
         Active Sessions    : 1
         Session Information
     Vlink far-end VN-Port-MAC: 0e:fd:00:00:0a:01
    Enode-MAC: 10:10:94:01:00:02,       Interface: xe-0/0/31
      Active VN_Ports : 1
      VN_Port Information
      VN-Port MAC: 0e:fd:00:00:0a:01
         Active Sessions    : 1
         Session Information
     Vlink far-end VN-Port-MAC: 0e:fd:00:00:0b:01
```

List the FIP snooping information on transit switch TS3 using the operational mode command **show fip snooping detail**

```
user@switch> show fip snooping detail
VLAN: vlan200,  Mode: VN2VN Snooping
  FC-MAP: 0e:fd:00
  Beacon_Period:  90000
  VN2VN Mode: Point-to-Point
    Enode Information
```



```
Enode-MAC: 10:10:94:01:00:02,       Interface: xe-0/0/10
  Active VN_Ports : 1
  VN_Port Information
  VN-Port MAC: 0e:fd:00:00:0b:01
     Active Sessions    : 1
     Session Information
 Vlink far-end VN-Port-MAC: 0e:fd:00:00:0a:01
Enode-MAC: 10:10:94:01:00:02,       Interface: xe-0/0/11
  Active VN_Ports : 1
  VN_Port Information
  VN-Port MAC: 0e:fd:00:00:0a:01
     Active Sessions    : 1
     Session Information
 Vlink far-end VN-Port-MAC: 0e:fd:00:00:0b:01
```

Meaning The **show fip snooping detail** command lists all of the transit switch information about VN2VN_Port FIP snooping and VN2VF_Port FIP snooping on each transit switch. The command shows that:

- The VLAN is **vlan200**.
- The mode is FIP snooping mode **VN2VN**, for VN2VN_Port FIP snooping. (If the Mode field shows **VN2VF**, then the FIP snooping mode is VN2VF_Port FIP snooping.)
- The beacon period is **90000**.
- The interfaces connected to the ENodes are **xe-0/0/20** and **xe-0/0/21** on transit switch TS1, **xe-0/0/30** and **xe-0/0/31** on aggregation layer transit switch TS2, and **xe-0/0/10** and **xe-0/0/11** on transit switch TS3. Because the transit switches are transparent passthrough switches, the network-facing trunk ports “see” the FCoE host ENodes at the far end of the VN2VN_Port virtual link.

In addition, this useful command shows information about the ENodes and the VN2VN_Port sessions.

Verifying the Interface Port Mode

Purpose Verify that the interface port modes are **tagged-access** for ENode-facing ports and **trunk** for network-facing ports on each transit switch.

Action List the Ethernet switching interfaces to confirm the port mode using the **show ethernet-switching interfaces detail** operational command for each interface. The output is truncated to show only the relevant portions.

List the Ethernet switching interface information on FCoE transit switch TS1 using the operational mode commands **show ethernet-switching interfaces xe-0/0/20.0 detail** and **show ethernet-switching interfaces xe-0/0/21.0 detail**:

```
user@switch> show ethernet-switching interfaces xe-0/0/20.0 detail
Interface: xe-0/0/20.0, Index: 75, State: up, Port mode: Tagged-Access
.
.
.

user@switch> show ethernet-switching interfaces xe-0/0/21.0 detail
```

```
Interface: xe-0/0/21.0, Index: 83, State: up, Port mode: Trunk
.
.
.
```

List the Ethernet switching interface information on aggregation layer FCoE transit switch TS2 using the operational mode commands **show ethernet-switching interfaces xe-0/0/30.0 detail** and **show ethernet-switching interfaces xe-0/0/31.0 detail**:

```
user@switch> show ethernet-switching interfaces xe-0/0/30.0 detail
Interface: xe-0/0/30.0, Index: 71, State: up, Port mode: Trunk
.
.
.

user@switch> show ethernet-switching interfaces xe-0/0/31.0 detail
Interface: xe-0/0/31.0, Index: 73, State: up, Port mode: Trunk
.
.
.
```

List the Ethernet switching interface information on FCoE transit switch TS3 using the operational mode commands **show ethernet-switching interfaces xe-0/0/10.0 detail** and **show ethernet-switching interfaces xe-0/0/11.0 detail**:

```
user@switch> show ethernet-switching interfaces xe-0/0/10.0 detail
Interface: xe-0/0/10.0, Index: 56, State: up, Port mode: Tagged-Access
.
.
.

user@switch> show ethernet-switching interfaces xe-0/0/11.0 detail
Interface: xe-0/0/11.0, Index: 59, State: up, Port mode: Trunk
.
.
.
```

Related Documentation

- Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Directly Connected to the Same FCoE Transit Switch) on page 228
- Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Directly Connected to Different FCoE Transit Switches) on page 233
- Enabling VN2VN_Port FIP Snooping and Configuring the Beacon Period on an FCoE Transit Switch on page 300
- Understanding VN_Port to VN_Port FIP Snooping on an FCoE Transit Switch on page 87

Example: Configuring Automated Fibre Channel Interface Load Rebalancing

Automated Fibre Channel (FC) interface (NP_Port) load rebalancing configures the switch to rebalance the session loads on the native FC interfaces automatically on a load-rebalancing trigger event. (Alternatively, you can rebalance the link load on the FC interfaces on demand so that you control when the link load is rebalanced.) Rebalancing the FC link load is a disruptive action that causes some or all of the current sessions to log out, then log in again to be placed on the active FC links in a balanced manner.

This example shows you how to configure and verify automated FC link load rebalancing on an FCoE-FC gateway local FC fabric.

Requirements

This example uses the following hardware and software components:

- Juniper Networks QFX3500 Switch
- Junos OS Release 12.3 or later for the QFX Series

Overview

When a load rebalancing trigger occurs, the switch automatically rebalances the link loads by redistributing the sessions across the active NP_Port links.

There are three possible load-rebalancing triggers:

1. When you enable automated load rebalancing, the switch checks the load balance on the existing NP_Port links. If the links are already balanced, the switch does not rebalance the link load. If the links are not balanced, the switch rebalances the link loads using the configured load-balancing algorithm.
2. When a new NP_Port link comes up on a local FCoE-FC gateway fabric, the switch rebalances the link load using the configured load-balancing algorithm if automated load balancing is enabled.
3. When the port speed is changed (unless the port speed change does not change the actual port speed, for example, changing the port speed from auto to 8 Gbps).

Automated load rebalancing logs out sessions in accordance with the configured load-balancing algorithm. Disabling automated load rebalancing is not disruptive because the link load is already balanced.

Use automated load rebalancing if you want link loads to be rebalanced automatically instead of at times of your choosing. Keep in mind that load rebalancing is a disruptive event (sessions are logged out).

Topology

This example configures automated load rebalancing on a local FC fabric on an FCoE-FC gateway. This example does not show you how to configure the load-balancing algorithm or any other load-balancing characteristics. The load-balancing configuration for this example is:

- FC fabric name—fc_fabric_100
- FC fabric ID—100

- FC fabric type—Proxy
- FC fabric interfaces—fc-0/0/0, fc-0/0/1, fc-0/0/42, fc-0/0/43, vlan.100, vlan.20
- Load-balancing algorithm—Simple
- No fabric WWN verify—Configured
- Traceoptions—Configured to log in file fc_fabric_100_proxy.log

Configuration

To configure automated load balancing on a local FC fabric, perform this task:

- [xref target has no title]
- Results on page 252

CLI Quick Configuration

To quickly configure automated load balancing, copy the following commands, paste them in a text file, remove line breaks, change variables and details to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level:

```
[edit]
set fc-fabrics fc_fabric_100 proxy auto-load-rebalance
```

Step-by-Step Procedure

- Configure automated load balancing on FC fabric fc_fabric_100:

  ```
  user@switch# set fc-fabrics fc_fabric_100 proxy auto-load-rebalance
  ```

Results

Display the results of the configuration:

```
user@switch> show configuration fc-fabrics
fc_fabric_100 {
    fabric-id 100;
    fabric-type proxy;
    interface {
        fc-0/0/0.0;
        fc-0/0/1.0;
        vlan.100;
        vlan.20;
        fc-0/0/42.0;
        fc-0/0/43.0;
    }
    proxy {
        traceoptions {
            file fc_fabric_100_proxy.log size 20m;
            flag all;
        }
        load-balance-algorithm simple;
        auto-load-rebalance;
        no-fabric-wwn-verify;
    }
}
```


Verification

Verifying That Automated Load Rebalancing Is Enabled

Purpose Verify that automated load rebalancing is configured on local FC fabric **fc_fabric_100**.

Action Verify the results of the automated load-rebalancing configuration using the operational mode command **show fibre-channel proxy fabric-state fabric fc_fabric_100**:

```
user@switch> show fibre-channel proxy fabric-state fabric fc_fabric_100
Fabric: fc_fabric_100, Fabric-id: 100
Proxy load balance algorithm: Simple, Fabric WWN verification: No
Auto load rebalance enabled  : Yes
Last rebalance start-time    : Never
Last rebalance end-time      : Never
Last rebalance trigger       : None
Last rebalance trigger-time  : Mon Sep 10 21:42:30 2012 usec: 814602
Last rebalance trigger-result: None
```

Meaning The **show fibre-channel proxy fabric-state fabric fc_fabric_100** operational command displays information about the specified local FC fabric. The output shows that the **Auto load rebalance enabled** field value is **Yes**, which indicates that automated load rebalancing is enabled on fabric fc_fabric_100.

Related Documentation

- Defining the Proxy Load-Balancing Algorithm on page 279
- Simulating On-Demand Fibre Channel Link Load Rebalancing (Dry Run Test) on page 280
- Understanding Load Balancing in an FCoE-FC Gateway Proxy Fabric on page 63
- Monitoring Fibre Channel Interface Load Balancing on page 383

Example: Configuring an FCoE LAG on a Redundant Server Node Group

This example shows how to configure a Fibre Channel over Ethernet (FCoE) link aggregation group (LAG) on a redundant server Node group (RSNG) to transport FCoE traffic and regular Ethernet traffic across the same link aggregation bundle. The FCoE servers have converged network adapters (CNAs) and communicate with the Fibre Channel (FC) storage area network (SAN). FCoE servers are usually connected to a switch that performs FIP snooping, such as an FCoE transit switch or an FCoE-FC gateway switch that performs FIP snooping. This example provides a common FCoE LAG configuration for an FCoE transit switch and an FCoE-FC gateway, and shows how to disable FIP snooping scaling on an FCoE untrusted FCoE-FC gateway fabric (fc-fabric).

Requirements

This example uses the following hardware and software components:

- Two Juniper Networks QFabric System Node devices configured as an RSNG. The Node devices can be configured as FCoE transit switches or as FCoE-FC gateways. (A configuration with one Node device as an FCoE transit switch and the other Node device as an FCoE-FC gateway is possible providing that the transit switch and the FCoE-FC gateway use different FCoE VLANs.)
- Junos OS Release 13.2X52-D10 or later for the QFX Series
- One FCoE server with two CNA ports

Overview

Standard LAGs use a hashing algorithm to determine which physical link in the LAG is used for a transmission, so a series of communications between two devices might use different physical links in the LAG for different transmissions. However, FCoE traffic requires a point-to-point link (or a virtual point-to-point link) between the FCoE device and the Fibre Channel (FC) storage area network (SAN) switch.

An FCoE LAG solves this problem by ensuring that the same LAG link is used for communication between a given FCoE device and the QFabric system Node device, preserving point-to-point link emulation. At the same time, regular Ethernet traffic (traffic that is not FCoE traffic) on the LAG is distributed across member interfaces in the same way as on a standard LAG. FCoE traffic is treated properly in terms of maintaining a virtual point-to-point link with the FC SAN, and regular Ethernet traffic enjoys the usual LAG benefits of load balancing and link redundancy.

NOTE: **Configuring a LAG as an FCoE LAG does not provide link redundancy for FCoE traffic, and does not load balance FCoE traffic.**

On FCoE-FC gateway untrusted Fibre Channel fabrics (fc-fabrics), if you configure an FCoE LAG, you must also disable enhanced FIP snooping scaling (scaling up to 2,500 sessions), which reduces the number of supported FIP snooping sessions to 376 sessions. On an FCoE-FC gateway, disabling enhanced FIP snooping scaling is global to the Node device. Trusted fc-fabrics on an FCoE-FC gateway support enhanced FIP snooping scaling.

This example shows you how to:

- Configure the RSNG and its Node devices
- Configure the FCoE LAG on the RSNG
- Configure a dedicated VLAN for FCoE traffic (an FCoE VLAN) and a native VLAN for untagged FCoE initialization protocol (FIP) traffic
- Enable VN2VF_Port FIP snooping on the FCoE VLAN
- Disable FIP snooping scaling on an untrusted FCoE-FC gateway fabric

NOTE: FCoE traffic requires lossless transport across the Ethernet network to comply with the requirements for transporting storage traffic. This example describes how to configure an FCoE LAG to provide redundancy for FCoE traffic. See "Example: Configuring CoS PFC for FCoE Traffic" on page 186 for how to configure lossless transport for FCoE traffic.

NOTE: On a Node device that is configured as an FCoE-FC gateway, you must create a Fibre Channel fabric, configure native FC interfaces, configure an FCoE VLAN interface (a Layer 3 RVI) for the FCoE VLAN (which includes the FCoE LAG as a member interface), and add the native FC interfaces and FCoE VLAN interface to the FC fabric. For an example of FCoE-FC gateway interface configuration, see "Example: Setting Up Fibre Channel and FCoE VLAN Interfaces in an FCoE-FC Gateway Fabric" on page 171.

Topology

Table 31 on page 255 shows the configuration components for this example.

Table 31: Components of the FCoE LAG Configuration Example

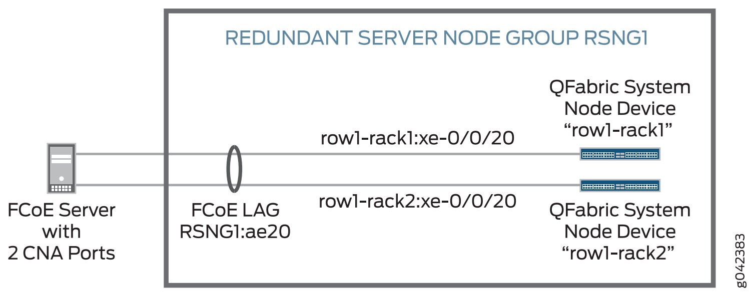
Component	Settings
Hardware	Two QFabric system Node devices configured as an RSNG (the Node devices can be configured as FCoE transit switches or as FCoE-FC gateways; this example is valid for both modes): • RSNG name—RSNG1 • First Node device—Serial number ABCD1234, alias name row1-rack1 • Second Node device—Serial number ABCD1235, alias name row1-rack2 NOTE: The alias names chosen for this example indicate the physical locations of the Node devices. You can use any aliasing system you want to make identifying Node devices easier, or you can use the default Node device names (the Node device serial numbers). One FCoE server with two CNA ports.
LAG configuration	RSNG device count—48 FCoE LAG name—RSNG1:ae20 FCoE LAG member interfaces—row1rack1:xe-0/0/20 and row1rack2:xe-0/0/20 FCoE LAG LACP—active FCoE LAG port mode—trunk MTU—2180 FCoE LAG VLAN memberships—FCoE VLAN (**fcoe-vlan1**) and native VLAN

Table 31: Components of the FCoE LAG Configuration Example *(continued)*

Component	Settings
FCoE VLAN	Name—**fcoe-vlan1** VLAN ID—**2000** Member interfaces—**RSNG1:ae20**
Native VLAN	Name—**native** VLAN ID—**1** Member interfaces—**RSNG1:ae20**
VN2VF_Port FIP snooping	Enabled on the FCoE VLAN (**fcoe-vlan1**)
FIP snooping scaling	Enabled for FCoE transit switch portion of the example. Disabled for the FCoE-FC gateway portion of the example (gateway FC fabric is FCoE untrusted).

Figure 16 on page 256 shows the network topology for this example.

Figure 16: FCoE LAG Example Topology



Configuration

To configure an FCoE LAG between an FCoE server with two CNA ports and the two Node device members of an RSNG, perform these tasks:

- Configuring an FCoE LAG on an RSNG (FCoE Transit Switch or FCoE-FC Gateway) on page 258
- Disabling Enhanced FIP Snooping Scaling on an FCoE-FC Gateway on page 259
- Results on page 259

CLI Quick Configuration

In this example, the enhanced FIP snooping scaling is disabled (376 sessions) n the FCoE-FC gateway because the gateway fabric is an untrusted fc-fabric.

Most of the FCoE LAG configuration is common to both the FCoE transit switch and FCoE-FC gateway modes of operation. The CLI Quick Configuration shows the common configuration statements first, followed by the additional configuration statement that disables FIP snooping scaling on the FCoE-FC gateway. Disabling FIP snooping scaling on an FCoE-FC gateway is a global configuration that affects all of the fc-fabrics on the gateway. (On an FCoE transit switch, you can disable FIP snooping scaling on an individual FCoE VLAN without affecting other FCoE VLANs.)

NOTE: This example does not include configuring the FC fabric, the native FC fabric ports, and the Layer 3 FCoE VLAN interface.

Common Configuration

To quickly configure an FCoE LAG, copy the following commands, paste them in a text file, remove line breaks, change variables and details to match your network configuration, and then copy and paste the commands into the CLI at the [edit] hierarchy level.

```
set fabric aliases node-device ABCD1234 row1-rack1
set fabric aliases node-device ABCD1235 row1-rack2
set fabric resources node-group RSNG1 node-device row1-rack1
set fabric resources node-group RSNG1 node-device row1-rack2
set chassis node-group RSNG1 aggregated-devices ethernet device-count 48
set interfaces RSNG1:ae20 unit 0 family ethernet-switching port-mode trunk vlan members
fcoe-vlan1
set interfaces RSNG1:ae20 unit 0 family ethernet-switching native-vlan-id 1
set interfaces RSNG1:ae20 mtu 2180
set interfaces RSNG1:ae20 aggregated-ether-options fcoe-lag
set interfaces RSNG1:ae20 aggregated-ether-options lacp active
set interfaces row1-rack1:xe-0/0/20 ether-options 802.3ad RSNG1:ae20
set interfaces row1-rack2:xe-0/0/20 ether-options 802.3ad RSNG1:ae20
set vlans fcoe-vlan1 vlan-id 2000
set vlans native vlan-id 1
set vlans fcoe-vlan1 interface RSNG1:ae20
set ethernet-switching-options secure-access-port vlan fcoe-vlan1 examine-fip
```

NOTE: If you want to configure an FCoE-FC gateway fabric as a trusted fabric so that you can leave enhanced FIP snooping scaling enabled on the gateway, add the following statement to the configuration, replacing the variable *fc-fabric-name* with the name of the FC fabric (if you do this, do not disable FIP snooping scaling as shown in the FCoE-FC Gateway Additional Configuration):

set fc-fabrics *fc-fabric-name* protocols fip fcoe-trusted

FCoE-FC Gateway Additional Configuration (Untrusted FC Fabric)

To disable enhanced FIP snooping scaling on an FCoE-FC gateway untrusted FC fabric, copy the following command, paste it in a text file, remove line breaks, and then copy and paste the command into the CLI at the [edit] hierarchy level.

```
set fc-options no-fip-snooping-scaling
```

Configuring an FCoE LAG on an RSNG (FCoE Transit Switch or FCoE-FC Gateway)

Step-by-Step Procedure

To configure the RSNG member Node devices, the FCoE LAG, the FCoE VLAN, and VN2VF_Port FIP snooping on an FCoE transit switch or an FCoE-FC gateway:

1. Define aliases for the two Node devices that will be in the RSNG (aliases are easier to remember and more descriptive than the Node device serial number). Name the Node device with serial number **ABCD1234** as **row1-rack1** and the Node device with the serial number **ABCD1235** as **row1-rack2**:

   ```
   admin@qfabric# set fabric aliases node-device ABCD1234 row1-rack1
   admin@qfabric# set fabric aliases node-device ABCD1235 row1-rack2
   ```

2. Configure the Node device membership for **row1-rack1** and **row1-rack2** in the RSNG **RSNG1**:

   ```
   admin@qfabric# set fabric resources node-group RSNG1 node-device row1-rack1
   admin@qfabric# set fabric resources node-group RSNG1 node-device row1-rack2
   ```

3. Configure the number of LAG interfaces that RSNG **RSNG1** can support. (Each Node device in the RSNG has 48 server-facing ports. If we used one port from each Node device to provide Node device redundancy for each LAG, we might need to support a maximum of 48 LAGs, so we set the device count to **48** LAGs.)

   ```
   admin@qfabric# set chassis node-group RSNG1 aggregated-devices ethernet device-count
   48
   ```

4. Configure the LAG interface (**ae20**) on RSNG1 and set the port mode to **trunk** mode. In the same statement, configure the LAG interface membership in the dedicated FCoE VLAN **fcoe-vlan1**:

   ```
   admin@qfabric# set interfaces RSNG1:ae20 unit 0 family ethernet-switching port-mode
   trunk vlan members fcoe-vlan1
   ```

5. Configure the LAG interface membership in the native VLAN:

   ```
   admin@qfabric# set interfaces RSNG1:ae20 unit 0 family ethernet-switching native-vlan-id
   1
   ```

6. Configure the LAG interface with an MTU of **2180** to accommodate the size of the FCoE frame and headers.

   ```
   admin@qfabric# set interfaces RSNG1:ae20 mtu 2180
   ```

7. Configure the LAG **RSNG1:ae20** as an FCoE LAG:

   ```
   admin@qfabric# set interfaces RSNG1:ae20 aggregated-ether-options fcoe-lag
   ```

8. Enable LACP on the FCoE LAG:

   ```
   admin@qfabric# set interfaces RSNG1:ae20 aggregated-ether-options lacp active
   ```

9. Assign one Ethernet interface on each RSNG Node device to the FCoE LAG:

   ```
   admin@qfabric# set interfaces row1-rack1:xe-0/0/20 ether-options 802.3ad RSNG1:ae20
   admin@qfabric# set interfaces row1-rack2:xe-0/0/20 ether-options 802.3ad RSNG1:ae20
   ```

10. Configure a dedicated VLAN for FCoE traffic (an FCoE VLAN) named **fcoe-vlan1** with the VLAN ID **2000**:

    ```
    admin@qfabric# set vlans fcoe-vlan1 vlan-id 2000
    ```


11. Configure a native VLAN with the VLAN ID 1 to carry untagged FIP traffic:

    `admin@qfabric#` **set vlans native vlan-id 1**

12. Assign the FCoE LAG interface to the FCoE VLAN:

    `admin@qfabric#` **set vlans fcoe-vlan1 interface RSNG:ae20**

13. Assign the FCoE LAG interface to the native VLAN:

    `admin@qfabric#` **set vlans native interface RSNG:ae20**

14. Enable VN2VF_Port FIP snooping on the FCoE VLAN:

    `admin@qfabric#` **set ethernet-switching-options secure-access-port vlan fcoe-vlan1 examine-fip**

Disabling Enhanced FIP Snooping Scaling on an FCoE-FC Gateway

Step-by-Step Procedure

To disable enhanced FIP snooping scaling on an FCoE-FC gateway:

1. Disable FIP snooping scaling on the gateway fabrics. Disabling FIP snooping scaling on an FCoE-FC gateway is global to the gateway, so every FC fabric on the gateway reverts to supporting 376 sessions (instead of 2,500 sessions as with FIP snooping scaling enabled).

   `admin@qfabric#` **set fc-options no-fip-snooping-scaling**

Results

Display the results of the configuration. The results below show the configuration on an FCoE transit switch and have been edited to include only the components configured in the example:

```
admin@qfabric> show configuration
root@qfabric>fabric {
    resources {
        node-group RSNG1 {
            node-device row1-rack1;
            node-device row1-rack2;
        }
    }
    aliases {
        node-device ABCD1234 {
            row1-rack1;
        }
        node-device ABCD1235 {
            row1-rack2;
        }
    }
}
chassis {
    node-group RSNG1 {
        aggregated-devices {
            ethernet {
                device-count 48;
            }
        }
    }
}
```

```
interfaces {
    RSNG1:ae20 {
        aggregated-ether-options {
            fcoe-lag;
            lacp {
                active;
            }
        }
        unit 0 {
            family ethernet-switching {
                port-mode trunk;
                vlan {
                    members fcoe-vlan1;
                }
                native-vlan-id 1;
            }
        }
    }
    row1-rack1:xe-0/0/20 {
        ether-options {
            802.3ad RSNG1:ae20;
        }
    }
    row1-rack2:xe-0/0/20 {
        ether-options {
            802.3ad RSNG1:ae20;
        }
    }
}
ethernet-switching-options {
    secure-access-port {
        vlan fcoe-vlan1 {
            examine-fip;
        }
    }
}
vlans {
    fcoe-vlan1 {
        vlan-id 2000;
        interface {
            RSNG1:ae20.0;
        }
    }
    native {
        vlan-id 1;
        interface {
            RSNG1:ae20.0;
        }
    }
}
```

Verification

To verify the configuration of the QFabric system Node device resources, FCoE LAG, FCoE VLAN, native VLAN, and FIP snooping, perform these tasks:

- Verifying the Node Device Aliases (Names) on page 261
- Verifying the Node device Assignment to the Node Group on page 261

Verifying the Node Device Aliases (Names)

Purpose Verify that the Node device alias names are configured.

Action List the Node device inventory on the QFabric system using the **show fabric administration inventory node-devices** command:

```
admin@qfabric> show fabric administration inventory node-devices
root@qfabric>show fabric administration inventory node-devices
Item                    Identifier              Connection      Model
Node device
  row1-rack1            ABCD1234                Connected       qfx3500
  row1-rack2            ABCD1235                Connected       qfx3500
```

Meaning The **show fabric administration inventory node-devices** command lists the Node device names in the *Node device* column and lists the Node device serial numbers in the *Identifier* column. The *Connection* column shows if the Director device has detected the Node device, and the *Model* column lists QFX switch model type.

The command output shows that Node device **ABCD1234** is configured with the name (alias) **row1-rack1**, and the Node device **ABCD1235** is configured with the name **row1-rack2**.

Verifying the Node device Assignment to the Node Group

Purpose Verify that the redundant server Node group includes the two Node devices.

Action Verify that the QFabric system Node group **RSNG1** is configured with the correct Node devices using the **show configuration fabric resources** command:

```
admin@qfabric> show configuration fabric resources
root@qfabric> show configuration fabric resources
node-group RSNG1 {
    node-device row1-rack1;
    node-device row1-rack2;
}
```

Meaning The **show configuration fabric resources** command lists the Node groups and the Node devices in the Node groups. The command output shows that Node group **RSNG1** consists of the Node devices **row1-rack1** and **row1-rack2**.

Verifying the Number of Aggregated Ethernet Logical Devices (LAG Interfaces) That the Node Group Can Support

Purpose Verify the number of LAG interfaces that the redundant server node group supports.

Action List the LAG interface device count using the **show configuration chassis** command:

```
admin@qfabric> show configuration chassis
node-group RSNG1 {
    aggregated-devices {
        ethernet {
            device-count 48;
        }
    }
}
```

Meaning The **show configuration chassis** command displays the Ethernet device count (the number of LAG interfaces supported) as **48** devices.

Verifying the FCoE LAG Interface Configuration

Purpose Verify that the FCoE LAG interface, port mode, interface VLAN membership, and Node device interface membership in the FCoE LAG are correctly configured.

Action List the FCoE LAG interface and Node device interface information using the **show configuration interfaces** command:

```
admin@qfabric> show configuration interfaces
RSNG1:ae20 {
    aggregated-ether-options {
        fcoe-lag;
        lacp {
            active;
        }
    }
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members fcoe-vlan1;
            }
            native-vlan-id 1;
        }
    }
}
row1-rack1:xe-0/0/20 {
    ether-options {
        802.3ad RSNG1:ae20;
    }
}
row1-rack2:xe-0/0/20 {
    ether-options {
        802.3ad RSNG1:ae20;
    }
}
```

Meaning The **show configuration interfaces** command lists both the LAG interfaces and the individual Node device interfaces, and their configuration.

The command output shows a lot of information about the interfaces:

- The LAG interface name is **RSNG1:ae20**
- **fcoe-lag** confirms the LAG is an FCoE LAG
- **lacp** is configured in **active** mode
- Port mode is **trunk**
- The LAG has membership in the **fcoe-vlan1** VLAN and in the native VLAN with the VLAN ID **1**.
- Interface **row1-rack1:xe-0/0/20** is a member of FCoE LAG **RSNG1:ae20**
- Interface **row1-rack2:xe-0/0/20** is a member of FCoE LAG **RSNG1:ae20**

Verifying the FCoE VLAN and Native VLAN Configuration

Purpose Verify that the FCoE VLAN **fcoe-vlan1** and the native VLAN **native** are configured with the correct VLAN tags (**2000** and **1**, respectively) and that the FCoE LAG interface **RSNG1:ae20** is assigned to the VLANs.

Action List the VLAN information using the **show configuration vlans** command:

```
admin@qfabric> show configuration vlans
fcoe-vlan1 {
    vlan-id 2000;
    interface {
        RSNG1:ae20.0;
    }
}
native {
    vlan-id 1;
    interface {
        RSNG1:ae20.0;
}
```

Meaning The **show configuration vlans** command lists the configured VLANs, their VLAN IDs, and the interfaces assigned to the VLANs.

The command output shows that the FCoE VLAN **fcoe-vlan1** is configured with the VLAN ID **2000** and is assigned to the FCoE LAG interface **RSNG1:ae20**.

The command output also shows that the native VLAN **native** is configured with the VLAN ID **1** and is assigned to the FCoE LAG interface **RSNG1:ae20**.

Verifying the FIP Snooping Configuration

Purpose Verify that VN2VF_Port FIP snooping is enabled on the FCoE VLAN (**fcoe-vlan1**).

Action List the FIP snooping information using the **show configuration ethernet-switching-options** command:

```
admin@qfabric> show configuration ethernet-switching-options
secure-access-port {
    vlan fcoe-vlan1 {
        examine-fip;
    }
}
```

Meaning The **show configuration ethernet-switching-options** command lists the security options configured on VLANs. The command output shows that on VLAN **fcoe-vlan1**, VN2VF_Port FIP snooping is enabled (**examine-fip** output).

Related Documentation

- Configuring an FCoE LAG on page 302
- Configuring VLANs for FCoE Traffic on an FCoE Transit Switch on page 289
- Example: Setting Up Fibre Channel and FCoE VLAN Interfaces in an FCoE-FC Gateway Fabric on page 171
- Example: Configuring CoS PFC for FCoE Traffic on page 186
- *Example: Configuring CoS Hierarchical Port Scheduling (ETS)*
- Understanding FCoE LAGs on page 158
- Understanding CoS Flow Control (Ethernet PAUSE and PFC) on page 121

CHAPTER 6

Configuration Tasks

- Disabling the ETS Recommendation TLV on page 309
- Defining an Application for DCBX Application Protocol TLV Exchange on page 309
- Configuring an Application Map for DCBX Application Protocol TLV Exchange on page 311
- Applying an Application Map to an Interface for DCBX Application Protocol TLV Exchange on page 312

Configuring an FCoE-FC Gateway Fibre Channel Fabric

Fibre Channel (FC) fabric configuration consists of creating a unique name and identifier for each FC fabric you want to create and configuring it as an FCoE-FC gateway.

You can create a maximum of 12 FC fabrics on a QFX3500 switch. After you create a fabric, you can create and assign interfaces to the fabric, configure FIP parameters for the fabric, and set proxy traceoptions.

To configure an FC fabric using the CLI, specify a unique name and identification number for the fabric:

1. Configure the fabric name and fabric ID:

   ```
   [edit]
   user@switch# set fc-fabrics fabric-name fabric-id fabric-id
   ```

   NOTE: **Changing the fabric name or the fabric ID causes all logins to drop and forces the ENodes to log in again.**

   For example, to configure an FC fabric with the name **fab_ulous** and the fabric ID **10** (the range of **fabric-id** values is 1 through 4095):

   ```
   [edit]
   user@switch# set fc-fabrics fab_ulous fabric-id 10
   ```
2. Configure the fabric as a gateway fabric:

   ```
   [edit fc-fabrics fabric-name]
   user@switch# set fabric-type proxy
   ```

   For example, to configure the FC fabric with the name **fab_ulous** as a gateway fabric:

   ```
   [edit fc-fabrics fab_ulous]
   user@switch# set fabric-type proxy
   ```

Related Documentation

- Configuring a Fibre Channel Interface on page 269
- Configuring an FCoE VLAN Interface on an FCoE-FC Gateway on page 272
- Assigning Interfaces to a Fibre Channel Fabric on page 276
- Configuring FIP on an FCoE-FC Gateway on page 283
- Example: Setting Up Fibre Channel and FCoE VLAN Interfaces in an FCoE-FC Gateway Fabric on page 171
- Example: Configuring CoS PFC for FCoE Traffic on page 186
- Understanding an FCoE-FC Gateway on page 29

Disabling the Fabric WWN Verification Check

When a QFX Series NP_Port sends a fabric login (FLOGI) request to a Fibre Channel (FC) switch, the FLOGI accept (FLOGI-ACC) reply from the FC switch contains the SAN fabric worldwide name (WWN). The QFX Series uses the SAN fabric WWN in the multicast discovery advertisement (MDA) that the QFX Series sends to the ENodes in the FCoE network.

However, some FC switches substitute their own WWN (often the FC switch's virtual WWN) for the SAN fabric WWN in the FLOGI-ACC message. In this case, different NP_Ports that log in to the same FC fabric might receive different fabric WWNs in the FLOGI-ACC messages if the NP_Ports are connected to different FC switches in the SAN fabric.

If the QFX Series receives different fabric WWNs on NP_Ports that are connected to the same SAN fabric, the QFX Series uses the first fabric WWN it receives in the MDA it sends to the ENodes. The QFX Series isolates the NP_Ports that receive a different fabric WWN from other FC switches in that SAN fabric. No ENode sessions are assigned to the isolated NP_Ports. FC traffic is assigned only to NP_Ports that receive a fabric WWN in the FLOGI-ACC message that matches the fabric WWN received by the first NP_Port to log in to the FC fabric. (If an NP_Port receives a fabric WWN that does not match the fabric WWN received by the first NP_Port to log in to the FC fabric, it does not carry traffic to the SAN fabric.)

To prevent ENodes from being isolated due to a mismatched fabric WWN, you can disable the gateway fabric WWN verification check. Disabling the fabric WWN verification check enables all NP_Ports connected to a SAN fabric are used to carry traffic between the gateway and the FC switch, regardless of the fabric WWN the NP_Port receives in the FLOGI-ACC message.

NOTE: **Disabling or enabling the fabric WWN verification check logs out all FCoE sessions.**

To disable the fabric WWN verification check:

- `[edit fc-fabrics` *`fabric-name`* `proxy]`
  `user@switch#` **set no-fabric-wwn-verify**

Related Documentation

- Understanding FCoE-FC Gateway Functions on page 33
- **show fibre-channel proxy fabric-state on page 479**

Configuring a Physical Fibre Channel Interface

When you configure the switch as an FCoE-FC gateway, you must configure either 6 or 12 of the physical interfaces as native FC interfaces. Native FC interfaces connect to the storage area network (SAN) FC switch.

You can configure ports xe-0/0/0 through xe-0/0/5 as fc-0/0/0 through fc-0/0/5, and ports xe-0/0/42 through xe-0/0/47 as fc-0/0/42 through fc-0/0/47 to create blocks of native FC interfaces. You cannot individually configure a single port as a native FC interface. Within these port blocks, you cannot mix FC interfaces with Ethernet interfaces. All of the ports in a block must be either native FC interfaces or Ethernet interfaces.

You can configure:

- Six native FC interfaces by configuring either ports xe-0/0/0 through xe-0/0/5 as fc-0/0/0 through fc-0/0/5, or ports xe-0/0/42 through xe-0/0/47 as fc-0/0/42 through fc-0/0/47.
- Twelve native FC interfaces by configuring ports xe-0/0/0 through xe-0/0/5 as fc-0/0/0 through fc-0/0/5 and ports xe-0/0/42 through xe-0/0/47 as fc-0/0/42 through fc-0/0/47.
- No native FC interfaces by leaving ports xe-0/0/0 through xe-0/0/5 and ports xe-0/0/42 through xe-0/0/47 in their default state as Ethernet interfaces.

- To configure physical FC interfaces using the CLI, specify the physical port block you want to configure on the switch as native FC interfaces:

  ```
  [edit chassis]
  user@switch# set fpc fpc pic pic fibre-channel port-range port-range-low port-range-high
  ```

  For example, to configure six native FC interfaces, you can configure ports 0 through 5 as physical FC interfaces:

  ```
  [edit chassis]
  user@switch# set fpc 0 pic 0 fibre-channel port-range 0 5
  ```

  To configure 12 native FC interfaces requires two separate statements:

  ```
  [edit chassis]
  user@switch# set fpc 0 pic 0 fibre-channel port-range 0 5
  user@switch# set fpc 0 pic 0 fibre-channel port-range 42 47
  ```

Related Documentation

- Configuring an FCoE-FC Gateway Fibre Channel Fabric on page 266
- Configuring an FCoE VLAN Interface on an FCoE-FC Gateway on page 272
- Configuring a Fibre Channel Interface on page 269
- Assigning Interfaces to a Fibre Channel Fabric on page 276
- Example: Setting Up Fibre Channel and FCoE VLAN Interfaces in an FCoE-FC Gateway Fabric on page 171
- Understanding Interfaces on an FCoE-FC Gateway on page 50

Configuring a Fibre Channel Interface

When a QFX3500 acts as an FCoE-FC gateway, native Fibre Channel (FC) traffic flows between the switch and the storage area network (SAN) FC switch. When you configure a port as an FC interface, it transports only FC traffic. It does not transport Ethernet traffic.

You can configure ports xe-0/0/0 through xe-0/0/5 as fc-0/0/0 through fc-0/0/5 and ports xe-0/0/42 through xe-0/0/47 as fc-0/0/42 through fc-0/0/47 to create blocks of native FC interfaces.

Each of these blocks of ports must be configured either as all Ethernet ports or as all native FC ports. Within each block of ports, you cannot mix FC and Ethernet interfaces. This means that you can configure 0, 6, or 12 ports as native FC ports. "Configuring a Physical Fibre Channel Interface" on page 268 describes how to configure the port blocks as physical FC interfaces.

NOTE: **Do not configure ports that you want to use for native FC traffic as part of an Ethernet VLAN or as Ethernet ports.**

Configure a port as an FC interface when the port connects to the F_Port of an FC switch.

FC interface configuration includes:

- Explicitly specifying one or more ports as an FC family interface in NP_Port mode (mandatory).
- Configuring the FC interface options port speed and buffer-to-buffer credit state change number (BB_SC_N) (optional).
- Configuring the interface as a loopback interface (optional).

The buffer-to-buffer state change number feature prevents the loss of buffer-to-buffer credits between the two interfaces on either end of an FC link. The state change number determines the number of frames and receiver ready (R_RDY) primitives the interfaces exchange between the state change send (BB_SCs) and the state change receive (BB_SCr) primitives used to track these transactions.

Enabling BB_SC_N by configuring BB_SC_N on both of the FC link interfaces:

- Requests that $2^{BB\_SC\_N}$ number of frames be sent between two consecutive BB_SCs primitives, and
- Requests that $2^{BB\_SC\_N}$ number of R_RDY primitives be sent between two consecutive BB_SCr primitives.

When the number of R_RDY primitives received equals $2^{BB\_SC\_N}$, the R_RDY counter resets to zero. When the number of frames received equals $2^{BB\_SC\_N}$, the frame counter resets to zero. The interfaces calculate the number of buffer-to-buffer credits lost based on counter discrepancies and take corrective action to recover the lost credits.

If you enable BB_SC_N, the recommended BB_SC_N setting is eight. Setting the BB_SC_N number to zero (0) disables the feature. If either of the two connected FC interfaces is configured with zero as the BB_SC_N value, then both interfaces disable the feature. If the two connected FC interfaces have different nonzero BB_SC_N numbers configured, both interfaces use the higher number.

For the port to transport FC traffic, you must also set the physical port as an FC port using the port-range command.

To configure an FC interface using the CLI:

1. Specify the interface as family FC and set the port mode to NP_Port (setting the port mode to NP_Port is a mandatory configuration):

   ```
   [edit]
   user@switch# set interfaces interface-name unit unit family fibre-channel port-mode np-port
   ```

   For example, to configure the interface **fc-0/0/3** as an FC interface and set the port mode to **np-port**:

   ```
   [edit]
   user@switch# set interfaces fc-0/0/3 unit 0 family fibre-channel port-mode np-port
   ```
2. Configure the FC interface speed option:

   ```
   [edit]
   user@switch: set interfaces interface-name fibrechannel-options speed (auto-negotiation | 2g | 4g | 8g)
   ```

   For example, to set the FC interface speed option to **8g** for the interface **fc-0/0/3**:

   ```
   [edit]
   user@switch: set interfaces fc-0/0/3 fibrechannel-options speed 8g
   ```

   The default port mode is **auto-negotiation**, which sets the port speed to match the speed of the attached FC F_Port interface (2 Gbps, 4 Gbps, or 8 Gbps).

3. Configure the optional buffer-to-buffer credit state change number:

   ```
   [edit]
   user@switch: set interfaces interface-name fibrechannel-options bb-sc-n 0..15
   ```

   For example, to set the FC interface buffer-to-buffer credit state change number to **8** for the interface **fc-0/0/3**:

   ```
   [edit]
   user@switch: set interfaces fc-0/0/3 fibrechannel-options bb-sc-n 8
   ```

After you configure one or more FC interfaces, assign them and an FCoE VLAN to an FC fabric.

Related Documentation

- Assigning Interfaces to a Fibre Channel Fabric on page 276
- Configuring an FCoE VLAN Interface on an FCoE-FC Gateway on page 272
- Configuring a Physical Fibre Channel Interface on page 268
- Deleting a Fibre Channel Interface on page 277
- Example: Setting Up Fibre Channel and FCoE VLAN Interfaces in an FCoE-FC Gateway Fabric on page 171

- Understanding Interfaces on an FCoE-FC Gateway on page 50
- Understanding an FCoE-FC Gateway on page 29

Configuring an FCoE VLAN Interface on an FCoE-FC Gateway

When you configure the switch as an FCoE-FC gateway, a Layer 3 FCoE VLAN interface transmits and receives Fibre Channel over Ethernet (FCoE) traffic between the gateway and FCoE-capable servers on the Ethernet network. Configuring a Layer 3 FCoE VLAN interface on the switch creates virtual fabric port (VF_Port) interfaces facing the FCoE server virtual node ports (VN_Ports).

The FCoE VLAN interface is the interface for the dedicated VLAN the FCoE servers use for FCoE traffic. Each FC fabric requires at least one dedicated FCoE VLAN and at least one Layer 3 FCoE VLAN interface to transport FCoE traffic. On QFabric systems, the FCoE VLAN interface, the FCoE VLAN, and the interfaces that are members of the FCoE VLAN must be on the same Node device.

NOTE: FCoE VLANs (any VLAN that carries FCoE traffic) support only Spanning Tree Protocol (STP) and link aggregation group (LAG) Layer 2 features.

FCoE traffic cannot use a standard LAG because traffic might be hashed to different physical LAG links on different transmissions. This breaks the (virtual) point-to-point link that Fibre Channel traffic requires. If you configure a standard LAG interface for FCoE traffic, FCoE traffic might be rejected by the FC SAN.

QFabric systems support a special LAG called an FCoE LAG, which enables you to transport FCoE traffic and regular Ethernet traffic (traffic that is not FCoE traffic) across the same link aggregation bundle. Standard LAGs use a hashing algorithm to determine which physical link in the LAG is used for a transmission, so communication between two devices might use different physical links in the LAG for different transmissions. An FCoE LAG ensures that FCoE traffic uses the same physical link in the LAG for requests and replies in order to preserve the virtual point-to-point link between the FCoE device converged network adapter (CNA) and the FC SAN switch across a QFabric system Node device. An FCoE LAG does not provide load balancing or link redundancy for FCoE traffic. However, regular Ethernet traffic uses the standard hashing algorithm and receives the usual LAG benefits of load balancing and link redundancy in an FCoE LAG.

If the member interfaces of an FCoE VLAN belong to an FCoE LAG and are part of an FCoE untrusted FC fabric on the gateway, you must disable FIP snooping scaling on the gateway. FCoE untrusted gateway fabrics that include FCoE LAGs do not support enhanced FIP snooping scaling.

NOTE: To configure an FCoE VLAN on a device that you are using as transit switch, you do not use an FCoE VLAN interface. Instead, use the procedure described in "Configuring VLANs for FCoE Traffic on an FCoE Transit Switch" on page 289.

Before you configure an FCoE VLAN interface, create the FCoE VLAN and assign 10-Gigabit

Ethernet interfaces configured in tagged-access port mode to the VLAN. These 10-Gigabit Ethernet interfaces are the physical interfaces that transport the FCoE traffic to and from the FCoE devices in the Ethernet network.

Each Ethernet interface that connects to FCoE devices must also include the native VLAN to transport FIP traffic, because FIP VLAN discovery and notification frames are exchanged as untagged packets. The FCoE VLAN must carry only FCoE traffic. A VLAN cannot transport a mix of FCoE and standard Ethernet traffic.

FCoE VLAN interface configuration includes:

- Configuring a VLAN to use as a dedicated FCoE VLAN.
- Configuring a native VLAN for FIP traffic.
- Configuring member interfaces for the FCoE VLAN.
- Configuring the FCoE VLAN as a Fibre Channel (family) VLAN and setting the port mode value to **f-port**. Explicitly configuring the FCoE VLAN interface in F_Port mode is mandatory. The switch interface with which the FCoE server VN_Ports communicate must present a VF_Port to the servers.
- Configuring the FCoE VLAN interface as the Layer 3 interface for FCoE traffic.

To configure an FCoE VLAN interface:

1. Configure a dedicated FCoE VLAN:

   ```
   [edit vlans]
   user@switch# set vlan-name vlan-id vlan-id
   ```

   For example, to configure a VLAN named **fcoe_vlan** with a VLAN ID of **100** as the FCoE VLAN:

   ```
   [edit vlans]
   user@switch# set fcoe_vlan vlan-id 100
   ```

2. Configure a native VLAN for FIP traffic:

   ```
   [edit vlans]
   user@switch# set native vlan-id vlan-id
   ```

   For example, to configure the native VLAN with a VLAN ID of **1**:

   ```
   [edit vlans]
   user@switch# set native vlan-id 1
   ```

3. Configure member interfaces for the FCoE VLAN (use **ethernet-switching** as the family and **tagged-access** as the port mode):

   ```
   [edit interfaces]
   user@switch# set interface-name unit unit family family port-mode mode vlan members
   vlan-name
   ```

   For example, to configure the interface **xe-0/0/10** as a member of the FCoE VLAN **fcoe_vlan**:

   ```
   [edit interfaces]
   user@switch# set xe-0/0/10 unit 0 family ethernet-switching port-mode tagged-access vlan
   members fcoe_vlan
   ```


4. Configure the native VLAN on the FCoE VLAN member interfaces:

   ```
   [edit interfaces]
   user@switch# set interface-name unit unit family family native-vlan-id vlan-id
   ```

   For example, to configure the interface **xe-0/0/10** as a member of the native VLAN with the native VLAN ID **1**:

   ```
   [edit interfaces]
   user@switch# set xe-0/0/10 unit 0 family ethernet-switching native-vlan-id 1
   ```

5. Assign the Ethernet interfaces to the FCoE VLAN:

   ```
   [edit vlans]
   user@switch# set vlan-name interface interface-name
   ```

   For example, to assign the interface **xe-0/0/10.0** to the FCoE VLAN named **fcoe_vlan**:

   ```
   [edit vlans]
   user@switch# set fcoe_vlan interface xe-0/0/10.0
   ```

6. Define an interface as an FCoE VLAN interface in F_Port mode (to present a VF_Port to the FCoE servers):

   ```
   [edit interfaces]
   user@switch# set vlan unit unit family fibre-channel port-mode f-port
   ```

   For example, to configure VLAN unit **100** as an FCoE VLAN interface and set the port mode to **f-port**:

   ```
   [edit interfaces]
   user@switch# set vlan unit 100 family fibre-channel port-mode f-port
   ```

7. Define the Layer 3 FCoE VLAN interface:

   ```
   [edit vlans]
   user@switch# set vlan-name l3-interface vlan-interface-name
   ```

   For example, to configure VLAN interface unit **100** (the FCoE VLAN interface defined earlier in this example) as the Layer 3 FCoE VLAN interface for FCoE VLAN **fcoe_vlan**:

   ```
   [edit vlans]
   user@switch# set fcoe_vlan l3-interface vlan.100
   ```

Related Documentation

- Understanding Interfaces on an FCoE-FC Gateway on page 50
- Configuring an FCoE-FC Gateway Fibre Channel Fabric on page 266
- Configuring a Physical Fibre Channel Interface on page 268
- Configuring a Fibre Channel Interface on page 269
- Assigning Interfaces to a Fibre Channel Fabric on page 276
- Disabling VN2VF_Port FIP Snooping on an FCoE-FC Gateway Switch Interface on page 299
- Configuring VLANs for FCoE Traffic on an FCoE Transit Switch on page 289
- Configuring an FCoE LAG on page 302
- Example: Setting Up Fibre Channel and FCoE VLAN Interfaces in an FCoE-FC Gateway Fabric on page 171

Assigning Interfaces to a Fibre Channel Fabric

When you configure the switch as an FCoE-FC gateway, you assign one or more (up to 12) native Fibre Channel (FC) interfaces and at least one FCoE VLAN interface to each FC fabric. FC interfaces transport native FC traffic between the proxy gateway and the storage area network (SAN) FC switch. FCoE VLAN interfaces transport FCoE traffic between FCoE-capable servers and the gateway.

Each FC fabric needs both types of interfaces to transport traffic between FCoE servers on the Ethernet network and FC storage devices in the core FC network behind the FC switch. FCoE traffic between the FCoE servers and the gateway must travel in a dedicated FCoE VLAN. Native FC traffic passes between the gateway and the FC switch on the native FC interfaces.

You must configure the FC interfaces and the FCoE VLAN interfaces that you assign to a particular fabric on the same Juniper Networks QFX3500 Switch. Traffic between an FCoE device and the FC switch must ingress and egress the same gateway.

To assign core-facing native FC interfaces and a server-facing FCoE VLAN interface to an FC fabric, configure a fabric and then specify the interfaces:

1. Assign the native FC interfaces to the FC fabric:

```
[edit fc-fabrics fabric-name]
user@switch: set interface interface-name
user@switch: set interface interface-name
user@switch: set interface interface-name
...
```

2. Assign an FCoE VLAN interface to the FC fabric:

```
[edit fc-fabrics fabric-name]
user@switch: set interface vlan-name
```

For example, to assign the native FC interfaces **fc-0/0/0.0**, **fc-0/0/1.0**, and **fc-0/0/2.0** and the FCoE VLAN interface **vlan.100** to an FC fabric named **san_tana**:

```
user@switch: set fc-fabrics san_tana interface fc-0/0/0.0
user@switch: set fc-fabrics san_tana interface fc-0/0/1.0
user@switch: set fc-fabrics san_tana interface fc-0/0/2.0
user@switch: set fc-fabrics san_tana interface vlan.100
```

Related Documentation

- Configuring a Fibre Channel Interface on page 269
- Configuring an FCoE VLAN Interface on an FCoE-FC Gateway on page 272
- Example: Setting Up Fibre Channel and FCoE VLAN Interfaces in an FCoE-FC Gateway Fabric on page 171
- Understanding Interfaces on an FCoE-FC Gateway on page 50
- Understanding an FCoE-FC Gateway on page 29

Deleting a Fibre Channel Interface

Before you delete a Fibre Channel (FC) interface, you must first delete the interface from the FC fabric configuration. This prevents configuration errors that would result if you deleted an FC interface from the **[edit interfaces]** hierarchy level but did not delete the interface from the FC fabric.

When you configure the switch as an FCoE-FC gateway, FC interfaces transmit and receive native FC traffic between the gateway and the FC switch. You can configure ports xe-0/0/0 through xe-0/0/5 as fc-0/0/0 through fc-0/0/5 and ports xe-0/0/42 through xe-0/0/47 as fc-0/0/42 through fc-0/0/47 to create one or two blocks of six native FC interfaces.

To delete an FC interface using the CLI:

1. Delete the FC interface from the FC fabric to which it belongs:

   ```
   [edit]
   user@switch# delete fc-fabrics fabric-name interface interface-name
   ```

   For example, to delete the FC interface **fc-0/0/3.0** from an FC fabric named **sanfab1**:

   ```
   [edit]
   user@switch# delete fc-fabrics sanfab1 interface fc-0/0/3.0
   ```
2. Delete the FC interface from the switch **[edit interfaces]** hierarchy:

   ```
   [edit]
   user@switch: delete interfaces interface-name
   ```

   For example, to delete the interface **fc-0/0/3.0** from the switch:

   ```
   [edit]
   user@switch: delete interfaces fc-0/0/3.0
   ```

The FC interface has been deleted from the FC fabric and from the switch.

Related Documentation

- Assigning Interfaces to a Fibre Channel Fabric on page 276
- Configuring a Physical Fibre Channel Interface on page 268
- Example: Setting Up Fibre Channel and FCoE VLAN Interfaces in an FCoE-FC Gateway Fabric on page 171
- Understanding Interfaces on an FCoE-FC Gateway on page 50

Disabling Storm Control on FCoE Interfaces on an FCoE-FC Gateway

Storm control is not supported on the FCoE interfaces of an FCoE-FC gateway VLAN. Enabling storm control on an FCoE-FC gateway VLAN interface may cause FCoE packet loss. Storm control is disabled by default on all interfaces. However, if you enabled storm control globally on all switch interfaces or on any interfaces that are part of the FCoE VLAN interface, you must disable storm control on the Ethernet interfaces of the FCoE VLAN.

If storm control is enabled on only a few interfaces of the FCoE VLAN, you can disable storm control on individual interfaces by including the **delete ethernet-switching-options storm-control interface *interface-name*** statement in the configuration, where ***interface-name*** is the name of the interface on which you want to disable storm control.

If storm control is enabled globally on the switch when the switch is acting as an FCoE-FC gateway, it is often easiest to disable storm control on all interfaces, then enable storm control only on Ethernet interfaces that are not part of the FCoE VLAN interface.

If storm control is enabled globally, you can disable storm control in either of two ways:

- Disable storm control on all interfaces, then enable storm control on the interfaces you want to use storm control. (From the default configuration, you cannot disable storm control on individual interfaces because the default configuration enables storm control on **all** interfaces, not on individual interfaces.)

  For example, if you want interfaces xe-0/0/20, xe-0/0/21, and xe-0/0/22 to use storm control, disable storm control on all interfaces, then enable storm control on those three interfaces:

  1. Disable storm control on all interfaces:

     ```
     user@switch# delete ethernet-switching-options storm-control interface all
     ```
  2. Enable storm control on interfaces xe-0/0/20, xe-0/0/21, and xe-0/0/22:

     ```
     user@switch# set ethernet-switching-options storm-control interface xe-0/0/20
     user@switch# set ethernet-switching-options storm-control interface xe-0/0/21
     user@switch# set ethernet-switching-options storm-control interface xe-0/0/22
     ```
- Disable storm control for all unknown unicast traffic on all interfaces by including the following statement in your configuration:

  ```
  user@switch# set ethernet-switching-options storm-control interface all no-unknown-unicast
  ```

Related Documentation

- Understanding Interfaces on an FCoE-FC Gateway on page 50
- *Understanding Storm Control*
- Example: Setting Up Fibre Channel and FCoE VLAN Interfaces in an FCoE-FC Gateway Fabric on page 171

Defining the Proxy Load-Balancing Algorithm

When the QFX Series is configured as an FCoE-FC gateway, it balances the FCoE session load assigned to each NP_Port link between the gateway and the FC switch in the FC SAN to avoid overloading or underutilizing each link. The QFX Series supports three types of load-balancing mechanisms:

- Simple load balancing—Load balancing is based on the weighted utilization (session load) of the NP_Ports connected to an FC fabric. The session load is the sum of the FLOGI and FDISC sessions on each link. Each new ENode fabric login (FLOGI) or VN_Port fabric discovery (FDISC) session is assigned to the least-loaded link, so an FDISC session initiated by the VN_Port on an ENode might not be assigned to the same link as the parent ENode's FLOGI session. Simple load balancing is the default algorithm. Simple load balancing is the default load-balancing algorithm. Rebalancing the link load disrupts only selected sessions to minimize the impact (the switch uses an algorithm to log out only the sessions that need to be moved to other links to balance the load when those sessions log in again).
- ENode-based load balancing—Load balancing is based on the weighted utilization (session load) of the NP_Ports connected to an FC fabric. The session load is the sum of the FLOGI and FDISC sessions on each link. However, when an ENode logs in to the fabric, the switch places all subsequent VN_Port FDISC sessions associated with that ENode on the same link as the ENode FLOGI session, regardless of the link load. New ENode FLOGIs are placed on the least-loaded link. The switch calculates the link load based on the combined total of FLOGIs and FDISCs on each NP_Port link. Rebalancing the link load disrupts all sessions (all sessions log out and then log in again).
- FLOGI-based load balancing—Load balancing is based on the weighted utilization (session load) of the NP_Ports connected to an FC fabric. The session load is the sum of the FLOGI sessions on each link. FDISC sessions are not counted. When an ENode logs in to the fabric, the switch places all subsequent VN_Port FDISC sessions associated with that ENode on the same link as the ENode FLOGI session, regardless of the link load. New ENode FLOGIs are placed on the least-loaded link. Rebalancing the link load disrupts only selected sessions to minimize the impact (the switch uses an algorithm to log out only the sessions that need to be moved to other links to balance the load when those sessions log in again).

To define the proxy load-balancing algorithm for a proxy fabric on the FCoE-FC gateway, set the algorithm as **enode-based**, **simple**, or **flogi-based**:

- `[edit fc-fabrics` *`fabric-name`* `proxy]`
  `user@switch#` **set load-balance-algorithm (enode-based | simple | flogi-based)**

  For example, to configure a gateway fabric named **san_fab1** to use **enode-based** load balancing:

  `user@switch#` **set fc-fabrics san_fab1 proxy load-balance-algorithm enode-based**

Related Documentation

- Example: Configuring Automated Fibre Channel Interface Load Rebalancing on page 250
- Simulating On-Demand Fibre Channel Link Load Rebalancing (Dry Run Test) on page 280

- Understanding Load Balancing in an FCoE-FC Gateway Proxy Fabric on page 63
- Monitoring Fibre Channel Interface Load Balancing on page 383

Simulating On-Demand Fibre Channel Link Load Rebalancing (Dry Run Test)

On-demand Fibre Channel (FC) link load rebalancing is a disruptive action that causes sessions to log out of the network, then log back in to be placed on FC links (NP_Ports) in a balanced manner. The number of sessions logged out to rebalance the links depends on the load-balancing algorithm used (simple, ENode-based, or FLOGI-based) and whether or not the load is already balanced. (If the link load is already balanced, the switch does not rebalance the loads when you request on-demand load rebalancing.)

You can use the **dry-run** option to list the sessions that might be affected (logged out to be redistributed among the active FC interface links) by on-demand load rebalancing *before* you actually rebalance the link load. (Because new sessions might log in between the time you perform a dry run and the time you request on-demand load rebalancing, the affected sessions may change. Therefore, the sooner that you perform an on-demand load rebalance after you perform a dry run, the more accurate the dry run results are likely to be.)

To request a link load rebalancing dry run:

```
user@switch> request fibre-channel proxy load-rebalance dry-run fabric fabric-name
```

For example, to request a dry run on an FC fabric named *fc_fabric_100* to display a list of sessions that might be disrupted if you request an actual link load rebalance:

```
user@switch> request fibre-channel proxy load-rebalance dry-run fabric fc_fabric_100
Fabric: fc_fabric_100, Fabric-id: 100
F-Port                        FCID     Port-WWN                NP-Port
vlan.100                      0x8a013a 02:01:00:64:00:00:00:2a fc-0/0/1.0
vlan.100                      0x8a013c 02:01:00:64:00:00:00:2b fc-0/0/1.0
vlan.100                      0x8a0146 02:01:00:64:00:00:00:2e fc-0/0/1.0
vlan.100                      0x8a014c 02:01:00:64:00:00:00:2f fc-0/0/1.0
```

Related Documentation

- **request fibre-channel proxy load-rebalance on page 401**
- Defining the Proxy Load-Balancing Algorithm on page 279
- Example: Configuring Automated Fibre Channel Interface Load Rebalancing on page 250
- Understanding Load Balancing in an FCoE-FC Gateway Proxy Fabric on page 63
- Monitoring Fibre Channel Interface Load Balancing on page 383

Enabling and Disabling CoS OxID Hash Control on QFX Series Standalone Switches

The originator exchange identifier (OxID) field is one of several fields that the switch can use in its hash function computation for FCoE traffic load balancing over multiple outgoing links in an Ethernet link aggregation group (LAG) on ports that face an FCoE forwarder (FCF). You can configure whether or not the switch uses the OxID in the hash computation.

Including the OxID field in the load-balancing hash computation allows different exchanges between a pair of Fibre Channel (FC) endpoints (such as an FCoE host and an FC storage device) to take different paths across the network, thus improving the aggregate network throughput.

However, if the paths between different sets of FC endpoints have common links, congestion on one set of FC endpoints can affect the other set of endpoints. Such congestion can happen if the FCoE traffic on the two sets of endpoints uses the same priority (IEEE 802.1p code point). It is common for networks to use priority 3 (IEEE 802.1p code point 011) for FCoE traffic. However, on the QFX3500, you can assign different IEEE priorities to different lossless FCoE flows as described in *Understanding CoS IEEE 802.1p Priorities for Lossless Traffic Flows* to further separate the traffic flows.

OxID hash control is enabled by default.

- To enable OxID hash control field for FCoE traffic load balancing:

```
[edit forwarding-options hash-key]
user@switch# set family fcoe oxid enable
```

- To disable OxID hash control field for FCoE traffic load balancing:

```
[edit forwarding-options hash-key]
user@switch# set family fcoe oxid disable
```

Related Documentation

- Understanding OxID Hash Control for FCoE Traffic Load Balancing on QFX Series Standalone Switches on page 79
- Understanding OxID Hash Control for FCoE Traffic Load Balancing on QFabric Systems on page 163

- *Understanding CoS IEEE 802.1p Priorities for Lossless Traffic Flows*

Enabling and Disabling CoS OxID Hash Control on QFabric Systems

The originator exchange identifier (OxID) field is one of several fields used in the hash function computation for FCoE traffic load balancing over multiple outgoing links in an Ethernet link aggregation group (LAG) on ports that face an FCoE forwarder (FCF). The QFabric system Node device ports can be 10-Gigabit Ethernet ports or 40-Gigabit fabric ports. (The 40-Gigabit fabric ports that connect a QFabric system Node device to QFabric system Interconnect devices function as a LAG even though they are not explicitly configured as a LAG.)

The originator of an exchange between a pair of Fibre Channel (FC) endpoints (such as an FCoE host and an FC storage device) uses the OxID field as an identifier for that exchange. The originator also uses the OxID field to track the progress of the series of sequences that comprise the exchange.

OxID hash control is enabled by default.

You can enable or disable OxID hash control on the 10-Gigabit Ethernet interfaces and on the 40-Gigabit fabric interfaces of a QFabric system Node group. OxID hash control is either enabled or disabled on all of the fabric or Ethernet interfaces of a Node group. For example, you cannot disable OxID hash control on some fabric interfaces in a Node group and leave OxID hash control enabled on other fabric interfaces of the same Node group.

1. To enable or disable OxID hash control on all of the 10-Gigabit Ethernet interfaces of a specified Node group or on all Node groups:

   ```
   [edit forwarding-options hash-key]
   admin@qfabric# set family fcoe ethernet-interfaces node-group [node-group-name | all] oxid [enable | disable]
   ```

   For example, to disable OxID hash control on all of the 10-Gigabit Ethernet interfaces of a Node group named **RSNG1**:

   ```
   admin@qfabric# set family fcoe ethernet-interfaces node-group RSNG1 oxid disable
   ```

2. To enable or disable OxID hash control on all of the 40-Gigabit fabric interfaces of a specified Node group or on all Node groups:

   ```
   [edit forwarding-options hash-key]
   admin@qfabric# set family fcoe fabric-interfaces node-group [node-group-name | all] oxid [enable | disable]
   ```

   For example, to disable OxID hash control on the fabric interfaces of all Node groups:

   ```
   admin@qfabric# set family fcoe fabric-interfaces node-group all oxid disable
   ```

Related Documentation

- Enabling and Disabling CoS OxID Hash Control on QFX Series Standalone Switches on page 281
- Understanding OxID Hash Control for FCoE Traffic Load Balancing on QFabric Systems on page 163

- Understanding OxID Hash Control for FCoE Traffic Load Balancing on QFX Series Standalone Switches on page 79

Configuring FIP on an FCoE-FC Gateway

Fibre Channel over Ethernet (FCoE) Initialization Protocol (FIP) establishes and maintains Fibre Channel (FC) virtual links between pairs of FCoE devices. A virtual link emulates the physical point-to-point link that FC requires between two FC devices.

FIP is enabled by default and uses the default FIP settings on all FCoE interfaces that are part of the gateway FC fabric. You can use the default FIP parameter values, or you can configure FIP parameters globally or on a per-interface basis. Configuring FIP on an individual interface overrides the global FIP configuration.

You can configure the following parameters globally for the fabric and per interface:

- FIP keepalive message transmission interval—This interval is the time period between sending FIP keepalive messages.
- Priority—If an FCoE node (ENode) connects to more than one switch, the priority value determines the switch to which the ENode connects. The switch with the lowest priority number has the highest priority.

You can only configure the following parameters globally on an FC fabric:

- FC-MAP—The 24-bit FCoE mapped address prefix that identifies the attached FC switch in the SAN fabric. The FC-MAP value is used in the fabric provided MAC address (FPMA) created for each ENode that logs in. This value must be the same for the FC switch and the QFX Series.

  NOTE: **Changing the FC-MAP value causes all logins to drop and forces the ENodes to log in again.**

- FCoE trusted—You can globally configure all of the Ethernet ports in a specified FC fabric to be FCoE trusted. You might want to configure interfaces as FCoE trusted if the interfaces are connected to a transit switch that is performing FIP snooping. For interfaces that are directly connected to FCoE hosts, FIP snooping should be enabled, and you should not configure the fabric as FCoE trusted.

  NOTE: **Do not configure interfaces with FIP snooping enabled as FCoE trusted.**

  Configuring interfaces as FCoE trusted reduces system overhead by eliminating the need for filters. The total number of sessions the system can support is 2500 sessions. Sessions are defined as the combined number of VN_Port to VF_Port sessions and VN_Port to VN_Port sessions. (Although VN2VF and VN2VN sessions run in different FCoE VLANs, the session limit is a system limit, not a per-VLAN limit.)

NOTE: A session is a FLOGI or FDISC login to the FC SAN fabric. Session does not refer to end-to-end storage sessions. There is no limit to the number of end-to-end storage sessions.

NOTE: Changing the fabric ports from untrusted to trusted removes any existing FIP snooping filters from the ports. Changing the fabric ports from trusted to untrusted forces all of the FCoE sessions on those ports to log out so that when the ENodes and VN_Ports log in again, the switch can build the appropriate FIP snooping filters.

- Maximum number of FCoE sessions per ENode—You can globally configure the maximum number of FCoE sessions (FLOGI plus FDISC) permitted from an ENode. The maximum number of sessions per ENode is 2000 sessions. The total number of sessions (VN2VF_Port sessions and VN2VN_Port sessions combined) cannot exceed the gateway fabric's maximum limit of 2500 sessions.

To configure FIP options globally using the CLI:

1. Specify the fabric on which you want to configure FIP:

   ```
   [edit]
   user@switch# set fc-fabrics fabric-name protocols fip
   ```

2. Configure the FIP keepalive message transmission interval in milliseconds to specify the amount of time between periodic FIP discovery advertisements for the fabric interfaces (the default is 8000 ms; the range is 250 through 90000 ms):

   ```
   [edit fc-fabrics fabric-name protocols fip]
   user@switch# set fka-adv-period milliseconds
   ```

3. Configure the priority value the switch advertises to ENodes in the range from 0 through 255; the default value is 128:

   ```
   [edit fc-fabrics fabric-name protocols fip]
   user@switch# set priority priority
   ```

4. Configure the FC-MAP value to match the FC-MAP value of the attached FC switch in the FC SAN fabric; the range of possible values is 0EFC00 through 0EFCFF, and the default value is 0EFC00:

   ```
   [edit fc-fabrics fabric-name protocols fip]
   user@switch# set fc-map fc-map
   ```

5. Configure the interfaces in the FC fabric as FCoE trusted (in this example, we assume that the interfaces have not been enabled for FIP snooping):

   ```
   [edit fc-fabrics fabric-name protocols fip]
   user@switch# set fcoe-trusted
   ```

6. Configure the maximum number of FCoE sessions for each ENode in the fabric:

   ```
   [edit fc-fabrics fabric-name protocols fip]
   user@switch# set max-sessions-per-enode
   ```

For example, to configure all FCoE interfaces associated with an FC fabric called **movieco_san** with a FIP keepalive interval value of **25000** milliseconds, a priority of **70**, an FC-MAP value of **0EFC01**, as FCoE trusted, and with a maximum number of FCoE sessions per ENode of 200 sessions:

```
[edit fc-fabrics movieco_san protocols fip]
user@switch# set fka-adv-period 25000
user@switch# set priority 70
user@switch# set fc-map 0EFC01
user@switch# set fcoe-trusted
user@switch# set max-sessions-per-enode 200
```

To override the global FC fabric FIP configuration for a specific FCoE interface using the CLI:

1. Specify the fabric and interface on which you want to configure FIP:

   ```
   [edit fc-fabrics fabric-name protocols fip interface interface-name]
   ```

2. Configure the FIP keepalive message transmission interval and priority:

   ```
   [edit fc-fabrics fabric-name protocols fip interface interface-name]
   user@switch# set fka-adv-period milliseconds
   user@switch# set priority priority
   ```

Related Documentation

- Configuring an FCoE-FC Gateway Fibre Channel Fabric on page 266
- Configuring an FCoE VLAN Interface on an FCoE-FC Gateway on page 272
- Example: Setting Up Fibre Channel and FCoE VLAN Interfaces in an FCoE-FC Gateway Fabric on page 171
- Understanding FIP Parameters on an FCoE-FC Gateway on page 46

Setting the Maximum Number of FIP Login Sessions per ENode

When the switch acts as an FCoE-FC gateway, FCoE node (ENode) devices in the Ethernet network use the gateway to connect to the Fibre Channel (FC) storage area network (SAN). You can limit the maximum number of FIP login sessions permitted on each ENode. Limiting the number of login sessions can prevent login session rejections caused when the connected FC switch port configuration limits the number of FIP login sessions.

The maximum number of FIP sessions per ENode is 2000 sessions (FLOGI plus FDISC sessions). The limit you set applies to every ENode in the specified gateway fabric. Each ENode in the fabric can have up to the maximum number of sessions, but the total number of active sessions cannot exceed the session limits you apply to the fabric or the Node device.

There are also configurable FIP login session limits that you can apply to the gateway FC fabric, to the QFX3500 switch or QFabric system Node device, and to the interfaces in each FC fabric.

- To set a maximum number of FIP login sessions per ENode using the CLI:

  ```
  [edit fc-fabrics fc-fabric-name protocols fip]
  user@switch# set max-sessions-per-enode max-login-sessions
  ```

For example, to configure the ENodes on an FC fabric named **sanfab1** with a maximum FIP login session limit of **250** sessions:

```
[edit fc-fabrics sanfab1]
user@switch# set protocols fip max-sessions-per-enode 250
```

Related Documentation

- Setting the Maximum Number of FIP Login Sessions per FC Interface on page 286
- Setting the Maximum Number of FIP Login Sessions per FC Fabric on page 287
- Setting the Maximum Number of FIP Login Sessions per Node Device on page 288
- Understanding Interfaces on an FCoE-FC Gateway on page 50

Setting the Maximum Number of FIP Login Sessions per FC Interface

When the switch acts as an FCoE-FC gateway, NP_Ports are the native FC interfaces the gateway uses to connect to the FC switch. You can limit the maximum number of FIP login sessions permitted on an NP_Port interface. Limiting the number of login sessions on an interface can prevent login session rejections caused when the connected FC switch port configuration limits the number of FIP login sessions.

TIP: **A good practice is to configure a maximum number of login sessions on each NP_Port that is less than or equal to the maximum number of login sessions permitted on the connected FC switch port.**

The maximum number of FIP sessions is 2500 sessions. (This is the combined total of all VN2VF_Port and VN2VN_Port sessions on the system.)

There are also configurable FIP login session limits that you can apply to the gateway FC fabric, to the QFX3500 switch or QFabric system Node device, and to the ENodes in each FC fabric. To prevent unexpected FIP login rejections, the sum of the maximum FIP login sessions on all of the NP_Port interfaces that belong to an FC fabric should not exceed the maximum number of sessions the FC fabric supports or the device supports.

- To set a maximum number of FIP login sessions on an NP_Port using the CLI:

  ```
  [edit fc-fabrics fc-fabric-name interface interface-name]
  user@switch# set max-login-sessions max-login-sessions
  ```

  For example, to configure NP_Port interface **fc-0/0/5** with a maximum FIP login session limit of **500** sessions on an FC fabric named **sanfab1**:

  ```
  [edit fc-fabrics sanfab1]
  user@switch# set interface fc-0/0/5 max-login-sessions 500
  ```

Related Documentation

- Setting the Maximum Number of FIP Login Sessions per ENode on page 285
- Setting the Maximum Number of FIP Login Sessions per FC Fabric on page 287
- Setting the Maximum Number of FIP Login Sessions per Node Device on page 288
- Understanding Interfaces on an FCoE-FC Gateway on page 50

Setting the Maximum Number of FIP Login Sessions per FC Fabric

When the QFX Series acts an FCoE-FC gateway, you configure at least one local FC fabric on the gateway. A gateway FC fabric creates associations that connect FCoE devices on an Ethernet network to an FC switch on a Fibre Channel network. Each FC fabric on a gateway includes native FC interfaces (NP_Ports) that connect the gateway to the FC switch. When FCoE devices want to log in to the FC switch, the gateway sends the FIP login requests to the FC switch on the NP_Port links.

You can limit the maximum number of FIP login sessions permitted on a gateway FC fabric. If a QFX3500 switch or QFabric system Node device has more than one FC fabric, limiting the number of login sessions on an FC fabric can prevent one FC fabric from using all of the login sessions available on the device.

The maximum number of FIP sessions is 2500 sessions. (This is the combined total of all VN2VF_Port and VN2VN_Port sessions on the system.)

There are also configurable FIP login session limits that you can apply to the FC fabric NP_Port interfaces, to the QFX3500 switch or QFabric system Node device, and to the ENodes in each FC fabric. To prevent unexpected FIP login rejections:

- The sum of the maximum FIP login sessions on all of the NP_Port interfaces that belong to an FC fabric should not exceed the maximum number of sessions the FC fabric supports or the device supports.
- The sum of the maximum FIP login sessions on all of the FC fabrics on a device should not exceed the maximum number of sessions per device.

- To set a maximum number of FIP login sessions on an FC fabric using the CLI:

  ```
  [edit fc-fabrics fc-fabric-name]
  user@switch# set max-login-sessions max-login-sessions
  ```

  For example, to configure an FC fabric named **sanfab1** with a maximum FIP login session limit of **2000** sessions:

  ```
  [edit fc-fabrics sanfab1]
  user@switch# set fc-fabrics sanfab1 max-login-sessions 2000
  ```

Related Documentation

- Setting the Maximum Number of FIP Login Sessions per ENode on page 285
- Setting the Maximum Number of FIP Login Sessions per FC Interface on page 286
- Setting the Maximum Number of FIP Login Sessions per Node Device on page 288
- Understanding Interfaces on an FCoE-FC Gateway on page 50

Setting the Maximum Number of FIP Login Sessions per Node Device

When a QFX3500 switch or QFabric system Node device acts an FCoE-FC gateway, it connects FCoE devices on an Ethernet network to an FC switch in a Fibre Channel network. You can limit the maximum number of FIP login sessions for the FCoE devices on each Node device.

For QFX3500 switches, the maximum limit means that the sum of the FIP login sessions on all of the local FC fabrics on that QFX3500 switch cannot exceed the device maximum.

For the QFabric system, the limit applies to each Node device in the QFabric system. For example, if you configure a maximum FIP login session value of 2000 sessions, each Node device in the QFabric system can have a total of up to 2000 FIP login sessions running on its FC fabrics.

The maximum number of FIP sessions a device can support is 2500 sessions. (This is the combined total of all VN2VF_Port and VN2VN_Port sessions on the system.)

There are also configurable FIP login session limits that you can apply to the FC fabrics on the devices, to the NP_Port interfaces in each FC fabric, and to the ENodes in each FC fabric. To prevent unexpected FIP login rejections:

- The sum of the maximum FIP login sessions for all of the FC fabrics on a device should not exceed the maximum number of sessions per device.
- The sum of the maximum FIP login sessions on all of the NP_Port interfaces that belong to an FC fabric should not exceed the maximum number of sessions the FC fabric supports or the device supports.

- To set a maximum number of FIP login sessions for Node devices using the CLI:

  ```
  [edit fc-options]
  user@switch# set max-login-sessions-per-node max-login-sessions-per-node
  ```

  For example, to configure a maximum FIP login limit of **2000** sessions on a QFX3500 switch or on all Node devices in a QFabric system:

  ```
  [edit fc-options]
  user@switch# set max-login-sessions-per-node 2000
  ```

Related Documentation

- Setting the Maximum Number of FIP Login Sessions per ENode on page 285
- Setting the Maximum Number of FIP Login Sessions per FC Interface on page 286
- Setting the Maximum Number of FIP Login Sessions per FC Fabric on page 287
- Understanding Interfaces on an FCoE-FC Gateway on page 50

Configuring VLANs for FCoE Traffic on an FCoE Transit Switch

When you configure a QFX Series as a Fibre Channel over Ethernet (FCoE) transit switch, you must configure a VLAN that transports only FCoE traffic. FCoE traffic requires a dedicated VLAN and cannot share a VLAN with any other type of traffic. Because FCoE traffic is tagged traffic, the port (or interface) mode cannot be access mode, it must be either tagged-access port-mode (for switches that run the original CLI, such as QFX3500 and QFX3600 standalone switches and QFabric system Node devices) or trunk interface-mode (for switches that run the Enhanced Layer 2 Software (ELS) CLI, such as the QFX5100 switch).

However, each interface that belongs to an FCoE VLAN must not only transport the tagged FCoE traffic, it must also transport the untagged FCoE Initialization Protocol (FIP) traffic. FIP communicates with the storage area network (SAN) Fibre Channel (FC) switch to set up the FCoE session for the FCoE client.

To transport untagged traffic on a tagged-access or trunk mode interface, the interface must have a native VLAN configured on it. Therefore, each interface that belongs to an FCoE VLAN must also have a native VLAN on it.

There are slight differences in the way you configure a native VLAN on an interface, depending on whether the switch uses the ELS CLI or the original CLI. This topic describes both methods.

NOTE: FCoE VLANs (any VLAN that carries FCoE traffic) support only Spanning Tree Protocol (STP) and link aggregation group (LAG) Layer 2 features.

FCoE traffic cannot use a standard LAG because traffic might be hashed to different physical LAG links on different transmissions. This breaks the (virtual) point-to-point link that Fibre Channel traffic requires. If you configure a standard LAG interface for FCoE traffic, FCoE traffic might be rejected by the FC SAN.

QFabric systems support a special LAG called an FCoE LAG, which enables you to transport FCoE traffic and regular Ethernet traffic (traffic that is not FCoE traffic) across the same link aggregation bundle. Standard LAGs use a hashing algorithm to determine which physical link in the LAG is used for a transmission, so communication between two devices might use different physical links in the LAG for different transmissions. An FCoE LAG ensures that FCoE traffic uses the same physical link in the LAG for requests and replies in order to preserve the virtual point-to-point link between the FCoE device converged network adapter (CNA) and the FC SAN switch across the QFabric system Node device. An FCoE LAG does not provide load balancing or link redundancy for FCoE traffic. However, regular Ethernet traffic uses the standard hashing algorithm and receives the usual LAG benefits of load balancing and link redundancy in an FCoE LAG.

NOTE: To configure an FCoE VLAN on a QFX3500 switch or a QFabric system Node device that you are using as an FCoE-FC gateway, you must also configure an FCoE VLAN interface as described in "Configuring an FCoE VLAN Interface on an FCoE-FC Gateway" on page 272. (QFX3600, QFX5100, and Virtual Chassis switches do not support FCoE-FC gateway configuration.)

FCoE VLAN configuration includes:

- Configuring a VLAN to use as a dedicated FCoE VLAN
- Configuring the interface members of the FCoE VLAN.
- Configuring a native VLAN for FIP traffic.

This topic includes two configuration procedures, one for switches that run the original CLI, and one for switches that run the ELS CLI.

Original CLI Configuration

To configure an FCoE VLAN on a non-ELS switch (QFX3500, QFX3600, QFabric system Node devices):

1. Configure a dedicated FCoE VLAN:

   `[edit vlans]`
   `user@switch#` **set *vlan-name* vlan-id *vlan-id***

   For example, to configure a VLAN named **fcoe_vlan** with a VLAN ID of **100** as the FCoE VLAN:

   `[edit vlans]`
   `user@switch#` **set fcoe_vlan vlan-id 100**

2. Configure the FCoE VLAN on the interface (use **ethernet-switching** as the family and **tagged-access** as the port mode):

   `[edit interfaces]`
   `user@switch#` **set *interface-name* unit *unit* family *family* port-mode *mode* vlan members *vlan-name***

   For example, to configure the interface **xe-0/0/10** as a member of the FCoE VLAN **fcoe_vlan**:

   `[edit interfaces]`
   `user@switch#` **set xe-0/0/10 unit 0 family ethernet-switching port-mode tagged-access vlan members fcoe_vlan**

3. Configure the Ethernet interface membership in the FCoE VLAN:

   `[edit vlans]`
   `user@switch#` **set *vlan-name* interface *interface-name***

   For example, to assign the interface **xe-0/0/10.0** to the FCoE VLAN named **fcoe_vlan**:

   `[edit vlans]`
   `user@switch#` **set fcoe_vlan interface xe-0/0/10.0**

4. Configure a native VLAN for the untagged FIP traffic:

   `[edit vlans]`
   `user@switch#` **set native vlan-id *vlan-id***

   For example, to configure the native VLAN with a VLAN ID of **1**:

   `[edit vlans]`
   `user@switch#` **set native vlan-id 1**

5. Assign member interfaces to the native VLAN:

   `[edit interfaces]`
   `user@switch#` **set *interface-name* unit *unit* family *family* native-vlan-id *vlan-id***

   For example, to configure the interface **xe-0/0/10** as a member of the native VLAN with the native VLAN ID **1**:

   `[edit interfaces]`
   `user@switch#` **set xe-0/0/10 unit 0 family ethernet-switching native-vlan-id 1**

ELS CLI Configuration

To configure an FCoE VLAN on a QFX5100 switch running ELS:

1. Configure a dedicated FCoE VLAN:

   [edit vlans]
   user@switch# **set *vlan-name* vlan-id *vlan-id***

   For example, to configure a VLAN named **fcoe_vlan** with a VLAN ID of **100** as the FCoE VLAN:

   [edit vlans]
   user@switch# **set fcoe_vlan vlan-id 100**

2. Configure the FCoE VLAN on the interface (use **ethernet-switching** as the family and **trunk** as the interface mode):

   [edit interfaces]
   user@switch# **set *interface-name* unit *unit* family *family* interface-mode *mode* vlan members *vlan-name***

   For example, to configure the interface **xe-0/0/10** as a member of the FCoE VLAN **fcoe_vlan**:

   [edit interfaces]
   user@switch# **set xe-0/0/10 unit 0 family ethernet-switching interface-mode trunk vlan members fcoe_vlan**

3. Configure the Ethernet interface membership in the FCoE VLAN:

   [edit vlans]
   user@switch# **set *vlan-name* interface *interface-name***

   For example, to assign the interface **xe-0/0/10.0** to the FCoE VLAN named **fcoe_vlan**:

   [edit vlans]
   user@switch# **set fcoe_vlan interface xe-0/0/10.0**

4. Configure a native VLAN on the physical Ethernet interface for the untagged FIP traffic:

   [edit interfaces]
   user@switch# **set *interface-name* native-vlan-id *vlan-id***

   For example, to configure the native VLAN on interface **xe-0/0/10** with a VLAN ID of **1**:

   [edit interfaces]
   user@switch# **set xe-0/0/10 native-vlan-id 1**

5. Configure the Ethernet interface as a member of the native VLAN:

   [edit interfaces]
   user@switch# **set *interface-name* unit *unit* family *family* vlan members *native-vlan-id***

   NOTE: The *native-vlan-id* number must be the same as the native VLAN ID number that you configured on the physical Ethernet interface (see step 4).

   For example, to configure the interface **xe-0/0/10** as a member of the native VLAN with the native VLAN ID **1**:

   [edit interfaces]

```
user@switch# set xe-0/0/10 unit 0 family ethernet-switching vlan members 1
```

Related Documentation

- Understanding FCoE on page 19
- Understanding FCoE Transit Switch Functionality on page 25
- Example: Configuring CoS PFC for FCoE Traffic on page 186
- Configuring VN2VF_Port FIP Snooping and FCoE Trusted Interfaces on an FCoE Transit Switch on page 294
- Enabling VN2VN_Port FIP Snooping and Configuring the Beacon Period on an FCoE Transit Switch on page 300
- Disabling Enhanced FIP Snooping Scaling on page 298
- Configuring an FCoE LAG on page 302
- Configuring an FCoE VLAN Interface on an FCoE-FC Gateway on page 272

Configuring VN2VF_Port FIP Snooping and FCoE Trusted Interfaces on an FCoE Transit Switch

VN_Port to VF_Port (VN2VF_Port) Fibre Channel over Ethernet (FCoE) Initialization Protocol (FIP) snooping uses information gathered during FIP discovery and login to create firewall filters that provide security against unauthorized access to the FC switch or FCoE forwarder (FCF) through the QFX Series or EX4500 when the switch is acting as an FCoE transit switch. The firewall filters allow only FCoE devices that successfully log in to the FC fabric to access the FCF through the transit switch. VN2VF_Port FIP snooping provides security for the point-to-point virtual links that connect host FCoE Nodes (ENodes) and FCFs in the FCoE VLAN by denying access to any device that does not successfully log in to the FCF.

VN2VF_Port FIP snooping is disabled by default. You enable VN2VF_Port FIP snooping on a per-VLAN basis for VLANs that carry FCoE traffic. Ensure that a VLAN that carries FCoE traffic carries only FCoE traffic, because enabling VN2VF_Port FIP snooping denies access for all other Ethernet traffic.

NOTE: All of the transit switch ports are untrusted by default. If an ENode on an FCoE device logs in to an FCF before you enable VN2VF_Port FIP snooping on the VLAN and you then enable VN2VF_Port FIP snooping, the transit switch denies traffic from the ENode because the transit switch has not snooped (learned) the ENode state. The following process automatically logs the ENode back in to the FCF to reestablish the connection:

1. VN2VF_Port FIP snooping is enabled on an FCoE VLAN on the switch.
2. The switch denies existing connections between servers and the FCF on the FCoE VLAN by filtering the FCoE traffic and FIP traffic, so no keepalive messages from the ENodes reach the FCF.
3. The FCF port timer for each ENode and for each VN_Port on each ENode expires.
4. The FCF sends each ENode whose port timer has expired a Clear Virtual LInks (CVL) message.
5. The CVL message causes the ENode to log in again.

Because the FCF is a trusted source, you configure interfaces that connect to the FCF as FCoE trusted interfaces. FCoE trusted interfaces do not filter traffic (FIP snooping filtering should occur only at the FCoE access edge), but VN2VF_Port FIP snooping continues to run on trusted interfaces so that the switch learns the FCF state.

NOTE: Do not configure ENode-facing interfaces both with FIP snooping enabled and as trusted interfaces. FCoE VLANs with interfaces that are directly connected to FCoE hosts should be configured with FIP snooping enabled and the interfaces should *not* be trusted interfaces. Ethernet interfaces that are connected to an FCF should be configured as trusted interfaces and should not have FIP snooping enabled. Interfaces that are connected to a transit switch that is performing FIP snooping can be configured as trusted interfaces if the FCoE VLAN is not enabled for FIP snooping.

Optionally, you can specify an FC-MAP value for each FCoE VLAN. On a given FCoE VLAN, the switch learns only FCFs that have a matching FC-MAP value. The default FC-MAP value is 0EFC00h for all FC devices. (Enter hexadecimal values for FC-MAP preceded by the hexadecimal indicator “0x”—for example, 0x0EFC00.) If you change the FC-MAP value of an FCF, change the FC-MAP value for the FCoE VLAN it belongs to on the switch and on the servers you want to communicate with the FCF. An FCoE VLAN can have one and only one FC-MAP value.

NOTE: The default enhanced FIP snooping scaling supports 2,500 sessions. On QFabric systems, starting with Junos OS Release 13.2X52, you can disable enhanced FIP snooping scaling on a per-VLAN basis if you want to do so, but only 376 sessions are supported if you disable enhanced FIP snooping scaling.

There are differences in the way you configure FIP snooping and FCoE trusted interfaces on a switch that depend on whether the switch uses the original QFX/QFabric CLI (for example, a standalone QFX3500 or QFX3600 switch or a QFabric system Node device) or the Enhanced Layer 2 Software (ELS) CLI (for example, a standalone QFX5100 switch). This topic includes two configuration procedures, one for switches that run the original CLI, and one for switches that run the ELS CLI.

Original CLI Configuration

To enable VN2VF_Port FIP snooping:

- To enable VN2VF_Port FIP snooping on a single VLAN and specify the optional FC-MAP value:

  ```
  [edit ethernet-switching-options secure-access-port]
  user@switch# set vlan vlan-name examine-fip fc-map fc-map-value
  ```

  For example, to enable VN2VF_Port FIP snooping on a VLAN named **san1_vlan** and change the FC-MAP value to **0x0EFC03**:

  ```
  [edit ethernet-switching-options secure-access-port]
  user@switch# set vlan san1_vlan examine-fip fc-map 0x0EFC03
  ```

  NOTE: Changing the FC-MAP value causes all logins to drop and forces ENodes to log in again.

- To enable VN2VF_Port FIP snooping on all VLANs and use the default FC-MAP value:

  ```
  [edit ethernet-switching-options secure-access-port]
  user@switch# set vlan all examine-fip
  ```

- To configure an interface as an FCoE trusted interface:

  ```
  [edit ethernet-switching-options secure-access-port]
  user@switch# set interface interface-name fcoe-trusted
  ```

  For example, to configure interface **xe-0/0/30** as an FCoE trusted interface:

  ```
  [edit ethernet-switching-options secure-access-port]
  user@switch# set interface xe-0/0/30 fcoe-trusted
  ```

ELS CLI Configuration

To enable VN2VF_Port FIP snooping:

- To enable VN2VF_Port FIP snooping on a VLAN and specify the optional FC-MAP value:

  ```
  [edit]
  user@switch# set vlans vlan-name forwarding-options fip-security fc-map fc-map-value
  examine-vn2vf
  ```

For example, to enable VN2VF_Port FIP snooping on a VLAN named **san1_vlan** and change the FC-MAP value to **0x0EFC03**:

```
[edit]
user@switch# set vlans san1_vlan forwarding-options fip-security fc-map 0x0EFC03
examine-vn2vf
```

NOTE: Changing the FC-MAP value causes all logins to drop and forces ENodes to log in again.

- To configure an interface as an FCoE trusted interface:

```
[edit]
user@switch# set vlans vlan-name forwarding-options fip-security interface interface-name
fcoe-trusted
```

For example, to configure interface **xe-0/0/30** on VLAN named **san1_vlan** as an FCoE trusted interface:

```
[edit]
user@switch# set vlans san1_vlan forwarding-options fip-security interface xe-0/0/30
fcoe-trusted
```

Related Documentation

- *Example: Configuring an FCoE Transit Switch*
- Configuring an FCoE VLAN Interface on an FCoE-FC Gateway on page 272
- Configuring VLANs for FCoE Traffic on an FCoE Transit Switch on page 289
- Configuring an FCoE LAG on page 302
- Disabling Enhanced FIP Snooping Scaling on page 298
- *Understanding FIP Snooping*
- Understanding VN_Port to VF_Port FIP Snooping on an FCoE Transit Switch on page 80
- Understanding FCoE LAGs on page 158

Disabling Enhanced FIP Snooping Scaling

Enhanced FIP snooping scaling (introduced in Junos OS Release 12.3) scales up to 2,500 sessions and is the default FIP snooping scaling mode. On QFabric systems only, you can disable enhanced FIP snooping scaling. Disabling FIP snooping scaling reduces the number of supported FIP snooping sessions to 376 sessions.

On QFabric system Node device in FCoE-FC gateway mode, you disable FIP snooping scaling globally, on all of the Fibre Channel (FC) fabrics (fc-fabrics) on the Node device. Either all FC fabrics on a Node device use enhanced FIP snooping scaling (2,500 sessions), or all FC fabrics on a Node device disable FIP snooping scaling (376 sessions). On an FCoE-FC gateway, you must disable FIP snooping scaling if the member interfaces of an FCoE VLAN are configured as members of an FCoE LAG *and* if the FC fabric is an FCoE untrusted fabric. If the FC fabric is an FCoE trusted fabric, then you do not need to disable FIP snooping scaling on the gateway.

On a QFabric system Node device in FCoE transit switch mode, you do not need to disable FIP snooping scaling. However, you can disable FIP snooping scaling on a per-VLAN basis if you want to do so.

Disabling FIP snooping scaling is done differently on an FCoE-FC gateway than on an FCoE transit switch. This document provides the method for each mode.

Disabling Enhanced FIP Snooping Scaling on an FCoE-FC Gateway

If you configure an FCoE LAG on an FCoE untrusted gateway fabric, you must disable FIP snooping scaling. Disabling FIP snooping scaling is global and affects all FC fabrics on the gateway.

1. `admin@qfabric#` **set fc-options no-fip-snooping-scaling**

Disabling Enhanced FIP Snooping Scaling on an FCoE Transit Switch

If you choose to disable FIP snooping scaling on an FCoE transit switch, you can disable it on individual FCoE VLANs:

1. `admin@qfabric#` **set ethernet-switching-options secure-access-port vlan *fcoe-vlan-name* examine-fip no-fip-snooping-scaling**

   For example, if the FCoE VLAN name is **fcoe-vlan-blue**:

   `admin@qfabric#` **set ethernet-switching-options secure-access-port vlan fcoe-vlan-blue examine-fip no-fip-snooping-scaling**

Related Documentation

- Configuring an FCoE LAG on page 302
- Example: Configuring an FCoE LAG on a Redundant Server Node Group on page 253

- Understanding VN_Port to VF_Port FIP Snooping on an FCoE Transit Switch on page 80
- Understanding FCoE LAGs on page 158

Disabling VN2VF_Port FIP Snooping on an FCoE-FC Gateway Switch Interface

When the switch acts an FCoE-FC gateway, the FCoE-network-facing Ethernet interfaces in the FCoE VLAN are automatically enabled for VN_Port to VF_Port (VN2VF_Port) FIP snooping. You can disable VN2VF_Port FIP snooping on an individual Ethernet interface or you can disable VN2VF_Port FIP snooping globally for all Ethernet interfaces in a gateway Fibre Channel (FC) fabric.

Disable VN2VF_Port FIP snooping on an Ethernet interface by configuring it as an FCoE trusted interface. Disable VN2VF_Port FIP snooping on all Ethernet interfaces in an FC fabric by configuring the FC fabric as FCoE trusted.

Do not disable VN2VF_Port FIP snooping on an interface unless you are certain that the interface is connected to a trusted device. Do not disable VN2VF_Port FIP snooping on an FC fabric unless all of the FCoE-network-facing interfaces in the fabric are either connected to a transit switch that is performing VN2VF_Port FIP snooping on the FCoE devices as they log in to the FC network or all of the interfaces are connected to trusted devices.

VN2VF_Port FIP snooping installs firewall filters that block FIP and FCoE frames from sources that have not logged in to the switch and prevents unauthorized access to the network. Disabling VN2VF_Port FIP snooping disables these firewall filters and permits access to all FIP and FCoE frames transported on that interface.

- To disable VN2VF_Port FIP snooping on an FCoE-device-facing Ethernet interface in an FCoE VLAN, configure that interface as a trusted interface:

  ```
  [edit ethernet-switching-options secure-access-port]
  user@switch# set interface interface-name fcoe-trusted
  ```

  For example, to configure interface **xe-0/0/7** as a trusted FC interface:

  ```
  [edit ethernet-switching-options secure-access-port]
  user@switch# set interface xe-0/0/7 fcoe-trusted
  ```
- To disable VN2VF_Port FIP snooping on all FCoE-device-facing interfaces in a gateway FC fabric, configure that fabric as a trusted fabric:

  ```
  [edit]
  user@switch# set fc-fabrics fabric-name protocols fip fcoe-trusted
  ```

  For example, to configure an FC fabric named *santastic* as an FCoE trusted fabric:

  ```
  [edit]
  user@switch# set fc-fabrics santastic protocols fip fcoe-trusted
  ```

Related Documentation

- Configuring VN2VF_Port FIP Snooping and FCoE Trusted Interfaces on an FCoE Transit Switch on page 294
- Understanding VN_Port to VF_Port FIP Snooping on an FCoE Transit Switch on page 80
- Understanding Interfaces on an FCoE-FC Gateway on page 50

- Understanding an FCoE-FC Gateway on page 29

Enabling VN2VN_Port FIP Snooping and Configuring the Beacon Period on an FCoE Transit Switch

VN_Port to VN_Port (VN2VN_Port) FIP snooping on an FCoE transit switch provides security to help prevent unauthorized access and data transmission on a bridge that connects ENodes in the Ethernet network. VN2VN_Port FIP snooping provides security for virtual links by creating filters based on information gathered (snooped) about FCoE devices during FIP transactions.

VN2VN_Port FIP snooping is conceptually similar to VN2VF_Port FIP snooping between VN_Ports and VF_Ports, but VN2VN_Port FIP snooping does not require traffic between VN_Ports to traverse the Fibre Channel (FC) switch or FCoE forwarder (FCF). Instead, a VN_Port communicates transparently through the transit switch on a virtual link that emulates a direct connection to the VN_Port at the other end of the virtual link.

VN2VN_Port FIP snooping is disabled by default. You enable VN2VN_Port FIP snooping on a per-VLAN basis on VLANs that carry VN2VN_Port FCoE traffic. Ensure that the VLAN carries only FCoE traffic between VN_Ports, because enabling VN2VN_Port FIP snooping denies access for all other traffic, including VN2VF_Port FIP snooping traffic.

All ENodes that you want to communicate using VN2VN_Port FIP snooping must use an FCoE VLAN dedicated to VN2VN_Port traffic. You cannot mix VN2VN_Port FIP snooping traffic with VN2VF_Port FIP snooping traffic in the same FCoE VLAN.

NOTE: **An FCoE VLAN can support either VN2VF_Port FIP snooping or VN2VN_Port FIP snooping, but not both. Configure separate FCoE VLANs for VN2VF_Port FIP snooping traffic and for VN2VN_Port FIP snooping traffic. On FCoE VLANs that are configured as VN2VN_Port FIP snooping VLANs, VN2VF_Port traffic is dropped.**

The *beacon period* is conceptually similar to the FIP keepalive period (timer) for VN2VF_Port FIP snooping virtual link maintenance. The beacon period performs virtual link maintenance for VN2VN_Port FIP snooping. It is the time interval between messages that verify the connection is still valid and the device at the other end of the virtual link is still reachable. You set the beacon period value for each FCoE VLAN that you configure to do VN2VN_Port FIP snooping.

NOTE: **In addition to enabling VN2VN_Port FIP snooping and configuring the beacon period, you must also configure a dedicated FCoE VLAN for the VN2VN_Port traffic, and set the FCoE transit switch ports in the proper port mode and trusted or untrusted state (interfaces are untrusted by default). See the VN2VN_Port FIP snooping configuration example topics for complete configurations of several common network topologies.**

There are differences in the way you configure a native VLAN on an interface that depend on whether the switch uses the original QFX/QFabric CLI (for example, a standalone QFX3500 or QFX3600 switch or a QFabric system Node device) or the Enhanced Layer 2 Software (ELS) CLI (for example, a standalone QFX5100 switch). This topic includes two configuration procedures, one for switches that run the original CLI, and one for switches that run the ELS CLI.

Original CLI Configuration

To enable VN2VN_Port FIP snooping and set the beacon period on an FCoE VLAN that is dedicated to VN2VN_Port traffic:

- `[edit ethernet-switching-options secure-access-port]`
  `user@switch#` **set vlan *vlan-name* examine-fip examine-vn2vn beacon-period *milliseconds***

  For example, to enable VN2VN_Port FIP snooping on a VLAN named **vlan200** and set the beacon period to **90000** milliseconds:

  `[edit ethernet-switching-options secure-access-port]`
  `user@switch#` **set vlan vlan200 examine-fip examine-vn2vn beacon-period 90000**

ELS CLI Configuration

To enable VN2VN_Port FIP snooping and set the beacon period on an FCoE VLAN that is dedicated to VN2VN_Port traffic:

- `[edit]`
  `user@switch#` **set vlans *vlan-name* forwarding-options fip-security examine-vn2vn beacon-period *milliseconds***

  For example, to enable VN2VN_Port FIP snooping on a VLAN named **vlan200** and set the beacon period to **90000** milliseconds:

  `[edit]`
  `user@switch#` **set vlans vlan200 forwarding-options fip-security examine-vn2vn beacon-period 90000**

Related Documentation

- Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Directly Connected to the Same FCoE Transit Switch) on page 228
- Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Directly Connected to Different FCoE Transit Switches) on page 233
- Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Indirectly Connected Through an Aggregation Layer FCoE Transit Switch) on page 241
- Configuring VN2VF_Port FIP Snooping and FCoE Trusted Interfaces on an FCoE Transit Switch on page 294
- Understanding VN_Port to VN_Port FIP Snooping on an FCoE Transit Switch on page 87

Configuring an FCoE LAG

A Fibre Channel over Ethernet (FCoE) link aggregation group (LAG) is a special LAG that enables you to transport FCoE traffic and regular Ethernet traffic across the same link aggregation bundle. Standard LAGs use a hashing algorithm to determine which physical link in the LAG is used for a transmission, so a series of communications between two devices might use different physical links in the LAG for different transmissions.

However, FCoE traffic requires a point-to-point link (or a virtual point-to-point link) between the FCoE device and the Fibre Channel (FC) storage area network (SAN) switch. This requirement means that communication between an FCoE device and a QFabric system Node device must use the same physical link in a LAG to maintain the virtual point-to-point connection.

An FCoE LAG solves the problem by ensuring that the same LAG link is used for communication between an FC SAN switch and a given FCoE device across a QFabric system Node device, preserving point-to-point link emulation. At the same time, regular Ethernet traffic (traffic that is not FCoE traffic) on the LAG is distributed across member interfaces in the same way as on a standard LAG. FCoE traffic is treated properly in terms of maintaining a virtual point-to-point link with the FC SAN, and regular Ethernet traffic enjoys the usual LAG benefits of load balancing and link redundancy.

NOTE: **Configuring a LAG as an FCoE LAG does not provide link redundancy for FCoE traffic, and does not load balance FCoE traffic.**

On FCoE-FC gateway untrusted Fibre Channel fabrics (fc-fabrics), if you configure an FCoE LAG, you must also disable enhanced FIP snooping scaling (scaling up to 2,500 sessions), which reduces the number of supported FIP snooping sessions to 376 sessions. On an FCoE-FC gateway, disabling enhanced FIP snooping scaling is global to the Node device. Trusted fc-fabrics on an FCoE-FC gateway support enhanced FIP snooping scaling.

This example shows how to configure an FCoE LAG with enhanced FIP snooping scaling enabled and also with enhanced FIP snooping scaling disabled.

The steps required to create the FCoE LAG are:

- Configuring an FCoE LAG interface.
- Assigning the Ethernet interfaces connected to the FCoE device to the FCoE LAG.
- Configuring FIP snooping.

In addition to configuring the FCoE LAG and FIP snooping scaling, you also need to configure:

- Configure a dedicated FCoE VLAN for the FCoE traffic.
- Configure a native VLAN for the untagged FIP traffic.
- Enable FIP snooping on the FCoE VLAN.

- Configure the FCoE LAG interface membership in the FCoE VLAN and the native VLAN.
- For FCoE-FC gateway switches, configure a Layer 3 FCoE VLAN interface, and add the FCoE VLAN interface to the Fibre Channel fabric.
- For FCoE-FC gateway switches, configure the fc-fabric as an FCoE trusted fabric if you are using enhanced FIP snooping scaling (and if the FCoE traffic is trusted).

"Example: Configuring an FCoE LAG on a Redundant Server Node Group" on page 253 includes an example of this configuration.

Configuring an FCoE LAG When Enhanced FIP Snooping Scaling is Enabled

This example shows how to configure an FCoE LAG when you can use enhanced FIP snooping scaling, such as when the FCoE-FC gateway fabrics are trusted, or on an FCoE transit switch.

1. Specify the number of LAGs (Ethernet devices) the QFabric system Node group will support:

   `admin@qfabric#` **set chassis node-group *node-group-name* aggregated-devices ethernet-device-count *device-count***

   For example, to configure the Node group **RSNG1** to allow up to ten LAGs:

   `admin@qfabric#` **set chassis node-group RSNG1 aggregated-devices ethernet-device-count 10**

2. Configure the LAG interface on the RSNG:

   `admin@qfabric#` **set interfaces *lag-interface-name* family ethernet-switching port-mode trunk**

   For example, to configure a LAG interface named **ae3** on Node group **RSNG1**:

   `admin@qfabric#` **set interfaces RSNG1:ae3 family ethernet-switching port-mode trunk**

3. Configure the LAG interface as an FCoE LAG:

   `admin@qfabric#` **set interfaces *lag-interface-name* aggregated-ether-options fcoe-lag**

   For example, to configure LAG **ae3** on a Node group named **RSNG1** as an FCoE LAG:

   `admin@qfabric#` **set interfaces RSNG1:ae3 aggregated-ether-options fcoe-lag**

4. Enable LACP on the FCoE LAG:

   `admin@qfabric#` **set interfaces *fcoe-lag-interface-name* aggregated-ether-options lacp active**

   For example, to configure LACP on FCoE LAG **RSNG1:ae3**:

   `admin@qfabric#` **set interfaces RSNG1:ae3 aggregated-ether-options lacp active**

5. Assign the Ethernet interfaces connected to the FCoE device converged network adapter (CNA) to the FCoE LAG:

   `admin@qfabric#` **set interfaces *interface-name* ether-options 802.3ad *fcoe-lag-name***

   For example, to assign interfaces **xe-0/0/20** and **xe-0/0/21** on Node device **row1-rack1** (which is part of the Node group **RSNG1**) to the FCoE LAG **ae3** (on Node group **RSNG1**):

admin@qfabric# **set interfaces row1-rack1:xe-0/0/20 ether-options 802.3ad RSNG1:ae3**
admin@qfabric# **set interfaces row1-rack1:xe-0/0/21 ether-options 802.3ad RSNG1:ae3**

NOTE: On QFabric system Node groups that have two or more member nodes, you can assign interfaces from any Node in the Node group to the FCoE LAG. Adding to the example, if Node device **row2-rack1** is part of Node group **RSNG1**, then you can add interfaces from **row2-rack1** to the FCoE LAG. For example, **set interfaces row2-rack1:xe-0/0/20 ether-options 802.3ad RSNG1:ae3** adds an interface on a second Node device to the FCoE LAG.

6. Enable FIP snooping on the FCoE VLAN:

   admin@qfabric# **set ethernet-switching-options secure-access-port vlan *fcoe-vlan-name* examine-fip**

   For example, to enable FIP snooping on an FCoE VLAN named **fcoe-vlan-blue**:

   admin@qfabric# **set ethernet-switching-options secure-access-port vlan fcoe-vlan-blue examine-fip**

7. On an FCoE-FC gateway only, enable FCoE trusted mode on the fc-fabric:

   admin@qfabric# **set fc-fabrics *fc-fabric-name* protocols fip fcoe-trusted**

   For example, to configure an fc-fabric named **sanfab1** as an FCoE trusted fabric:

   admin@qfabric# **set fc-fabrics sanfab1 protocols fip fcoe-trusted**

Configuring an FCoE LAG When Enhanced FIP Snooping Scaling Must be Disabled

This example shows how to configure an FCoE LAG when you need to disable enhanced FIP snooping scaling, for example, when an FCoE-FC gateway fabric is untrusted.

Follow steps 1-6 of the preceding example to configure the FCoE LAG and enable FIP snooping on the FCoE VLAN. Next, disable enhanced FIP snooping scaling:

1. On an FCoE-FC gateway switch, disable FIP snooping scaling on all FCoE LAGs in the Fibre Channel fabric options configuration:

   admin@qfabric# **set fc-options no-fip-snooping-scaling**

   This global statement disables FIP snooping scaling on all FCoE LAGs associated with all FC fabrics on the switch.

Related Documentation

- Configuring VLANs for FCoE Traffic on an FCoE Transit Switch on page 289
- Configuring VN2VF_Port FIP Snooping and FCoE Trusted Interfaces on an FCoE Transit Switch on page 294
- Example: Configuring an FCoE LAG on a Redundant Server Node Group on page 253
- Understanding FCoE LAGs on page 158

Configuring the DCBX Mode

You can configure the DCBX mode that an interface uses to communicate with the connected peer. QFX Systems support three DCBX modes:

- Autonegotiation—The interface negotiates with the connected peer to determine the DCBX mode. This is the default DCBX mode.
- IEEE DCBX—The interface uses IEEE DCBX type, length, and value (TLV) to exchange DCBX information with the connected peer. QFX3500 Node devices come up with IEEE DCBX enabled by default and then autonegotiate with the connected peer to determine the final DCBX mode.
- DCBX Version 1.01—The interface uses Converged Enhanced Ethernet (CEE) DCBX version 1.01 TLVs to exchange DCBX information with the connected peer. QFabric Node devices come up with DCBX version 1.01 enabled by default and then autonegotiate with the connected peer to determine the final DCBX mode.

NOTE: **QFX Systems do not support pre-CEE (pre-DCB) versions of DCBX such as DCBX version 1.00. If a QFX Series interface receives an LLDP frame with pre-CEE DCBX TLVs, the system drops the frame.**

Configure the DCBX mode by specifying the mode for one interface or for all interfaces.

- To configure the DCBX mode, specify the interface and the mode:

  ```
  [edit protocols dcbx]
  user@switch# set interface interface-name mode (auto-negotiate | ieee-dcbx | dcbx-version-1.01)
  ```

  For example, to configure DCBX version 1.01 on interface **xe-0/0/21**:

  ```
  user@switch# set protocols dcbx interface xe-0/0/21 mode dcbx-version-1.01
  ```

  To configure IEEE DCBX on all interfaces:

  ```
  user@switch# set protocols dcbx interface all mode ieee-dcbx
  ```

Related Documentation

- Configuring DCBX Autonegotiation on page 306
- Disabling the ETS Recommendation TLV on page 309
- Understanding DCBX on page 108
- Understanding DCBX Application Protocol TLV Exchange on page 117
- **show dcbx neighbors on page 415**

Configuring DCBX Autonegotiation

Data Center Bridging Capability Exchange protocol (DCBX) discovers the data center bridging (DCB) capabilities of peers by exchanging feature configuration information. DCBX also detects feature misconfiguration and mismatches, and can configure DCB on peers. DCBX is an extension of the Link Layer Discovery Protocol (LLDP), and LLDP must remain enabled on every interface for which you want to use DCBX. If you attempt to enable DCBX on an interface on which LLDP is disabled, the configuration commit operation fails.

NOTE: **LLDP and DCBX are enabled by default on all interfaces.**

The switch supports DCBX autonegotiation for:

- Priority-based flow control (PFC) configuration
- Layer 2 and Layer 4 applications such as Fibre Channel over Ethernet (FCoE) and Internet Small Computer System Interface (iSCSI)
- Enhanced transmission selection (ETS) advertisement

DCBX autonegotiation is configured on a per-interface basis for each supported feature or application. The PFC and application DCBX exchanges use autonegotiation by default. The default autonegotiation behavior is:

- DCBX is enabled on the interface if the connected peer device also supports DCBX.
- DCBX is disabled on the interface if the connected peer device does not support DCBX.

You can override the default behavior for each feature by turning off autonegotiation to force an interface to enable or disable the feature.

Autonegotiation of ETS means that when ETS is enabled on an interface (priority groups are configured), the interface advertises its ETS configuration to the peer device. In this case, priorities (forwarding classes) that are not part of a priority group (forwarding class set) receive no bandwidth and are advertised in an automatically generated default forwarding class. If ETS is not enabled on an interface (no priority groups are configured), all of the priorities are advertised in one automatically generated default priority group that receives 100 percent of the port bandwidth.

Disabling ETS autonegotiation prevents the interface from sending the Recommendation TLV or the Configuration TLV to the connected peer.

On interfaces that use IEEE DCBX mode to exchange DCBX parameters, you can disable autonegotiation of the enhanced transmission selection (ETS) Recommendation TLV to the peer if you want an asymmetric ETS configuration between the peers. DCBX still exchanges the ETS Configuration TLV if you disable the ETS Recommendation TLV.

Autonegotiation of PFC means that when PFC is enabled on an interface, if the peer device connected to the interface supports PFC and is provisioned compatibly with the switch, DCBX sets the PFC operational state to enabled. If the peer device connected to the interface does not support PFC or is not provisioned compatibly with the switch, DCBX sets the operational state to disabled.

In addition, if the peer advertises that it is "willing" to learn its PFC configuration from the switch, DCBX pushes the switch's PFC configuration to the peer and does not check the peer's administrative state. The switch does not learn PFC configuration from peers (the switch does not advertise its state as "willing").

Disabling PFC autonegotiation prevents the interface from exchanging PFC configuration information with the peer. It forces the interface to enable PFC if PFC is configured on the interface or to disable PFC if PFC is not configured on the interface. If you disable PFC autonegotiation, the assumption is that the peer is also configured manually.

Autonegotiation of applications depends on whether or not you apply an application map to an interface. If you apply an application map to an interface, the interface autonegotiates DCBX for each application in the application map. PFC must be enabled on the FCoE priority (the FCoE IEEE 802.1p code point) for the interface to advertise the FCoE application. The interface only advertises applications that are included in the application map.

For example, if you apply an application map to an interface and the application map does not include the FCoE application, then that interface does not perform DCBX advertisement of FCoE.

If you do not apply an application map to an interface, DCBX does not advertise applications on that interface, with the exception of FCoE, which is handled differently than other applications.

NOTE: **If you do not apply an application map to an interface, the interface performs autonegotiation of FCoE if the interface carries traffic in the FCoE forwarding class and also has PFC enabled on the FCoE priority. On such interfaces, if DCBX detects that the peer device connected to the interface supports FCoE, the switch advertises its FCoE capability and IEEE 802.1p code point on that interface. If DCBX detects that the peer device connected to the interface does not support FCoE, DCBX marks that interface as "FCoE down" and disables FCoE on the interface.**

When DCBX marks an interface as "FCoE down," the behavior of the switch depends on how you use it in the network:

- When the switch acts as an FCoE-FC gateway, it does not send or receive FCoE Initialization Protocol (FIP) packets.
- When the switch acts as an FCoE transit switch, the interface drops all of the FIP packets it receives. In addition, FIP packets received from an FCoE forwarder (FCF) are not forwarded to interfaces marked as "FCoE down."

Disabling autonegotiation prevents the interface from exchanging application information with the peer. In this case, the assumption is that the peer is also configured manually.

To disable DCBX autonegotiation of PFC, applications (including FCoE), and ETS using the CLI:

1. Turn off autonegotiation for PFC.

   [edit]
   user@switch# **set protocols dcbx interface *interface-name* priority-flow-control no-auto-negotiation**

2. Turn off autonegotiation for applications.

   [edit]
   user@switch# **set protocols dcbx interface *interface-name* applications no-auto-negotiation**

3. Turn off autonegotiation for ETS.

   [edit]
   user@switch# **set protocols dcbx interface *interface-name* enhanced-transmission-selection no-auto-negotiation**

To disable autonegotiation of the ETS Recommendation TLV so that DCBX exchanges only the ETS Configuration TLV:

- [edit protocols dcbx interface *interface-name*]
  user@switch# **set enhanced-transmission-selection no-recommendation-tlv**

Related Documentation

- Example: Configuring DCBX Application Protocol TLV Exchange on page 217
- Example: Configuring CoS PFC for FCoE Traffic on page 186
- Disabling the ETS Recommendation TLV on page 309
- Understanding DCBX Application Protocol TLV Exchange on page 117

Disabling the ETS Recommendation TLV

The enhanced transmission selection (ETS) Recommendation TLV communicates the ETS settings that the switch wants the connected peer interface to use. If the peer interface is "willing," the peer interface changes its configuration to match the configuration in the ETS Recommendation TLV. By default, the switch interfaces send the ETS Recommendation TLV to the peer. The settings communicated are the egress ETS settings defined by configuring hierarchical scheduling on the interface.

We recommend that you use the same ETS settings on the connected peer that you use on the switch interface and that you leave the ETS Recommendation TLV enabled. However, on interfaces that use IEEE DCBX as the DCBX mode, if you want an asymmetric configuration between the switch interface and the connected peer, you can disable the ETS Recommendation TLV.

NOTE: **Disabling the ETS Recommendation TLV on interfaces that use DCBX version 1.01 as the DCBX mode has no effect and does not change DCBX behavior.**

If you disable the ETS Recommendation TLV, the switch still sends the ETS Configuration TLV to the connected peer. The result is that the connected peer is informed about the switch DCBX ETS configuration, but even if the peer is "willing," the peer does not change its configuration to match the switch configuration. This is asymmetric configuration—the two interfaces can have different parameter values for the ETS attribute.

To disable the ETS Recommendation TLV:

- `[edit protocols dcbx interface` *`interface-name`*`]`
  `user@switch#` **set enhanced-transmission-selection no-recommendation-tlv**

Related Documentation

- Configuring the DCBX Mode on page 305
- Configuring DCBX Autonegotiation on page 306
- Understanding DCBX on page 108
- *Understanding Data Center Bridging Capability Exchange Protocol for EX Series Switches*

Defining an Application for DCBX Application Protocol TLV Exchange

Define each application for which you want DCBX to exchange application protocol information. You can define Layer 2 and Layer 4 applications. After you define applications, you map them to IEEE 802.1p code points, and then apply the application map to the interfaces on which you want DCBX to exchange application protocol information with connected peers. (See *Related Documentation* for how to configure application maps and apply them to interfaces, and for an example of the entire procedure that also includes classifier configuration.)

NOTE: In Junos OS Release 12.1, the FCoE application was configured by default, so you did not need to configure it in an application map. In Junos OS Release 12.2, if you want DCBX to advertise the FCoE application on an interface and you apply an application map to that interface, you must explicitly configure FCoE in the application map. You also must enable priority-based flow control (PFC) on the FCoE code point on all interfaces that you want to advertise FCoE. If you apply an application map to an interface, the interface sends DCBX TLVs only for the applications configured in the application map.

Define Layer 2 applications by mapping an application name to an EtherType. Define Layer 4 applications by mapping an application name to a protocol (TCP or UDP) and a destination port.

- To define a Layer 2 application, specify the name of the application and its EtherType:

  ```
  [edit applications]
  user@switch# set application application-name ether-type ether-type
  ```

  For example, to configure an application named **PTP** (for Precision Time Protocol) that uses the EtherType **0x88F7**:

  ```
  user@switch# set applications application ptp ether-type 0x88F7
  ```

- To define a Layer 4 application, specify the name of the application, its protocol (TCP or UDP), and its destination port:

  ```
  [edit]
  user@switch# set applications application application-name protocol (tcp | udp)
  destination-port port-value
  ```

  For example, to configure an application named **iscsi** (for Internet Small Computer System Interface) that uses the protocol **TCP** and the destination port **3260**:

  ```
  user@switch# set applications application iscsi protocol tcp destination-port 3260
  ```

Related Documentation

- Configuring an Application Map for DCBX Application Protocol TLV Exchange on page 311
- Applying an Application Map to an Interface for DCBX Application Protocol TLV Exchange on page 312
- Configuring DCBX Autonegotiation on page 306
- Example: Configuring DCBX Application Protocol TLV Exchange on page 217
- *Example: Configuring DCBX to Support an iSCSI Application*
- Understanding DCBX Application Protocol TLV Exchange on page 117
- **show dcbx neighbors on page 415**

Configuring an Application Map for DCBX Application Protocol TLV Exchange

After you define applications for which you want to exchange DCBX application protocol information, map the applications to IEEE 802.1p code points. The IEEE 802.1p code points identify incoming traffic and allow you to map that traffic to the desired application. You then apply the application map to the interfaces on which you want DCBX to exchange application protocol information with connected peers. (See *Related Documentation* for how to define applications and apply the application map to interfaces, and for an example of the entire procedure that also includes classifier configuration.)

NOTE: **In Junos OS Release 12.1, the FCoE application was configured by default, so you did not need to configure it in an application map. In Junos OS Release 12.2, if you want DCBX to advertise the FCoE application on an interface and you apply an application map to that interface, you must explicitly configure FCoE in the application map. You also must enable priority-based flow control (PFC) on the FCoE code point on all interfaces that you want to advertise FCoE. If you apply an application map to an interface, the interface sends DCBX TLVs only for the applications configured in the application map.**

Configure an application map by creating an application map name and mapping an application to one or more IEEE 802.1p code points.

- To define an application map, specify the name of the application map, the name of the application, and the IEEE 802.1p code points of the incoming traffic that you want to associate with the application in the application map:

  `[edit policy-options]`
  `user@switch#` **set application-maps** ***application-map-name*** **application** ***application-name*** **code-points [** ***aliases*** **] [** ***bit-patterns*** **]**

  For example, to configure an application map named **ptp-app-map** that includes an application named **PTP** (for Precision Time Protocol) and map the application to IEEE 802.1p code points **001** and **101**:

  `user@switch#` **set policy-options application-maps ptp-app-map application ptp code points [001 101]**

Related Documentation

- Defining an Application for DCBX Application Protocol TLV Exchange on page 309
- Applying an Application Map to an Interface for DCBX Application Protocol TLV Exchange on page 312
- Configuring DCBX Autonegotiation on page 306
- Example: Configuring DCBX Application Protocol TLV Exchange on page 217
- *Example: Configuring DCBX to Support an iSCSI Application*
- **show dcbx neighbors on page 415**

Applying an Application Map to an Interface for DCBX Application Protocol TLV Exchange

After you define applications and map them to IEEE 802.1p code points in an application map, apply the application map to the interfaces on which you want DCBX to exchange the application protocol information with connected peers. (See *Related Documentation* for how to define applications and configure application maps to interfaces, and for an example of the entire procedure that also includes classifier configuration.)

NOTE: In Junos OS Release 12.1, the FCoE application was configured by default, so you did not need to configure it in an application map. In Junos OS Release 12.2, if you want DCBX to advertise the FCoE application on an interface and you apply an application map to that interface, you must explicitly configure FCoE in the application map. You also must enable priority-based flow control (PFC) on the FCoE code point on all interfaces that you want to advertise FCoE. If you apply an application map to an interface, the interface sends DCBX TLVs only for the applications configured in the application map.

- To apply an application map to a DCBX interface, specify the DCBX interface and the application map name:

  ```
  [edit protocols]
  user@switch# set dcbx interface interface-name application-map application-map-name
  ```

  For example, to apply an application map named **ptp-app-map** on interface **xe-0/0/11**:

  ```
  user@switch# set protocols dcbx interface xe-0/0/11 application-map ptp-app-map
  ```

Related Documentation

- Defining an Application for DCBX Application Protocol TLV Exchange on page 309
- Configuring an Application Map for DCBX Application Protocol TLV Exchange on page 311
- Configuring DCBX Autonegotiation on page 306
- Example: Configuring DCBX Application Protocol TLV Exchange on page 217
- *Example: Configuring DCBX to Support an iSCSI Application*
- **show dcbx neighbors on page 415**

CHAPTER 7

Configuration Statements

application (Application Maps)

Syntax

```
application application-name {
  code-points [ aliases ] [ bit-patterns ];
}
```

Hierarchy Level [edit policy-options application-maps *application-map-name*]

Release Information Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Add an application to an application map and define the application's code points.

Options ***application-name***—Name of the application.

The remaining statement is explained separately.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring an Application Map for DCBX Application Protocol TLV Exchange on page 311
- Example: Configuring DCBX Application Protocol TLV Exchange on page 217
- *Example: Configuring DCBX to Support an iSCSI Application*
- Understanding DCBX Application Protocol TLV Exchange on page 117
- *Understanding DCBX Application Protocol TLV Exchange on EX Series Switches*

application (Applications)

Syntax

```
application application-name {
  destination-port port-value;
  protocol (tcp | udp);
  ether-type type;
}
```

Hierarchy Level [edit applications]

Release Information Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Configure properties to define an application.

Options *application-name*—Name of the application.

The statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Defining an Application for DCBX Application Protocol TLV Exchange on page 309
- Example: Configuring DCBX Application Protocol TLV Exchange on page 217
- *Example: Configuring DCBX to Support an iSCSI Application*
- Understanding DCBX Application Protocol TLV Exchange on page 117
- *Understanding DCBX Application Protocol TLV Exchange on EX Series Switches*

application-map

Syntax application-map *application-map-name*;

Hierarchy Level [edit protocols dcbx interface *interface-name*]

Release Information Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Specify an application map to apply to an interface.

Options *application-map-name*—Name of the application map.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- **show dcbx neighbors on page 415**
- Applying an Application Map to an Interface for DCBX Application Protocol TLV Exchange on page 312
- Example: Configuring DCBX Application Protocol TLV Exchange on page 217
- *Example: Configuring DCBX to Support an iSCSI Application*
- Understanding DCBX Application Protocol TLV Exchange on page 117
- *Understanding DCBX Application Protocol TLV Exchange on EX Series Switches*

application-maps

Syntax

```
application-maps application-map-name {
  application application-name {
    code-points [ aliases ] [ bit-patterns ];
  }
}
```

Hierarchy Level [edit policy-options]

Release Information Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Define an application map by specifying the applications that belong to the application map.

Options *application-map-name*—Name of the application map.

The remaining statements are explained separately.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring an Application Map for DCBX Application Protocol TLV Exchange on page 311
- Example: Configuring DCBX Application Protocol TLV Exchange on page 217
- *Example: Configuring DCBX to Support an iSCSI Application*
- Understanding DCBX Application Protocol TLV Exchange on page 117
- *Understanding DCBX Application Protocol TLV Exchange on EX Series Switches*

applications (Applications)

Syntax

```
applications {
  application application-name {
    destination-port port-value;
    protocol (tcp | udp);
    ether-type type;
  }
}
```

Hierarchy Level [edit]

Release Information Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Define applications that DCBX advertises.

Options The statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Defining an Application for DCBX Application Protocol TLV Exchange on page 309
- Example: Configuring DCBX Application Protocol TLV Exchange on page 217
- *Example: Configuring DCBX to Support an iSCSI Application*
- Understanding DCBX Application Protocol TLV Exchange on page 117
- *Understanding DCBX Application Protocol TLV Exchange on EX Series Switches*

applications (DCBX)

Syntax

```
applications {
  no-auto-negotiation;
}
```

Hierarchy Level [edit protocols dcbx interface *interface-name*]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced in Junos OS Release 12.1 for the EX Series

Description Configure Data Center Bridging Capability Exchange protocol (DCBX) applications on an interface.

Options The remaining statements are explained separately.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

-
- **show dcbx neighbors on page 415**
- Understanding DCB Features and Requirements on page 16

auto-load-rebalance

Syntax auto-load-rebalance;

Hierarchy Level [edit fc-fabrics *fabric-name* proxy]

Release Information Command introduced in Junos OS Release 12.3 for the QFX Series.

Description Configure the system to rebalance NP_Port link loads automatically on an FCoE-FC gateway proxy fabric if the link loads become unbalanced. Load rebalancing is a disruptive action that forces some or all sessions (depending on the configured load-balancing algorithm) to log out and then log in again. When sessions log in again, they are placed on NP_Port interfaces so that the link loads are balanced.

Required Privilege Level storage—To view this statement in the configuration.
storage-control—To add this statement to the configuration.

Related Documentation

- Defining the Proxy Load-Balancing Algorithm on page 279
- Example: Configuring Automated Fibre Channel Interface Load Rebalancing on page 250
- Simulating On-Demand Fibre Channel Link Load Rebalancing (Dry Run Test) on page 280
- Understanding Load Balancing in an FCoE-FC Gateway Proxy Fabric on page 63
- Monitoring Fibre Channel Interface Load Balancing on page 383

bb-sc-n

Syntax bb-sc-n *bb-sc-n*;

Hierarchy Level [edit interfaces *interface-name* fibrechannel-options]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure the buffer-to-buffer credit state change number to prevent the permanent loss of Fibre Channel credits over time (buffer-to-buffer credit recovery).

Options ***bb-sc-n***—Number of buffer-to-buffer state change credits.

Range: 0 through 15

Default: 0 (disabled)

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- **show fibre-channel interfaces on page 472**
- Example: Setting Up Fibre Channel and FCoE VLAN Interfaces in an FCoE-FC Gateway Fabric on page 171
- Configuring a Fibre Channel Interface on page 269
- Configuring a Physical Fibre Channel Interface on page 268
- Understanding Interfaces on an FCoE-FC Gateway on page 50

beacon-period

Syntax beacon-period *milliseconds*;

Hierarchy Level Original CLI

[edit ethernet-switching options secure-access-port vlan (all | *vlan-name*) examine-fip examine-vn2vn]

ELS CLI for Platforms that Support FCoE

[edit vlans *vlan-name* forwarding-options fip-security]

NOTE: The beacon-period configuration statement is in a different hierarchy on the original CLI than on the Enhanced Layer 2 Software (ELS) CLI.

Release Information Statement introduced in Junos OS Release 12.2 for the QFX Series.
Statement introduced for the ELS CLI in Junos OS Release 13.2 for the QFX Series.

Description Set the interval between periodic beacons. Beacons perform virtual link maintenance for VN_Ports in a way that is similar to FIP keepalive advertisements.

The ENode sends periodic beacons every 90 seconds on behalf of the VN_Port. Each received beacon resets the session timer for the virtual link connection to the other VN_Port. If the FCF does not receive a beacon before the beacon timer expires, the VN_Port is considered as “down” and the virtual link is terminated. The beacon timer expires in 2.5 times the configured beacon timer value.

Options *milliseconds*—Time in milliseconds between beacons.

Range: 250 through 90000 milliseconds

Default: 8000 milliseconds

Required Privilege Level storage—To view this statement in the configuration.
storage-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Directly Connected to the Same FCoE Transit Switch) on page 228
- Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Directly Connected to Different FCoE Transit Switches) on page 233
- Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Indirectly Connected Through an Aggregation Layer FCoE Transit Switch) on page 241
- *Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Directly Connected to the Same FCoE Transit Switch)*
- *Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Directly Connected to Different FCoE Transit Switches)*

- *Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Indirectly Connected Through an Aggregation Layer FCoE Transit Switch)*

code-points (Application Maps)

Syntax code-points [*aliases*] [*bit-patterns*];

Hierarchy Level [edit policy-options application-maps *application-map-name* application *application-name*]

Release Information Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Define one or more code-point aliases or bit sets for an application.

Options ***aliases***—Name of the alias or aliases.

bit-patterns—Value of the code-point bits, in decimal form.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring an Application Map for DCBX Application Protocol TLV Exchange on page 311
- Example: Configuring DCBX Application Protocol TLV Exchange on page 217
- *Example: Configuring DCBX to Support an iSCSI Application*
- Understanding DCBX Application Protocol TLV Exchange on page 117
- *Understanding DCBX Application Protocol TLV Exchange on EX Series Switches*

dcbx

Syntax

```
dcbx {
  disable;
  interface (interface-name | all) {
    disable;
    application-map application-map-name;
    applications {
      no-auto-negotiation;
    }
    enhanced-transmission-selection {
      no-auto-negotiation;
      no-recommendation-tlv;
      recommendation-tlv {
        no-auto-negotiation;
      }
    }
    dcbx-version (auto-negotiate | ieee-dcbx | dcbx-version-1.01);
    priority-flow-control {
      no-auto-negotiation;
    }
  }
}
```

Hierarchy Level [edit protocols]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced in Junos OS Release 11.3 for EX Series switches.
mode and **recommendation-tlv** statements introduced in Junos OS Release 12.2 for the QFX Series.

Description Configure DCBX properties.

Options The statements are explained separately.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- **show dcbx neighbors on page 415**
- Understanding DCB Features and Requirements on page 16
- Configuring DCBX Autonegotiation on page 306

- *Understanding DCB Features and Requirements on EX Series Switches*
- *Disabling DCBX to Disable PFC Autonegotiation on EX Series Switches (CLI Procedure)*

dcbx-version

Syntax dcbx-version (auto-negotiate | ieee-dcbx | dcbx-version-1.01);

Hierarchy Level [edit protocols dcbx interface (all | *interface-name*)]

Release Information Statement introduced in Junos OS Release 12.2 for the QFX Series.

Description Set the DCBX version for the specified interface or interfaces.

QFX3500 switches come up in IEEE DCBX mode and then autonegotiate with the connected peer to set the DCBX version.

QFabric system Node devices come up using DCBX version 1.01, and then autonegotiate with the connected peer to set the DCBX mode.

Default The default DCBX mode is autonegotation.

Options **auto-negotiate**—Automatically negotiate the DCBX version with the connected peer.

ieee-dcbx—Force the interface to use IEEE DCBX mode, regardless of the peer configuration.

dcbx-version-1.01—Force the interface to use version 1.01 DCBX mode, regardless of the peer configuration.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- **show dcbx neighbors on page 415**
- Configuring DCBX Autonegotiation on page 306
- Understanding DCBX on page 108

description (Fibre Channel Fabrics)

Syntax description *description*

Hierarchy Level [edit fc-fabrics *fabric-name*]

Description Text string that describes the Fibre Channel fabric. The text string has no effect on the operation of the fabric.

Options *description*—Text that describes the fabric. Text can include letters, numbers, and hyphens (-) and can be up to 255 characters in length. If the text includes spaces, enclose the entire text string in quotation marks.

Required Privilege Level storage—To view this statement in the configuration.
storage-control—To add this statement to the configuration.

Related Documentation

- show fibre-channel fabric on page 437

destination-port (Applications)

Syntax destination-port *port-value*;

Hierarchy Level [edit applications application *application-name*]

Release Information Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) destination port number, which combines with **protocol** to identify an application type. The Internet Assigned Numbers Authority (IANA) assigns port numbers. See the IANA *Service Name and Transport Protocol Port Number Registry* at http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml for a list of assigned port numbers.

NOTE: To create an application for iSCSI, use the protocol **tcp** with the destination port number **3260**.

Options *port-value*—Identifier for the port.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Defining an Application for DCBX Application Protocol TLV Exchange on page 309
- Example: Configuring DCBX Application Protocol TLV Exchange on page 217
- *Example: Configuring DCBX to Support an iSCSI Application*
- Understanding DCBX Application Protocol TLV Exchange on page 117
- *Understanding DCBX Application Protocol TLV Exchange on EX Series Switches*

disable (DCBX)

Syntax disable

Hierarchy Level [edit protocols dcbx]

[edit protocols dcbx interface *interface-name*]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced in Junos OS Release 11.3 for EX Series switches.

Description Disable Data Center Bridging Capability Exchange protocol (DCBX) on one or more 10-Gigabit Ethernet interfaces.

Default DCBX is enabled by default on all 10-Gigabit or higher Ethernet interfaces.

DCBX is enabled by default on all 10-Gigabit Ethernet interfaces on EX4500 CEE-enabled switches.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Configuring DCBX Autonegotiation on page 306
- *Disabling DCBX to Disable PFC Autonegotiation on EX Series Switches (CLI Procedure)*
- Understanding DCB Features and Requirements on page 16
- *Understanding DCB Features and Requirements on EX Series Switches*

enhanced-transmission-selection

Syntax

```
enhanced-transmission-selection {
  no-auto-negotiation;
  no-recommendation-tlv;
  recommendation-tlv {
    no-auto-negotiation;
  }
}
```

Hierarchy Level [edit protocols dcbx interface *interface-name*]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Disable advertising the enhanced transmission selection (ETS) state of the interface to the peer. To disable ETS on the interface, do not enable ETS on the interface in the class-of-service (CoS) configuration.

Disabling ETS autonegotiation stops the QFX Series from advertising the ETS Configuration TLV and the ETS Recommendation TLV.

Disabling the ETS recommendation TLV stops the QFX Series from advertising the ETS Recommendation TLV, but the ETS Configuration TLV is still advertised.

Options **no-auto-negotiation**—Disable automatic negotiation of ETS (Configuration TLV and Recommendation TLV)

no-recommendation-tlv—Disable automatic negotiation of the ETS Recommendation TLV

recommendation-tlv—Enable automatic negotiation of ETS Recommendation TLV

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- **show dcbx neighbors on page 415**
- Configuring DCBX Autonegotiation on page 306
- *Example: Configuring CoS Hierarchical Port Scheduling (ETS)*
- Understanding DCB Features and Requirements on page 16

ether-type

Syntax ether-type *ether-type*;

Hierarchy Level [edit applications application *application-name*]

Release Information Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Two-octet field in an Ethernet frame that defines the protocol encapsulated in the frame payload. See http://standards.ieee.org/develop/regauth/ethertype/eth.txt for a list of Institute of Electrical and Electronics Engineers (IEEE) EtherTypes.

NOTE: To create a FIP application, use the EtherType 0x8914.

Options ***type***—Identifier for the EtherType.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Defining an Application for DCBX Application Protocol TLV Exchange on page 309
- Example: Configuring DCBX Application Protocol TLV Exchange on page 217
- Understanding DCBX Application Protocol TLV Exchange on page 117

ethernet-interfaces

Syntax

```
ethernet-interfaces {
  node-group (node-group-name | all) {
    oxid (enable | disable);
  }
}
```

Hierarchy Level [edit forwarding-options hash-key family fcoe]

Release Information Statement introduced in Junos OS Release 13.2X52-D10 for the QFX Series.

Description Specify that you are enabling or disabling OxID hash control on the Ethernet (FCoE) LAG ports of a QFabric system Node group. OxID hash control is enabled or disabled on the FCoE LAG ports that face an FCoE forwarder (FCF).

Options The statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Enabling and Disabling CoS OxID Hash Control on QFabric Systems on page 282
- Understanding OxID Hash Control for FCoE Traffic Load Balancing on QFabric Systems on page 163

examine-fip

Syntax

```
examine-fip {
  examine-vn2vn {
    beacon-period milliseconds;
  }
  fc-map fc-map-value;
  no-fip-snooping-scaling;
}
```

Hierarchy Level

[edit ethernet-switching-options secure-access-port vlan (all | *vlan-name*)]

Release Information

Statement introduced in Junos OS Release 10.4 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement **examine-vn2vn** introduced in Junos OS Release 12.2 for the QFX Series.
Statement **no-fip-snooping-scaling** introduced in Junos OS Release 13.2X52-D10 for the QFX Series.

Description

NOTE: This statement supports the original CLI. If your switch runs the Enhanced Layer 2 Software (ELS) CLI, see *examine-vn2vf* for VN_Port to VF_Port (VN2VF_Port) FIP snooping, and see examine-vn2vn for VN_Port to VN_Port (VN2VN_Port) FIP snooping. For ELS details, see *Getting Started with Enhanced Layer 2 Software*.

Enable FIP snooping on a specified VLAN. Ensure that the VLAN is a dedicated FCoE VLAN that transports only FCoE traffic.

(QFX Series only) Enable VN2VN_Port FIP snooping on the specified VLAN. The VLAN must be a dedicated FCoE VLAN that transports only VN2VN_Port traffic. One FCoE VLAN cannot support both VN2VF_Port FIP snooping and VN2VN_Port FIP snooping. Configure separate, dedicated FCoE VLANs for VN2VN_Port FIP snooping and VN2VN_Port FIP snooping.

The remaining statements are explained separately.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- *vlan*
- *Example: Configuring an FCoE Transit Switch*
- Configuring VN2VF_Port FIP Snooping and FCoE Trusted Interfaces on an FCoE Transit Switch on page 294

examine-vn2vn

Syntax

```
examine-vn2vn {
  beacon-period milliseconds;
}
```

Hierarchy Level

Original CLI

[edit ethernet-switching options secure-access-port vlan (all | *vlan-name*) examine-fip]

ELS CLI for Platforms that Support FCoE

[edit vlans *vlan-name* forwarding-options fip-security]

NOTE: The examine-vn2vn configuration statement is in a different hierarchy on the original CLI than on the Enhanced Layer 2 Software (ELS) CLI.

Release Information

Statement introduced in Junos OS Release 12.2 for the QFX Series.
Statement introduced for the ELS CLI in Junos OS Release 13.2 for the QFX Series.

Description

Enable VN_Port to VN_Port (VN2VN) FIP snooping on a specified VLAN. The VLAN must be a dedicated FCoE VLAN that transports only FCoE traffic. A VLAN cannot support VN2VN FIP snooping and VN_Port to VF_Port FIP snooping (VN2VF) simultaneously. Configure separate VLANs for VN2VN FIP snooping and VN2VF FIP snooping.

When you enable VN2VN FIP snooping on a VLAN, the VN2VF session filters are removed and the all existing VN2VF sessions are terminated.

The remaining statement is explained separately.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Directly Connected to the Same FCoE Transit Switch) on page 228
- Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Directly Connected to Different FCoE Transit Switches) on page 233
- Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Indirectly Connected Through an Aggregation Layer FCoE Transit Switch) on page 241
- *Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Directly Connected to the Same FCoE Transit Switch)*
- *Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Directly Connected to Different FCoE Transit Switches)*
- *Example: Configuring VN2VN_Port FIP Snooping (FCoE Hosts Indirectly Connected Through an Aggregation Layer FCoE Transit Switch)*

fabric-id

Syntax fabric-id *fc-fabric-id*;

Hierarchy Level [edit fc-fabrics *fc-fabric-name*]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure a unique identifier for the FC fabric.

NOTE: Changing the ID of an FC fabric causes all logins to drop and forces the ENodes to log in again.

Options *fc-fabric-id* —Unique identifier of the FC fabric.

Required Privilege Level storage—To view this statement in the configuration.
storage-control—To add this statement to the configuration.

Related Documentation

- show fibre-channel fabric on page 437
- Configuring an FCoE-FC Gateway Fibre Channel Fabric on page 266
- Understanding an FCoE-FC Gateway on page 29

fabric-interfaces

Syntax

```
fabric-interfaces {
  node-group (node-group-name | all) {
    oxid (enable | disable);
  }
}
```

Hierarchy Level [edit forwarding-options hash-key family fcoe]

Release Information Statement introduced in Junos OS Release 13.2X52-D10 for the QFX Series.

Description Specify that you are enabling or disabling OxID hash control on the fabric ports of a QFabric system Node group. OxID hash control is enabled or disabled on the fabric ports that face an FCoE forwarder (FCF).

Options The statements are explained separately.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Enabling and Disabling CoS OxID Hash Control on QFabric Systems on page 282
- Understanding OxID Hash Control for FCoE Traffic Load Balancing on QFabric Systems on page 163

fabric-type

Syntax fabric-type proxy;

Hierarchy Level [edit fc-fabrics *fc-fabric-name*]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Specify that the FC fabric be an FCoE-FC gateway fabric.

Options **proxy**—Specify that the switch be an FCoE-FC gateway fabric.

Required Privilege Level storage—To view this statement in the configuration.
storage-control—To add this statement to the configuration.

Related Documentation

- **show fibre-channel fabric on page 437**
- Configuring an FCoE-FC Gateway Fibre Channel Fabric on page 266
- Understanding an FCoE-FC Gateway on page 29

family fcoe

Syntax

QFX Series Standalone Switches

```
family fcoe {
  oxid (enable | disable);
}
```

QFabric Systems

```
family fcoe {
  ethernet-interfaces {
    node-group (node-group-name | all) {
      oxid (enable | disable);
    }
  }
  fabric-interfaces {
    node-group (node-group-name | all) {
      oxid (enable | disable);
    }
  }
}
```

Hierarchy Level

[edit forwarding-options hash-key]

Release Information

Statement introduced in Junos OS Release 12.3 for the QFX Series.
Ethernet-interfaces and fabric-interfaces statements introduced in Junos OS Release 13.2X52-D10 for the QFX Series.

Description

Configure whether or not to use the originator exchange identifier (OxID) field for hash control for FCoE traffic load balancing.

Options

The statement is explained separately.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Enabling and Disabling CoS OxID Hash Control on QFX Series Standalone Switches on page 281
- Enabling and Disabling CoS OxID Hash Control on QFabric Systems on page 282
- Understanding OxID Hash Control for FCoE Traffic Load Balancing on QFX Series Standalone Switches on page 79
- Understanding OxID Hash Control for FCoE Traffic Load Balancing on QFabric Systems on page 163

fc2

Syntax

```
fc2 {
  traceoptions {
    file filename <replace> <size size> <files number> <no-stamp>;
    <world-readable | no-world-readable>;
    flag flag <flag-modifier>;
  }
}
```

Hierarchy Level [edit fc-fabrics *fc-fabric-name*]

Description Fibre Channel network layer (FC2) configuration.

Options The statements are explained separately.

Required Privilege Level storage—To view this statement in the configuration.
storage-control—To add this statement to the configuration.

fc-fabrics

Syntax

```
fc-fabrics {
  fc-fabric-name {
    description
    fabric-id fc-fabric-id;
    fabric-type proxy;
    interface {
      interface-name {
        max-login-sessions max-login-sessions;
      }
      interface-name {
        max-login-sessions max-login-sessions;
      }
      <...>;
        max-login-sessions max-login-sessions;
      }
      vlan.interface-name;
    }
    fc2 {
      traceoptions {
        file filename <replace> <size size> <files number> <no-stamp>;
        <world-readable | no-world-readable>;
        flag flag <flag-modifier>;
      }
    }
    max-login-sessions max-login-sessions;
    protocols {
      fip {
        fcoe-trusted;
        fc-map fc-map-value;
        fka-adv-period milliseconds;
        interface {
          interface-name {
            fka-adv-period milliseconds;
            priority priority;
          }
        }
        max-sessions-per-enode max-sessions-per-enode;
        priority priority;
        traceoptions {
          file filename <replace> <size size> <files number> <no-stamp>;
          <world-readable | no-world-readable>;
          flag flag <flag-modifier> <disable>;
        }
      }
    }
    proxy {
      auto-load-rebalance
      load-balance-algorithm (simple | enode-based | flogi-based);
      no-fabric-wwn-verify;
      traceoptions {
        file filename <replace> <size size> <files number> <no-stamp>;
        <world-readable | no-world-readable>;
```



```
            flag flag <flag-modifier> <disable>;
          }
        }
      }
    }
```

Hierarchy Level [edit]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure an FC fabric. You can configure a maximum of 12 FC fabrics, one per native FC port.

NOTE: Changing the name of an FC fabric causes all logins to drop and forces the ENodes to log in again.

Options *fc-fabric-name* —Unique name of the FC fabric.

The other statements are explained separately.

Required Privilege Level storage—To view this statement in the configuration.
storage-control—To add this statement to the configuration.

Related Documentation

- **show fibre-channel fabric on page 437**
- Configuring VN2VF_Port FIP Snooping and FCoE Trusted Interfaces on an FCoE Transit Switch on page 294
- Configuring a Physical Fibre Channel Interface on page 268
- Configuring a Fibre Channel Interface on page 269
- Configuring an FCoE VLAN Interface on an FCoE-FC Gateway on page 272
- Assigning Interfaces to a Fibre Channel Fabric on page 276
- Configuring an FCoE-FC Gateway Fibre Channel Fabric on page 266
- Configuring FIP on an FCoE-FC Gateway on page 283
- Configuring DCBX Autonegotiation on page 306
- Overview of Fibre Channel on the QFX Series on page 4
- Understanding FCoE-FC Gateway Functions on page 33

fc-map

Syntax fc-map *fc-map-value*;

Hierarchy Level Original CLI

[edit ethernet-switching options secure-access-port vlan (all | *vlan-name*) examine-fip]

ELS CLI for Platforms that Support FCoE

[edit vlans *vlan-name* forwarding-options fip-security]

NOTE: The fc-map configuration statement is in a different hierarchy on the original CLI than on the Enhanced Layer 2 Software (ELS) CLI.

QFX Series that Support FCoE-FC Gateway Configuration

[edit fc-fabrics *fc-fabric-name* protocols fip]

Release Information Statement introduced in Junos OS Release 10.4 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced for the ELS CLI in Junos OS Release 13.2 for the QFX Series.

Description Set the FCoE mapped address prefix (FC-MAP) value for the FCoE VLAN to match the FC switch (or FCoE forwarder) FC-MAP value for the FC fabric. The FC-MAP value is a unique MAC address prefix an FC switch uses to identify FCoE traffic for a given FC fabric (traffic on a particular FCoE VLAN).

You can configure the FC-MAP value or use the default value. The default FC-MAP value is different for VN_Port to VF_Port (VN2VF_Port) FIP snooping (0x0EFC00) than for VN_Port to VN_Port (VN2VN_Port) FIP snooping.

The FC switch provides the FC-MAP value to FCoE nodes (ENodes) in the FIP discovery advertisement message. If the EX Series switch or the QFX Series FCoE VLAN FC-MAP value does not match the FC switch FC-MAP value, neither device discovers the FC switch on that VLAN, and the ENodes on that VLAN cannot access the FC switch. The FC switch accepts only FCoE traffic that uses the correct FC-MAP value as part of the VN_Port MAC address.

When the QFX Series acts as an FCoE-FC gateway, the FC-MAP value for the gateway and the FCoE devices must match the FC switch FC-MAP value in order to communicate with the FC switch.

NOTE: Changing the FC-MAP value causes all logins to drop and forces the ENodes to log in again.

Options *fc-map-value*—FC-MAP value, hexadecimal value preceded by “0x”.

Range: 0x0EFC00 through 0x0EFCFF

Default: 0x0EFC00 for VN2VF_Port FIP snooping0x0EFD00 for VN2VN_Port FIP snooping

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- examine-fip on page 332
- show fip snooping on page 492
- *Example: Configuring an FCoE Transit Switch*
- Configuring VN2VF_Port FIP Snooping and FCoE Trusted Interfaces on an FCoE Transit Switch on page 294

fc-options

Syntax

```
fc-options
  max-login-sessions-per-node max-login-sessions-per-node;
  no-fip-snooping-scaling;
  traceoptions {
    file filename <replace> <size size> <files number> <no-stamp>;
    <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
  }
```

Hierarchy Level

[edit]

Release Information

Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement **no-fip-snooping-scaling** introduced in Junos OS Release 13.2X52-D10 for the QFabric system.

Description

Set Fibre Channel options.

Required Privilege Level

storage—To view this statement in the configuration.
storage-control—To add this statement to the configuration.

fcoe-lag

Syntax fcoe-lag;

Hierarchy Level [edit interfaces *lag-interface-name* aggregated-ether-options]

Release Information Statement introduced in Junos OS Release 13.2X52-D10 for the QFX Series.

Description Configure a special link aggregation group (LAG) to transport Fibre Channel over Ethernet (FCoE) traffic and regular Ethernet traffic across the same link aggregation bundle.

An FCoE LAG ensures that FCoE traffic uses the same link within a LAG to transmit and receive information between an FCoE device and a Fibre Channel (FC) SAN switch across a QFabric system Node device. This preserves the point-to-point link emulation that FC requires. A standard LAG uses a hashing algorithm to determine the LAG link used for each communication, so with a standard LAG, you cannot guarantee that communication between an FCoE device and the QFabric system Node device always uses the same link. If communication between the FCoE device and the QFabric system Node device uses different physical links, the SAN terminates the link.

An FCoE LAG treats regular Ethernet traffic (traffic that is not FCoE traffic) in the same way as on a standard LAG, providing link redundancy and load-balancing for the regular Ethernet traffic. An FCoE LAG does not provide link redundancy or load balancing for FCoE traffic.

On FCoE-FC gateways, if the gateway has one or more untrusted FC fabrics, you must also disable FIP snooping scaling on the gateway by including the **no-fip-snooping-scaling** option in the **[edit fc-options]** hierarchy.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- **no-fcoe-lag on page 357**
- **no-fip-snooping-scaling on page 358**
- Configuring an FCoE LAG on page 302
- Configuring VLANs for FCoE Traffic on an FCoE Transit Switch on page 289
- Configuring VN2VF_Port FIP Snooping and FCoE Trusted Interfaces on an FCoE Transit Switch on page 294
- Understanding FCoE LAGs on page 158

fcoe-trusted

Syntax fcoe-trusted;

Hierarchy Level Original CLI

[edit ethernet-switching-options secure-access-port interface *interface-name*]

ELS CLI for Platforms that Support FCoE

[edit vlans *vlan-name* forwarding-options fip-security interface *interface-name*]

NOTE: **The fcoe-trusted configuration statement is in a different hierarchy on the original CLI than on the Enhanced Layer 2 Software (ELS) CLI.**

QFX Series that Support FCoE-FC Gateway Configuration

[edit fc-fabrics *fc-fabric-name* protocols fip]

Release Information Statement introduced in Junos OS Release 10.4 for EX Series switches.
Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced for the FC fabric in Junos OS Release 11.3 for the QFX Series.
Statement introduced for the ELS CLI in Junos OS Release 13.2 for the QFX Series.

Description Configure the specified 10-Gigabit Ethernet interface to trust Fibre Channel over Ethernet (FCoE) traffic. If an interface is connected to another switch such as an FCoE forwarder (FCF) or a transit switch, you can configure the interface as trusted so that the interface forwards FCoE traffic from the switch to the FCoE devices without installing FIP snooping filters.

(QFX Series FCoE-FC gateway) Configure the specified local Fibre Channel fabric to trust FCoE traffic on all ports in the fabric. Changing the fabric ports from untrusted to trusted removes any existing FIP snooping filters from the ports. Changing the fabric ports from trusted to untrusted by removing the **fcoe-trusted** configuration from the fabric forces all of the FCoE sessions on those ports to log out so that when the ENodes and VN_Ports log in again, the switch can build the appropriate FIP snooping filters.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- show fip snooping on page 492
- *Example: Configuring an FCoE Transit Switch*
- Configuring VN2VF_Port FIP Snooping and FCoE Trusted Interfaces on an FCoE Transit Switch on page 294
- Configuring VN2VF_Port FIP Snooping and FCoE Trusted Interfaces on an FCoE Transit Switch on page 294

fibre-channel (Family Interfaces)

Syntax

```
fibre-channel {
  port-mode (f-port | np-port);
}
```

Hierarchy Level

[edit interfaces vlan unit *logical-unit-number* family],
[edit interfaces *interface-name* unit *logical-unit-number* family]

Release Information

Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description

Configure the port mode for FCoE VLAN interfaces and native FC interfaces.

Options

The statements are explained separately.

Required Privilege Level

routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Setting Up Fibre Channel and FCoE VLAN Interfaces in an FCoE-FC Gateway Fabric on page 171
- Configuring an FCoE VLAN Interface on an FCoE-FC Gateway on page 272
- Configuring a Fibre Channel Interface on page 269
- **show fibre-channel interfaces on page 472**
- *show vlans*
- Understanding Interfaces on an FCoE-FC Gateway on page 50

fibre-channel (Port)

Syntax

```
fibre-channel {
  port-range {
    port-range-low port-range-high;
  }
}
```

Hierarchy Level [edit chassis fpc *fpc-id* pic *pic-id*]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Specify a range of ports to carry FC traffic when the switch is configured as an FCoE-FC gateway.

Options The statement is explained separately.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- Example: Setting Up Fibre Channel and FCoE VLAN Interfaces in an FCoE-FC Gateway Fabric on page 171
- Configuring a Physical Fibre Channel Interface on page 268
- **show fibre-channel interfaces on page 472**
- Understanding Interfaces on an FCoE-FC Gateway on page 50

fibrechannel-options

Syntax

```
fibrechannel-options {
  bb-sc-n
  (loopback | no-loopback);
  speed (auto-negotiation | 2g | 4g | 8g);
}
```

Hierarchy Level [edit interfaces *interface-name*]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure FC interface properties such as speed and loopback mode.

Options The statements are explained separately.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- **show fibre-channel interfaces on page 472**
- Configuring a Fibre Channel Interface on page 269

fip

Syntax

```
fip {
  fcoe-trusted;
  fc-map fc-map-value;
  fka-adv-period milliseconds;
  interface {
    interface-name {
      fka-adv-period milliseconds;
      priority priority;
    }
  }
  max-sessions-per-enode max-sessions-per-enode;
  priority priority;
  traceoptions {
    file filename <replace> <size size> <files number> <no-stamp>;
    <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
  }
}
```

Hierarchy Level [edit fc-fabrics *fc-fabric-name* protocols]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure global or interface-specific FIP options. Individual interface settings override global settings.

Options The statements are explained separately.

Required Privilege Level storage—To view this statement in the configuration.
storage-control—To add this statement to the configuration.

Related Documentation

- show fibre-channel fip on page 443
- Configuring FIP on an FCoE-FC Gateway on page 283
- Overview of FIP on page 9

fka-adv-period

Syntax

fka-adv-period *milliseconds*;

Hierarchy Level

[edit fc-fabrics *fc-fabric-name* protocols fip],
[edit fc-fabrics *fc-fabric-name* protocols fip interface *interface-name*]

Release Information

Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description

Set the global or interface-specific interval between periodic FIP keepalive advertisements. An interval set at the interface level overrides the global setting.

Options

milliseconds—Time in milliseconds between FIP keepalive advertisements.

Range: 250 through 90000 milliseconds

Default: 8000 milliseconds

Required Privilege Level

storage—To view this statement in the configuration.
storage-control—To add this statement to the configuration.

Related Documentation

- show fibre-channel fip on page 443
- show fibre-channel fip interface on page 458
- Overview of FIP on page 9

interface (DCBX)

Syntax

```
interface (interface-name | all) {
  disable;
  application-map application-map-name;
  applications {
    no-auto-negotiation;
  }
  enhanced-transmission-selection {
    no-auto-negotiation;
    no-recommendation-tlv;
    recommendation-tlv {
      no-auto-negotiation;
    }
  }
  dcbx-version (auto-negotiate | ieee-dcbx | dcbx-version-1.01);
  priority-flow-control {
    no-auto-negotiation;
  }
}
```

Hierarchy Level [edit protocols dcbx]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced in Junos OS Release 11.3 for the EX Series switches.
Mode and **recommendation-tlv** statements introduced in Junos OS Release 12.2 for the QFX Series.

Description Configure DCBX properties on an interface.

Options ***interface-name***—Name of the interface.

The remaining statements are explained separately.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- **show dcbx neighbors on page 415**
- Configuring DCBX Autonegotiation on page 306
- *Example: Configuring DCBX to Support an iSCSI Application*
- Understanding DCB Features and Requirements on page 16
- *Understanding DCB Features and Requirements on EX Series Switches*
- *Understanding DCBX Application Protocol TLV Exchange on EX Series Switches*

interface (Fibre Channel Fabric)

Syntax

```
interface {
  interface-name {
    max-login-sessions max-login-sessions;
  }
  interface-name {
    max-login-sessions max-login-sessions;
  }
  <...> {
    max-login-sessions max-login-sessions;
  }
  vlan.interface-name;
}
```

Hierarchy Level [edit fc-fabrics *fc-fabric-name*]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Associate one or more native Fibre Channel (FC) interfaces with an FC fabric and one VLAN interface for FCoE traffic. An FC interface can be associated with only one FC fabric.

Options ***interface-name***—Name of the native FC interface. You can assign one or more FC interfaces to an FC fabric.

vlan.*vlan-interface-name*—Name of the VLAN interface for FCoE traffic. You can assign one VLAN interface to an FC fabric.

The remaining statement is explained separately.

Required Privilege Level storage—To view this statement in the configuration.
storage-control—To add this statement to the configuration.

Related Documentation

- Example: Setting Up Fibre Channel and FCoE VLAN Interfaces in an FCoE-FC Gateway Fabric on page 171
- Configuring a Fibre Channel Interface on page 269
- Configuring a Physical Fibre Channel Interface on page 268
- Configuring an FCoE VLAN Interface on an FCoE-FC Gateway on page 272
- Understanding Interfaces on an FCoE-FC Gateway on page 50

interface (FIP)

Syntax

```
interface {
  interface-name {
    fka-adv-period milliseconds;
    priority priority;
  }
}
```

Hierarchy Level [edit fc-fabrics *fc-fabric-name* protocols fip]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure FIP options on a per-interface basis. (Override global FIP configuration for a specified interface.)

Options ***interface-name***—Name of the interface.

The statements are explained separately.

Required Privilege Level storage—To view this statement in the configuration.
storage-control—To add this statement to the configuration.

Related Documentation

- **show fibre-channel fip on page 443**
- Configuring FIP on an FCoE-FC Gateway on page 283
- Overview of FIP on page 9

load-balance-algorithm

Syntax load-balance-algorithm (simple | enode-based | flogi-based);

Hierarchy Level [edit fc-fabrics *fabric-name* proxy]

Release Information Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Set the load-balancing algorithm that the QFX Series uses to distribute FCoE sessions (FLOGI and FDISC sessions from the FCoE devices in the Ethernet network) among the NP_Port links to the FC switch.

NOTE: **Changing the load-balancing algorithm when FCoE sessions are running forces the FCoE sessions to log out, then log in again.**

Options **simple**—Load balancing is based on the weighted utilization (load) of the NP_Ports connected to an FC fabric. Each new FLOGI or FDISC is assigned to the least-loaded link. When a link load rebalance occurs, the system minimizes disruption by using an algorithm to log out only the sessions that need to be moved to other links to balance the link load. To further minimize disruption, the algorithm logs out the sessions with the fewest dependencies (for example, FDISC sessions are logged out before FLOGI sessions). When the sessions log in again, they are placed on NP_Port interfaces in a manner that balances the link loads. This is the default load-balancing algorithm.

enode-based—Load balancing is based on the ENode FLOGI. When an ENode logs in to the fabric, all subsequent FDISC sessions (VN_Port sessions) associated with that ENode are placed on the same link as the ENode FLOGI session, regardless of the link load. New ENode FLOGIs are placed on the least-loaded link. When a link load rebalance occurs, the system logs off all sessions. The sessions log in again and are placed on NP_Port interfaces in a balanced manner.

flogi-based—FLOGI-based load balancing is similar to ENode-based load balancing, but the behavior when the loads are rebalanced is different. Load balancing is based on the ENode FLOGI. When an ENode logs in to the fabric, all subsequent FDISC sessions associated with that ENode are placed on the same link as the ENode FLOGI session, regardless of the link load. New ENode FLOGIs are placed on the least-loaded link. When a link load rebalance occurs, the system minimizes disruption by using an algorithm to log out only the sessions that need to be moved to other links to balance the link load. When the logged out sessions log back in, they are placed on NP_Port interfaces in a manner that balances the link loads.

Required Privilege Level storage—To view this statement in the configuration.
storage-control—To add this statement to the configuration.

Related Documentation

- Defining the Proxy Load-Balancing Algorithm on page 279
- Example: Configuring Automated Fibre Channel Interface Load Rebalancing on page 250

- Simulating On-Demand Fibre Channel Link Load Rebalancing (Dry Run Test) on page 280
- Monitoring Fibre Channel Interface Load Balancing on page 383
- Understanding Load Balancing in an FCoE-FC Gateway Proxy Fabric on page 63

loopback (Fibre Channel Interface)

Syntax (loopback | no-loopback);

Hierarchy Level [edit interfaces *interface-name* fibrechannel-options]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Enable or disable loopback mode for FC interfaces.

Default By default, loopback mode is disabled on FC interfaces.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- **show fibre-channel interfaces on page 472**
- Configuring a Fibre Channel Interface on page 269

max-login-sessions

Syntax
max-login-sessions *max-login-sessions*;

Hierarchy Level
[edit fc-fabrics *fc-fabric-name*];
[edit fc-fabrics *fc-fabric-name* interface *interface-name*];

Release Information
Statement introduced in Junos OS Release 12.2 for the QFX Series.

Description
Set the maximum number of FCoE initialization protocol (FIP) session logins permitted for an individual NP_Port interface in an FCoE-FC gateway fabric (FC fabric) or for the entire FCoE-FC gateway fabric. You can set a maximum FIP session limit for each NP_Port interface connected to an FC switch. You can also set a maximum FIP session limit for the entire FC fabric. The sum of the maximum login sessions permitted on the NP_Port interfaces in an FC fabric should not exceed the maximum login sessions configured for that FC fabric.

The maximum number of FIP sessions (the combined total of all VN2VF_Port and VN2VN_Port sessions on the system) is 2500 sessions.

Options
max-login-sessions—Maximum number of FIP login sessions.

Range: 128 through 2500

Default: 2500

Required Privilege Level
storage—To view this statement in the configuration.
storage-control—To add this statement to the configuration.

Related Documentation

- **max-login-sessions-per-node on page 354**
- Setting the Maximum Number of FIP Login Sessions per FC Interface on page 286
- Setting the Maximum Number of FIP Login Sessions per FC Fabric on page 287
- Understanding Interfaces on an FCoE-FC Gateway on page 50

max-login-sessions-per-node

Syntax max-login-sessions-per-node *max-login-sessions-per-node*;

Hierarchy Level [edit fc-options]

Release Information Statement introduced in Junos OS Release 12.2 for the QFX Series.

Description Set the maximum number of FCoE initialization protocol (FIP) session logins permitted on a Node device. (This is the combined total of all VN2VF_Port and VN2VN_Port sessions on the Node device.)

On a QFX3500 switch, the **max-login-sessions-per-node** command sets the maximum FIP session login limit for all of the FC fabrics configured on the device. The combined number of FIP sessions on all FC fabrics on the device should not exceed this limit.

On a QFabric system, the **max-login-sessions-per-node** command globally sets the maximum FIP session login limit for each QFabric system Node device in the QFabric system. For example, if you set the Node limit to 2000 login sessions, then each QFabric Node device supports up to 2000 FIP login sessions. The total configured maximum number of login sessions of all of the FC fabrics on a Node device should not exceed the Node session limit.

NOTE: FIP login session limits configured at the FC fabric level or at the FC fabric interface level might limit a Node device to fewer total sessions than the configured Node limit.

Options *max-login-sessions-per-node*—Maximum number of FIP login sessions.

Range: 128 through 2500

Default: 2500

Required Privilege Level storage—To view this statement in the configuration.
storage-control—To add this statement to the configuration.

Related Documentation

- max-login-sessions on page 353
- Setting the Maximum Number of FIP Login Sessions per Node Device on page 288
- Understanding Interfaces on an FCoE-FC Gateway on page 50

max-sessions-per-enode

Syntax max-sessions-per-enode *max-sessions-per-enode*;

Hierarchy Level [edit fc-fabrics *fc-fabric-name* protocols fip]

Release Information Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Set the maximum number of FCoE login sessions (FLOGI plus FDISC) from a single ENode allowed on the gateway FC fabric (the fabric configured on the QFabric system). The maximum number of logins per ENode is 2000 sessions.

NOTE: A session is a FLOGI or FDISC login to the FC SAN fabric. Session does not refer to end-to-end storage sessions. There is no limit to the number of end-to-end storage sessions.

Options *max-sessions-per-enode*—Maximum number of FCoE sessions a single ENode can establish on the switch.

Range: 32 through 2000

Default: 32

Required Privilege Level storage—To view this statement in the configuration.
storage-control—To add this statement to the configuration.

Related Documentation

- **fcoe-trusted on page 343**
- **show fibre-channel fip on page 443**
- Configuring FIP on an FCoE-FC Gateway on page 283
- Understanding FIP Parameters on an FCoE-FC Gateway on page 46

no-fabric-wwn-verify

Syntax no-fabric-wwn-verfiy;

Hierarchy Level [edit fc-fabrics *fabric-name* proxy]

Release Information Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Disable the fabric worldwide name (WWN) verification check in the fabric login accept message (FLOGI-ACC) for implicit FLOGIs. If you enable this option, when a QFX Series NP_Port performs a FLOGI to the FC fabric, the QFX Series does not verify the fabric WWN in the FLOGI-ACC against the current fabric WWN.

NOTE: **Disabling or enabling the fabric WWN verification check logs out all FCoE sessions.**

Default Disabled. By default, all implicit FLOGIs from the QFX Series NP_Ports to the FC fabric are verified against the current fabric WWN.

Required Privilege Level storage—To view this statement in the configuration.
storage-control—To add this statement to the configuration.

Related Documentation

- **show fibre-channel proxy fabric-state on page 479**
- Understanding FCoE-FC Gateway Functions on page 33

no-fcoe-lag

Syntax no-fcoe-lag;

Hierarchy Level [edit interfaces *lag-interface-name* aggregated-ether-options]

Release Information Statement introduced in Junos OS Release 13.2X52-D10 for the QFX Series.

Description Convert an FCoE LAG into a standard LAG. When you convert an FCoE LAG into a standard LAG, the standard LAG no longer works reliably for FCoE traffic. This is because FCoE traffic must use the same physical link within a LAG interface for communication between the FCoE device and the Fibre Channel SAN across a QFabric system Node device. A standard LAG uses a hashing algorithm to determine the link to use for each transmission, so there is no way to guarantee that a response will use the same link on which a device receives a request. An FCoE LAG guarantees that the same physical LAG link is used for communication between an FCoE device and the QFabric system Node device.

If you convert an FCoE LAG into a standard LAG, do not use the standard LAG for FCoE traffic.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- **fcoe-lag on page 342**
- **no-fip-snooping-scaling on page 358**
- Configuring an FCoE LAG on page 302
- Configuring VLANs for FCoE Traffic on an FCoE Transit Switch on page 289
- Configuring VN2VF_Port FIP Snooping and FCoE Trusted Interfaces on an FCoE Transit Switch on page 294
- Understanding FCoE LAGs on page 158

no-fip-snooping-scaling

Syntax no-fip-snooping-scaling

Hierarchy Level (FCoE-FC gateway) [edit fc-options]

Hierarchy Level (FCoE Transit Switch) [edit ethernet-switching-options secure-access-port vlan (all | *vlan-name*) examine-fip]

Release Information Statement introduced in Junos OS Release 13.2X52-D10 for the QFX Series.

Description Disable FIP snooping scaling on all FCoE VLANs on an FCoE-FC gateway, or disable FIP snooping scaling on the specified FCoE VLAN on an FCoE transit switch.

Disabling FIP snooping scaling reduces the maximum number of FIP snooping sessions from 2,500 sessions (the maximum with FIP snooping scaling enabled) to 376 sessions. FIP snooping scaling is enabled by default.

Use this statement to disable FIP snooping scaling if you want to configure an FCoE LAG on an FCoE-FC gateway that contains one or more untrusted FC fabrics. Untrusted FC fabrics do not support FIP snooping scaling.

On an FCoE transit switch, you can use this statement to disable FIP snooping scaling on a specified FCoE VLAN.

Default FIP snooping scaling is enabled by default.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- **fcoe-lag on page 342**
- Configuring an FCoE LAG on page 302
- Configuring VN2VF_Port FIP Snooping and FCoE Trusted Interfaces on an FCoE Transit Switch on page 294
- Configuring VLANs for FCoE Traffic on an FCoE Transit Switch on page 289
- Understanding FCoE LAGs on page 158

no-recommendation-tlv

Syntax no-recommendation-tlv;

Hierarchy Level [edit protocols dcbx interface *interface-name* enhanced-transmission-selection]

Release Information Statement introduced in Junos OS Release 12.2 for the QFX Series.

Description Disable DCBX to send the ETS Recommendation TLV (also known as the Information TLV) on egress. This feature is valid only if the interface DCBX mode is IEEE DCBX. If the interface DCBX mode is DCBX version 1.01, this statement has no effect. (DCBX version 1.01 does not advertise separate TLVs for individual attributes.)

Default DCBX-enabled interfaces send the ETS recommendation TLV unless it is disabled.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- **show dcbx neighbors on page 415**
- Configuring DCBX Autonegotiation on page 306

node-group (OxID Hash Control)

Syntax

```
node-group (node-group-name | all) {
   oxid (enable | disable);
}
```

Hierarchy Level

[edit forwarding-options hash-key family fcoe ethernet-interfaces]
[edit forwarding-options hash-key family fcoe fabric-interfaces]

Release Information Statement introduced in Junos OS Release 13.2X52-D10 for the QFX Series.

Description Specify a QFabric system Node group on which to enable or disable OxID hash control. OxID hash control is enabled or disabled on the fabric ports or on the Ethernet (FCoE) LAG ports that face an FCoE forwarder (FCF).

Options ***node-group***—Name of the Node group on which you want to enable OxID hash control.

all—All Node groups on the QFabric system (OxID hash control will be enabled or disabled on all Node groups).

The remaining statement is explained separately.

Required Privilege Level

interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Enabling and Disabling CoS OxID Hash Control on QFabric Systems on page 282
- Understanding OxID Hash Control for FCoE Traffic Load Balancing on QFabric Systems on page 163

oxid

Syntax oxid (enable | disable)

Hierarchy Level QFX Series Standalone Switches

[edit forwarding-options hash-key family fcoe]

QFabric Systems

[edit forwarding-options hash-key family fcoe ethernet-interfaces node-group (*node-group-name* | all) {]
[edit forwarding-options hash-key family fcoe fabric-interfaces node-group (*node-group-name* | all) {]

Release Information Statement introduced in Junos OS Release 12.3 for the QFX Series.
Statement introduced in Junos OS Release 13.2X52-D10 for the QFabric System.

Description Enable or disable whether the switch uses the originator exchange identifier (OxID) field for hash control for FCoE traffic load balancing.

Default OxID hash control is enabled by default.

Options **oxid (enable | disable)**—Enable or disable whether the switch uses the OxID hash control field for FCoE traffic load balancing.

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Enabling and Disabling CoS OxID Hash Control on QFX Series Standalone Switches on page 281
- Enabling and Disabling CoS OxID Hash Control on QFabric Systems on page 282
- Understanding OxID Hash Control for FCoE Traffic Load Balancing on QFX Series Standalone Switches on page 79
- Understanding OxID Hash Control for FCoE Traffic Load Balancing on QFabric Systems on page 163

policy-options

Syntax

```
policy-options
  application-maps application-map-name {
    application application-name {
      code-points [ aliases ] [ bit-patterns ];
    }
  }
  policy-statement policy-name {
    term term-name {
      from {
        family family-name;
        match-conditions;
        policy subroutine-policy-name;
        prefix-list prefix-list-name;
        prefix-list-filter prefix-list-name match-type <actions>;
        route-filter destination-prefix match-type <actions>;
        source-address-filter source-prefix match-type <actions>;
      }
      to {
        match-conditions;
        policy subroutine-policy-name;
      }
      then actions;
    }
  }
```

Hierarchy Level [edit]

Release Information Statement introduced in Junos OS Release 12.1 for the QFX Series.
Statement introduced in Junos OS Release 12.1 for the EX Series.

Description Configure options such as application maps for DCBX application protocol exchange and policy statements.

Required Privilege Level storage—To view this statement in the configuration.
storage-control—To add this statement to the configuration.

Related Documentation

- Defining an Application for DCBX Application Protocol TLV Exchange on page 309
- Example: Configuring DCBX Application Protocol TLV Exchange on page 217
- *Example: Configuring DCBX to Support an iSCSI Application*
- Understanding DCBX Application Protocol TLV Exchange on page 117
- *Understanding DCBX Application Protocol TLV Exchange on EX Series Switches*

port-mode (Fibre Channel Interfaces)

Syntax port-mode (f-port | np-port);

Hierarchy Level [edit interfaces vlan unit *unit* family fibre-channel],
[edit interfaces *interface-name* unit *logical-unit-number* family fibre-channel]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure the FCoE VLAN interface port mode to F_Port to connect the switch to FCoE initiators, or configure the native FC interface port mode to proxy N_Port (NP_Port) to connect the switch to an FC switch fabric port (F_Port).

Options **f-port**—Configure an FCoE VLAN interface to connect to FCoE initiator Virtual N_Ports (VN_Ports).

np-port—Configure a native FC port to connect to an FC switch F_Port.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- **show fibre-channel interfaces on page 472**
- Example: Setting Up Fibre Channel and FCoE VLAN Interfaces in an FCoE-FC Gateway Fabric on page 171
- Configuring an FCoE VLAN Interface on an FCoE-FC Gateway on page 272
- Configuring a Fibre Channel Interface on page 269
- Understanding Interfaces on an FCoE-FC Gateway on page 50

port-range

Syntax port-range *port-range-low port-range-high*;

Hierarchy Level [edit chassis fpc *fpc-id* pic *pic-id* fibre-channel]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure a contiguous block of ports as FC ports. You can configure the FC-capable ports xe-0/0/0 through xe-0/0/5 as fc-0/0/0 through fc-0/0/5, and ports xe-0/0/42 through xe-0/0/47 as fc-0/0/42 through fc-0/0/47 to create blocks of native FC interfaces. You cannot individually configure a single port as a native FC interface. Within these port blocks, you cannot mix FC interfaces with Ethernet interfaces. All of the ports in a block must be either native FC interfaces or Ethernet interfaces.

You can configure:

- Six native FC interfaces by configuring either ports xe-0/0/0 through xe-0/0/5 as fc-0/0/0 through fc-0/0/5, or ports xe-0/0/42 through xe-0/0/47 as fc-0/0/42 through fc-0/0/47.
- Twelve native FC interfaces by configuring ports xe-0/0/0 through xe-0/0/5 as fc-0/0/0 through fc-0/0/5 and ports xe-0/0/42 through xe-0/0/47 as fc-0/0/42 through fc-0/0/47.
- No native FC interfaces by leaving ports xe-0/0/0 through xe-0/0/5 and ports xe-0/0/42 through xe-0/0/47 in their default state as Ethernet interfaces.

Options ***port-range-low***—Lowest-numbered port in the block of native FC interfaces, either **0** or **42**.

port-range-high—Highest-numbered port in the block of native FC interfaces. The value is **5** if the ***port-range-low*** value is **0**. The value is **47** if the ***port-range-low*** value is **42**.

NOTE: **Only a complete block of ports, xe-0/0/0 through xe-0/0/5, xe-0/0/42 through xe0/0/47, or both, can be configured as FC ports.**

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- **show fibre-channel interfaces on page 472**
- Example: Setting Up Fibre Channel and FCoE VLAN Interfaces in an FCoE-FC Gateway Fabric on page 171
- Configuring a Physical Fibre Channel Interface on page 268
- Understanding Interfaces on an FCoE-FC Gateway on page 50

priority (FIP)

Syntax priority *priority*;

Hierarchy Level [edit fc-fabrics *fc-fabric-name* protocols fip],
[edit fc-fabrics *fc-fabric-name* protocols fip interface *interface-name*]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Sets the global or interface-specific priority value associated with the switch FCF-MAC. CNAs use the priority value to determine the switch with which they will perform FIP FLOGI. The lower the value, the higher the priority. The switch advertises this value to the server ENodes on the FCoE network. A priority value set at the interface level overrides the global setting.

Options *priority* —Value that determines the FCF an ENode selects to perform FIP FLOGI. The lower the priority number, the higher the priority of the FCF.

Range: 0 through 255

Default: 128

Required Privilege Level storage—To view this statement in the configuration.
storage-control—To add this statement to the configuration.

Related Documentation

- show fibre-channel fip on page 443
- show fibre-channel fip interface on page 458
- Overview of FIP on page 9
- Configuring FIP on an FCoE-FC Gateway on page 283

priority-flow-control

Syntax

```
priority-flow-control {
  no-auto-negotiation;
}
```

Hierarchy Level [edit protocols dcbx interface (all | *interface-name*)]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.
Statement introduced in Junos OS Release 11.3 for EX Series switches.

Description Disable autonegotiation of priority-based flow control (PFC) on one or more Ethernet interfaces. Autonegotiation enables PFC on an interface only if the switch and the peer device connected to the switch both support PFC and have the same PFC configuration. Disabling autonegotiation on an interface forces the interface to use the PFC state (enabled or disabled) that is configured on the switch by the configuration and assignment of the congestion notification profile.

Options **no-auto-negotiation**—Disable automatic negotiation of PFC.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- **show dcbx neighbors on page 415**
- *Configuring CoS PFC (Congestion Notification Profiles)*
- *Configuring Priority-Based Flow Control for an EX Series Switch (CLI Procedure)*
- Configuring DCBX Autonegotiation on page 306
- Example: Configuring CoS PFC for FCoE Traffic on page 186
- *Understanding Data Center Bridging Capability Exchange Protocol for EX Series Switches*
- *Understanding Priority-Based Flow Control*
- Understanding DCB Features and Requirements on page 16

protocol (Applications)

Syntax protocol (tcp | udp);

Hierarchy Level [edit applications application *application-name*]

Release Information Statement introduced in Junos OS Release 12.1 for EX Series switches.
Statement introduced in Junos OS Release 12.1 for the QFX Series.

Description Networking protocol type, which combines with **destination-port** to identify an application type.

NOTE: To create an application for iSCSI, use the protocol **tcp** with the destination port number **3260**.

Options **tcp**—Transmission Control Protocol

udp—User Datagram Protocol

Required Privilege Level interface—To view this statement in the configuration.
interface-control—To add this statement to the configuration.

Related Documentation

- Defining an Application for DCBX Application Protocol TLV Exchange on page 309
- Example: Configuring DCBX Application Protocol TLV Exchange on page 217
- *Example: Configuring DCBX to Support an iSCSI Application*
- Understanding DCBX Application Protocol TLV Exchange on page 117
- *Understanding DCBX Application Protocol TLV Exchange on EX Series Switches*

protocols (FIP)

Syntax

```
protocols {
  fip {
    fcoe-trusted;
    fc-map fc-map-value;
    fka-adv-period milliseconds;
    interface {
      interface-name {
        fka-adv-period milliseconds;
        priority priority;
      }
    }
    max-sessions-per-enode max-sessions-per-enode;
    priority priority;
    traceoptions {
      file filename <replace> <size size> <files number> <no-stamp>;
      <world-readable | no-world-readable>;
      flag flag <flag-modifier> <disable>;
    }
  }
}
```

Hierarchy Level [edit fc-fabrics *fc-fabric-name*]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure global or interface-specific FC protocol options. Individual interface settings override global settings.

Options The statements are explained separately.

Required Privilege Level storage—To view this statement in the configuration.
storage-control—To add this statement to the configuration.

Related Documentation

- **show fibre-channel fip on page 443**
- Configuring FIP on an FCoE-FC Gateway on page 283
- Overview of FIP on page 9

proxy (Fibre Channel)

Syntax

```
proxy {
  auto-load-rebalance
  load-balance-algorithm (simple | enode-based | flog-based);
  no-fabric-wwn-verify;
  traceoptions {
    file filename <replace> <size size> <files number> <no-stamp>
    <world-readable | no-world-readable>;
    flag flag <flag-modifier> <disable>;
  }
}
```

Hierarchy Level [edit fc-fabrics *fabric-name*]

Description Configure proxy fabric operations.

Options The statement is explained separately.

Required Privilege Level storage—To view this statement in the configuration.
storage-control—To add this statement to the configuration.

Related Documentation

- Understanding FCoE-FC Gateway Functions on page 33

recommendation-tlv

Syntax

```
recommendation-tlv {
  no-auto-negotiation;
}
```

Hierarchy Level [edit protocols dcbx interface *interface-name* enhanced-transmission-selection]

Release Information Statement introduced in Junos OS Release 12.2 for the QFX Series.

Description Enable DCBX to send the ETS Recommendation TLV (also known as the Information TLV) on egress. This feature is valid only if the interface DCBX mode is IEEE DCBX. If the interface DCBX mode is DCBX version 1.01, this statement has no effect. (DCBX version 1.01 does not advertise separate TLVs for individual attributes.)

Default DCBX-enabled interfaces send the ETS recommendation TLV unless it is disabled.

Options **no-auto-negotiation**—Disable sending of the ETS recommendation TLV.

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- show dcbx neighbors on page 415
- Configuring DCBX Autonegotiation on page 306

speed (Fibre Channel Interfaces)

Syntax speed (auto-negotiation | 2g | 4g | 8g);

Hierarchy Level [edit interfaces *interface-name* fibrechannel-options]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Configure FC interface speed.

Options **auto-negotiation**—Automatically negotiate interface speed to match the speed of the attached link (2 Gbps, 4 Gbps, 8 Gbps).

2g—2 Gbps link speed

4g—4 Gbps link speed

8g—8 Gbps link speed

Default: auto-negotiation

Required Privilege Level routing—To view this statement in the configuration.
routing-control—To add this statement to the configuration.

Related Documentation

- **show fibre-channel interfaces on page 472**
- Example: Setting Up Fibre Channel and FCoE VLAN Interfaces in an FCoE-FC Gateway Fabric on page 171
- Configuring a Fibre Channel Interface on page 269
- Configuring a Physical Fibre Channel Interface on page 268
- Understanding Interfaces on an FCoE-FC Gateway on page 50

traceoptions (FC-2 Fibre Channel)

Syntax

traceoptions {
  file *filename* <replace> <size *size*> <files *number*> <no-stamp>;
  <world-readable | no-world-readable>;
  flag *flag* <*flag-modifier*>;
}

Hierarchy Level [edit fc-fabrics *fabric-name* fc2]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Set FC-2 protocol tracing options.

NOTE: **The traceoptions statement is not supported on the QFabric system.**

Default Traceoptions is disabled.

Options

file *name*—Name of the file to receive the tracing operation output. Enclose the name in quotation marks. Traceoption output files are located in the **/var/log/** directory.

files ***number***—(Optional) Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file*.0**. The traceoption output continues in a second trace file named ***trace-file*.1**. When ***trace-file*.1** reaches its maximum size, output continues in a third file named ***trace-file*.2**, and so on. When the maximum number of trace files is reached, the oldest trace file is overwritten.

If you specify a maximum number of files, you must also specify a maximum file size with the **size** option.

Range: 2 through 1000 files

Default: 1 trace file

flag—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements:

- **all** —Trace all operations.
- **error**—Trace all error events
- **normal**—Trace all normal events.

Default: If you do not specify the **normal** option, only unusual or abnormal operations are traced.

- **parse**—Trace configuration parsing.
- **rx-frame**—(Optional) Trace received frames.
- **rx-frame-header**—(Optional) Trace received frame headers.

- **tx-frame**—(Optional) Trace transmitted frames.
- **tx-frame-header**—(Optional) Trace transmitted frame headers.

no-stamp—(Optional) Do not place timestamp information at the beginning of each line in the trace file.

Default: If you omit this option, timestamp information is placed at the beginning of each line of the tracing output.

no-world-readable—(Optional) Prevent any user from reading the log file.

replace—(Optional) Replace an existing trace file if there is one.

Default: If you do not include this option, tracing output is appended to an existing trace file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file* .0**. Incoming tracefile data is logged in the now empty ***trace-file***. When ***trace-file*** again reaches its maximum size, ***trace-file*.0** is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you must also specify a maximum number of trace files with the **files** option.

Syntax: ***x*k** to specify KB, ***x*m** to specify MB, or ***x*g** to specify GB

Range: 10 KB through the maximum file size of 4 GB (the maximum is lower if 4 GB is not supported on your system)

Default: 1 MB

world-readable—(Optional) Allow any user to read the log file.

Required Privilege Level

storage—To view this statement in the configuration.
storage-control—To add this statement to the configuration.

traceoptions (Fibre Channel)

Syntax

```
traceoptions {
  file filename <replace> <size size> <files number> <no-stamp>
  <world-readable | no-world-readable>;
  flag flag <flag-modifier>;
}
```

Hierarchy Level [edit fc-options]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Set FC protocol tracing options.

NOTE: **The traceoptions statement is not supported on the QFabric system.**

Default Traceoptions is disabled.

Options **file** ***name***—Name of the file to receive the tracing operation output. Enclose the name in quotation marks. Traceoption output files are located in the **/var/log/** directory.

files ***number***—(Optional) Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file*.0**. The traceoption output continues in a second trace file named ***trace-file*.1**. When ***trace-file*.1** reaches its maximum size, output continues in a third file named ***trace-file*.2**, and so on. When the maximum number of trace files is reached, the oldest trace file is overwritten.

If you specify a maximum number of files, you must also specify a maximum file size with the **size** option.

Range: 2 through 1000 files

Default: 1 trace file

flag—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements:

- **all**—Trace all operations.
- **fabric**—Trace virtual fabric events.
- **fc2**—Trace the FC2 (network layer protocols) events.
- **fip**—Trace the Fibre Channel over Ethernet (FCoE) Initialization Protocol events.
- **flogi** —Trace the fabric login server events.
- **forwarding-database**—Trace the forwarding database and next-hop events.
- **interface**—Trace the interface events.

- **krt**—Trace the communication over the routing socket.
- **lib**—Trace library calls.
- **lif**—Trace Fibre Channel logical interface (fc-lif) events.
- **vswitch**—Trace virtual switch events.

The following are the global tracing options:

- **all**—All trace operations.
- **config-internal**—Trace configuration internals.
- **general**—Trace general events.
- **normal**—All normal events.

Default: If you do not specify this option, only unusual or abnormal operations are traced.

- **parse**—Trace configuration parsing.
- **state**—Trace state transitions.
- **task**—Trace protocol task processing.
- **timer**—Trace protocol task timer processing.

no-stamp—(Optional) Do not place timestamp information at the beginning of each line in the trace file.

Default: If you omit this option, timestamp information is placed at the beginning of each line of the tracing output.

no-world-readable—(Optional) Prevent any user from reading the log file.

replace—(Optional) Replace an existing trace file if there is one.

Default: If you do not include this option, tracing output is appended to an existing trace file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file* .0.** Incoming tracefile data is logged in the now empty ***trace-file***. When ***trace-file*** again reaches its maximum size, ***trace-file*.0** is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you must also specify a maximum number of trace files with the **files** option.

Syntax: ***x*k** to specify KB, ***x*m** to specify MB, or ***x*g** to specify GB

Range: 10 KB through the maximum file size of 4 GB (maximum is lower if 4 GB is not supported on your system)

Default: 1 MB

world-readable—(Optional) Allow any user to read the log file.

Required Privilege Level
storage—To view this statement in the configuration.
storage-control—To add this statement to the configuration.

traceoptions (FIP Protocol Fibre Channel)

Syntax

traceoptions {
  file *filename* <replace> <size *size*> <files *number*> <no-stamp>
  <world-readable | no-world-readable>;
  flag *flag* <*flag-modifier*>
}

Hierarchy Level

[edit fc-fabrics *fabric-name* protocols fip]

Release Information

Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description

Set proxy FC protocol tracing options.

NOTE: **The traceoptions statement is not supported on the QFabric system.**

Default

Traceoptions is disabled.

Options

file ***name***—Name of the file to receive the tracing operation output. Enclose the name in quotation marks. Traceoption output files are located in the **/var/log/** directory.

files ***number*** —(Optional) Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file*.0**. The traceoption output continues in a second trace file named ***trace-file*.1**. When ***trace-file*.1** reaches its maximum size, output continues in a third file named ***trace-file*.2**, and so on. When the maximum number of trace files is reached, the oldest trace file is overwritten.

If you specify a maximum number of files, you must also specify a maximum file size with the size option.

Range: 2 through 1000 files

Default: 1 trace file

flag—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements:

- **all**—Trace all operations.
- **error**—Trace all error events
- **normal**—Trace all normal events.

Default: If you do not specify this option, only unusual or abnormal operations are traced.

- **packet**—Trace packet decoding operations
- **parse**—Trace configuration parsing.
- **state**—Trace state transitions.

no-stamp—(Optional) Do not place timestamp information at the beginning of each line in the trace file.

Default: If you omit this option, timestamp information is placed at the beginning of each line of the tracing output.

no-world-readable—(Optional) Prevent any user from reading the log file.

replace—(Optional) Replace an existing trace file if there is one.

Default: If you do not include this option, tracing output is appended to an existing trace file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file*.0.** Incoming tracefile data is logged in the now empty ***trace-file***. When ***trace-file*** again reaches its maximum size, ***trace-file*.0** is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you must also specify a maximum number of trace files with the **files** option.

Syntax: ***x*k** to specify KB, ***x*m** to specify MB, or ***x*g** to specify GB

Range: 10 KB through the maximum file size of 4 GB (maximum is lower if 4 GB is not supported on your system)

Default: 1 MB

world-readable—(Optional) Allow any user to read the log file.

Required Privilege Level

storage—To view this statement in the configuration.
storage-control—To add this statement to the configuration.

traceoptions (Proxy Fibre Channel)

Syntax

traceoptions {
  file *filename* <replace> <size *size*> <files *number*> <no-stamp>
  <world-readable | no-world-readable>;
  flag *flag* <*flag-modifier*>
}

Hierarchy Level [edit fc-fabrics *fabric-name* proxy]

Release Information Statement introduced in Junos OS Release 11.1 for the QFX Series.

Description Set proxy FC protocol tracing options.

NOTE: **The traceoptions statement is not supported on the QFabric system.**

Default Traceoptions is disabled.

Options

file ***name***—Name of the file to receive the tracing operation output. Enclose the name in quotation marks. Traceoption output files are located in the **/var/log/** directory.

files ***number***—(Optional) Maximum number of trace files. When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file*.0**. The traceoption output continues in a second trace file named ***trace-file*.1**. When ***trace-file*.1** reaches its maximum size, output continues in a third file named ***trace-file*.2**, and so on. When the maximum number of trace files is reached, the oldest trace file is overwritten.

If you specify a maximum number of files, you must also specify a maximum file size with the **size** option.

Range: 2 through 1000 files

Default: 1 trace file

flag—Tracing operation to perform. To specify more than one tracing operation, include multiple **flag** statements:

- **all**—Trace all operations.
- **error**—Trace all error events.
- **interface**—Trace the interface events.
- **normal**—Trace all normal events.

Default: If you do not specify this option, only unusual or abnormal operations are traced.

- **packet**—Trace packet decoding operations
- **parse**—Trace configuration parsing.

- **state**—Trace state transitions.

no-stamp—(Optional) Do not place timestamp information at the beginning of each line in the trace file.

Default: If you omit this option, timestamp information is placed at the beginning of each line of the tracing output.

no-world-readable—(Optional) Prevent any user from reading the log file.

replace—(Optional) Replace an existing trace file if there is one.

Default: If you do not include this option, tracing output is appended to an existing trace file.

size *size*—(Optional) Maximum size of each trace file, in kilobytes (KB), megabytes (MB), or gigabytes (GB). When a trace file named ***trace-file*** reaches its maximum size, it is renamed ***trace-file*.0**. Incoming tracefile data is logged in the now empty ***trace-file***. When ***trace-file*** again reaches its maximum size, ***trace-file*.0** is renamed ***trace-file*.1** and ***trace-file*** is renamed ***trace-file*.0**. This renaming scheme continues until the maximum number of trace files is reached. Then the oldest trace file is overwritten.

If you specify a maximum file size, you must also specify a maximum number of trace files with the **files** option.

Syntax: ***x*k** to specify KB, ***x*m** to specify MB, or ***x*g** to specify GB

Range: 10 KB through the maximum file size of 4 GB (maximum is lower if 4 GB is not supported on your system)

Default: 1 MB

world-readable—(Optional) Allow any user to read the log file.

Required Privilege Level

storage—To view this statement in the configuration.
storage-control—To add this statement to the configuration.

PART 3

Administration

CHAPTER 8

Routine Monitoring

Monitoring Fibre Channel Interface Load Balancing

You can use operational mode commands to monitor load balancing when the switch is in FCoE-FC gateway mode:

Monitoring the Interface Load-Balancing State

Purpose Monitor the number of sessions, whether load balancing is enabled or disabled, and the load-balancing weight for each native Fibre Channel (FC) interface.

NOTE: A session is a FLOGI or FDISC login to the FC SAN fabric. Session does not refer to end-to-end storage sessions.

Action To monitor the load-balancing state of the native FC interfaces in the CLI, enter the following CLI command:

user@switch> **show fibre-channel proxy np-port**

For example:

```
user@switch> show fibre-channel proxy np-port
Fabric: sanfab1, Fabric-id: 10
NP-Port      State              Sessions       LB state   LB weight
fc-0/0/0.0  online             5              ON         4
fc-0/0/1.0  online             5              ON         4
fc-0/0/2.0  online             10             ON         8

Fabric: fc_fab2, Fabric-id: 200
NP-Port      State              Sessions       LB state   LB weight
fc-0/0/44.0 isolated           0              OFF        0

Fabric: fc_fabric_100, Fabric-id: 100
NP-Port      State              Sessions       LB state   LB weight
fc-0/0/46.0 online             1              ON         8
```

Meaning Table 32 on page 384 summarizes key output fields for the FC interface load-balancing state.

Table 32: Summary of Key FC Interface Load-Balancing Output Fields

Field	Values
Fabric	Name of the fabric.
Fabric-id	Fabric ID number.
NP-Port	NP_Port interface connected to the FCoE forwarder (FCF) or the FC switch.
State	FCID state of the NP_Port interface: • **online**—The port is online and connected to the FC switch. FCoE devices can log in to the FC switch using this port. • **isolated**—The port is isolated and is not part of the load-balancing function. FCoE devices cannot log in to the FC switch using this port. • **offline**—The port is offline.
Sessions	Number of active sessions on the NP_Port interface.
LB state	Load-balancing state: • **On**—Load balancing is on • **Off**—Load balancing is off.
LB weight	Load-balancing weight, which reflects the port speed: • **2**—Port speed is 2 Gbps. • **4**—Port speed is 4 Gbps. • **8**—Port speed is 8 Gbps.

The gateway determines the least-loaded interface using the following weighted round-robin (WRR) algorithm:

(number-of-sessions * max-weight) / weight

where *max-weight* is an internal constant. If the load on the FC interfaces is equal, the session is assigned to the interface with the highest link speed (the greatest weight).

Monitoring the Fabric Load-Balancing Algorithm

Purpose Monitor the type of load-balancing algorithm (simple, ENode-based, or FLOGI-based) used on the native FC interfaces, whether or not automated load rebalancing is enabled, and the load rebalancing state of the fabric.

Action To monitor the load-balancing algorithm used on the native FC interfaces and the load rebalancing state in the CLI, enter the following CLI command:

user@switch> **show fibre-channel proxy fabric-state**

For example:

```
user@switch> show fibre-channel proxy fabric-state
Fabric: sanfab1, Fabric-id: 10
Proxy load balance algorithm: Simple, Fabric WWN verification: Yes
Auto load rebalance enabled  : No
Last rebalance start-time    : Never
Last rebalance end-time      : Never
Last rebalance trigger       : Link-up
Last rebalance trigger-time  : Mon Sep 10 21:42:30 2012 usec: 814602
Last rebalance trigger-result: Not-configured

Fabric: fc_fab2, Fabric-id: 200
Proxy load balance algorithm: ENode based, Fabric WWN verification: Yes
Auto load rebalance enabled  : No
Last rebalance start-time    : Never
Last rebalance end-time      : Never
Last rebalance trigger       : Link-up
Last rebalance trigger-time  : Mon Sep 17 17:23:35 2012 usec: 619684
Last rebalance trigger-result: Not-configured

Fabric: fc_fabric_100, Fabric-id: 100
Proxy load balance algorithm: FLOGI based, Fabric WWN verification: No
Auto load rebalance enabled  : Yes
Last rebalance start-time    : Never
Last rebalance end-time      : Never
Last rebalance trigger       : Config-CLI
Last rebalance trigger-time  : Fri Nov  2 08:56:16 2012 usec: 004487
Last rebalance trigger-result: Not-required
```

Meaning

You can configure each local FC fabric on an FCoE-FC gateway to use one of three types of load-balancing algorithms, *simple*, *ENode-based*, or *FLOGI-based*. All of the native FC interfaces (NP_Ports) in a particular gateway FC fabric use the same load-balancing algorithm (the load-balancing algorithm is applied on a per-fabric basis).

Table 33 on page 385 summarizes key output fields for the FC interface load-balancing algorithm and state.

Table 33: show fibre-channel proxy fabric-state Output Fields

Field Name	Field Description
Fabric	Name of the fabric.
Fabric-id	Fabric ID number.

Table 33: show fibre-channel proxy fabric-state Output Fields *(continued)*

Field Name	Field Description
Proxy load balance algorithm	Load-balancing algorithm used on the FCoE-FC gateway FC fabric: • Simple—Load balancing is based on the weighted utilization (load) of the NP_Ports connected to an FC fabric. Each new FLOGI or FDISC is assigned to the least-loaded link. On a link load rebalance, only the sessions that need to be moved to another link are logged out. When those sessions log in again, they are placed on active NP_Port interfaces in a balanced manner. • ENode-based—Load balancing is based on the ENode FLOGI. When an ENode logs in to the fabric, all subsequent FDISC sessions (VN_Port sessions) associated with that ENode are placed on the same link as the ENode FLOGI session, regardless of the link load. New ENode FLOGIs are placed on the least-loaded link. On a link load rebalance, all sessions are logged out. When the sessions log in again, they are placed on active NP_Port interfaces in a balanced manner. • FLOGI-based—Load balancing is based on the ENode FLOGI. When an ENode logs in to the fabric, all subsequent FDISC sessions (VN_Port sessions) associated with that ENode are placed on the same link as the ENode FLOGI session, regardless of the link load. New ENode FLOGIs are placed on the least-loaded link. On a link load rebalance, only the sessions that need to be moved to another link are logged out. When those sessions log in again, they are placed on active NP_Port interfaces in a balanced manner.
Fabric WWN verification	Fabric worldwide name (WWN) verification check state on the FCoE-FC gateway fabric: • Yes—Fabric WWN verification check is enabled. • No—Fabric WWN verification check is disabled.
Auto load rebalance enabled	Automated link load rebalancing configuration for the FCoE-FC gateway fabric: • No—Automated load balancing is disabled (default state). • Yes—Automated load balancing is enabled.
Last rebalance start-time	Time that the last link load rebalance began on the FCoE-FC gateway fabric: • Never—The link load has never been rebalanced. • Timestamp value—Time the last link load rebalancing started.
Last rebalance end-time	Time that the last link load rebalance ended on the FCoE-FC gateway fabric: • Never—The link load has never been rebalanced. • Timestamp value—Time the last link load rebalancing ended.

Table 33: show fibre-channel proxy fabric-state Output Fields *(continued)*

Field Name	Field Description
Last rebalance trigger	Event that triggered the last link load rebalance on the FCoE-FC gateway fabric: • None—The link load has never been rebalanced. • Config-CLI—Configure (enable) automated load balancing. • Request-CLI—Rebalance requested from the CLI using the **request fibre-channel proxy load-rebalance fabric *fabric-name*** operational command. • Preview-CLI—Rebalancing *dry run* requested from the CLI using the **request fibre-channel proxy load-rebalance dry-run fabric *fabric-name*** operational command. Indicates that the switch completed the dry run. A dry run simulates a link load rebalance and displays a list of sessions that might be affected if you request an actual rebalance. • Link-up—New FC link (NP_Port) up on the FCoE-FC gateway fabric, which causes a rebalance to distribute sessions to the new link. • Restore-complete—If the FC process on the switch restarts, the switch attempts to restore the session state that existed before the restart. When automated rebalance is enabled, restore-complete indicates that the sessions have been restored and rebalanced.
Last rebalance trigger-time	Time that the last link load rebalance was triggered on the FCoE-FC gateway fabric: • Never—Link load rebalancing has never been triggered. • Timestamp value—Time the last link load rebalancing was triggered.
Last rebalance trigger-result	Result of the last trigger event on the FCoE-FC gateway fabric: • Never—Link load rebalancing has never been triggered. • Not-configured—Automated rebalancing is not configured on the FCoE-FC gateway fabric. • Not-required—Last rebalance trigger did not require rebalancing the link load (the link load was already balanced across the active NP_Port links). • In-progress—Link load rebalancing is in progress and has not finished yet. • Restore-in-progress—The switch is recovering from an FC process restart and is in the process of restoring the sessions to the active NP_Port links. • Success—Link load rebalancing was successful. • Logged-out-all—All sessions have been logged out. • Preview-complete—The switch has finished simulating a dry run rebalancing request from the CLI (**request fibre-channel proxy load-rebalance dry-run fabric *fabric-name*** operational command) and reported the sessions that might be affected if you request an actual link load rebalance. • Fabric-deletion-in-progress—FCoE-FC gateway fabric is in the process of being deleted. NOTE: A trigger event does not necessarily result in a rebalance action. Link load rebalancing only occurs if the NP_Port interface session load is not balanced at the time of the trigger event.

Related Documentation

- show fibre-channel proxy fabric-state on page 479
- show fibre-channel proxy np-port on page 486

- Configuring a Fibre Channel Interface on page 269
- Defining the Proxy Load-Balancing Algorithm on page 279
- Example: Configuring Automated Fibre Channel Interface Load Rebalancing on page 250
- Understanding Load Balancing in an FCoE-FC Gateway Proxy Fabric on page 63
- Understanding FCoE-FC Gateway Functions on page 33

CHAPTER 9

Operational Commands

- clear fibre-channel fc2 statistics
- clear fibre-channel fip enode
- clear fibre-channel fip statistics
- clear fibre-channel fip vn-port
- clear fibre-channel flogi statistics
- clear fibre-channel proxy statistics
- clear fip snooping enode
- clear fip snooping statistics
- clear fip snooping vlan
- clear fip vlan-discovery statistics
- request fibre-channel proxy load-rebalance
- restart
- show dcbx
- show dcbx neighbors
- show fibre-channel fabric
- show fibre-channel fc2 sessions
- show fibre-channel fc2 statistics
- show fibre-channel fip
- show fibre-channel fip enode
- show fibre-channel fip fabric
- show fibre-channel fip fcf
- show fibre-channel fip interface
- show fibre-channel fip statistics
- show fibre-channel flogi fport
- show fibre-channel flogi nport
- show fibre-channel flogi statistics
- show fibre-channel interfaces
- show fibre-channel next-hops

- show fibre-channel routes
- show fibre-channel proxy fabric-state
- show fibre-channel proxy login-table
- show fibre-channel proxy np-port
- show fibre-channel proxy statistics
- show fip snooping
- show fip snooping enode
- show fip snooping fcf
- show fip snooping interface
- show fip snooping statistics
- show fip snooping vlan
- show fip vlan-discovery
- show route forwarding-table family fibre-channel

clear fibre-channel fc2 statistics

Syntax

clear fibre-channel fc2 statistics
<fabric *fabric-name*>

Release Information

Command introduced in Junos OS Release 11.1 for the QFX Series.

Description

Clear FC-2 (network layer) Fibre Channel statistics globally or on a specified Fibre Channel fabric.

Options

fabric *fabric-name*—(Optional) Clear FC-2 statistics only on the specified fabric.

Required Privilege Level

view

Related Documentation

- show fibre-channel fc2 statistics on page 441
- show fibre-channel fc2 sessions on page 439

List of Sample Output

clear fibre-channel fc2 statistics on page 391

Sample Output

clear fibre-channel fc2 statistics

```
user@switch> clear fibre-channel fc2 statistics
```

clear fibre-channel fip enode

Syntax **clear fibre-channel fip enode *enode-mac***

Release Information Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Clear Fibre Channel over Ethernet (FCoE) node (ENode) information for a specified ENode. This operation deletes the ENode state from the switch database and from the FIP snooping firewall filters, which causes the ENode to lose the connection to the Fibre Channel (FC) fabric and to log in to the fabric again. If you clear an ENode, all VN_Ports associated with that ENode are also cleared and lose their connection to the FC fabric and must log in to the fabric again.

Options ***enode-mac***—MAC address of the ENode.

Required Privilege Level view

Related Documentation

- show fibre-channel fip enode on page 448
- clear fibre-channel fip statistics on page 393
- clear fibre-channel fip vn-port on page 394

List of Sample Output clear fibre-channel fip enode on page 392

Sample Output

clear fibre-channel fip enode

```
user@switch> clear fibre-channel fip enode 00:10:94:00:00:02
```


clear fibre-channel fip statistics

Syntax

clear fibre-channel fip statistics
<**fabric** *fabric-name*>

Release Information

Command introduced in Junos OS Release 11.1 for the QFX Series.

Description

Clear Fibre Channel over Ethernet (FCoE) initialization protocol (FIP) statistics.

Options

fabric ***fabric-name***—(Optional) Clear FIP statistics only on the specified fabric.

Required Privilege Level

view

Related Documentation

- show fibre-channel fip statistics on page 461
- show fibre-channel fip on page 443

List of Sample Output

clear fibre-channel fip statistics on page 393

Sample Output

clear fibre-channel fip statistics

```
user@switch> clear fibre-channel fip statistics
```

clear fibre-channel fip vn-port

Syntax **clear fibre-channel fip vn-port** ***vn-port--mac***

Release Information Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Clear virtual N_Port (VN_Port) information for a specified VN_Port. This operation deletes the VN_Port state from the switch database and from the FIP snooping firewall filters, which causes the VN_Port to lose its connection to the Fibre Channel fabric and to log in to the fabric again. When you clear a VN_Port, other VN_Ports associated with the same Fibre Channel over Ethernet (FCoE) Node (ENode) are not affected and are not cleared.

Options ***vn-port-mac***—MAC address of the VN_Port.

Required Privilege Level view

Related Documentation

- show fibre-channel fip enode on page 448
- clear fibre-channel fip enode on page 392
- clear fibre-channel fip statistics on page 393

List of Sample Output clear fibre-channel fip vn-port on page 394

Sample Output

clear fibre-channel fip vn-port

```
user@switch> clear fibre-channel fip vn-port 00:10:94:00:00:08
```


clear fibre-channel flogi statistics

Syntax **clear fibre-channel flogi statistics**
<**fabric** *fabric-name*>

Release Information Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Clear fabric login (FLOGI) statistics globally or on a specified Fibre Channel fabric.

Options **fabric** ***fabric-name***—(Optional) Clear FLOGI statistics only on the specified fabric.

Required Privilege Level view

Related Documentation

- show fibre-channel flogi statistics on page 469
- show fibre-channel flogi fport on page 465
- show fibre-channel flogi nport on page 467

List of Sample Output clear fibre-channel flogi statistics on page 395

Sample Output

clear fibre-channel flogi statistics

`user@switch>` **clear fibre-channel flogi statistics**

clear fibre-channel proxy statistics

Syntax

clear fibre-channel proxy statistics
<fabric *fabric-name*>

Release Information

Command introduced in Junos OS Release 11.1 for the QFX Series.

Description

Clear Fibre Channel gateway statistics globally or on a specified Fibre Channel fabric.

Options

fabric *fabric-name*—(Optional) Clear proxy statistics only on the specified fabric.

Required Privilege Level

view

Related Documentation

- show fibre-channel proxy statistics on page 489
- show fibre-channel proxy login-table on page 483
- show fibre-channel proxy np-port on page 486

List of Sample Output

clear fibre-channel proxy statistics on page 396

Sample Output

clear fibre-channel proxy statistics

`user@switch>` **clear fibre-channel proxy statistics**

clear fip snooping enode

Syntax **clear fip snooping enode** ***enode-mac***
<vlan *vlan-name*>

Release Information Command introduced in Junos OS Release 10.4 for EX Series switches.
Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Clear FIP snooping information for the specified FCoE Node (ENode) or (optionally) only on a specified VLAN. This operation deletes the ENode state from the switch database and from the FIP snooping firewall filters, which causes the ENode to lose its connection to the FCoE forwarder (FCF) and to log in to the FCF again.

Options ***enode-mac***—MAC address of the ENode.

vlan ***vlan-name***—(Optional) Name of the VLAN.

Required Privilege Level view

Related Documentation

• show fip snooping enode on page 497

List of Sample Output clear fip snooping enode enode-mac on page 397

Sample Output

clear fip snooping enode enode-mac

```
user@switch> clear fip snooping enode 00:10:94:00:00:02
```

clear fip snooping statistics

Syntax	**clear fip snooping statistics** <vlan *vlan-name*>
Release Information	Command introduced in Junos OS Release 10.4 for EX Series switches. Command introduced in Junos OS Release 11.1 for the QFX Series.
Description	Clear FIP snooping statistics globally or on a specified VLAN.
Required Privilege Level	view
Related Documentation	• show fip snooping statistics on page 507
List of Sample Output	clear fip snooping statistics on page 398

Sample Output

clear fip snooping statistics

```
user@switch> clear fip snooping statistics
```

clear fip snooping vlan

Syntax **clear fip snooping vlan *vlan-name***

Release Information Command introduced in Junos OS Release 10.4 for EX Series switches.
Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Clear FIP snooping information for the specified VLAN. This operation deletes all ENode and FCF information for the VLAN from the switch database and causes the ENodes to lose their connections to the FCFs. After clearing a VLAN, the switch relearns all of the FCFs and ENodes on the VLAN, and the ENodes must log in to the FCF again.

Options ***vlan-name***—Name of the VLAN.

Required Privilege Level view

Related Documentation

- show fip snooping vlan on page 510

List of Sample Output clear fip snooping vlan vlan-name on page 399

Sample Output

clear fip snooping vlan vlan-name

```
user@switch> clear fip snooping vlan fcoevlan1
```

clear fip vlan-discovery statistics

Syntax **clear fip vlan-discovery statistics**

Release Information Command introduced in Junos OS Release 12.1 for the QFX Series.

Description Clear FIP VLAN discovery statistics.

Required Privilege Level view

Related Documentation
- show fip vlan-discovery on page 514

List of Sample Output clear fip vlan-discovery statistics on page 400

Sample Output

clear fip vlan-discovery statistics

```
user@switch> clear fip vlan-discovery statistics
```


request fibre-channel proxy load-rebalance

Syntax

request fibre-channel proxy load-rebalance
<dry-run>
fabric <*fabric-name*>
<brief | detail>

Release Information

Command introduced in Junos OS Release 12.3 for the QFX Series.

Description

Rebalance the link load on one or more FCoE-FC gateway proxy fabrics (local Fibre Channel fabrics on the gateway) on demand. Load rebalancing is a disruptive action that forces some or all sessions (depending on the configured load-balancing algorithm) to log out and then log in again. When sessions log in again, they are placed on NP_Port interfaces so that the link loads are balanced.

Link load rebalancing occurs 10 seconds after you run the rebalancing command, unless another rebalancing trigger occurs before the 10 seconds elapse. If another rebalancing event occurs before the 10-second timer elapses, the timer is extended. Rebalancing occurs a maximum of 30 seconds after you run the rebalancing command, regardless of whether more rebalancing events occur.

You can also perform a *dry run* to see a list of sessions that might be affected (logged out) if you request a load rebalance. A dry run does not rebalance the link loads; it only lists the sessions that might be affected if you rebalance.

Options

dry-run—(Optional) Simulates performing link load rebalancing and displays a list of sessions that might be affected if you rebalance the link loads.

fabric *fabric-name*—Name of the fabric on which you want to rebalance the link loads. If you do not specify a fabric name with the fabric keyword, all fabrics on the FCoE-FC gateway rebalance their link loads.

brief | detail—(Optional) Display the specified level of output.

Additional Information

Requesting link load rebalancing is a one-time, on-demand operation. You must explicitly request load rebalancing every time you want to rebalance the link loads. Alternatively, you can configure automated load rebalancing if you want the NP_Port links to be rebalanced automatically whenever a load-rebalancing trigger occurs.

Required Privilege Level

maintenance

Related Documentation

- Monitoring Fibre Channel Interface Load Balancing on page 383
- Simulating On-Demand Fibre Channel Link Load Rebalancing (Dry Run Test) on page 280
- Defining the Proxy Load-Balancing Algorithm on page 279
- Example: Configuring Automated Fibre Channel Interface Load Rebalancing on page 250
- Understanding Load Balancing in an FCoE-FC Gateway Proxy Fabric on page 63

List of Sample Output request fibre-channel proxy load-rebalance dry-run fabric fc_fabric_100 on page 402

Output Fields Table 34 on page 402 lists the output fields for the **request fibre-channel proxy load-rebalance dry-run** command. Output fields are listed in the approximate order in which they appear.

Table 34: request fibre-channel proxy load-rebalance dry-run Output Fields

Field Name	Field Description
Fabric	Name of the fabric.
Fabric-id	Fabric ID number.
F-Port	FCoE VLAN interface (VF_Port interface to the FCoE network).
FCID	VN_Port Fibre Channel identifier provided by the Fibre Channel over Ethernet Forwarder (FCoE forwarder) or the Fibre Channel switch.
Port-WWN	Unique worldwide name (WWN) of the VN_Port.
NP-Port	Name of the native Fibre Channel interface.

Sample Output

request fibre-channel proxy load-rebalance dry-run fabric fc_fabric_100

```
user@host> request fibre-channel proxy load-rebalance dry-run fabric fc_fabric_100
Fabric: fc_fabric_100, Fabric-id: 100
F-Port                        FCID     Port-WWN                NP-Port
vlan.100                      0x8a013a 02:01:00:64:00:00:00:2a fc-0/0/1.0
vlan.100                      0x8a013c 02:01:00:64:00:00:00:2b fc-0/0/1.0
vlan.100                      0x8a0146 02:01:00:64:00:00:00:2e fc-0/0/1.0
vlan.100                      0x8a014c 02:01:00:64:00:00:00:2f fc-0/0/1.0
```


restart

List of Syntax

Syntax

restart
<adaptive-services |ancpd-service | application-identification |audit-process | auto-configuration |captive-portal-content-delivery |ce-l2tp-service |chassis-control | class-of-service |clksyncd-service |database-replication|datapath-trace-service |dhcp-service | diameter-service | disk-monitoring | dynamic-flow-capture | ecc-error-logging | ethernet-connectivity-fault-management |ethernet-link-fault-management |event-processing | firewall |general-authentication-service | gracefully | iccp-service |idp-policy | immediately |interface-control | ipsec-key-management | kernel-replication | l2-learning | l2cpd-service | l2tp-service | l2tp-universal-edge | lacp | license-service |link-management |local-policy-decision-function |mac-validation |mib-process | mobile-ip | mountd-service |mpls-traceroute |mspd | multicast-snooping |named-service | nfsd-service | packet-triggered-subscribers |peer-selection-service |pgcp-service | pgm | pic-services-logging | pki-service |ppp | ppp-service |pppoe | protected-system-domain-service | redundancy-interface-process | remote-operations | root-system-domain-service | routing <logical-system *logical-system-name*> | sampling | sbc-configuration-process | sdk-service |service-deployment | services | services pgcp gateway *gateway-name* | snmp |soft |static-subscribers |statistics-service| subscriber-management | subscriber-management-helper | tunnel-oamd |usb-control| vrrp |web-management>
<gracefully | immediately | soft>

Syntax (ACX Series Routers)

restart
<adaptive-services |audit-process | auto-configuration | autoinstallation |chassis-control | class-of-service |clksyncd-service |database-replication| dhcp-service | diameter-service | disk-monitoring | dynamic-flow-capture | ethernet-connectivity-fault-management |ethernet-link-fault-management |event-processing | firewall |general-authentication-service | gracefully | immediately |interface-control | ipsec-key-management | l2-learning | lacp |link-management |mib-process | mobile-ip | mountd-service |mpls-traceroute |mspd | named-service | nfsd-service | pgm | pki-service |ppp | pppoe | redundancy-interface-process | remote-operations | routing | sampling | sdk-service |secure-neighbor-discovery | service-deployment | services | snmp |soft |statistics-service| subscriber-management | subscriber-management-helper | tunnel-oamd | vrrp>

Syntax (EX Series Switches)

restart
<autoinstallation | chassis-control | class-of-service | database-replication | dhcp | dhcp-service | diameter-service | dot1x-protocol | ethernet-link-fault-management | ethernet-switching | event-processing | firewall | general-authentication-service | interface-control | kernel-replication | l2-learning | lacp | license-service | link-management

| lldpd-service | mib-process | mountd-service | multicast-snooping | pgm |
redundancy-interface-process | remote-operations | routing | secure-neighbor-discovery
| service-deployment | sflow-service | snmp | vrrp | web-management>

Syntax (Routing Matrix)

restart
<adaptive-services | audit-process | chassis-control | class-of-service | disk-monitoring |
dynamic-flow-capture | ecc-error-logging | event-processing | firewall | interface-control
| ipsec-key-management | kernel-replication | l2-learning | l2tp-service | lacp |
link-management | mib-process | pgm | pic-services-logging | ppp | pppoe |
redundancy-interface-process | remote-operations | routing <logical-system
logical-system-name> | sampling | service-deployment | snmp>
<all | all-lcc | lcc *number*>
<gracefully | immediately | soft>

Syntax (J Series Routing Platform)

restart
<adaptive-services | audit-process | chassis-control | class-of-service | dhcp | dialer-services
| dlsw | event-processing | firewall | interface-control | ipsec-key-management |
isdn-signaling | l2-learning | l2tp-service | mib-process | network-access-service | pgm |
ppp | pppoe | remote-operations | routing <logical-system *logical-system-name*> | sampling
| service-deployment | snmp | usb-control | web-management>
<gracefully | immediately | soft>

Syntax (TX Matrix Routers)

restart
<adaptive-services | audit-process | chassis-control | class-of-service | dhcp-service |
diameter-service | disk-monitoring | dynamic-flow-capture | ecc-error-logging |
event-processing | firewall | interface-control | ipsec-key-management | kernel-replication
| l2-learning | l2tp-service | lacp | link-management | mib-process |pgm | pic-services-logging
| ppp | pppoe | redundancy-interface-process | remote-operations | routing <logical-system
logical-system-name> | sampling | service-deployment | snmp| statistics-service>
<all-chassis | all-lcc | lcc *number* | scc>
<gracefully | immediately | soft>

Syntax (TX Matrix Plus Routers)

restart
<adaptive-services | audit-process | chassis-control | class-of-service | dhcp-service |
diameter-service | disk-monitoring | dynamic-flow-capture | ecc-error-logging |
event-processing | firewall | interface-control | ipsec-key-management | kernel-replication
| l2-learning | l2tp-service | lacp | link-management | mib-process | pgm |
pic-services-logging | ppp | pppoe | redundancy-interface-process | remote-operations |
routing <logical-system *logical-system-name*> | sampling | service-deployment | snmp|
statistics-service>
<all-chassis | all-lcc | all-sfc | lcc *number* | sfc *number*>
<gracefully | immediately | soft>

Syntax (MX Series Routers)

restart
<adaptive-services | ancpd-service |application-identification|audit-process |
auto-configuration |captive-portal-content-delivery |ce-l2tp-service |chassis-control |
class-of-service |clksyncd-service |database-replication| datapath-trace-service
|dhcp-service | diameter-service | disk-monitoring | dynamic-flow-capture |
ecc-error-logging | ethernet-connectivity-fault-management
|ethernet-link-fault-management |event-processing | firewall |
general-authentication-service | gracefully |iccp-service |idp-policy |immediately
|interface-control | ipsec-key-management | kernel-replication | l2-learning |l2cpd-service
|l2tp-service | l2tp-universal-edge | lacp | license-service |link-management
|local-policy-decision-function |mac-validation |mib-process | mobile-ip |mountd-service
|mpls-traceroute |mspd | multicast-snooping |named-service | nfsd-service |

packet-triggered-subscribers |peer-selection-service | pgcp-service | pgm | pic-services-logging | pki-service |ppp | ppp-service |pppoe | protected-system-domain-service | redundancy-interface-process | remote-operations |root-system-domain-service | routing |routing <logical-system *logical-system-name*> | sampling | sbc-configuration-process | sdk-service |service-deployment |services | services pgcp gateway *gateway-name* |snmp |soft |static-subscribers |statistics-service| subscriber-management | subscriber-management-helper | tunnel-oamd | usb-control| vrrp |web-management>
<all-members>
<gracefully | immediately | soft>
<local>
<member *member-id*>

Syntax (J Series Routers)

restart
<adaptive-services | audit-process | chassis-control | class-of-service | dhcp | dhcp-service | dialer-services | diameter-service | dlsw | event-processing | firewall | interface-control | ipsec-key-management | isdn-signaling | l2ald | l2-learning | l2tp-service | mib-process | network-access-service | pgm | ppp | pppoe | remote-operations | routing <logical-system *logical-system-name*> | sampling | service-deployment | snmp | usb-control | web-management>
<gracefully | immediately | soft>

Syntax (QFX Series)

restart
<adaptive-services | audit-process | chassis-control | class-of-service | dialer-services | diameter-service | dlsw | ethernet-connectivity | event-processing | fibre-channel | firewall | general-authentication-service | igmp-host-services | interface-control | ipsec-key-management | isdn-signaling | l2ald | l2-learning | l2tp-service | mib-process | named-service | network-access-service | nstrace-process | pgm | ppp | pppoe | redundancy-interface-process | remote-operations |*logical-system-name*> | routing | sampling |secure-neighbor-discovery | service-deployment | snmp | usb-control | web-management>
<gracefully | immediately | soft>

Release Information

Command introduced before Junos OS Release 7.4.
Command introduced in Junos OS Release 9.0 for EX Series switches.
Command introduced in Junos OS Release 11.1 for the QFX Series.
Command introduced in Junos OS Release 12.2 for ACX Series routers.
Options added:

- **dynamic-flow-capture** in Junos OS Release 7.4.
- **dlsw** in Junos OS Release 7.5.
- **event-processing** in Junos OS Release 7.5.
- **ppp** in Junos OS Release 7.5.
- **l2ald** in Junos OS Release 8.0.
- **link-management** in Release 8.0.
- **pgcp-service** in Junos OS Release 8.4.
- **sbc-configuration-process** in Junos OS Release 9.5.
- **services pgcp gateway** in Junos OS Release 9.6.
- **sfc** and **all-sfc** for the TX Matrix Router in Junos OS Release 9.6.

Description Restart a Junos OS process.

CAUTION: Never restart a software process unless instructed to do so by a customer support engineer. A restart might cause the router or switch to drop calls and interrupt transmission, resulting in possible loss of data.

Options **none**—Same as **gracefully**.

adaptive-services—(Optional) Restart the configuration management process that manages the configuration for stateful firewall, Network Address Translation (NAT), intrusion detection services (IDS), and IP Security (IPsec) services on the Adaptive Services PIC.

all-chassis—(TX Matrix and TX Matrix Plus routers only) (Optional) Restart the software process on all chassis.

all-lcc—(TX Matrix and TX Matrix Plus routers only) (Optional) For a TX Matrix router, restart the software process on all T640 routers connected to the TX Matrix router. For a TX Matrix Plus router, restart the software process on all T1600 routers connected to the TX Matrix Plus router.

all-members—(MX Series routers only) (Optional) Restart the software process for all members of the Virtual Chassis configuration.

all-sfc—(TX Matrix Plus routers only) (Optional) For a TX Matrix Plus router, restart the software processes for the TX Matrix Plus router (or switch-fabric chassis).

ancpd-service—(Optional) Restart the Access Node Control Protocol (ANCP) process, which works with a special Internet Group Management Protocol (IGMP) session to collect outgoing interface mapping events in a scalable manner.

application-identification—(Optional) Restart the process that identifies an application using intrusion detection and prevention (IDP) to allow or deny traffic based on applications running on standard or nonstandard ports.

audit-process—(Optional) Restart the RADIUS accounting process that gathers statistical data that can be used for general network monitoring, analyzing, and tracking usage patterns, for billing a user based on the amount of time or type of services accessed.

auto-configuration—(Optional) Restart the Interface Auto-Configuration process.

autoinstallation—(EX Series switches only) (Optional) Restart the autoinstallation process.

captive-portal-content-delivery—(Optional) Restart the HTTP redirect service by specifying the location to which a subscriber's initial Web browser session is redirected, enabling initial provisioning and service selection for the subscriber.

ce-l2tp-service—(M10, M10i, M7i, and MX Series routers only) (Optional) Restart the Universal Edge Layer 2 Tunneling Protocol (L2TP) process, which establishes L2TP tunnels and Point-to-Point Protocol (PPP) sessions through L2TP tunnels.

chassis-control—(Optional) Restart the chassis management process.

class-of-service—(Optional) Restart the class-of-service (CoS) process, which controls the router's or switch’s CoS configuration.

clksyncd-service—(Optional) Restart the external clock synchronization process, which uses synchronous Ethernet (SyncE).

database-replication—(EX Series switches and MX Series routers only) (Optional) Restart the database replication process.

datapath-trace-service—(Optional) Restart the packet path tracing process.

dhcp—(J Series routers and EX Series switches only) (Optional) Restart the software process for a Dynamic Host Configuration Protocol (DHCP) server. A DHCP server allocates network IP addresses and delivers configuration settings to client hosts without user intervention.

dhcp-service—(Optional) Restart the Dynamic Host Configuration Protocol process.

dialer-services—(J Series routers and EX Series switches only) (Optional) Restart the ISDN dial-out process.

diameter-service—(Optional) Restart the diameter process.

disk-monitoring—(Optional) Restart disk monitoring, which checks the health of the hard disk drive on the Routing Engine.

dlsw—(J Series routers and QFX Series only) (Optional) Restart the data link switching (DLSw) service.

dot1x-protocol—(EX Series switches only) (Optional) Restart the port-based network access control process.

dynamic-flow-capture—(Optional) Restart the dynamic flow capture (DFC) process, which controls DFC configurations on Monitoring Services III PICs.

ecc-error-logging—(Optional) Restart the error checking and correction (ECC) process, which logs ECC parity errors in memory on the Routing Engine.

ethernet-connectivity-fault-management—(Optional) Restart the process that provides IEEE 802.1ag Operation, Administration, and Management (OAM) connectivity fault management (CFM) database information for CFM maintenance association end points (MEPs) in a CFM session.

ethernet-link-fault-management—(EX Series switches and MX Series routers only) (Optional) Restart the process that provides the OAM link fault management (LFM) information for Ethernet interfaces.

ethernet-switching—(EX Series switches only) (Optional) Restart the Ethernet switching process.

event-processing—(Optional) Restart the event process (eventd).

fibre-channel—(QFX Series only) (Optional) Restart the Fibre Channel process.

firewall—(Optional) Restart the firewall management process, which manages the firewall configuration and enables accepting or rejecting packets that are transiting an interface on a router or switch.

general-authentication-service—(EX Series switches and MX Series routers only) (Optional) Restart the general authentication process.

gracefully—(Optional) Restart the software process.

iccp-service—(Optional) Restart the Inter-Chassis Communication Protocol (ICCP) process.

idp-policy—(Optional) Restart the intrusion detection and prevention (IDP) protocol process.

immediately—(Optional) Immediately restart the software process.

interface-control—(Optional) Restart the interface process, which controls the router's or switch’s physical interface devices and logical interfaces.

ipsec-key-management—(Optional) Restart the IPsec key management process.

isdn-signaling—(J Series routers and QFX Series only) (Optional) Restart the ISDN signaling process, which initiates ISDN connections.

kernel-replication—(Optional) Restart the kernel replication process, which replicates the state of the backup Routing Engine when graceful Routing Engine switchover (GRES) is configured.

l2-learning—(Optional) Restart the Layer 2 address flooding and learning process.

l2cpd-service—(Optional) Restart the Layer 2 Control Protocol process, which enables features such as Layer 2 protocol tunneling and nonstop bridging.

l2tp-service— (M10, M10i, M7i, and MX Series routers only) (Optional) Restart the Layer 2 Tunneling Protocol (L2TP) process, which sets up client services for establishing Point-to-Point Protocol (PPP) tunnels across a network and negotiating Multilink PPP if it is implemented.

l2tp-universal-edge—(MX Series routers only) (Optional) Restart the L2TP process, which establishes L2TP tunnels and PPP sessions through L2TP tunnels.

lacp—(Optional) Restart the Link Aggregation Control Protocol (LACP) process. LACP provides a standardized means for exchanging information between partner systems on a link to allow their link aggregation control instances to reach agreement on the identity of the LAG to which the link belongs, and then to move the link to that LAG, and to enable the transmission and reception processes for the link to function in an orderly manner.

lcc ***number***—(TX Matrix and TX Matrix Plus routers only) (Optional) For a TX Matrix router, restart the software process for a specific T640 router that is connected to the TX Matrix router. For a TX Matrix Plus router, restart the software process for a specific router that is connected to the TX Matrix Plus router.

Replace *number* with the following values depending on the LCC configuration:

- 0 through 3, when T640 routers are connected to a TX Matrix router in a routing matrix.
- 0 through 3, when T1600 routers are connected to a TX Matrix Plus router in a routing matrix.
- 0 through 7, when T1600 routers are connected to a TX Matrix Plus router with 3D SIBs in a routing matrix.
- 0, 2, 4, or 6, when T4000 routers are connected to a TX Matrix Plus router with 3D SIBs in a routing matrix.

license-service—(EX Series switches only) (Optional) Restart the feature license management process.

link-management— (TX Matrix and TX Matrix Plus routers and EX Series switches only) (Optional) Restart the Link Management Protocol (LMP) process, which establishes and maintains LMP control channels.

lldpd-service—(EX Series switches only) (Optional) Restart the Link Layer Discovery Protocol (LLDP) process.

local—(MX Series routers only) (Optional) Restart the software process for the local Virtual Chassis member.

local-policy-decision-function— (Optional) Restart the process for the Local Policy Decision Function, which regulates collection of statistics related to applications and application groups and tracking of information about dynamic subscribers and static interfaces.

mac-validation— (Optional) Restart the Media Access Control (MAC) validation process, which configures MAC address validation for subscriber interfaces created on demux interfaces in dynamic profiles on MX Series routers.

member ***member-id***—(MX Series routers only) (Optional) Restart the software process for a specific member of the Virtual Chassis configuration. Replace ***member-id*** with a value of **0** or **1**.

mib-process—(Optional) Restart the Management Information Base (MIB) version II process, which provides the router's MIB II agent.

mobile-ip—(Optional) Restart the Mobile IP process, which configures Junos OS Mobile IP features.

mountd-service—(EX Series switches and MX Series routers only) (Optional) Restart the service for NFS mount requests.

mpls-traceroute—(Optional) Restart the MPLS Periodic Traceroute process.

mspd—(Optional) Restart the Multiservice process.

multicast-snooping—(EX Series switches and MX Series routers only) (Optional) Restart the multicast snooping process, which makes Layer 2 devices, such as VLAN switches, aware of Layer 3 information, such as the media access control (MAC) addresses of members of a multicast group.

named-service—(Optional) Restart the DNS Server process, which is used by a router or a switch to resolve hostnames into addresses.

network-access-service—(J Series routers and QFX Series only) (Optional) Restart the network access process, which provides the router's Challenge Handshake Authentication Protocol (CHAP) authentication service.

nfsd-service—(Optional) Restart the Remote NFS Server process, which provides remote file access for applications that need NFS-based transport.

packet-triggered-subscribers—(Optional) Restart the packet-triggered subscribers and policy control (PTSP) process, which allows the application of policies to dynamic subscribers that are controlled by a subscriber termination device.

peer-selection-service—(Optional) Restart the Peer Selection Service process.

pgcp-service—(Optional) Restart the pgcpd service process running on the Routing Engine. This option does not restart pgcpd processes running on mobile station PICs. To restart pgcpd processes running on mobile station PICs, use the **services pgcp gateway** option.

pgm—(Optional) Restart the process that implements the Pragmatic General Multicast (PGM) protocol for assisting in the reliable delivery of multicast packets.

pic-services-logging—(Optional) Restart the logging process for some PICs. With this process, also known as fsad (the file system access daemon), PICs send special logging information to the Routing Engine for archiving on the hard disk.

pki-service—(Optional) Restart the PKI Service process.

ppp—(Optional) Restart the Point-to-Point Protocol (PPP) process, which is the encapsulation protocol process for transporting IP traffic across point-to-point links.

ppp-service—(Optional) Restart the Universal Edge PPP process, which is the encapsulation protocol process for transporting IP traffic across Universal Edge routers.

pppoe—(Optional) Restart the Point-to-Point Protocol over Ethernet (PPPoE) process, which combines PPP that typically runs over broadband connections with the Ethernet link-layer protocol that allows users to connect to a network of hosts over a bridge or access concentrator.

protected-system-domain-service—(Optional) Restart the Protected System Domain (PSD) process.

redundancy-interface-process—(Optional) Restart the ASP redundancy process.

remote-operations—(Optional) Restart the remote operations process, which provides the ping and traceroute MIBs.

root-system-domain-service—(Optional) Restart the Root System Domain (RSD) service.

routing—(ACX Series routers, QFX Series, EX Series switches, and MX Series routers only) (Optional) Restart the routing protocol process.

routing <logical-system *logical-system-name*>—(Optional) Restart the routing protocol process, which controls the routing protocols that run on the router or switch and maintains the routing tables. Optionally, restart the routing protocol process for the specified logical system only.

sampling—(Optional) Restart the sampling process, which performs packet sampling based on particular input interfaces and various fields in the packet header.

sbc-configuration-process—(Optional) Restart the session border controller (SBC) process of the border signaling gateway (BSG).

scc—(TX Matrix routers only) (Optional) Restart the software process on the TX Matrix router (or switch-card chassis).

sdk-service—(Optional) Restart the SDK Service process, which runs on the Routing Engine and is responsible for communications between the SDK application and Junos OS. Although the SDK Service process is present on the router, it is turned off by default.

secure-neighbor-discovery—(QFX Series, EX Series switches, and MX Series routers only) (Optional) Restart the secure Neighbor Discovery Protocol (NDP) process, which provides support for protecting NDP messages.

sfc *number*—(TX Matrix Plus routers only) (Optional) Restart the software process on the TX Matrix Plus router (or switch-fabric chassis). Replace ***number*** with **0**.

service-deployment—(Optional) Restart the service deployment process, which enables Junos OS to work with the Session and Resource Control (SRC) software.

services—(Optional) Restart a service.

services pgcp gateway *gateway-name*—(Optional) Restart the pgcpd process for a specific border gateway function (BGF) running on an MS-PIC. This option does not restart the pgcpd process running on the Routing Engine. To restart the pgcpd process on the Routing Engine, use the **pgcp-service** option.

sflow-service—(EX Series switches only) (Optional) Restart the flow sampling (sFlow technology) process.

snmp—(Optional) Restart the SNMP process, which enables the monitoring of network devices from a central location and provides the router's or switch's SNMP master agent.

soft—(Optional) Reread and reactivate the configuration without completely restarting the software processes. For example, BGP peers stay up and the routing table stays constant. Omitting this option results in a graceful restart of the software process.

static-subscribers—(Optional) Restart the static subscribers process, which associates subscribers with statically configured interfaces and provides dynamic service activation and activation for these subscribers.

statistics-service—(Optional) Restart the process that manages the Packet Forwarding Engine statistics.

subscriber-management—(Optional) Restart the Subscriber Management process.

subscriber-management-helper—(Optional) Restart the Subscriber Management Helper process.

tunnel-oamd—(Optional) Restart the Tunnel OAM process, which enables the Operations, Administration, and Maintenance of Layer 2 tunneled networks. Layer 2 protocol tunneling (L2PT) allows service providers to send Layer 2 PDUs across the provider's cloud and deliver them to Juniper Networks EX Series Ethernet Switches that are not part of the local broadcast domain.

usb-control—(J Series routers and MX Series routers only) (Optional) Restart the USB control process.

vrrp—(ACX Series routers, EX Series switches, and MX Series routers only) (Optional) Restart the Virtual Router Redundancy Protocol (VRRP) process, which enables hosts on a LAN to make use of redundant routing platforms on that LAN without requiring more than the static configuration of a single default route on the hosts.

web-management—(J Series routers, QFX Series, EX Series switches, and MX Series routers only) (Optional) Restart the Web management process.

Required Privilege Level reset

Related Documentation

- *Overview of Junos OS CLI Operational Mode Commands*

List of Sample Output restart interfaces on page 412

Output Fields When you enter this command, you are provided feedback on the status of your request.

Sample Output

restart interfaces

```
user@host> restart interfaces
interfaces process terminated
interfaces process restarted
```


show dcbx

Syntax show dcbx

Release Information Command introduced in Junos OS Release 11.3 for the QFX Series.

Description List DCBX status (enabled or disabled) and the interfaces on which DCBX is enabled.

Required Privilege Level view

Related Documentation

- **show dcbx neighbors on page 415**
- Configuring DCBX Autonegotiation on page 306

Output Fields Table 35 on page 414 lists the output fields for the **show dcbx** command. Output fields are listed in the approximate order in which they appear.

Table 35: show dcbx output fields

Field Name	Field Description
DCBX	Status of DCBX on the switch or for the specified interface: • Enabled—DCBX is enabled on the switch or on the specified interface • Disabled—DCBX is disabled on the switch or on the specified interface
Interface	Name of the interface

Sample Output

show dcbx

```
user@switch> show dcbx
DCBX                       : Enabled
Interface        DCBX
xe-0/0/9.0       enabled
xe-0/0/32.0      enabled
xe-0/0/36.0      enabled
```


show dcbx neighbors

Syntax

show dcbx neighbors
<**interface** *interface-name*>
<**terse**>

Release Information

Command introduced in Junos OS Release 11.1 for the QFX Series.
Command introduced in Junos OS Release 11.3 for EX Series switches.

Description

Display information about Data Center Bridging Capability Exchange protocol (DCBX) neighbor interfaces.

Options

none—Display information about all DCBX neighbor interfaces.

interface-name—(Optional) Display information for the specified interface.

terse—Display the specified level of output.

Required Privilege Level

view

Related Documentation

- Configuring DCBX Autonegotiation on page 306
- Example: Configuring DCBX Application Protocol TLV Exchange on page 217
- *Example: Configuring an FCoE Transit Switch*
- *Example: Configuring DCBX to Support an iSCSI Application*
- Understanding DCB Features and Requirements on page 16
- *Understanding Data Center Bridging Capability Exchange Protocol for EX Series Switches*
- dcbx on page 324

List of Sample Output

show dcbx neighbors interface (QFX Series, DCBX Version 1.01 Mode) on page 428
show dcbx neighbors interface (QFX Series, IEEE DCBX Mode) on page 430
show dcbx neighbors terse (QFX Series) on page 432
show dcbx neighbors (EX4500 Switch: FCoE Interfaces on Both Local and Peer with PFC Configured Compatibly) on page 432
show dcbx neighbors (EX4500 Switch: DCBX Interfaces on Local and Peer Are Configured Compatibly with iSCSI Application) on page 433
show dcbx neighbors (EX4500 Switch: Includes ETS) on page 434

Output Fields

Table 36 on page 415 lists the output fields for the **show dcbx neighbors** command. Output fields are listed in the approximate order in which they appear.

Table 36: show dcbx neighbors Output Fields

Field Name	Field Description
Interface	Name of the interface.

Table 36: show dcbx neighbors Output Fields *(continued)*

Field Name		Field Description
Parent Interface		Name of the link aggregation group (LAG) interface to which the DCBX interface belongs.
Active-application-map		Name of the application map applied to the interface.
Protocol-Mode		(QFX Series) DCBX protocol mode the interface uses: • IEEE DCBX Version—The interface uses IEEE DCBX mode. • DCBX Version 1.01—The interface uses DCBX version 1.01. NOTE: On interfaces that use the IEEE DCBX mode, the **show dcbx neighbors interface** ***interface-name*** operational command does not include application, PFC, or ETS operational state in the output.
Protocol-State		(DCBX Version 1.01 only) DCBX protocol state synchronization status: • **in-sync**—The local interface received an acknowledge message from the peer to indicate that the peer received a state change message sent by the local interface. • **ack-pending**—The local interface has not yet received an acknowledge message from the peer to indicate that the peer received a state change message sent by the local interface.
Local-Advertisement		(DCBX Version 1.01 only) Status of advertisements that the local interface sends to the peer.
	Operational version	Version of the DCBX standard used.
	sequence-number	Number of state change messages sent to the peer. If the interface **Protocol-State** value is **in-sync**, this number should match the **acknowledge-id** number in the **Peer-Advertisement** section. If the interface **Protocol-State** value is **ack-pending**, this number does not match the **acknowledge-id** number in the **Peer-Advertisement** section.
	acknowledge-id	Number of acknowledge messages received from the peer. If the **Protocol-State** value is **in-sync**, this number should match the **sequence-number** value in the **Peer-Advertisement** section. If the **Protocol-State** value is **ack-pending**, this number does not match the **sequence-number** value in the **Peer-Advertisement** section.

Table 36: show dcbx neighbors Output Fields *(continued)*

Field Name		Field Description
Peer-Advertisement		(DCBX Version 1.01 only) Status of advertisements that the peer sends to the local interface.
	Operational version	Version of the DCBX standard used.
	sequence-number	Number of state change messages the peer sent to the local interface. If this number matches the **acknowledge-id** number in the **Local-Advertisement** field, this indicates that the local interface has acknowledged all of the peer's state change messages and is synchronized. If this number does not match the **acknowledge-id** number in the **Local-Advertisement** field, this indicates that the peer has not yet received an acknowledgment for a state change message from the local interface.
	acknowledge-id	Number of acknowledge messages the peer has received from the local interface. If this number matches the **sequence-number** value in the **Local-Advertisement** field, this indicates that the peer has acknowledged all of the local interface's state change messages and is in synchronization. If this number does not match the **sequence-number** value in the **Local-Advertisement** field, this indicates that the peer has not yet sent an acknowledgment for a state change message from the local interface.

Table 36: show dcbx neighbors Output Fields *(continued)*

Field Name			Field Description
Feature: PFC			Priority-based flow control (PFC) feature DCBX state information.
	Protocol-State		(DCBX Version 1.01 only) DCBX protocol state synchronization status: • **ack-pending**—The local interface has not yet received an acknowledge message from the peer to indicate that the peer received a PFC state change message sent by the local interface. • **in-sync**—The local interface received an acknowledge message from the peer to indicate that the peer received a PFC state change message sent by the local interface. • **not-applicable**—PFC autonegotiation is disabled.
	Operational State		(DCBX Version 1.01 only) Operational state of the feature: **enabled** or **disabled**.
	Local-Advertisement		Status of advertisements that the local interface sends to the peer.
		Enable	(DCBX Version 1.01 only) State that the local interface advertises to the peer: • **Yes**—The feature is enabled. • **No**—The feature is disabled.
		Willing	Willingness of the local interface to learn the PFC configuration from the peer using DCBX: • **Yes**—The local interface is willing to learn the PFC configuration from the peer. • **No**—The local interface is not willing to learn the PFC configuration from the peer.
		Mac auth Bypass Capability	(IEEE DCBX only) (QFX Series) Media access controller (MAC) authentication bypass provides access to devices based on MAC address authentication. This is not supported, so the only value seen in the local advertisement field is **no**.
		Error	(DCBX Version 1.01 only) Configuration compatibility error status: • **No**—No error detected. Local and peer configuration are compatible. • **Yes**—Error detected. Local and peer configuration are not compatible.

Table 36: show dcbx neighbors Output Fields *(continued)*

Field Name			Field Description
		Operational State	PFC operational state on the interface: • Enabled—PFC is enabled on the interface • Disabled—PFC is disabled on the interface
		Maximum Traffic Classes capable to support PFC	Largest number of traffic classes the local interface supports for PFC: • **6** (EX Series switches) • **8** (QFX Series)
		Code Point	PFC code point, which is specified in the 3-bit class-of-service field in the VLAN header.
		Admin Mode	PFC administrative state for each code point on the local interface: • **Enabled**—PFC is enabled for the code point. • **Disabled**—PFC is disabled for the code point.
		Operational Mode	(QFX Series) PFC operational mode for each code point: • Enable—PFC is enabled on the code point. • Disable—PFC is disabled on the code point.
	Peer-Advertisement		Status of advertisements that the peer sends to the local interface.
		Enable	(DCBX Version 1.01 only) State that the peer advertises to the local interface: • **Yes**—The feature is enabled. • **No**—The feature is disabled.
		Willing	Willingness of the peer to learn the PFC configuration from the local interface using DCBX: • **Yes**—The peer is willing to learn the PFC configuration from the local interface. • **No**—The peer is not willing to learn the PFC configuration from the local interface.
		Error	(DCBX Version 1.01 only) Configuration compatibility error status: • **No**—No error detected. Local and peer configuration are compatible. • **Yes**—Error detected. Local and peer configuration are not compatible.

Table 36: show dcbx neighbors Output Fields *(continued)*

Field Name		Field Description
	Operational State	PFC operational state on the interface: • Enabled—PFC is enabled on the interface • Disabled—PFC is disabled on the interface
	Mac auth Bypass Capability	(IEEE DCBX only) (QFX Series) Media access controller (MAC) authentication bypass provides access to devices based on MAC address authentication. Although the QFX Series does not support this feature, the connected peer might support it. This field reports the peer state: • **Yes**—The connected peer supports MAC authentication bypass. • **No**—The connected peer does not support MAC authentication bypass.
	Maximum Traffic Classes capable to support PFC	Largest number of traffic classes the peer supports for PFC: • **6** (EX Series switches) • **8** (QFX Series)
	Code Point	PFC code point, which is specified in the 3-bit class-of-service field in the VLAN header.
	Admin Mode	PFC administrative state for each code point on the peer: • **Enabled**—PFC is enabled for the code point. • **Disabled**—PFC is disabled for the code point.

Table 36: show dcbx neighbors Output Fields *(continued)*

Field Name			Field Description
Feature: Application			State information for the DCBX application.
	Protocol-State		(DCBX Version 1.01 only) DCBX protocol state synchronization status: • **in-sync**—The local interface received an acknowledge message from the peer to indicate that the peer received an FCoE state change message sent by the local interface. • **ack-pending**—The local interface has not yet received an acknowledge message from the peer to indicate that the peer received an FCoE state change message sent by the local interface. • **not-applicable**—The local interface is set to **no-auto-negotiation** (autonegotiation is disabled). If the interface is associated with an FCoE forwarding class, the interface advertises FCoE capability even if the connected peer does not advertise FCoE capability.
	Local-Advertisement		Status of advertisements that the local interface sends to the peer. If the local interface is set to **no-auto-negotiation** (autonegotiation is disabled), the local advertisement portion of the output is not shown.
		Enable	(DCBX Version 1.01 only) State that the local interface advertises to the peer: • **Yes**—The feature is enabled. • **No**—The feature is disabled.
		Willing	(DCBX Version 1.01 only) Willingness of the local interface to learn the FCoE interface state from the peer using DCBX: • **Yes**—The local interface is willing to learn the FCoE interface state from the peer. • **No**—The local interface is not willing to learn the FCoE interface state from the peer.
		Error	(DCBX Version 1.01 only) Configuration compatibility error status: • **No**—No error detected. The local and peer configuration are compatible. • **Yes**—Error detected. The local and peer configuration are not compatible.
		Appl-Name	Name of the application:

Table 36: show dcbx neighbors Output Fields *(continued)*

Field Name		Field Description
	Ethernet-Type	(DCBX Version 1.01 only) Ethernet type (EtherType) of the application. For example, **0x8906** indicates the EtherType for the FCoE application. Either the EtherType (for Layer 2 applications) or the Socket Number (for Layer 4 applications) of the application is displayed in the output.
	Socket-Number	Destination port socket number of the application, if applicable. Either the EtherType (for Layer 2 applications) or the Socket Number (for Layer 4 applications) of the application is displayed in the output.
	Priority-Field or **Priority-Map**	Priority assigned to the application. For EX Series switches, the priority of the FCoE application is determined by the PFC congestion notification profile that has been configured and associated with the FCoE interface. For other applications, the priority is based on the application map.
	Status	(DCBX Version 1.01 only) Local status when autonegotiation is enabled: • **Enabled**—The application feature is enabled on both the local interface and the peer interface. (The local configuration and the peer configuration match.) • **Disabled**—The local configuration and the peer configuration do not match. NOTE: If there is a configuration mismatch in one application between the switch and the peer, all the other applications including FCoE are disabled.
Peer-Advertisement		Status of advertisements that the peer sends to the local interface.
	Enable	(DCBX Version 1.01 only) State that the peer advertises to the local interface: • **Yes**—The feature is enabled. • **No**—The feature is disabled.
	Willing	(DCBX Version 1.01 only) Willingness of the peer to learn the FCoE interface state from the local interface using DCBX: • **Yes**—The peer is willing to learn the FCoE interface state from the local interface. • **No**—The peer is not willing to learn the FCoE interface state from the local interface.

Table 36: show dcbx neighbors Output Fields *(continued)*

Field Name		Field Description
	Error	(DCBX Version 1.01 only) Configuration compatibility error status: • **No**—No error detected. Local and peer configuration are compatible. • **Yes**—Error detected. Local and peer configuration are not compatible.
	Appl-Name	Name of the application: • **FCoE**—Fibre Channel over Ethernet
	Ethernet-Type	Ethernet type (EtherType) of the application. For example, **0x8906** indicates the EtherType for the FCoE application. Either the EtherType (for Layer 2 applications) or the Socket-Number (for Layer 4 applications) of the application is displayed in the output.
	Socket-Number	Destination port socket number of the application, if applicable. Either the EtherType (for Layer 2 applications) or the Socket Number (for Layer 4 applications) of the application is displayed in the output.
	Priority-Field or **Priority-Map**	Priority assigned to the application.
	Status	(DCBX Version 1.01 only) Peer interface status: • **Enabled**—The application feature is enabled on both the local interface and the peer interface. (The local configuration and the peer configuration match.) • **Disabled**—The local configuration and the peer configuration do not match.

Table 36: show dcbx neighbors Output Fields *(continued)*

Field Name			Field Description
Feature: ETS			Enhanced Transmission Selection (ETS) DCBX state information.
	Protocol-State		(DCBX Version 1.01 only) ETS protocol state synchronization status: • **in-sync**—The local interface received an acknowledge message from the peer to indicate that the peer received an ETS state change message sent by the local interface. • **ack-pending**—The local interface has not yet received an acknowledge message from the peer to indicate that the peer received an ETS state change message sent by the local interface.
	Operational State		(DCBX Version 1.01 only) Operational state of the feature, **enabled** or **disabled**.
	Local-Advertisement		Status of advertisements that the local interface sends to the peer.
		Enable	(DCBX Version 1.01 only) State that the local interface advertises to the peer: • **Yes**—The feature is enabled. • **No**—The feature is disabled.
		TLV Type	(IEEE DCBX only) Type of ETS TLV: • Configuration—Advertises the Configuration TLV, which communicates the local ETS configuration to the peer but does not ask the peer to use the configuration. • Recommendation—Advertises the Recommendation TLV, which communicates the local ETS configuration to the peer, and if the peer is "willing," configures the peer interface to match the local ETS configuration. • Recommendation-or-Configuration—Advertises both TLVs.
		Willing	Willingness of the local interface to learn the ETS state from the peer using DCBX (EX Series switches always advertise **No** for this field): • **Yes**—Local interface is willing to learn the ETS state from the peer. • **No**—Local interface is not willing to learn the ETS state from the peer.
		Credit Based Shaper	

Table 36: show dcbx neighbors Output Fields *(continued)*

Field Name		Field Description
		(IEEE DCBX only) Alternative method of flow control to buffer-to-buffer credit. The QFX Series does not support a credit-based shaper, so the value of this field is always **No**.
	Error	(DCBX Version 1.01 only) Configuration error status: • **No**—No error. This should always be the switch ETS error state. • **Yes**—Error detected.
	Maximum Traffic Classes capable to support PFC	(DCBX Version 1.01 only) Largest number of traffic classes the local interface supports for PFC.
	Maximum Traffic Classes supported	(IEEE DCBX only) Largest number of traffic classes the local interface supports for ETS. (EX Series switches support only one traffic class for ETS. However, a different value might be shown for this field.)
	Code Point	PFC code point, which is specified in the 3-bit class-of-service field in the VLAN header.
	Priority-Group	Class-of-service (CoS) priority group (forwarding class set) identification number.
	Percentage B/W	Configured minimum percentage of link bandwidth allocated to the priority group. Only explicitly configured values appear in this output column. If the link bandwidth is the default percentage, it is not shown. (EX Series switches allocate 100% of link bandwidth to the default priority group, group 7.)
	Transmission Selection Algorithm	(IEEE DCBX only) The transmission selection algorithm used by the interface. The QFX Series supports ETS but does not support using the credit-based shaper algorithm, so the only value shown in this field is **ETS**.
Peer-Advertisement		Status of advertisements that the peer sends to the local interface.
	Enable	

Table 36: show dcbx neighbors Output Fields *(continued)*

Field Name	Field Description
	(DCBX Version 1.01 only) State that the peer advertises to the local interface: • **Yes**—The feature is enabled. • **No**—The feature is disabled.
TLV Type	(IEEE DCBX only) Type of ETS TLV: • Configuration—Advertises the Configuration TLV, which communicates the local ETS configuration to the peer but does not ask the peer to use the configuration. • Recommendation—Advertises the Recommendation TLV, which communicates the local ETS configuration to the peer, and if the peer is “willing,” configures the peer interface to match the local ETS configuration. • Configuration/Recommendation—Advertises both TLVs.
Willing	Willingness of the peer to learn the ETS state from the local interface using DCBX: • **Yes**—Peer is willing to learn the ETS state from the local interface. • **No**—Peer is not willing to learn the ETS state from the local interface.
Credit Based Shaper	(IEEE DCBX only) Alternative method of flow control to buffer-to-buffer credit. The QFX Series does not support a credit-based shaper, so the value of this field is always **No**.
Error	(DCBX Version 1.01 only) Configuration error status of the peer: • **No**—No error in peer ETS TLV. • **Yes**—Error in peer ETS TLV.
Maximum Traffic Classes capable to support PFC	(DCBX Version 1.01 only) Largest number of traffic classes the local interface supports for PFC.
Maximum Traffic Classes supported	(IEEE DCBX only) Largest number of traffic classes the local interface supports for ETS. (EX Series switches support only one traffic class for ETS. However, a different value might be shown for this field.)
Code Point	

Table 36: show dcbx neighbors Output Fields *(continued)*

Field Name		Field Description
		PFC code point, which is specified in the 3-bit class-of-service field in the VLAN header.
	Priority-Group	CoS priority group (forwarding class set) identification number.
	Percentage B/W	Configured minimum percentage of link bandwidth allocated to the priority group. (EX Series switches allocate 100% of link bandwidth to the default priority group, group 7.)
	Transmission Selection Algorithm	(IEEE DCBX only) Transmission selection algorithm used by the interface. The QFX Series supports ETS but does not support using the credit-based shaper algorithm, so the only value shown in this field is **ETS**.
PFC		(QFX Series, **terse** option only) DCBX TLV advertisement state for PFC: • Disabled—PFC configuration matches the configuration on the connected peer and PFC is disabled • Enabled—PFC configuration matches the configuration on the connected peer and PFC is enabled • Not Advt—Interface does not advertise PFC to the connected peer
ETS		(**terse** option only) Local DCBX TLV advertisement state for ETS: • Advt—Interface advertises ETS TLVs • Disabled—ETS is disabled on the interface (interface does not advertise ETS)
ETS Rec		(**terse** option only) DCBX TLV peer advertisement state for ETS (state received from the connected DCBX peer): • Advt—Peer interface advertises ETS TLVs • Not Advt—Peer interface does not advertise ETS NOTE: When the DCBX mode is DCBX version 1.01, no peer information is displayed.

Table 36: show dcbx neighbors Output Fields *(continued)*

Field Name	Field Description
Version	(**terse** option only) The DCBX version used on the interface and whether the DCBX version was autonegotiated or explicitly configured: • **IEEE**—The interface uses IEEE DCBX. • **1.01**—The interface uses DCBX version 1.01. When the DCBX version used is the result of autonegotiation, the term **(Auto)** appears next to the version. For example, **IEEE (Auto)** indicates that the interface autonegotiated with the connected peer to use IEEE DCBX. Autonegotiation is enabled by default.

Sample Output

show dcbx neighbors interface (QFX Series, DCBX Version 1.01 Mode)

```
user@switch> show dcbx neighbors interface xe-0/0/0
Interface : xe-0/0/0.0 - Parent Interface: ae0.0
    Active-application-map: app-map-1
    Protocol-State: in-sync
    Protocol-Mode: DCBX Version 1.01

    Local-Advertisement:
        Operational version: 1
        sequence-number: 130, acknowledge-id: 102

    Peer-Advertisement:
        Operational version: 1
        sequence-number: 102, acknowledge-id: 130

    Feature: PFC, Protocol-State: in-sync

      Operational State: Enabled

      Local-Advertisement:
        Enable: Yes, Willing: No, Error: No
        Maximum Traffic Classes capable to support PFC: 8

        Code Point             Admin Mode   Operational Mode
          000                   Disabled    Disable
          001                   Disabled    Disable
          010                   Disabled    Disable
          011                   Enabled     Enable
          100                   Enabled     Enable
          101                   Disabled    Disable
          110                   Disabled    Disable
          111                   Disabled    Disable

      Peer-Advertisement:
        Enable: Yes, Willing: No, Error: No
        Maximum Traffic Classes capable to support PFC: 8

        Code Point             Admin Mode
          000                   Disabled
```



```
    001               Disabled
    010               Disabled
    011               Enabled
    100               Enabled
    101               Disabled
    110               Disabled
    111               Disabled

Feature: Application, Protocol-State: in-sync

  Local-Advertisement:
    Enable: Yes, Willing: No, Error: No

    Appl-Name    Ethernet-Type    Socket-Number    Priority-Map    Status
      FCoE         0x8906                            00001110        Enabled

      iSCSI                         3260            10000000        Enabled


  Peer-Advertisement:
    Enable: Yes, Willing: Yes, Error: No

    Appl-Name    Ethernet-Type    Socket-Number    Priority-Map    Status

      FCoE         0x8906            N/A            00001110        Enabled


Feature: ETS, Protocol-State: in-sync

  Operational State: Enabled

  Local-Advertisement:
    Enable: Yes, Willing: No, Error: No
    Maximum Traffic Classes capable to support PFC: 8

    Code Point          Priority-Group
      000                 0
      001                 7
      010                 7
      011                 7
      100                 0
      101                 1
      110                 1
      111                 7

    Priority-Group      Percentage B/W
      0                   40%
      1                   5%

  Peer-Advertisement:
    Enable: Yes, Willing: No, Error: No
    Maximum Traffic Classes capable to support PFC: 8

    Code Point          Priority-Group
      000                 0
      001                 7
      010                 7
      011                 7
      100                 0
      101                 1
      110                 1
```

```
      111                  7

    Priority-Group         Percentage B/W
      0                      40%
      1                      5%
```

show dcbx neighbors interface (QFX Series, IEEE DCBX Mode)

```
user@switch> show dcbx neighbors interface xe-0/0/0
Interface : xe-0/0/0.0 - Parent Interface: ae0.0
    Active-application-map: app-map-1
    Protocol-Mode: IEEE-DCBX Version

    Feature: PFC

      Local-Advertisement:
        Willing: No
        Mac auth Bypass Capability: No
        Operational State: Enabled

        Maximum Traffic Classes capable to support PFC: 8

        Code Point           Admin Mode
          000                 Disabled
          001                 Disabled
          010                 Disabled
          011                 Enabled
          100                 Enabled
          101                 Disabled
          110                 Disabled
          111                 Disabled

      Peer-Advertisement:
        Willing: No
        Mac auth Bypass Capability: No
        Operational State: Enabled

        Maximum Traffic Classes capable to support PFC: 8

        Code Point          Admin Mode
          000                 Disabled
          001                 Disabled
          010                 Disabled
          011                 Enabled
          100                 Enabled
          101                 Disabled
          110                 Disabled
          111                 Disabled

    Feature: Application

      Local-Advertisement:

        Appl-Name    Ethernet-Type    Socket-Number    Priority-field
          FCoE         0x8906                            00001110
          iSCSI                           3260           10000000

      Peer-Advertisement:
        Appl-Name    Ethernet-Type    Socket-Number    Priority-field
```

```
      FCoE         0x8906             N/A            00001110

Feature: ETS

  Local-Advertisement:
    TLV Type: Configuration/Recommendation
    Willing: No
    Credit Based Shaper: No
    Maximum Traffic Classes supported: 3

    Code Point           Priority-Group
      000                  0
      001                  7
      010                  7
      011                  7
      100                  0
      101                  1
      110                  1
      111                  7

    Priority-Group       Percentage B/W
      0                    40%
      1                    5%

    Priority-Group   Transmission Selection Algorithm
      0              Enhanced Transmission Selection
      1              Enhanced Transmission Selection

  Peer-Advertisement:
    TLV Type: Configuration
    Willing: No
    Credit Based Shaper: No

    Code Point           Priority-Group
      000                  0
      001                  7
      010                  7
      011                  7
      100                  0
      101                  1
      110                  1
      111                  7

    Priority-Group       Percentage B/W
      0                    40%
      1                    5%

    Priority-Group   Transmission Selection Algorithm
      0              Enhanced Transmission Selection
      1              Enhanced Transmission Selection

  Peer-Advertisement:

    TLV Type: Recommendation

    Code Point           Priority-Group
      000                  0
      001                  7
      010                  7
      011                  7
      100                  0
```



```
    101                   1
    110                   1
    111                   7

  Priority-Group        Percentage B/W
    0                     40%
    1                     5%

  Priority-Group    Transmission Selection Algorithm
    0               Enhanced Transmission Selection
    1               Enhanced Transmission Selection
```

show dcbx neighbors terse (QFX Series)

```
user@switch> show dcbx neighbors terse
Interface    Parent      PFC        ETS        ETS        Version
             Interface                                    Rec
xe-0/0/8.0   -           Enabled    Advt       Advt       IEEE (Auto)
xe-0/0/9.0   -           Disabled   Disabled              1.01
xe-0/0/11.0 ae0.0        Enabled    Advt       Advt       IEEE (Auto)
xe-0/0/12.0 ae0.0        Enabled    Advt       Advt       IEEE (Auto)
xe-0/0/32.0 -            Enabled    Advt       Not Advt   IEEE
xe-0/0/36.0 -            Not Advt   Advt       Advt       IEEE
```

show dcbx neighbors (EX4500 Switch: FCoE Interfaces on Both Local and Peer with PFC Configured Compatibly)

```
user@switch> show dcbx neighbors interface xe-0/0/14

   Interface : xe-0/0/14.0 - Parent Interface: ae0.0
    Protocol-State: in-sync

    Local-Advertisement:
        Operational version: 0
        sequence-number: 6, acknowledge-id: 6

    Peer-Advertisement:
        Operational version: 0
        sequence-number: 6, acknowledge-id: 6

    Feature: PFC, Protocol-State: in-sync

      Operational State: Enabled

      Local-Advertisement:
        Enable: Yes, Willing: No, Error: No
        Maximum Traffic Classes capable to support PFC: 6

         Code Point            Admin Mode
          000                 Disabled
          001                 Disabled
          010                 Disabled
          011                 Enabled
          100                 Disabled
          101                 Disabled
          110                 Disabled
          111                 Disabled
```



```
      Peer-Advertisement:
        Enable: Yes, Willing: No, Error: No
        Maximum Traffic Classes capable to support PFC: 6

        Code Point              Admin Mode
          000                  Disabled
          001                  Disabled
          010                  Disabled
          011                  Enabled
          100                  Disabled
          101                  Disabled
          110                  Disabled
          111                  Disabled


    Feature: Application, Protocol-State: in-sync

      Local-Advertisement:
        Enable: Yes, Willing: No, Error: No   <<< Error bit will not be set as
there is no miss configuration between local and peer.


        Appl-Name    Ethernet-Type    Socket-Number    Priority-Map    Status
        FCoE          0x8906                           00001000        Enabled


      Peer-Advertisement:
        Enable: Yes, Willing: No, Error: No

                 Appl-Name    Ethernet-Type    Socket-Number    Priority-Map
Status
                 FCoE          0x8906                           00001000
Enabled
```

show dcbx neighbors (EX4500 Switch: DCBX Interfaces on Local and Peer Are Configured Compatibly with iSCSI Application)

user@switch> **show dcbx neighbors interface xe-0/0/14**

```
  Interface : xe-0/0/14.0 - Parent Interface: ae0.0

    Protocol-State: in-sync

    Active-application-map: iscsi-map

     Local-Advertisement:
        Operational version: 0
        sequence-number: 9, acknowledge-id: 12


    Peer-Advertisement:
        Operational version: 0
        sequence-number: 12, acknowledge-id: 9


    Feature: PFC, Protocol-State: in-sync
      Operational State: Enabled
```

```
  Local-Advertisement:
   Enable: Yes, Willing: No, Error: No
   Maximum Traffic Classes capable to support PFC: 6


Code Point              Admin Mode
     000                 Disabled
     001                 Disabled
     010                 Disabled
     011                 Enabled
     100                 Disabled
     101                 Disabled
     110                 Disabled
     111                 Disabled

 Peer-Advertisement:
   Enable: Yes, Willing: No, Error: No
   Maximum Traffic Classes capable to support PFC: 6

Code Point              Admin Mode
     000                 Disabled
     001                 Disabled
     010                 Disabled
     011                 Enabled
     100                 Disabled
     101                 Disabled
     110                 Disabled
     111                 Disabled

Feature: Application, Protocol-State: in-sync
 Local-Advertisement:
   Enable: Yes, Willing: No, Error: No

       Appl-Name    Ethernet-Type    Socket-Number   Priority-Map    Status

        FCoE         0x8906                           00001000        Enabled

        iscsi                         3260            00100000        Enabled


 Peer-Advertisement:
   Enable: Yes, Willing: No, Error: No

       Appl-Name    Ethernet-Type    Socket-Number   Priority-Map    Status

        FCoE         0x8906                           00001000        Enabled

        iscsi                         3260            00100000        Enabled
```

show dcbx neighbors (EX4500 Switch: Includes ETS)

user@switch> **show dcbx neighbors interface xe-0/0/3**

```
Interface : xe-0/0/3.0
 Protocol-State: in-sync
 Active-application-map: map_iscsi

 Local-Advertisement:
     Operational version: 0
```



```
    sequence-number: 1, acknowledge-id: 5

Peer-Advertisement:
    Operational version: 0
    sequence-number: 5, acknowledge-id: 1

Feature: PFC, Protocol-State: in-sync

  Operational State: Enabled

  Local-Advertisement:
    Enable: Yes, Willing: No, Error: No
    Maximum Traffic Classes capable to support PFC: 6

    Code Point          Admin Mode
      000                 Enabled
      001                 Enabled
      010                 Disabled
      011                 Disabled
      100                 Disabled
      101                 Disabled
      110                 Disabled
      111                 Disabled

  Peer-Advertisement:
    Enable: Yes, Willing: Yes, Error: No
    Maximum Traffic Classes capable to support PFC: 8

    Code Point          Admin Mode
      000                 Enabled
      001                 Disabled
      010                 Disabled
      011                 Disabled
      100                 Enabled
      101                 Disabled
      110                 Disabled
      111                 Disabled

Feature: Application, Protocol-State: in-sync

  Local-Advertisement:
    Enable: Yes, Willing: No, Error: No

    Appl-Name    Ethernet-Type    Socket-Number    Priority-Map    Status
      FCoE         0x8906                            00000001        Enabled

      iscsi                         3260             00000010        Enabled


  Peer-Advertisement:
    Enable: Yes, Willing: Yes, Error: No

    Appl-Name    Ethernet-Type    Socket-Number    Priority-Map    Status
      FCoE         0x8906                            00001000        Enabled

      iscsi                         3260             00010000        Enabled


Feature: ETS, Protocol-State: in-sync

  Operational State: Enabled
```

```
Local-Advertisement:
  Enable: Yes, Willing: No, Error: No
  Maximum Traffic Classes supported : 3

  Code Point            Priority-Group
    000                   7
    001                   7
    010                   7
    011                   7
    100                   7
    101                   7
    110                   7
    111                   7

  Priority-Group        Percentage B/W
    7                     100%

Peer-Advertisement:
  Enable: Yes, Willing: Yes, Error: No
  Maximum Traffic Classes supported : 8

  Code Point            Priority-Group
    000                   0
    001                   1
    010                   0
    011                   0
    100                   2
    101                   0
    110                   0
    111                   0

  Priority-Group        Percentage B/W
    0                     30%
    1                     40%
    2                     30%
```


show fibre-channel fabric

Syntax

show fibre-channel fabric
<extensive | summary>
<*fabric-name*>
<sort-by (name | fabric-id)>

Release Information Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Display Fibre Channel fabric information.

Options

fabric-name—(Optional) Display output only for the specified fabric.

extensive | summary—(Optional) Display the specified level of output.

sort-by (name | fabric-id)—(Optional) Sort output by fabric name or fabric ID.

Required Privilege Level view

Related Documentation

- fc-fabrics on page 338
- Configuring an FCoE-FC Gateway Fibre Channel Fabric on page 266

List of Sample Output

show fibre-channel fabric on page 438
show fibre-channel fabric extensive on page 438

Output Fields Table 37 on page 437 lists the output fields for the **show fibre-channel fabric** command. Output fields are listed in the approximate order in which they appear.

Table 37: show fibre-channel fabric Output Fields

Field Name	Field Description	Level of Output
Fabric	Name of the fabric.	All
Fabric-ID	Identification number of the fabric.	All
Type	Type of fabric. All fabrics are **PROXY** fabrics.	All
Interfaces	Native Fibre Channel interfaces and FCoE interfaces assigned to the fabric.	All
Created at	Date and time the fabric was created.	**extensive**
Internal Index	Fabric index internal to Junos OS.	**extensive**
Origin	Origin information internal to Junos OS.	**extensive**
Description	Text description of the fabric.	**extensive**

Table 37: show fibre-channel fabric Output Fields *(continued)*

Field Name	Field Description	Level of Output
Fabric WWN	Unique WWN of the fabric generated by the FCF.	**extensive**
Login sessions	Number of FIP login sessions currently running on the fabric.	**extensive**
Configured max login sessions	Configured maximum number of FIP login sessions permitted on the fabric.	**extensive**

Sample Output

show fibre-channel fabric

```
user@switch> show fibre-channel fabric
Fabric         Fabric-ID     Type     Interfaces

proxy2         200           PROXY
                                     fc-0/0/0.0
                                     fc-0/0/1.0
```

show fibre-channel fabric extensive

```
user@switch> show fibre-channel fabric extensive
Fabric: proxy2, Created at: Mon Apr 19 14:02:58 2010
Fabric-ID: 200, Internal index: 2, Origin: Static
Description: srv-fabric,  Type: PROXY,  Fabric WWN: 10:00:00:05:33:51:d7:cd
Login sessions: 200,  Configured max login sessions: 500
      fc-0/0/0.0, (untagged)
      fc-0/0/1.0, (untagged)
```

show fibre-channel fc2 sessions

Syntax

show fibre-channel fc2 sessions
<fabric *fabric-name*>
<brief | detail>

Release Information Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Display Fibre Channel FC-2 information.

NOTE: A session is a FLOGI or FDISC login to the FC SAN fabric. Session does not refer to end-to-end storage sessions.

Options

fabric *fabric-name*—(Optional) Display output only for the specified fabric.

brief | detail—(Optional) Display the specified level of output.

Required Privilege Level view

Related Documentation

- show fibre-channel fc2 statistics on page 441
- clear fibre-channel fc2 statistics on page 391

List of Sample Output

show fibre-channel fc2 sessions on page 440
show fibre-channel fc2 sessions detail on page 440

Output Fields Table 38 on page 439 lists the output fields for the **show fibre-channel fc2 sessions** command. Output fields are listed in the approximate order in which they appear.

Table 38: show fibre-channel fc2 sessions Output Fields

Field Name	Field Description	Level of Output
Fabric	Name of the fabric.	All
Fabric-id	Identification number of the fabric.	All
Interface Name	Name of the interface.	All
Local FCID	Address of the local end of the connection.	All
Far FCID	Address of the far (remote) end of the connection.	All
# Pending Exchanges	Number of pending exchanges for the session.	All
Flags	Flags internal to Junos OS.	**detail**

Table 38: show fibre-channel fc2 sessions Output Fields *(continued)*

Field Name	Field Description	Level of Output
Refcount	Reference count internal to Junos OS.	detail
Users	Information internal to Junos OS.	detail

Sample Output

show fibre-channel fc2 sessions

```
user@switch> show fibre-channel fc2 sessions
Fabric: fip-proxy, Fabric-id: 1
Interface      Local     Far         # Pending
Name           FCID      FCID        Exchanges
fc-0/0/0.0     *         0xfffffe    0
fc-0/0/1.0     *         0xfffffe    0
fc-0/0/2.0     *         0xfffffe    0
```

show fibre-channel fc2 sessions detail

```
user@switch> show fibre-channel fc2 sessions detail
Fabric: fip-proxy, Fabric-id: 1
Interface Name      fc-0/0/0.0
Local FCID:         *
Far FCID:           0xfffffe
Exchanges:          0
Flags:              SELF_LOCK USER_SYNCHED
Refcount:           2
Users:              1

Interface Name      fc-0/0/1.0
Local FCID:         *
Far FCID:           0xfffffe
Exchanges:          0
Flags:              SELF_LOCK USER_SYNCHED
Refcount:           2

Interface Name      fc-0/0/2.0
Local FCID:         *
Far FCID:           0xfffffe
Exchanges:          0
Flags:              SELF_LOCK USER_SYNCHED
Refcount:           2
Users:              1
```

show fibre-channel fc2 statistics

Syntax

show fibre-channel fc2 statistics
<fabric ***fabric-name*****>**

Release Information

Command introduced in Junos OS Release 11.1 for the QFX Series.

Description

Display Fibre Channel FC-2 statistics.

Options

fabric ***fabric-name***—(Optional) Display output only for the specified fabric.

Required Privilege Level

view

Related Documentation

- show fibre-channel fc2 sessions on page 439
- clear fibre-channel fc2 statistics on page 391

List of Sample Output

show fibre-channel fc2 statistics on page 442

Output Fields

Table 39 on page 441 lists the output fields for the **show fibre-channel fc2 statistics** command. Output fields are listed in the approximate order in which they appear.

Table 39: show fibre-channel fc2 statistics Output Fields

Field Name	Field Description
Global statistics	Statistics for all fabrics.
Frame buffers allocated	Number of frame buffers currently allocated to all fabrics.
Frame buffers freed	Number of frame buffers freed.
Frames dropped	Number of dropped frames.
Fabric statistics	Fabric-specific statistics.
Fabric	Name of the fabric.
Fabric-id	Identification number of the fabric.
Tx-FRJTs	Number of fabric frame rejects (F_RJTs).
Tx-PRJTs	Number of port frame rejects (P_RJTs).
Tx-LSRJTs	Number of link service rejections.
Tx-ABTS	Number of abort sequence frames sent.
Rx-Drops	Number of received frames dropped.

Table 39: show fibre-channel fc2 statistics Output Fields *(continued)*

Field Name	Field Description
Rx-ABTS	Number of abort sequence frames received.

Sample Output

show fibre-channel fc2 statistics

```
user@switch> show fibre-channel fc2 statistics
Global statistics:

Frame buffers allocated: 60
Frame buffers freed:     60
Frames dropped:          0

Fabric statistics:

Fabric : fip-proxy, Fabric-id: 1
Tx-FRJTs:     0
Tx-PRJTs:     0
Tx-LSRJTs:    0
Tx-ABTS:      0
Rx-Drops:     0
Rx-ABTS:      0
```

show fibre-channel fip

Syntax

show fibre-channel fip
<brief | detail>

Release Information

Command introduced in Junos OS Release 11.1 for the QFX Series.

Description

Display Fibre Channel over Ethernet Initialization Protocol (FIP) information.

Options

brief | detail—(Optional) Display the specified level of output.

Required Privilege Level

view

Related Documentation

- Configuring FIP on an FCoE-FC Gateway on page 283
- show fibre-channel fip enode on page 448
- show fibre-channel fip fabric on page 452
- show fibre-channel fip fcf on page 455
- show fibre-channel fip interface on page 458
- show fibre-channel fip statistics on page 461
- clear fibre-channel fip statistics on page 393

List of Sample Output

show fibre-channel fip on page 445
show fibre-channel fip detail on page 446

Output Fields

Table 40 on page 443 lists the output fields for the **show fibre-channel fip** command. Output fields are listed in the approximate order in which they appear. A session is a FLOGI or FDISC login to the FC SAN fabric. Session does not refer to end-to-end storage sessions.

Table 40: show fibre-channel fip Output Fields

Field Name	Field Description	Level of Output
Configured max FIP sessions per Node Device	Configured maximum number of FIP sessions permitted on the Node device. For QFabric systems, this is the maximum number of FIP sessions permitted on each Node device in the fabric. For QFX3500 devices, this is the maximum number of FIP sessions permitted on the device.	**detail**
Node Device	Node device identifier.	**detail**
Total FIP sessions	Total number of FIP sessions on the FCoE-FC gateway switch.	**detail**

Table 40: show fibre-channel fip Output Fields *(continued)*

Field Name	Field Description	Level of Output
Total FCoE filters	Total number of FIP filters on the FCoE-FC gateway switch.	**detail**
Fabric Name	Name of the fabric and in parentheses the fabric ID.	All
FC-MAP	FCoE mapped address prefix of the FCoE forwarder for the fabric.	**detail**
FKA-ADV-PERIOD	Period of time in milliseconds between FIP keepalive advertisements configured for the FC fabric.	**detail**
MAX-SESSIONS-PER-ENODE	Maximum number of concurrent sessions (FLOGI and FDISC combined) that each ENode can instantiate.	**detail**
FCoE trusted	Whether ports on the FC fabric are trusted or untrusted: • **Yes**—Ports on the FC fabric are trusted; FIP snooping is turned off. • **No**—Ports on the FC fabric are not trusted; FIP snooping is turned on.	**detail**
Member	Information about an FCF that is a member of the fabric.	All
• **FCF-MAC**	MAC address used in discovery advertisements.	All
• **FKA-ADV-PERIOD**	Period of time in milliseconds between FIP keepalive advertisements configured for the FC interface.	**detail**
• **FKA-ADV-D-BIT**	Disable FIP keepalive advertisement monitoring bit. The state is always **off**.	**detail**
• **Type**	Type of interface: • **VF_Port Capable**—Interface can act as a VF_Port interface.	**detail**
• **Priority**	Priority value associated with the switch FCF-MAC. Converged network adapters (CNAs) use the priority value to determine the switch with which they will perform FIP FLOGI. The lower the value, the higher the priority. Value range: 0 through 255.	**detail**
• **State**	FIP state on the fabric: • **Enable**—FIP is enabled on the fabric. • **Disable**—FIP is disabled on the fabric.	**detail**

Table 40: show fibre-channel fip Output Fields *(continued)*

Field Name		Field Description	Level of Output
ENode		Information about a connected FCoE node (ENode).	All
	• **ENode-MAC**	MAC address of the connected ENode.	All
	• **Enode State**	Login state internal to Junos OS.	All
	• **Configured ENode timer**	User-configured FIP keepalive advertisement interval in milliseconds.	**detail**
	• **Running ENode timer**	Runtime interval in milliseconds of the last FIP keepalive advertisement received. This value changes every time an FKA_ADV is received.	**detail**
	• **Active FIP Sessions**	Number of active FIP sessions on the ENode.	**detail**
	• **VN-Port-MAC**	MAC address of a VN_Port on the ENode.	All
	• **Session State**	Session state internal to Junos OS.	**detail**
	• **Configured FKA-ADV**	User-configured FIP keepalive advertisement interval in milliseconds.	**detail**
	• **Running FKA-ADV**	Runtime interval in milliseconds of the last FIP keepalive advertisement received. This value changes every time an FKA_ADV is received.	**detail**
	• **Configured VN-Port Timer**	Configured state of the VN_Port keepalive timer in milliseconds. This value is always 90 and is not user-configurable.	**detail**
	• **Running VN-Port Timer**	Running state of the VN_Port keepalive timer in milliseconds.	**detail**
	• **FCID**	Fibre Channel ID of the VN_Port.	**detail**
	• **WWN**	Unique worldwide name of the VN_Port.	**detail**

Sample Output

show fibre-channel fip

```
user@switch> show fibre-channel fip
Fabric Name : proxy2 (200)
  Member
  FCF-MAC : 00:30:48:b0:ee:d2 (Interface vlan.100)
    Enode
    Enode-MAC : 00:10:94:00:00:02       State : Logged-in
        Session
        VN-Port-MAC              : 0e:fc:00:03:00:02
        VN-Port-MAC              : 0e:fc:00:03:00:01
```



```
    Enode-MAC : 00:10:94:00:00:03       State : Logged-in
        Session
        VN-Port-MAC               : 0e:fc:00:03:00:04
        VN-Port-MAC               : 0e:fc:00:03:00:03
```

show fibre-channel fip detail

```
user@switch> show fibre-channel fip detail
Configured max FIP sessions per Node Device: 2500
Node Device: 0  Total FIP sessions: 4  Total FCoE filters: 4

Fabric Name : proxy2 (200)
  FC-MAP          : 0e:fc:00
  FKA-ADV-PERIOD : 90000        MAX-SESSIONS-PER-ENODE : 32
  FCoE trusted    : No

    Member
    FCF-MAC: 00:30:48:b0:ee:d2  (Interface vlan.100)
      FKA-ADV-PERIOD  : 90000    FKA-ADV-D-BIT-bit : Off
      Type : VF_Port Capable
      Priority : 86              State : Enable

        ENode
        Enode-MAC : 00:10:94:00:00:02   ENode State : Logged-in
         Configured ENode timer: 8000    Running ENode timer: 12226
         Active FIP Sessions : 2

          Session details
          VN-Port-MAC               : 0e:fc:00:03:00:02
          Session state             : Up
          Configured FKA-ADV        : 90000
          Running FKA-ADV           : 0
          Configured VN-Port Timer : 90000
          Running VN-Port Timer     : 213193
          FCID                      : 0x2c1a01
          WWN                       : 10:00:00:00:c9:a4:a3:cf

          VN-Port-MAC               : 0e:fc:00:03:00:01
          Session state             : Up
          Configured FKA-ADV        : 90000
          Running FKA-ADV           : 0
          Configured VN-Port Timer : 90000
          Running VN-Port Timer     : 213632
          FCID                      : 0x2c1a02
          WWN                       : 10:00:00:00:d9:b4:e3:df

        ENode
        Enode-MAC : 00:10:94:00:00:03   ENode State : Logged-in
         Configured ENode timer: 8000    Running ENode timer: 12254
         Active FIP Sessions : 2

          Session details
          VN-Port-MAC               : 0e:fc:00:03:00:04
          Session state             : Up
          Configured FKA-ADV        : 90000
          Running FKA-ADV           : 0
          Configured VN-Port Timer : 90000
          Running VN-Port Timer     : 213480
          FCID                      : 0x2c1a03
          WWN                       : 21:00:00:c0:dd:11:09:13
```

```
VN-Port-MAC              : 0e:fc:00:03:00:03
Session state            : Up
Configured FKA-ADV       : 90000
Running FKA-ADV          : 0
Configured VN-Port Timer : 90000
Running VN-Port Timer    : 214004
FCID                     : 0x2c1a04
WWN                      : 21:00:00:c0:df:12:08:14
```

show fibre-channel fip enode

Syntax

show fibre-channel fip enode *enode-mac*
<brief | detail>
<vn-port-mac *vn-port-mac*>

Release Information

Command introduced in Junos OS Release 11.1 for the QFX Series.

Description

Display Fibre Channel over Ethernet (FCoE) Initialization Protocol (FIP) information for a specified ENode or a specified VN_Port on an ENode.

Options

brief | detail—(Optional) Display the specified level of output.

enode-mac—Display information for the ENode specified by the MAC address.

vn-port-mac *vn-port-mac*—(Optional) Display information only for the specified VN_Port.

Required Privilege Level

view

Related Documentation

- Configuring FIP on an FCoE-FC Gateway on page 283
- show fibre-channel fip on page 443
- show fibre-channel fip fabric on page 452
- show fibre-channel fip fcf on page 455
- show fibre-channel fip interface on page 458
- show fibre-channel fip statistics on page 461
- clear fibre-channel fip enode on page 392

List of Sample Output

show fibre-channel fip enode on page 450
show fibre-channel fip enode detail on page 451

Output Fields

Table 41 on page 448 lists the output fields for the **show fibre-channel fip enode** command. Output fields are listed in the approximate order in which they appear. A session is a FLOGI or FDISC login to the FC SAN fabric. Session does not refer to end-to-end storage sessions.

Table 41: show fibre-channel fip enode Output Fields

Field Name	Field Description	Level of Output
Fabric Name	Name of the fabric and in parentheses the fabric ID.	All
FC-MAP	FCoE mapped address prefix of the FCoE forwarder for the fabric.	**detail**
FKA-ADV-PERIOD	Period of time in milliseconds between FIP keepalive advertisements configured for the FC fabric.	**detail**

Table 41: show fibre-channel fip enode Output Fields *(continued)*

Field Name	Field Description	Level of Output
MAX-SESSIONS-PER-ENODE	Maximum number of concurrent sessions (FLOGI and FDISC combined) that each ENode can instantiate.	**detail**
FCoE trusted	Whether ports on the FC fabric are trusted or untrusted: • **Yes**—Ports on the FC fabric are trusted; FIP snooping is turned off. • **No**—Ports on the FC fabric are not trusted; FIP snooping is turned on.	**detail**
Member	Information about an FCF that is a member of the fabric.	All
• **FCF-MAC**	MAC address used in discovery advertisements.	All
• **FKA-ADV-PERIOD**	Period of time in milliseconds between FIP keepalive advertisements configured for the FC interface.	**detail**
• **FKA-ADV-D-BIT**	Disable FIP keepalive advertisement monitoring bit. The state is always **off**.	**detail**
• **Type**	Type of interface: • **VF_Port Capable**—Interface can act as a VF_Port interface.	**detail**
• **Priority**	Priority value associated with the switch FCF-MAC. Converged network adapters (CNAs) use the priority value to determine the switch with which they will perform FIP FLOGI. The lower the value, the higher the priority. Value range: 0 through 255.	**detail**
• **State**	FIP state on the fabric: • **Enable**—FIP is enabled on the fabric. • **Disable**—FIP is disabled on the fabric.	**detail**

Table 41: show fibre-channel fip enode Output Fields *(continued)*

Field Name	Field Description	Level of Output
ENode	Information about a connected FCoE node (ENode).	All
• **ENode-MAC**	MAC address of the connected ENode.	All
• **ENode State**	Login state internal to Junos OS.	All
• **Configured ENode timer**	User-configured FIP keepalive advertisement interval in milliseconds.	**detail**
• **Running ENode timer**	Runtime interval in milliseconds of the last FIP keepalive advertisement received. This value changes every time an FKA_ADV is received.	**detail**
• **Active FIP Sessions**	Number of active FIP sessions on the ENode.	**detail**
• **VN-Port-MAC**	MAC address of a VN_Port on the ENode.	**detail**
• **Session State**	Session state internal to Junos OS.	**detail**
• **Configured FKA-ADV**	User-configured FIP keepalive advertisement interval in milliseconds.	**detail**
• **Running FKA-ADV**	Runtime interval in seconds of the last FIP keepalive advertisement received. This value changes every time an FKA_ADV is received.	**detail**
• **Configured VN-Port Timer**	Configured state of the VN_Port keepalive timer in seconds. This value is always 90 and is not user-configurable.	**detail**
• **Running VN-Port Timer**	Running state of the VN_Port keepalive timer in seconds.	**detail**
• **FCID**	Fibre Channel ID of the VN_Port.	**detail**
• **WWN**	Unique worldwide name of the VN_Port.	**detail**

Sample Output

show fibre-channel fip enode

```
user@switch> show fibre-channel fip enode 00:10:94:00:00:02
Fabric Name : proxy2 (200)
  Member
  FCF-MAC : 00:30:48:b0:ee:d2 (Interface vlan.100)
    Enode
    Enode-MAC : 00:10:94:00:00:02       State : Logged-in
        Session
        VN-Port-MAC             : 0e:fc:00:03:00:02
        VN-Port-MAC             : 0e:fc:00:03:00:01
```

show fibre-channel fip enode detail

```
user@switch> show fibre-channel fip enode 00:10:94:00:00:02 detail
Fabric Name : proxy2 (200)
  FC-MAP         : 0e:fc:00
  FKA-ADV-PERIOD : 90000       MAX-SESSIONS-PER-ENODE : 32
  FCoE trusted   : No

    Member
    FCF-MAC: 00:30:48:b0:ee:d2  (Interface vlan.100)
      FKA-ADV-PERIOD  : 90000    FKA-ADV-D-BIT-bit : Off
      Type : VF_Port Capable
      Priority : 86             State : Enable

        ENode
        Enode-MAC : 00:10:94:00:00:02   ENode State : Logged-in
         Configured ENode timer: 8000    Running ENode timer: 12226
         Active FIP Sessions : 2

          Session details
          VN-Port-MAC               : 0e:fc:00:03:00:02
          Session state             : Up
          Configured FKA-ADV        : 90000
          Running FKA-ADV           : 0
          Configured VN-Port Timer : 90000
          Running VN-Port Timer     : 213193
          FCID                      : 0x2c1a01
          WWN                       : 10:00:00:00:c9:a4:a3:cf

          VN-Port-MAC               : 0e:fc:00:03:00:01
          Session state             : Up
          Configured FKA-ADV        : 90000
          Running FKA-ADV           : 0
          Configured VN-Port Timer : 90000
          Running VN-Port Timer     : 213632
          FCID                      : 0x2c1a02
          WWN                       : 10:00:00:00:d9:b4:e3:df
```

show fibre-channel fip fabric

Syntax **show fibre-channel fip fabric** ***fabric-name***
<brief | detail>

Release Information Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Display Fibre Channel over Ethernet (FCoE) Initialization Protocol (FIP) information for a specified Fibre Channel fabric.

Options **brief | detail**—(Optional) Display the specified level of output.

fabric-name—Display information for the specified fabric.

Required Privilege Level view

Related Documentation

- Configuring FIP on an FCoE-FC Gateway on page 283
- show fibre-channel fip on page 443
- show fibre-channel fip enode on page 448
- show fibre-channel fip fcf on page 455
- show fibre-channel fip interface on page 458
- show fibre-channel fip statistics on page 461

List of Sample Output

show fibre-channel fip fabric proxy2 on page 453
show fibre-channel fip fabric detail on page 454

Output Fields Table 42 on page 452 lists the output fields for the **show fibre-channel fip fabric** command. Output fields are listed in the approximate order in which they appear.

Table 42: show fibre-channel fip fabric Output Fields

Field Name	Field Description	Level of Output
Fabric Name	Name of the fabric and in parentheses the fabric ID.	All
FC-MAP	FCoE mapped address prefix of the FCoE forwarder for the fabric.	**detail**
FKA-ADV-PERIOD	Period of time in milliseconds between FIP keepalive advertisements configured for the FC fabric.	**detail**

Table 42: show fibre-channel fip fabric Output Fields *(continued)*

Field Name		Field Description	Level of Output
Member		Information about an FCF that is a member of the fabric.	All
	• **FCF-MAC**	MAC address used in discovery advertisements.	All
	• **FKA-ADV-PERIOD**	Period of time in milliseconds between FIP keepalive advertisements configured for the FC interface.	**detail**
	• **FKA-ADV-D-BIT**	Disable FIP keepalive advertisement monitoring bit. The state is always **off**.	**detail**
	• **Type**	Type of interface: • **VF_Port Capable**—Interface can act as a VF_Port interface.	**detail**
ENode		Information about a connected FCoE node (ENode).	All
	• **ENode-MAC**	MAC address of the connected ENode.	All
	• **State**	Login state internal to Junos OS.	All
	• **VN-Port-MAC**	MAC address of a VN_Port on the ENode.	**detail**
	• **Session State**	Session state internal to Junos OS.	**detail**
	• **Configured FKA-ADV**	User-configured FIP keepalive advertisement interval in milliseconds.	**detail**
	• **Running FKA-ADV**	Runtime interval in seconds of the last FIP keepalive advertisement received. This value changes every time an FKA_ADV is received.	**detail**
	• **Configured VN-Port Timer**	Configured state of the VN_Port keepalive timer in seconds. This value is always 90 and is not user-configurable.	**detail**
	• **Running VN-Port Timer**	Running state of the VN_Port keepalive timer in seconds.	**detail**

Sample Output

show fibre-channel fip fabric proxy2

```
user@switch> show fibre-channel fip fabric proxy2
Fabric Name : proxy2 (200)
  Member
  FCF-MAC : 00:30:48:b0:ee:d2 (Interface vlan.100)
    Enode
```

```
Enode-MAC : 00:10:94:00:00:02       State : Logged-in
Enode-MAC : 00:10:94:00:00:03       State : Logged-in
```

show fibre-channel fip fabric detail

```
user@switch> show fibre-channel fip fabric proxy2 detail
Fabric Name : proxy2 (200)
  FC-MAP         : 0e:fc:00
  FKA-ADV-PERIOD : 90000

    Member
    FCF-MAC: 00:30:48:b0:ee:d2  (Interface vlan.100)
      FKA-ADV-PERIOD  : 90000     FKA-ADV-D-bit : Off
      Type : VF_Port Capable

        ENode
        Enode-MAC : 00:10:94:00:00:02   State : Logged-in

          Session details
          VN-Port-MAC              : 0e:fc:00:03:00:02
          Session state            : Up
          Configured FKA-ADV       : 90000
          Running FKA-ADV          : 0
          Configured VN-Port Timer : 90
          Running VN-Port Timer    : 0

          VN-Port-MAC              : 0e:fc:00:03:00:01
          Session state            : Up
          Configured FKA-ADV       : 90000
          Running FKA-ADV          : 0
          Configured VN-Port Timer : 90
          Running VN-Port Timer    : 0

        ENode
        Enode-MAC : 00:10:94:00:00:03   State : Logged-in

          Session details
          VN-Port-MAC              : 0e:fc:00:03:00:04
          Session state            : Up
          Configured FKA-ADV       : 90000
          Running FKA-ADV          : 0
          Configured VN-Port Timer : 90
          Running VN-Port Timer    : 0

          VN-Port-MAC              : 0e:fc:00:03:00:03
          Session state            : Up
          Configured FKA-ADV       : 90000
          Running FKA-ADV          : 0
          Configured VN-Port Timer : 90
          Running VN-Port Timer    : 0
```


show fibre-channel fip fcf

Syntax

show fibre-channel fip fcf ***fcf-mac***
<brief | detail>
<fabric ***fabric-name*****>**

Release Information

Command introduced in Junos OS Release 11.1 for the QFX Series.

Description

Display Fibre Channel over Ethernet (FCoE) Initialization Protocol (FIP) information for a specified FCoE forwarder (FCF).

Options

brief | detail—(Optional) Display the specified level of output.

fabric ***fabric-name***—(Optional) Display FCF information only for the specified fabric.

fcf-mac—Display information for the FCF specified by the MAC address.

Required Privilege Level

view

Related Documentation

- Configuring FIP on an FCoE-FC Gateway on page 283
- show fibre-channel fip on page 443
- show fibre-channel fip enode on page 448
- show fibre-channel fip fabric on page 452
- show fibre-channel fip interface on page 458
- show fibre-channel fip statistics on page 461

List of Sample Output

show fibre-channel fip fcf on page 456
show fibre-channel fip fcf detail on page 457

Output Fields

Table 43 on page 455 lists the output fields for the **show fibre-channel fip fcf** command. Output fields are listed in the approximate order in which they appear.

Table 43: show fibre-channel fip fcf Output Fields

Field Name	Field Description	Level of Output
Fabric Name	Name of the fabric and in parentheses the fabric ID.	All
FC-MAP	FCoE mapped address prefix of the FCoE forwarder for the fabric.	**detail**
FKA-ADV-PERIOD	Period of time in milliseconds between FIP keepalive advertisements configured for the FC fabric.	**detail**

Table 43: show fibre-channel fip fcf Output Fields *(continued)*

Field Name	Field Description	Level of Output
Member	Information about an FCF that is a member of the fabric.	All
• **FCF-MAC**	MAC address used in discovery advertisements.	All
• **FKA-ADV-PERIOD**	Period of time in milliseconds between FIP keepalive advertisements configured for the FC interface.	**detail**
• **FKA-ADV-D-BIT**	Disable FIP keepalive advertisement monitoring bit. The state is always **off**.	**detail**
• **Type**	Type of interface: • **VF_Port Capable**—Interface can act as a VF_Port interface.	**detail**
ENode	Information about a connected FCoE node (ENode).	All
• **ENode-MAC**	MAC address of the connected ENode.	All
• **State**	Login state internal to Junos OS.	All
• **VN-Port-MAC**	MAC address of a VN_Port on the ENode.	**detail**
• **Session State**	Session state internal to Junos OS.	**detail**
• **Configured FKA-ADV**	User-configured FIP keepalive advertisement interval in milliseconds.	**detail**
• **Running FKA-ADV**	Runtime interval in seconds of the last FIP keepalive advertisement received. This value changes every time an FKA_ADV is received.	**detail**
• **Configured VN-Port Timer**	Configured state of the VN_Port keepalive timer in seconds. This value is always 90 and is not user-configurable.	**detail**
• **Running VN-Port Timer**	Running state of the VN_Port keepalive timer in seconds.	**detail**

Sample Output

show fibre-channel fip fcf

```
user@switch> show fibre-channel fip fcf 00:30:48:b0:ee:d2
Fabric Name : proxy2 (200)
  Member
  FCF-MAC : 00:30:48:b0:ee:d2 (Interface vlan.100)
    Enode
```



```
    Enode-MAC : 00:10:94:00:00:02       State : Logged-in
    Enode-MAC : 00:10:94:00:00:03       State : Logged-in
```

show fibre-channel fip fcf detail

```
user@switch> show fibre-channel fip fcf 00:30:48:b0:ee:d2 detail
Fabric Name : proxy2 (200)
  FC-MAP         : 0e:fc:00
  FKA-ADV-PERIOD : 90000

    Member
    FCF-MAC: 00:30:48:b0:ee:d2  (Interface vlan.100)
      FKA-ADV-PERIOD  : 90000     FKA-ADV-D-bit : Off
      Type : VF_Port Capable

        ENode
        Enode-MAC : 00:10:94:00:00:02   State : Logged-in

          Session details
          VN-Port-MAC             : 0e:fc:00:03:00:02
          Session state           : Up
          Configured FKA-ADV      : 90000
          Running FKA-ADV         : 0
          Configured VN-Port Timer : 90
          Running VN-Port Timer    : 0

          VN-Port-MAC             : 0e:fc:00:03:00:01
          Session state           : Up
          Configured FKA-ADV      : 90000
          Running FKA-ADV         : 0
          Configured VN-Port Timer : 90
          Running VN-Port Timer    : 0

        ENode
        Enode-MAC : 00:10:94:00:00:03   State : Logged-in

          Session details
          VN-Port-MAC             : 0e:fc:00:03:00:04
          Session state           : Up
          Configured FKA-ADV      : 90000
          Running FKA-ADV         : 0
          Configured VN-Port Timer : 90
          Running VN-Port Timer    : 0

          VN-Port-MAC             : 0e:fc:00:03:00:03
          Session state           : Up
          Configured FKA-ADV      : 90000
          Running FKA-ADV         : 0
          Configured VN-Port Timer : 90
          Running VN-Port Timer    : 0
```


show fibre-channel fip interface

Syntax

show fibre-channel fip interface *interface-name*
<**brief** | **detail**>
<**enode** *enode-mac*>
<**fabric** *fabric-name*>
<**vn-port** *vn-port-mac*>

Release Information

Command introduced in Junos OS Release 11.1 for the QFX Series.

Description

Display Fibre Channel over Ethernet (FCoE) Initialization Protocol (FIP) information for a specified interface.

Options

brief | detail—(Optional) Display the specified level of output.

enode-mac—MAC address of the ENode.

fabric *fabric-name*—(Optional) Display interface information only for the specified fabric.

interface-name—Display information for the specified interface.

vn-port-mac—MAC address of the VN_Port.

Required Privilege Level

view

Related Documentation

- Configuring FIP on an FCoE-FC Gateway on page 283
- show fibre-channel fip on page 443
- show fibre-channel fip enode on page 448
- show fibre-channel fip fabric on page 452
- show fibre-channel fip fcf on page 455
- show fibre-channel fip statistics on page 461
- clear fibre-channel fip vn-port on page 394

List of Sample Output

show fibre-channel fip interface on page 459
show fibre-channel fip interface detail on page 460

Output Fields

Table 44 on page 458 lists the output fields for the **show fibre-channel fip interface** command. Output fields are listed in the approximate order in which they appear.

Table 44: show fibre-channel fip interface Output Fields

Field Name	Field Description	Level of Output
Fabric Name	Name of the fabric and in parentheses the fabric ID.	All
FC-MAP	FCoE mapped address prefix of the FCoE forwarder for the fabric.	**detail**

Table 44: show fibre-channel fip interface Output Fields *(continued)*

Field Name		Field Description	Level of Output
FKA-ADV-PERIOD		Period of time in milliseconds between FIP keepalive advertisements configured for the FC fabric.	**detail**
Member		Information about an FCF that is a member of the fabric.	All
	• **FCF-MAC**	MAC address used in discovery advertisements.	All
	• **FKA-ADV-PERIOD**	Period of time in milliseconds between FIP keepalive advertisements configured for the FC interface.	**detail**
	• **FKA-ADV-D-BIT**	Disable FIP keepalive advertisement monitoring bit. The state is always **off**.	**detail**
	• **Type**	Type of interface: • **VF_Port Capable**—Interface can act as a VF_Port interface.	**detail**
ENode		Information about a connected FCoE node (ENode).	All
	• **ENode-MAC**	MAC address of the connected ENode.	All
	• **State**	Login state internal to Junos OS.	All
	• **VN-Port-MAC**	MAC address of a VN_Port on the ENode.	**detail**
	• **Session State**	Session state internal to Junos OS.	**detail**
	• **Configured FKA-ADV**	User-configured FIP keepalive advertisement interval in milliseconds.	**detail**
	• **Running FKA-ADV**	Runtime interval in seconds of the last FIP keepalive advertisement received. This value changes every time an FKA_ADV is received.	**detail**
	• **Configured VN-Port Timer**	Configured state of the VN_Port keepalive timer in seconds. This value is always 90 and is not user-configurable.	**detail**
	• **Running VN-Port Timer**	Running state of the VN_Port keepalive timer in seconds.	**detail**

Sample Output

show fibre-channel fip interface

```
user@switch> show fibre-channel fip interface vlan.100
Fabric Name : proxy2 (200)
  Member
```



```
  FCF-MAC : 00:30:48:b0:ee:d2 (Interface vlan.100)
    Enode
    Enode-MAC : 00:10:94:00:00:02       State : Logged-in
    Enode-MAC : 00:10:94:00:00:03       State : Logged-in
```

show fibre-channel fip interface detail

```
user@switch> show fibre-channel fip interface vlan.100 detail
Fabric Name : proxy2 (200)
  FC-MAP         : 0e:fc:00
  FKA-ADV-PERIOD : 90000

    Member
    FCF-MAC: 00:30:48:b0:ee:d2  (Interface vlan.100)
      FKA-ADV-PERIOD  : 90000     FKA-ADV-D-bit : Off
      Type : VF_Port Capable

        ENode
        Enode-MAC : 00:10:94:00:00:02   State : Logged-in

          Session details
          VN-Port-MAC              : 0e:fc:00:03:00:02
          Session state            : Up
          Configured FKA-ADV       : 90000
          Running FKA-ADV          : 0
          Configured VN-Port Timer : 90
          Running VN-Port Timer    : 0

          VN-Port-MAC              : 0e:fc:00:03:00:01
          Session state            : Up
          Configured FKA-ADV       : 90000
          Running FKA-ADV          : 0
          Configured VN-Port Timer : 90
          Running VN-Port Timer    : 0

        ENode
        Enode-MAC : 00:10:94:00:00:03   State : Logged-in

          Session details
          VN-Port-MAC              : 0e:fc:00:03:00:04
          Session state            : Up
          Configured FKA-ADV       : 90000
          Running FKA-ADV          : 0
          Configured VN-Port Timer : 90
          Running VN-Port Timer    : 0

          VN-Port-MAC              : 0e:fc:00:03:00:03
          Session state            : Up
          Configured FKA-ADV       : 90000
          Running FKA-ADV          : 0
          Configured VN-Port Timer : 90
          Running VN-Port Timer    : 0
```


show fibre-channel fip statistics

Syntax **show fibre-channel fip statistics**
<**fabric** *fabric-name*>

Release Information Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Display Fibre Channel over Ethernet Initialization Protocol (FIP) statistics.

Options **fabric** ***fabric-name***—(Optional) Display output only for the specified fabric.

Required Privilege Level view

Related Documentation

- show fibre-channel fip on page 443
- show fibre-channel fip enode on page 448
- show fibre-channel fip fabric on page 452
- show fibre-channel fip fcf on page 455
- show fibre-channel fip interface on page 458
- clear fibre-channel fip statistics on page 393

List of Sample Output show fibre-channel fip statistics on page 463

Output Fields Table 45 on page 461 lists the output fields for the **show fibre-channel fip statistics** command. Output fields are listed in the approximate order in which they appear.

Table 45: show fibre-channel fip statistics Output Fields

Field Name	Field Description
Fabric name	Name of the fabric.
Interface name	Name of the FCoE VLAN interface.

Table 45: show fibre-channel fip statistics Output Fields *(continued)*

Field Name		Field Description
FIP Message Type		Type of FIP message for the displayed row of statistics..
	• **MDS**	Number of multicast discovery solicitations.
	• **UDS**	Number of unicast discovery solicitations.
	• **FLOGI**	Number of fabric login (FLOGI) messages.
	• **FDISC**	Number of fabric discovery (FDISC) messages.
	• **LOGO**	Number of fabric logout (LOGO) messages.
	• **ENODE KA**	Number of ENode keepalive messages.
	• **VN_Port KA**	Number of VN_Port keepalive messages.
	• **MDA**	Number of multicast discovery advertisements.
	• **UDA**	Number of unicast discovery advertisements.
	• **FLOGI ACC**	Number of fabric login requests accepted.
	• **FLOGI RJT**	Number of fabric login requests rejected.
	• **FDISC ACC**	Number of fabric discovery requests accepted.
	• **FDISC RJT**	Number of fabric discovery requests rejected.
	• **LOGO ACC**	Number of logout requests accepted.
	• **LOGO RJT**	Number of logout requests rejected.
	• **CVL**	Number of clear virtual links (CVL) messages.
	• **CVL ALL**	Number of CVL all messages.
Received		Number of messages received.
Sent		Number of messages sent.
Rx errors		Number of receive errors.

Table 45: show fibre-channel fip statistics Output Fields *(continued)*

Field Name		Field Description
Dropped		Number of dropped messages. NOTE: One cause of dropped messages is that the system limits the number of discovery solicitations (MDS and UDS) it accepts to a maximum of 100 outstanding requests at any given time. If the system has 100 discovery solicitations outstanding, the system does not respond to new discovery solicitations. Instead, the system drops new discovery solicitations and reports the number of dropped discovery solicitations in this field. When there are fewer than 100 outstanding discovery solicitations, the system responds to new requests as usual with a discovery advertisement.
General Statistics	**Number of frames recvd with invalid src-mac**	Number of frames received that have an invalid source media access control (MAC) address.
	Number of frames recvd with invalid version	Number of FIP frames received with an Invalid FIP version.
	Number of frames recvd with invalid opcode	Number of FIP validation descriptors with an invalid opcode received.
	Number of frames recvd with invalid subcode	Number of FIP validation descriptors with an invalid subcode received.
	Number of frames recvd on inactive FCF	Number of frames received on a logical interface if FIP is not active on that logical interface (for example, if a WWN is not allocated to that logical interface).

Sample Output

show fibre-channel fip statistics

```
user@switch> show fibre-channel fip statistics
Fabric name: proxy2

Interface name: vlan.100
FIP Message type      Received      Sent      Rx errors     Dropped
MDS                   22236         0         0             17089
UDS                   0             0         0             0
FLOGI                 1257          0         8             0
FDISC                 0             0         0             0
LOGO                  0             0         0             0
ENODE KA              455           0         6             0
VN_Port KA            22            0         0             0
MDA                   0             243       0             0
UDA                   0             5147      0             0
FLOGI ACC             0             376       0             0
FLOGI RJT             0             881       0             0
FDISC ACC             0             0         0             0
FDISC RJT             0             0         0             0
LOGO ACC              0             0         0             0
LOGO RJT              0             0         0             0
```

```
CVL                     0            374      0           0
CVL ALL                 0            380      0           0


General Statistics:

Number of frame recvd with invalid src-mac:          0
Number of frame recvd with invalid version:          0
Number of frame recvd with invalid opcode:           0
Number of frame recvd with invalid subcode:          0
Number of frame recvd on inactive FCF:               0
```


show fibre-channel flogi fport

Syntax

show fibre-channel flogi fport
<**fabric** *fabric-name*>

Release Information

Command introduced in Junos OS Release 11.1 for the QFX Series.

Description

Display Fibre Channel fabric login (FLOGI) F_Port information.

Options

fabric ***fabric-name***—(Optional) Display output only for the specified fabric.

Required Privilege Level

view

Related Documentation

- show fibre-channel flogi nport on page 467
- show fibre-channel flogi statistics on page 469

List of Sample Output

show fibre-channel flogi fport on page 465

Output Fields

Table 46 on page 465 lists the output fields for the **show fibre-channel flogi fport** command. Output fields are listed in the approximate order in which they appear.

Table 46: show fibre-channel flogi fport Output Fields

Field Name	Field Description
Fabric	Name of the fabric.
Interface	Name of the switch VF_Port interface.
Mac-Address	Media access control (MAC) address of the ENode.
State	Interface physical state: **up** or **down**.
Logins	Number of logins to the VF_Port.
NPIV	N_Port ID virtualization (NPIV) state: **Yes** or **No**.
FLOGI-Port-WWN	Unique worldwide name (WWN) of the VN_Port performing fabric login (FLOGI) to the switch VF_Port.

Sample Output

show fibre-channel flogi fport

```
user@switch> show fibre-channel flogi fport
Fabric: proxy2
Interface       Mac-Address         State  Logins   NPIV  FLOGI-Port-WWN
vlan.100       00:10:94:00:00:02   Up     2        Yes   20:00:10:94:00:01:00:01
vlan.100       00:10:94:00:00:03   Up     2        Yes   20:00:10:94:00:02:00:01
```

show fibre-channel flogi nport

Syntax

show fibre-channel flogi nport
<brief | detail>
<fabric *fabric-name*>

Release Information

Command introduced in Junos OS Release 11.1 for the QFX Series.

Description

Display Fibre Channel fabric login (FLOGI) VN_Port information.

Options

brief | detail—(Optional) Display the specified level of output.

fabric *fabric-name*—(Optional) Display output only for the specified fabric.

Required Privilege Level

view

Related Documentation

- show fibre-channel flogi fport on page 465
- show fibre-channel flogi statistics on page 469

List of Sample Output

show fibre-channel flogi nport on page 468
show fibre-channel flogi nport detail on page 468

Output Fields

Table 47 on page 467 lists the output fields for the **show fibre-channel flogi nport** command. Output fields are listed in the approximate order in which they appear.

Table 47: show fibre-channel flogi nport Output Fields

Field Name	Field Description	Level of Output
Fabric	Name of the fabric.	All
Virtual-switch	Name of the fabric.	**detail**
Interface	Name of the VF_Port interface.	All
FCID	VN_Port Fibre Channel identifier provided by the Fibre Channel over Ethernet Forwarder (FCoE forwarder) or the Fibre Channel switch.	All
Port-WWN	Unique worldwide name (WWN) of the VN_Port.	All
Node-WWN	Unique WWN of the node hosting the VN_Port.	All
State or **Flogi-state**	Login state internal to Junos OS.	All
FLOGI-Port-WWN	Unique worldwide name (WWN) of the VN_Port performing fabric login (FLOGI) to the switch VF_Port.	**detail**

Sample Output

show fibre-channel flogi nport

```
user@switch> show fibre-channel flogi nport
Fabric: proxy2
Interface     FCID      Port-WWN                 Node-WWN                 State
vlan.100      0x030001  20:00:10:94:00:01:00:01  10:00:10:94:00:00:00:01 online
vlan.100      0x030002  20:00:10:94:00:01:00:05  10:00:10:94:00:00:00:01 online
vlan.100      0x030003  20:00:10:94:00:02:00:01  10:00:10:94:00:00:00:02 online
vlan.100      0x030004  20:00:10:94:00:02:00:05  10:00:10:94:00:00:00:02 online
```

show fibre-channel flogi nport detail

```
user@switch> show fibre-channel flogi nport detail
Fabric: proxy2
    Virtual-switch: proxy2

        Interface: vlan.100
        Flogi-state: online
        FCID: 0x030001
        Port-WWN: 20:00:10:94:00:01:00:01
        Node-WWN: 10:00:10:94:00:00:00:01
        FLOGI-Port-WWN: 20:00:10:94:00:01:00:01

        Interface: vlan.100
        Flogi-state: online
        FCID: 0x030002
        Port-WWN: 20:00:10:94:00:01:00:05
        Node-WWN: 10:00:10:94:00:00:00:01
        FLOGI-Port-WWN: 20:00:10:94:00:01:00:01

        Interface: vlan.100
        Flogi-state: online
        FCID: 0x030003
        Port-WWN: 20:00:10:94:00:02:00:01
        Node-WWN: 10:00:10:94:00:00:00:02
        FLOGI-Port-WWN: 20:00:10:94:00:02:00:01

        Interface: vlan.100
        Flogi-state: online
        FCID: 0x030004
        Port-WWN: 20:00:10:94:00:02:00:05
        Node-WWN: 10:00:10:94:00:00:00:02
        FLOGI-Port-WWN: 20:00:10:94:00:02:00:01
```

show fibre-channel flogi statistics

Syntax

show fibre-channel flogi statistics
<fabric *fabric-name*>

Release Information

Command introduced in Junos OS Release 11.1 for the QFX Series.

Description

Display Fibre Channel fabric login (FLOGI) statistics.

Options

fabric *fabric-name*—(Optional) Display output only for the specified fabric.

Required Privilege Level

view

Related Documentation

- show fibre-channel flogi fport on page 465
- show fibre-channel flogi nport on page 467
- clear fibre-channel flogi statistics on page 395

List of Sample Output

show fibre-channel flogi statistics on page 470

Output Fields

Table 48 on page 469 lists the output fields for the **show fibre-channel flogi statistics** command. Output fields are listed in the approximate order in which they appear.

Table 48: show fibre-channel flogi statistics Output Fields

Field Name	Field Description
Fabric	Name of the fabric.
FLOGI-Server Message type	Type of message: • **FLOGI**—Fabric login (FLOGI) messages. • **FDISC**—Fabric discovery (FDISC) messages. • **FLOGO**—Fabric logout messages. • **FLOGO-LS-ACC**—Fabric logout link service accept messages. • **LS-Accept**—Link service accept messages. • **LS-Reject**—Link service reject messages. • **invalid**—Invalid messages.
Received	Number of messages received for a given message type.
Sent	Number of messages sent for a given message type.
Fabric	Name of the fabric.
Rx errors	Number of receive errors for a given type of message.

Table 48: show fibre-channel flogi statistics Output Fields *(continued)*

Field Name		Field Description
General Statistics	• Number of FC2 Header Parse Errors	Number of errors parsing the FC-2 header.
	• Number of FLOGI Parse Errors	Number of errors parsing fabric login requests.
	• Number of FDISC Parse Errors	Number of errors parsing fabric discovery requests.
	• Number of FLOGO Parse Errors	Number of errors parsing fabric logout requests.
	• Number of Logins Discarded as Domain-ID not available	Number of discarded logins due to unavailability of a domain ID.
	• Number of Logins Discarded as FCID not available	Number of discarded logins due to the unavailability of a Fibre Channel ID.
	• Number of FCID requests deferred	Number of deferred FCID requests.
	• Number of deferred FCID requests failed	Number of deferred FCID requests that failed.

Sample Output

show fibre-channel flogi statistics

```
user@switch> show fibre-channel flogi statistics
Fabric: proxy2

FLOGI-Server Message type    Received       Sent   Rx errors
FLOGI                        2              0      0
FDISC                        2              0      0
FLOGO                        0              0      0
FLOGO-LS-ACC                 0              0      0
LS-Accept                    0              4      0
LS-Reject                    0              0      0
invalid                      0              0      0

General Statistics:

Number of FC2 Header Parse Errors:                         0
Number of FLOGI Parse Errors:                              0
Number of FDISC Parse Errors:                              0
Number of FLOGO Parse Errors:                              0
Number of Logins Discarded as Domain-ID not available:     0
Number of Logins Discarded as FCID not available:          0
Number of FCID requests deferred:                          0
Number of deferred FCID requests failed:                   0
```


show fibre-channel interfaces

Syntax

<brief | detail>
<fabric *fabric-name*>
show fibre-channel interfaces *interface-name*

Release Information Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Display information about Fibre Channel (FC) interfaces.

Options

brief | detail—(Optional) Display the specified level of output.

fabric *fabric-name*—(Optional) Display output only for the specified fabric.

interface-name—Display output for the specified interface.

Required Privilege Level view

Related Documentation

- Example: Setting Up Fibre Channel and FCoE VLAN Interfaces in an FCoE-FC Gateway Fabric on page 171
- Configuring a Physical Fibre Channel Interface on page 268
- Configuring an FCoE VLAN Interface on an FCoE-FC Gateway on page 272
- Configuring a Fibre Channel Interface on page 269
- Assigning Interfaces to a Fibre Channel Fabric on page 276

List of Sample Output

show fibre-channel interfaces on page 473
show fibre-channel interfaces detail on page 474

Output Fields Table 49 on page 472 lists the output fields for the **show fibre-channel interfaces** command. Output fields are listed in the approximate order in which they appear.

Table 49: show fibre-channel interfaces Output Fields

Field Name	Field Description	Level of Output
Interface	Name of the FC interface.	All
Idx or **Index**	Interface index internal to Junos OS.	All
Type	Type of interface: • **FC**—Native FC interface • **FCOE**—Fibre Channel over Ethernet interface	All
Native Fabric-id	Identification number of the QFX Series fabric.	All
NPIV	N_Port ID virtualization (NPIV) state: **Yes** or **No**.	All

Table 49: show fibre-channel interfaces Output Fields *(continued)*

Field Name	Field Description	Level of Output
Config-Mode	User-configured port mode: • **F**—The port is configured as a VF_Port, an FCoE port connected to FCoE devices. • **NP**—The port is configured as a proxy N_Port (NP_Port), a native FC port connected to an FC switch.	All
Oper-Mode	Operational port mode: • **F**—The port is operating as a VF_Port, an FCoE port connected to FCoE devices. • **NP**—The port is operating as an NP_Port, a native FC port connected to an FC switch or an FCoE forwarder (FCF).	All
State	Interface state: **up** or **down**.	All
WWN	Unique worldwide name (WWN) of the port.	**detail**
FSM-State	Finite state machine state, internal to Junos OS.	**detail**
Class ID	Fibre Channel interface class ID, internal to Junos OS.	**detail**
BB_SC_N	Buffer-to-buffer state change number.	**detail**
Tx B2B credits	Number of buffer-to-buffer credits advertised by the neighbor switch that is connected to the FC interface.	**detail**
Fabric	Name of the fabric.	**detail**
Remote-MAC	Media access control (MAC) address of the remotely connected FCoE device VN_Port interface.	**detail**
Tagging	Not used. Value is shown as **untagged**.	**detail**
Mode	Logical interface (LIF) mode of operation.	**detail**
H/W token	Unique identifier for the FCoE VLAN interface, internal to Junos OS.	**detail**

Sample Output

show fibre-channel interfaces

```
user@switch> show fibre-channel interfaces
                              Native               Config  Oper
Interface         Idx   Type  Fabric-id     NPIV   Mode    Mode    State
fc-0/0/1.0        70    FC    200           YES    NP      NP      up
vlan.100          84    FCOE  200           YES    F       F       up
```


show fibre-channel interfaces detail

```
user@switch> show fibre-channel interfaces detail
Interface: fc-0/0/1.0, Index: 70, Type: FC, Native Fabric-id: 200
NPIV: YES, Config-Mode: NP, Oper-Mode: NP, State: up
WWN: 10:00:00:15:17:a9:98:64, FSM-State: up, Class ID: 1, BB_SC_N: 0
Tx B2B credits: 32
  Fabric                Remote-MAC         Tagging    Mode   Oper state
  proxy2                -                  untagged   NP     up

Interface: vlan.100, Index: 84, Type: FCOE, Native Fabric-id: 200
NPIV: YES, Config-Mode: F, Oper-Mode: F, State: up
WWN: 10:00:00:30:48:b0:ee:d2, FSM-State: up
H/W token: 13
  Fabric                Remote-MAC         Tagging    Mode   Oper state
  proxy2                00:10:94:00:00:02  untagged   VF     up
  proxy2                00:10:94:00:00:03  untagged   VF     up
```


show fibre-channel next-hops

Syntax **show fibre-channel next-hops**

Release Information Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Display Fibre Channel next-hop route information.

Required Privilege Level view

Related Documentation

- show fibre-channel routes on page 477
- show route forwarding-table family fibre-channel on page 516

List of Sample Output show fibre-channel next-hops on page 475

Output Fields Table 50 on page 475 lists the output fields for the **show fibre-channel next-hops** command. Output fields are listed in the approximate order in which they appear.

Table 50: show fibre-channel next-hops Output Fields

Field Name	Field Description
Type	Type of next hop internal to Junos OS.
State	State of the NP_Port interface: • **Active**—The interface is online. • **Deleted**—The interface is deleted.
Interface	Name of the interface.
Mac-Address	Media access control (MAC) address of the interface.
Index	Next-hop index identifier.
Ref-count	Reference count internal to Junos OS.
Flags	Flags internal to Junos OS.

Sample Output

show fibre-channel next-hops

```
user@switch> show fibre-channel next-hops
Type     State  Interface     Mac-Address         Index  Ref-count Flags
intf     Active fc-0/0/0.0                        0       1
ucast    Active vlan.100      00:15:17:a9:98:64   674     1        kernel, self
ucast    Active vlan.100      0e:fc:00:03:00:01   675     1        kernel, self
ucast    Active vlan.100      0e:fc:00:03:00:02   676     1        kernel, self
ucast    Active vlan.100      0e:fc:00:03:00:03   677     1        kernel, self
ucast    Active vlan.100      0e:fc:00:03:00:04   678     1        kernel, self
```


show fibre-channel routes

Syntax

show fibre-channel routes
<fabric *fabric-name*>

Release Information Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Display Fibre Channel route information.

Options **fabric *fabric-name***—(Optional) Display output only for the specified fabric.

Required Privilege Level view

Related Documentation

- show fibre-channel next-hops on page 475
- show route forwarding-table family fibre-channel on page 516

List of Sample Output show fibre-channel routes on page 477

Output Fields Table 51 on page 477 lists the output fields for the **show fibre-channel routes** command. Output fields are listed in the approximate order in which they appear.

Table 51: show fibre-channel routes Output Fields

Field Name	Field Description
Fabric	Name of the fabric.
Route-prefix	Route destination.
State	State of the NP_Port interface: • **Active**—The interface is online. • **Deleted**—The interface is deleted.
Interface	Name of the interface.
Mac-Address	Media access control (MAC) address of the interface.
Index	Next-hop index identifier.
Flags	Flags internal to Junos OS.

Sample Output

show fibre-channel routes

```
user@switch> show fibre-channel routes
Fabric: proxy2
Route-prefix      State     Interface     Mac-Address        Index    Flags
0x030000/24       Active    fc-0/0/0.0    00:15:17:a9:98:64  674      kernel
```



```
0x030001/24        Active    vlan.100      0e:fc:00:03:00:01   675     kernel
0x030002/24        Active    vlan.100      0e:fc:00:03:00:02   676     kernel
0x030003/24        Active    vlan.100      0e:fc:00:03:00:03   677     kernel
0x030004/24        Active    vlan.100      0e:fc:00:03:00:04   678     kernel
```


show fibre-channel proxy fabric-state

Syntax **show fibre-channel proxy fabric-state**
<fabric *fabric-name*>

Release Information Command introduced in Junos OS Release 12.1 for the QFX Series.

Description Display Fibre Channel (FC) proxy fabric state information.

Options **fabric *fabric-name***—(Optional) Display output only for the specified fabric.

Required Privilege Level view

Related Documentation

- Monitoring Fibre Channel Interface Load Balancing on page 383
- **show fibre-channel proxy login-table on page 483**
- **show fibre-channel proxy np-port on page 486**
- **show fibre-channel proxy statistics on page 489**

List of Sample Output **show fibre-channel proxy fabric-state on page 481**
show fibre-channel proxy fabric-state fabric on page 481

Output Fields Table 52 on page 479 lists the output fields for the **show fibre-channel proxy fabric-state** command. Output fields are listed in the approximate order in which they appear.

Table 52: show fibre-channel proxy fabric-state Output Fields

Field Name	Field Description
Fabric	Name of the fabric.
Fabric-id	Fabric ID number.
Proxy load balance algorithm	Load-balancing algorithm used on the FCoE-FC gateway FC fabric: • Simple—Load balancing is based on the weighted utilization (load) of the NP_Ports connected to an FC fabric. Each new FLOGI or FDISC is assigned to the least-loaded link. On a link load rebalance, only the sessions that need to be moved to another link are logged out. When those sessions log in again, they are placed on active NP_Port interfaces in a balanced manner. • ENode-based—Load balancing is based on the ENode FLOGI. When an ENode logs in to the fabric, all subsequent FDISC sessions associated with that ENode are placed on the same link as the ENode FLOGI session, regardless of the link load. New ENode FLOGIs are placed on the least-loaded link. On a link load rebalance, all sessions are logged out. When the sessions log in again, they are placed on active NP_Port interfaces in a balanced manner. • FLOGI-based—Load balancing is based on the ENode FLOGI. When an ENode logs in to the fabric, all subsequent FDISC sessions associated with that ENode are placed on the same link as the ENode FLOGI session, regardless of the link load. New ENode FLOGIs are placed on the least-loaded link. On a link load rebalance, only the sessions that need to be moved to another link are logged out. When those sessions log in again, they are placed on active NP_Port interfaces in a balanced manner.

Table 52: show fibre-channel proxy fabric-state Output Fields *(continued)*

Field Name	Field Description
Fabric WWN verification	Fabric worldwide name (WWN) verification check state on the FCoE-FC gateway fabric: • Yes—Fabric WWN verification check is enabled. • No—Fabric WWN verification check is disabled.
Auto load rebalance enabled	Automated link load rebalancing configuration for the FCoE-FC gateway fabric: • No—Automated load balancing is disabled (default state). • Yes—Automated load balancing is enabled.
Last rebalance start-time	Time that the last link load rebalance began on the FCoE-FC gateway fabric: • Never—The link load has never been rebalanced. • Timestamp value—Time the last link load rebalancing started.
Last rebalance end-time	Time that the last link load rebalance ended on the FCoE-FC gateway fabric: • Never—The link load has never been rebalanced. • Timestamp value—Time the last link load rebalancing ended.
Last rebalance trigger	Event that triggered the last link load rebalance on the FCoE-FC gateway fabric: • None—The link load has never been rebalanced. • Config-CLI—Configure (enable) automated load balancing. • Request-CLI—Rebalance requested from the CLI using the **request fibre-channel proxy load-rebalance fabric *fabric-name*** operational command. • Preview-CLI—Rebalancing *dry run* requested from the CLI using the **request fibre-channel proxy load-rebalance dry-run fabric *fabric-name*** operational command. Indicates that the switch completed the dry run. A dry run simulates a link load rebalance and displays a list of sessions that might be affected if you request an actual rebalance. • Link-up—New FC link (NP_Port) up on the FCoE-FC gateway fabric, which causes a rebalance to distribute sessions to the new link. • Restore-complete—If the FC process on the switch restarts, the switch attempts to restore the session state that existed before the restart. When automated rebalance is enabled, restore-complete indicates that the sessions have been restored and rebalanced.
Last rebalance trigger-time	Time that the last link load rebalance was triggered on the FCoE-FC gateway fabric: • Never—Link load rebalancing has never been triggered. • Timestamp value—Time the last link load rebalancing was triggered.

Table 52: show fibre-channel proxy fabric-state Output Fields *(continued)*

Field Name	Field Description
Last rebalance trigger-result	Result of the last trigger event on the FCoE-FC gateway fabric: • Never—Link load rebalancing has never been triggered. • Not-configured—Automated rebalancing is not configured on the FCoE-FC gateway fabric. • Not-required—Last rebalance trigger did not require rebalancing the link load (the link load was already balanced across the active NP_Port links). • In-progress—Link load rebalancing is in progress and has not finished yet. • Restore-in-progress—The switch is recovering from an FC process restart and is in the process of restoring the sessions to the active NP_Port links. • Success—Link load rebalancing was successful. • Logged-out-all—All sessions have been logged out. • Preview-complete—The switch has finished simulating a dry run rebalancing request from the CLI (**request fibre-channel proxy load-rebalance dry-run fabric** ***fabric-name*** operational command) and reported the sessions that might be affected if you request an actual link load rebalance. • Fabric-deletion-in-progress—FCoE-FC gateway fabric is in the process of being deleted. NOTE: A trigger event does not necessarily result in a rebalance action. Link load rebalancing only occurs if the NP_Port interface session load is not balanced at the time of the trigger event.

Sample Output

show fibre-channel proxy fabric-state

```
user@switch> show fibre-channel proxy fabric-state
Fabric: san_fab1, Fabric-id: 10
Proxy load balance algorithm: Simple, Fabric WWN verification: Yes
Auto load rebalance enabled  : No
Last rebalance start-time    : Never
Last rebalance end-time      : Never
Last rebalance trigger       : Link-up
Last rebalance trigger-time  : Mon Sep 10 21:42:30 2012 usec: 814602
Last rebalance trigger-result: Not-configured

Fabric: san_fab2, Fabric-id: 20
Proxy load balance algorithm: ENode based, Fabric WWN verification: Yes
Auto load rebalance enabled  : No
Last rebalance start-time    : Never
Last rebalance end-time      : Never
Last rebalance trigger       : Link-up
Last rebalance trigger-time  : Mon Sep 17 17:23:35 2012 usec: 619684
Last rebalance trigger-result: Not-configured
```

show fibre-channel proxy fabric-state fabric

```
user@switch> show fibre-channel proxy fabric-state fabric fc_fabric_100
Fabric: fc_fabric_100, Fabric-id: 100
Proxy load balance algorithm: FLOGI based, Fabric WWN verification: No
Auto load rebalance enabled  : Yes
Last rebalance start-time    : Never
Last rebalance end-time      : Never
Last rebalance trigger       : Config-CLI
Last rebalance trigger-time  : Fri Nov  2 08:56:16 2012 usec: 004487
Last rebalance trigger-result: Not-required
```


show fibre-channel proxy login-table

Syntax

show fibre-channel proxy login-table
<brief | detail>
<fabric *fabric-name*>
<interface *interface-name*>

Release Information Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Display Fibre Channel (FC) proxy fabric login table information.

Options

brief | detail—(Optional) Display the specified level of output.

fabric *fabric-name*—(Optional) Display output only for the specified fabric.

interface *interface-name*—(Optional) Display output only for the specified interface.

Required Privilege Level view

Related Documentation

- Configuring an FCoE-FC Gateway Fibre Channel Fabric on page 266
- show fibre-channel proxy fabric-state on page 479
- show fibre-channel proxy np-port on page 486
- show fibre-channel proxy statistics on page 489

List of Sample Output

show fibre-channel proxy login-table on page 484
show fibre-channel proxy login-table detail on page 484

Output Fields Table 53 on page 483 lists the output fields for the **show fibre-channel proxy login-table** command. Output fields are listed in the approximate order in which they appear.

Table 53: show fibre-channel proxy login-table Output Fields

Field Name	Field Description	Level of Output
Fabric	Name of the fabric.	All
Fabric-id	Fabric ID number.	All
F-Port	One of the following two values: • VF_Port interface connected to the Fibre Channel over Ethernet (FCoE) host, shown as the FCoE VLAN interface. • QFX Series FC port that is logged in to the FC switch, shown by a hyphen (-) to indicate that it is not the FCoE device VN_Port.	All
FCID	VN_Port Fibre Channel identifier provided by the Fibre Channel over Ethernet (FCoE) forwarder (FCF) or the Fibre Channel switch.	All

Table 53: show fibre-channel proxy login-table Output Fields *(continued)*

Field Name	Field Description	Level of Output
Port-WWN	Unique worldwide name (WWN) of the VN_Port.	All
Node-WWN	Unique WWN of the node hosting the VN_Ports.	**detail**
NP-Port	NP_Port interface connected to the FCoE forwarder (FCF) or the Fibre Channel switch.	All
Class	FLOGI service class.	**detail**
Fabric port WWN	Unique WWN of the fabric port (VF_Port).	**detail**
Fabric WWN	Unique WWN of the fabric generated by the FCF.	**detail**

Sample Output

show fibre-channel proxy login-table

```
user@switch> show fibre-channel proxy login-table
Fabric: proxy2, Fabric-id: 200
F-Port                  FCID     Port-WWN                NP-Port
-                       0x030000 10:00:00:15:17:a9:98:64 fc-0/0/0.0
vlan.100                0x030001 20:00:10:94:00:01:00:01 fc-0/0/0.0
vlan.100                0x030002 20:00:10:94:00:01:00:05 fc-0/0/0.0
vlan.100                0x030003 20:00:10:94:00:02:00:01 fc-0/0/0.0
vlan.100                0x030004 20:00:10:94:00:02:00:05 fc-0/0/0.0
```

show fibre-channel proxy login-table detail

```
user@switch> show fibre-channel proxy login-table detail
Fabric: proxy2, Fabric-id: 200

    FCID:              0x030000
    F-Port:            -
    NP-Port:           fc-0/0/0.0
    Port WWN:          10:00:00:15:17:a9:98:64
    Node WWN:          20:c8:11:22:33:44:55:66
    Class:             3
    Fabric port WWN:   10:00:00:15:17:a9:99:48
    Fabric WWN:        00:0a:df:ff:0b:11:22:34

    FCID:              0x030001
    F-Port:            vlan.100
    NP-Port:           fc-0/0/0.0
    Port WWN:          20:00:10:94:00:01:00:01
    Node WWN:          10:00:10:94:00:00:00:01
    Class:             3
    Fabric port WWN:   10:00:00:15:17:a9:99:48
    Fabric WWN:        00:0a:df:ff:0b:11:22:34

    FCID:              0x030002
    F-Port:            vlan.100
    NP-Port:           fc-0/0/0.0
    Port WWN:          20:00:10:94:00:01:00:05
```

```
Node WWN:         10:00:10:94:00:00:00:01
Class:            3
Fabric port WWN:  10:00:00:15:17:a9:99:48
Fabric WWN:       00:0a:df:ff:0b:11:22:34

FCID:             0x030003
F-Port:           vlan.100
NP-Port:          fc-0/0/0.0
Port WWN:         20:00:10:94:00:02:00:01
Node WWN:         10:00:10:94:00:00:00:02
Class:            3
Fabric port WWN:  10:00:00:15:17:a9:99:48
Fabric WWN:       00:0a:df:ff:0b:11:22:34

FCID:             0x030004
F-Port:           vlan.100
NP-Port:          fc-0/0/0.0
Port WWN:         20:00:10:94:00:02:00:05
Node WWN:         10:00:10:94:00:00:00:02
Class:            3
Fabric port WWN:  10:00:00:15:17:a9:99:48
Fabric WWN:       00:0a:df:ff:0b:11:22:34
```

show fibre-channel proxy np-port

Syntax

show fibre-channel proxy np-port
<brief | detail>
<fabric *fabric-name*>
<interface *interface-name*>

Release Information Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Display Fibre Channel gateway fabric proxy Node Port (NP_Port) information.

Options

brief | detail—(Optional) Display the specified level of output.

fabric *fabric-name*—(Optional) Display output only for the specified fabric.

interface *interface-name*—(Optional) Display output only for the specified interface.

Required Privilege Level view

Related Documentation

- Configuring an FCoE-FC Gateway Fibre Channel Fabric on page 266
- Monitoring Fibre Channel Interface Load Balancing on page 383
- show fibre-channel proxy fabric-state on page 479
- show fibre-channel proxy login-table on page 483
- show fibre-channel proxy statistics on page 489

List of Sample Output

show fibre-channel proxy np-port on page 487
show fibre-channel proxy np-port detail on page 487

Output Fields Table 54 on page 486 lists the output fields for the **show fibre-channel proxy np-port** command. Output fields are listed in the approximate order in which they appear.

Table 54: show fibre-channel proxy np-port Output Fields

Field Name	Field Description	Level of Output
Fabric	Name of the fabric.	All
Fabric-id	Fabric ID number.	All
NP-Port	NP_Port interface connected to the FCoE forwarder (FCF) or the Fibre Channel switch.	All
State	FCID state of the NP_Port interface.	All
Sessions	Number of active sessions on the NP_Port interface. A session is a FLOGI or FDISC login to the FC SAN fabric. Session does not refer to end-to-end storage sessions.	All

Table 54: show fibre-channel proxy np-port Output Fields *(continued)*

Field Name	Field Description	Level of Output
Configured max login sessions	Configured maximum number of FIP login sessions permitted on the interface.	**detail**
Enodes	Number of ENodes with sessions on the NP_Port.	**detail**
LB state	Load-balancing state: • **On**—Load balancing is on • **Off**—Load balancing is off.	All
LB weight	Load balance weight, which reflects the port speed: • **2**—Port speed is 2 Gbps. • **4**—Port speed is 4 Gbps. • **8**—Port speed is 8 Gbps.	All
Ref-count	Reference count internal to Junos OS.	**detail**
Flags	Flags internal to Junos OS. NOTE: When an NP_Port interface reaches its configured maximum number of FIP sessions, the **Flags** field displays the flag **MAX-LOGINS-REACHED**.	**detail**

Sample Output

show fibre-channel proxy np-port

```
user@switch> show fibre-channel proxy np-port
Fabric: proxy1, Fabric-id: 10
NP-Port     State             Sessions      LB state   LB weight
fc-0/0/0.0  online            3             ON         4
fc-0/0/1.0  online            3             ON         4
fc-0/0/2.0  online            3             ON         4
root@junos1> show fibre-channel proxy np-port detail
```

show fibre-channel proxy np-port detail

```
user@switch> show fibre-channel proxy login-table detail
Fabric: proxy1, Fabric-id: 10

    NP-Port:                        fc-0/0/0.0
    State:                          online
    Sessions:                       3
    Configured max login sessions:  130
    Enodes                          1
    LB state:                       ON
    LB weight:                      4
    Ref-count:                      4
    Flags:                          UP LB
```



```
NP-Port:                        fc-0/0/1.0
State:                          online
Sessions:                       3
Configured max login sessions:  130
Enodes                          2
LB state:                       ON
LB weight:                      4
Ref-count:                      4
Flags:                          UP LB

NP-Port:                        fc-0/0/2.0
State:                          online
Sessions:                       130
Configured max login sessions:  130
Enodes                          17
LB state:                       OFF
LB weight:                      4
Ref-count:                      131
Flags:                          UP MAX-LOGINS-REACHED
```

show fibre-channel proxy statistics

Syntax **show fibre-channel proxy statistics**
<fabric *fabric-name*>

Release Information Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Display Fibre Channel proxy fabric statistics.

Options **fabric *fabric-name***—(Optional) Display output only for the specified fabric.

Required Privilege Level view

Related Documentation

- Configuring an FCoE-FC Gateway Fibre Channel Fabric on page 266
- show fibre-channel proxy fabric-state on page 479
- show fibre-channel proxy login-table on page 483
- show fibre-channel proxy np-port on page 486
- clear fibre-channel proxy statistics on page 396

List of Sample Output show fibre-channel proxy statistics on page 490

Output Fields Table 55 on page 489 lists the output fields for the **show fibre-channel proxy statistics** command. Output fields are listed in the approximate order in which they appear.

Table 55: show fibre-channel proxy statistics Output Fields

Field Name	Field Description
Fabric	Name of the fabric.
Fabric-id	Fabric ID number.

Table 55: show fibre-channel proxy statistics Output Fields *(continued)*

Field Name		Field Description
NP-Port Transmit Command Statistics		Transmitted command statistics for the NP_Port.
	• **Command**	Type of command issued on the NP_Port: • **FLOGI**—Fabric login commands issued. • **FDISC**—Fabric discovery commands issued. • **LOGO**—Logout commands issued. • **Others**—Other commands issued.
	• **Tx**	Number of times the command type was transmitted.
	• **Rx-ACC**	Number of times the NP_Port transmitted a receive accept message for the command type.
	• **Rx-RJT**	Number of times the NP_Port transmitted a receive reject message for the command type.
	• **Abort**	Number of times the NP_Port transmitted an abort message for the command type.
NP-Port Receive Command Statistics		Received command statistics for the NP_Port.
	• **Command**	The type of command received on the NP_Port: • **LOGO**—Logout commands issued. • **Others**—Other commands issued.
	• **Rx**	Number of times the command type was received.
	• **Tx-ACC**	Number of times the NP_Port received a transmit accept message for the command type.
	• **Tx-RJT**	Number of times the NP_Port received a transmit reject message for the command type.
	• **Abort**	Number of times the NP_Port received an abort message for the command type.

Sample Output

show fibre-channel proxy statistics

```
user@switch> show fibre-channel proxy statistics
Fabric: proxy1, Fabric-id: 10

NP-Port Transmit Command Statistics:
Command         Tx         Rx-ACC     Rx-RJT     Abort
FLOGI           3          3          0          0
FDISC           3          3          0          0
LOGO            0          0          0          0
```



```
Others          0         0         0         0

NP-Port Receive Command Statistics:
Command         Rx        Tx-ACC    Tx-RJT    Abort
LOGO            0         0         0         0
Others          0         0         0         0
```

show fip snooping

Syntax

show fip snooping
<brief | detail>

Release Information

Command introduced in Junos OS Release 10.4 for EX Series switches.
Command introduced in Junos OS Release 11.1 for the QFX Series.

Description

Display FIP snooping information.

Options

none—Display FIP snooping information.

brief | detail—(Optional) Display the specified level of output.

Required Privilege Level

view

Related Documentation

- Configuring VN2VF_Port FIP Snooping and FCoE Trusted Interfaces on an FCoE Transit Switch on page 294
- Configuring an FCoE LAG on page 302
- *Example: Configuring an FCoE Transit Switch*
- Example: Configuring an FCoE LAG on a Redundant Server Node Group on page 253
- **show fip snooping enode on page 497**
- **show fip snooping fcf on page 501**
- **show fip snooping interface on page 504**
- **show fip snooping statistics on page 507**
- **show fip snooping vlan on page 510**

List of Sample Output

show fip snooping on page 494
show fip snooping brief (QFX Series) on page 494
show fip snooping detail (QFX Series Switches) on page 495
show fip snooping detail (QFabric System FCoE with LAG Configured) on page 495
show fip snooping detail (EX Series Switches) on page 496

Output Fields

Table 56 on page 492 lists the output fields for the **show fip snooping** command. Output fields are listed in the approximate order in which they appear.

Table 56: show fip snooping Output Fields

Field Name	Field Description	Level of Output
VLAN	Name of the VLAN.	All

Table 56: show fip snooping Output Fields *(continued)*

Field Name	Field Description	Level of Output
Mode	(QFX Series only) Snooping mode enabled on the FCoE VLAN: • VN2VF Snooping—The FCoE VLAN is configured for FIP snooping between an ENode VN_Port and a switch VF_Port. • VN2VN Snooping—The FCoE VLAN is configured for VN_Port to VN_Port FIP snooping between ENode VN_Ports.	All
FC-MAP	FCoE mapped address prefix of the FCoE forwarder for the VLAN.	All
FCF or **FCF-MAC**	MAC address of the FCF.	All
Session Count or **Active Sessions**	Current number of virtual link sessions with VN_Ports.	All
VN_Port Count	(QFX Series only) Number of VN_Ports active on an ENode.	**brief**
Configured FKA-ADV	FIP keepalive interval in seconds configured on the FCF multiplied by three. For example, if the FKA_ADV period configured on the FCF is 86 seconds, the value of this field is 258. For the QFX Series only, the output of this field is always 0 (zero) if the VLAN is an FCoE-FC gateway VLAN. If the VLAN is a FIP snooping VLAN (a transit switch VLAN), then the output is accurate. This is because for an FCoE-FC gateway VLAN, FIP snooping is performed internally and the keepalive advertisements are not tracked by the switch's Ethernet module.	**detail**
Running FKA-ADV	Runtime interval in seconds of the last FIP keepalive advertisement the FCF received. This value changes every time the FCF receives an FKA_ADV. For the QFX Series only, the output of this field is always 0 (zero) if the VLAN is an FCoE-FC gateway VLAN. If the VLAN is a FIP snooping VLAN (a transit switch VLAN), then the output is accurate. This is because for an FCoE-FC gateway VLAN, FIP snooping is performed internally and the keepalive advertisements are not tracked by the switch's Ethernet module.	**detail**
Beacon Period	(QFX Series only) Beacon period interval in milliseconds.	**detail**

Table 56: show fip snooping Output Fields *(continued)*

Field Name	Field Description	Level of Output
VN2VN Mode	(QFX Series only) Mode of VN2VN_Port snooping: • Multi-Point—Multiple ENodes are connected to the network and form multiple virtual links. Each virtual link is created between one pair of VN_Ports. This is analogous to the loop mode in traditional FC networks. • Point-to-Point—Two ENodes are connected to the network and form a single VN_Port to VN_Port virtual link. This is analogous to the point-to-point FC link between an FC initiator and an FC target.	**detail**
ENode-MAC	MAC address of the connected FCoE node (ENode).	All
Interface	Interface connected to the ENode. (QFabric System only) When an FCoE LAG has been configured, LAG interface connected to the ENode and LAG member interface connected to ENode.	**detail**
VN-Port MAC	MAC address of a VN_Port on the ENode.	All
FKA-ADV	Runtime interval in seconds of the last FIP keepalive advertisement the ENode sent to the FCF on behalf of the VN_Port (VN_Port FKA_ADV). This value changes every time the ENode sends a VN_Port FKA_ADV to the FCF.	**detail**
Active VN_Ports	(QFX Series only) Number of VN_Ports active on an ENode.	**detail**
Vlink far-end VN-Port-MAC	(QFX Series only) Media access control (MAC) address of the VN_Port at the other end of the virtual link.	**detail**

Sample Output

show fip snooping

```
user@switch> show fip snooping
VLAN : fcoevlan1     FC-MAP : 0e:fc:00
  FCF : 00:10:94:00:00:01   Session Count : 2
    Enode-MAC : 00:10:94:00:00:02
      VN-Port-MAC : 0E:FC:00:01:00:05
      VN-Port-MAC : 0E:FC:00:01:00:01
```

show fip snooping brief (QFX Series)

```
user@switch> show fip snooping brief
VLAN: vlan100,    Mode: VN2VF Snooping
  FC-MAP: 0e:fc:00
  FCF: 30:10:94:01:00:00    Session Count: 2
    Enode-MAC: 10:10:94:01:00:01
```



```
      VN-Port-MAC: 0e:fc:00:01:0d:01
      VN-Port-MAC: 0e:fc:00:01:0e:01
VLAN: vlan101,    Mode: VN2VN Snooping
  FC-MAP: 0e:fc:00
    Enode-MAC: 10:10:94:01:00:02 VN_Port count: 1
      VN-Port-MAC: 0e:fc:00:01:0a:01  Session Count: 2
    Enode-MAC: 10:10:94:01:00:03 VN_Port count: 0
```

show fip snooping detail (QFX Series Switches)

```
user@switch> show fip snooping detail
root@sw-pa02v> show fip snooping detail
VLAN: vlan100,  Mode: VN2VF Snooping
  FC-MAP: 0e:fc:00
  FCF Information
  FCF-MAC           : 30:10:94:01:00:00
  Active Sessions    : 2
  Configured FKA-ADV : 258
  Running FKA-ADV    : 188
    Enode Information
    Enode-MAC: 10:10:94:01:00:01,      Interface: xe-0/0/10
    Configured FKA-ADV : 258
    Running FKA-ADV    : 230
      Session Information
      VN-Port MAC: 0e:fc:00:01:0d:01,   FKA-ADV : 230
      VN-Port MAC: 0e:fc:00:01:0e:01,   FKA-ADV : 245

VLAN: vlan101,  Mode: VN2VN Snooping
  FC-MAP: 0e:fd:00
  Beacon_Period:  90000
  VN2VN Mode: Multi-Point
    Enode Information
    Enode-MAC: 10:10:94:01:00:02,      Interface: xe-0/0/10
      Active VN_Ports : 1
      VN_Port Information
      VN-Port MAC: 0e:fd:00:01:0a:01
         Active Sessions    : 2
         Session Information
     Vlink far-end VN-Port-MAC: 0e:fd:00:01:0b:01
     Vlink far-end VN-Port-MAC: 0e:fd:00:01:0c:01
    Enode-MAC: 10:10:94:01:00:02,      Interface: xe-0/0/11
      Active VN_Ports : 0
```

show fip snooping detail (QFabric System FCoE with LAG Configured)

```
admin@qfabric> show fip snooping detail
VLAN: vlan_100, Mode: VN2VF Snooping
FC-MAP: 0e:fc:00
  FCF Information
  FCF-MAC           : 84:18:88:d1:f5:cc
  Active Sessions    : 2
  Configured FKA-ADV : 8000
  Running FKA-ADV    : 23962
    Enode Information
    Enode-MAC: 00:c0:dd:14:ae:6d,      Interface: P4546-C:ae0  P4546-C:xe-0/0/39

    Configured FKA-ADV : 8000
    Running FKA-ADV    : 16622
      Session Information
      VN-Port MAC: 0e:fc:00:6c:06:a5,   FKA-ADV : 246303
    Enode Information
```

```
Enode-MAC: 00:c0:dd:14:ae:6f,      Interface: P4546-C:ae0  P4546-C:xe-0/0/38

 Configured FKA-ADV : 8000
 Running FKA-ADV    : 16512
   Session Information
   VN-Port MAC: 0e:fc:00:6c:06:a4,   FKA-ADV : 238150
```

show fip snooping detail (EX Series Switches)

```
user@switch> show fip snooping detail
VLAN : fcoevlan1    FC-MAP : 0e:fc:00
  FCF Information
  FCF-MAC            : 00:10:94:00:00:01
  Active Sessions    : 2
  Configured FKA-ADV : 258
  Running FKA-ADV    : 244
    Enode Information
    Enode-MAC : 00:10:94:00:00:02       Interface : xe-0/0/1
    Configured FKA-ADV : 258
    Running FKA-ADV    : 248
      Session Information
      VN-Port MAC : 0E:FC:00:01:00:05   FKA-ADV : 264
      VN-Port MAC : 0E:FC:00:01:00:01   FKA-ADV : 260
```

show fip snooping enode

Syntax

show fip snooping enode ***enode-mac***
<brief | detail>
<vlan ***vlan-name*****>**

Release Information

Command introduced in Junos OS Release 10.4 for EX Series switches.
Command introduced in Junos OS Release 11.1 for the QFX Series.

Description

Display FIP snooping FCoE node (ENode) information.

Options

brief | detail—(Optional) Display the specified level of output.

enode-mac—Display information for the ENode specified by the MAC address.

vlan ***vlan-name***—(Optional) Display FIP snooping information for the ENode on only the specified VLAN.

Required Privilege Level

view

Related Documentation

- Configuring VN2VF_Port FIP Snooping and FCoE Trusted Interfaces on an FCoE Transit Switch on page 294
- *Example: Configuring an FCoE Transit Switch*
- show fip snooping on page 492
- show fip snooping fcf on page 501
- show fip snooping interface on page 504
- show fip snooping statistics on page 507
- show fip snooping vlan on page 510

List of Sample Output

show fip snooping enode on page 499
show fip snooping enode brief (QFX Series) on page 499
show fip snooping enode detail (QFX Series) on page 499
show fip snooping enode detail on page 499

Output Fields

Table 57 on page 497 lists the output fields for the **show fip snooping enode** command. Output fields are listed in the approximate order in which they appear.

Table 57: show fip snooping enode Output Fields

Field Name	Field Description	Level of Output
ENode and **ENode MAC**	MAC address of the ENode.	All
VLAN	Name of the VLAN.	All
Interface	Interface connected to the ENode.	All

Table 57: show fip snooping enode Output Fields *(continued)*

Field Name	Field Description	Level of Output
Mode	(QFX Series only) Snooping mode enabled on the FCoE VLAN: • VN2VF Snooping—The FCoE VLAN is configured for FIP snooping between an ENode VN_Port and a switch VF_Port. • VN2VN Snooping—The FCoE VLAN is configured for VN_Port to VN_Port FIP snooping between ENode VN_Ports.	All
VN_Port Count	(QFX Series only) Number of VN_Ports active on an ENode.	**brief**
Session Count	Current number of virtual link sessions with VN_Ports.	All
Configured FKA-ADV	FIP keepalive interval in seconds configured on the FCoE forwarder (FCF) multiplied by three. For example, if the FKA_ADV period configured on the FCF is 86 seconds, the value of this field is 258. This value remains constant. For the QFX Series only, the output of this field is always 0 (zero) if the VLAN is an FCoE-FC gateway VLAN. If the VLAN is a FIP snooping VLAN (a transit switch VLAN), then the output is accurate. This is because for an FCoE-FC gateway VLAN, FIP snooping is performed internally and the keepalive advertisements are not tracked by the switch's Ethernet module.	**detail**
Running FKA-ADV	Runtime interval in seconds of the last FIP keepalive advertisement the ENode sent to the FCF. This value changes every time the ENode sends an FKA_ADV to the FCF. For the QFX Series only, the output of this field is always 0 (zero) if the VLAN is an FCoE-FC gateway VLAN. If the VLAN is a FIP snooping VLAN (a transit switch VLAN), then the output is accurate. This is because for an FCoE-FC gateway VLAN, FIP snooping is performed internally and the keepalive advertisements are not tracked by the switch's Ethernet module.	**detail**
VN-Port or **VN-Port-MAC**	MAC address of a VN_Port on the ENode.	All
FKA-ADV	Runtime interval in seconds of the last FIP keepalive advertisement the ENode sent to the FCF on behalf of the VN_Port (VN_Port FKA_ADV). This value changes every time the ENode sends a VN_Port FKA_ADV to the FCF.	**detail**
FCF or **FCF-MAC**	MAC address of the FCF to which the VN_Port is connected.	All
Beacon Period	(QFX Series only) Beacon period interval in milliseconds.	**detail**

Table 57: show fip snooping enode Output Fields *(continued)*

Field Name	Field Description	Level of Output
VN2VN Mode	(QFX Series only) Mode of VN2VN_Port snooping: • Multi-Point—Multiple ENodes are connected to the network and form multiple virtual links. Each virtual link is created between one pair of VN_Ports. This is analogous to the loop mode in traditional FC networks. • Point-to-Point—Two ENodes are connected to the network and form a single VN_Port to VN_Port virtual link. This is analogous to the point-to-point FC link between an FC initiator and an FC target.	**detail**
Vlink far-end VN-Port-MAC	(QFX Series only) Media access control (MAC) address of the VN_Port at the other end of the virtual link.	**detail**

Sample Output

show fip snooping enode

```
user@switch> show fip snooping enode 00:10:94:00:00:02
Enode : 00:10:94:00:00:02   VLAN : vlan1   Interface : xe-0/0/1
        VN-Port-MAC           FCF-MAC
        0E:FC:00:00:00:05     00:10:94:00:00:01
        0E:FC:00:00:00:01     00:10:94:00:00:01
```

show fip snooping enode brief (QFX Series)

```
user@switch> show fip snooping enode 10:10:94:01:00:02 brief
Enode: 10:10:94:01:00:02 ,  VLAN: vlan101,  Interface: xe-0/0/10
  Mode: VN2VF Snooping      VN_Port Count: 1
        VN_Port Information
        VN_Port Mac: 0e:fc:00:01:0a:01     Session Count: 2
```

show fip snooping enode detail (QFX Series)

```
user@switch> show fip snooping enode 10:10:94:01:00:02 detail
Enode MAC: 10:10:94:01:00:02,   VLAN: vlan101,  Interface: xe-0/0/10
  Mode: VN2VF Snooping       VN_Port Count: 1
  Beacon_Period: 90000       VN2VN Mode: Multi-Point
        VN_Port Information
        VN_Port Mac: 0e:fc:00:01:0a:01     Session Count: 2
     Vlink far-end VN-Port-MAC: 0e:fc:00:01:0b:01
     Vlink far-end VN-Port-MAC: 0e:fc:00:01:0c:01
```

show fip snooping enode detail

```
user@switch> show fip snooping enode 00:10:94:00:00:02 detail
Enode MAC : 00:10:94:00:00:02   VLAN : vlan1    Interface : xe-0/0/1
Configured FKA-ADV : 258        Running FKA-ADV : 213
  Session Information
  VN-Port : 0E:FC:00:00:00:05   FKA-ADV : 229   FCF : 00:10:94:00:00:01
  VN-Port : 0E:FC:00:00:00:01   FKA-ADV : 225   FCF : 00:10:94:00:00:01
```


show fip snooping fcf

Syntax

show fip snooping fcf *fcf-mac*
<brief | detail>
<vlan *vlan-name*>

Release Information

Command introduced in Junos OS Release 10.4 for EX Series switches.
Command introduced in Junos OS Release 11.1 for the QFX Series.

Description

Display FIP snooping FCoE forwarder (FCF) information.

Options

brief | detail—(Optional) Display the specified level of output.

fcf-mac—Display information for the FCF specified by the MAC address.

vlan-name—(Optional) Display FIP snooping information for the FCF on only the specified VLAN.

Required Privilege Level

view

Related Documentation

- Configuring VN2VF_Port FIP Snooping and FCoE Trusted Interfaces on an FCoE Transit Switch on page 294
- *Example: Configuring an FCoE Transit Switch*
- show fip snooping on page 492
- show fip snooping enode on page 497
- show fip snooping interface on page 504
- show fip snooping statistics on page 507
- show fip snooping vlan on page 510

List of Sample Output

show fip snooping fcf on page 502
show fip snooping fcf detail on page 502

Output Fields

Table 58 on page 501 lists the output fields for the **show fip snooping fcf** command. Output fields are listed in the approximate order in which they appear.

Table 58: show fip snooping fcf Output Fields

Field Name	Field Description	Level of Output
FCF or **FCF-MAC**	MAC address of the FCoE forwarder.	All
VLAN	Name of the VLAN.	All
Session Count	Current number of virtual link sessions with VN_Ports.	None

Table 58: show fip snooping fcf Output Fields *(continued)*

Field Name	Field Description	Level of Output
Configured FKA-ADV	FIP keepalive interval in seconds configured on the FCF multiplied by three. For example, if the FKA_ADV period configured on the FCF is 86 seconds, the value of this field is 258.	**detail**
Running FKA-ADV	Runtime interval in seconds of the last FIP keepalive advertisement the FCF received. This value changes every time the FCF receives an FKA_ADV.	**detail**
ENode-MAC	MAC address of the connected ENode.	All
• **Interface**	Interface connected to the ENode.	**detail**
• **Configured FKA-ADV**	FIP keepalive interval in seconds configured on the FCF multiplied by three. For example, if the FKA_ADV period configured on the FCF is 86 seconds, the value of this field is 258. This value remains constant.	**detail**
• **Running FKA-ADV**	Runtime interval in seconds of the last FIP keepalive advertisement the ENode sent to the FCF. This value changes every time the ENode sends an FKA_ADV to the FCF.	**detail**
• **VN-Port MAC**	MAC address of a VN_Port on the ENode.	All
• **FKA-ADV**	Runtime interval in seconds of the last FIP keepalive advertisement the ENode sent to the FCF on behalf of the VN_Port (VN_Port FKA_ADV). This value changes every time the ENode sends a VN_Port FKA_ADV to the FCF.	**detail**

Sample Output

show fip snooping fcf

```
user@switch> show fip snooping fcf 00:10:94:00:00:01
FCF : 00:10:94:00:00:01   VLAN : vlan1   Session Count : 2
  Enode-MAC : 00:10:94:00:00:02
    VN-Port-MAC : 0E:FC:00:00:00:05
    VN-Port-MAC : 0E:FC:00:00:00:01
```

show fip snooping fcf detail

```
user@switch> show fip snooping fcf 00:10:94:00:00:01 detail
FCF-MAC : 00:10:94:00:00:01     VLAN : vlan1
Configured FKA-ADV : 258        Running FKA-ADV : 222
  Enode Information
  Enode-MAC : 00:10:94:00:00:02 Interface: xe-0/0/1
  Configured FKA-ADV : 258
  Running FKA-ADV    : 226
    Session Information
    VN-Port MAC : 0E:FC:00:00:00:05     FKA-ADV : 242
    VN-Port MAC : 0E:FC:00:00:00:01     FKA-ADV : 238
```

show fip snooping interface

Syntax

show fip snooping interface ***interface-name***
<brief | detail>

Release Information

Command introduced in Junos OS Release 12.1 for the QFX Series.

Description

Display FIP snooping information for the specified interface.

Options

brief | detail—(Optional) Display the specified level of output.

interface-name—Display information for the specified interface.

Required Privilege Level

view

Related Documentation

- Configuring VN2VF_Port FIP Snooping and FCoE Trusted Interfaces on an FCoE Transit Switch on page 294
- show fip snooping on page 492
- show fip snooping enode on page 497
- show fip snooping fcf on page 501
- show fip snooping statistics on page 507
- show fip snooping vlan on page 510

List of Sample Output

show fip snooping interface on page 506
show fip snooping interface detail on page 506

Output Fields

Table 59 on page 504 lists the output fields for the **show fip snooping interface** ***interface-name*** command. Output fields are listed in the approximate order in which they appear.

Table 59: show fip snooping interface Output Fields

Field Name	Field Description	Level of Output
VLAN	Name of the VLAN.	All
FC-MAP	FCoE mapped address prefix of the FCoE forwarder for the VLAN.	All
FCF or **FCF-MAC**	MAC address of the FCF.	All
Session Count or **Active Sessions**	Current number of virtual link sessions with VN_Ports.	All

Table 59: show fip snooping interface Output Fields *(continued)*

Field Name	Field Description	Level of Output
Configured FKA-ADV	FIP keepalive interval in seconds configured on the FCF multiplied by three. For example, if the FKA_ADV period configured on the FCF is 86 seconds, the value of this field is 258. For the QFX Series only, the output of this field is always 0 (zero) if the VLAN is an FCoE-FC gateway VLAN. If the VLAN is a FIP snooping VLAN (a transit switch VLAN), then the output is accurate. This is because for an FCoE-FC gateway VLAN, FIP snooping is performed internally and the keepalive advertisements are not tracked by the switch's Ethernet module.	**detail**
Running FKA-ADV	Runtime interval in seconds of the last FIP keepalive advertisement the FCF received. This value changes every time the FCF receives an FKA_ADV. For the QFX Series only, the output of this field is always 0 (zero) if the VLAN is an FCoE-FC gateway VLAN. If the VLAN is a FIP snooping VLAN (a transit switch VLAN), then the output is accurate. This is because for an FCoE-FC gateway VLAN, FIP snooping is performed internally and the keepalive advertisements are not tracked by the switch's Ethernet module.	**detail**
ENode-MAC	MAC address of the connected FCoE node (ENode).	All
Interface	Interface connected to the ENode.	**detail**
Configured FKA-ADV	FIP keepalive interval in seconds configured on the FCF multiplied by three. For example, if the FKA_ADV period configured on the FCF is 86 seconds, the value of this field is 258. This value remains constant. For the QFX Series only, the output of this field is always 0 (zero) if the VLAN is an FCoE-FC gateway VLAN. If the VLAN is a FIP snooping VLAN (a transit switch VLAN), then the output is accurate. This is because for an FCoE-FC gateway VLAN, FIP snooping is performed internally and the keepalive advertisements are not tracked by the switch's Ethernet module.	**detail**
Running FKA-ADV	Runtime interval in seconds of the last FIP keepalive advertisement the ENode sent to the FCF. This value changes every time the ENode sends an FKA_ADV to the FCF. For the QFX Series only, the output of this field is always 0 (zero) if the VLAN is an FCoE-FC gateway VLAN. If the VLAN is a FIP snooping VLAN (a transit switch VLAN), then the output is accurate. This is because for an FCoE-FC gateway VLAN, FIP snooping is performed internally and the keepalive advertisements are not tracked by the switch's Ethernet module.	**detail**
VN-Port MAC	MAC address of a VN_Port on the ENode.	All

Table 59: show fip snooping interface Output Fields *(continued)*

Field Name	Field Description	Level of Output
FKA-ADV	Runtime interval in seconds of the last FIP keepalive advertisement the ENode sent to the FCF on behalf of the VN_Port (VN_Port FKA_ADV). This value changes every time the ENode sends a VN_Port FKA_ADV to the FCF.	**detail**

Sample Output

show fip snooping interface

```
user@switch> show fip snooping interface xe-0/0/9.0
VLAN: vlan_100,    FC-MAP: 0e:fc:00
  FCF: 30:10:94:01:00:00    Session Count: 1
    Enode-MAC: 10:10:94:01:00:01
      VN-Port-MAC: 0e:fc:00:01:0a:01
```

show fip snooping interface detail

```
user@switch> show fip snooping interface xe-0/0/9.0 detail
VLAN: vlan_100, FC-MAP: 0e:fc:00
  FCF Information
  FCF-MAC            : 30:10:94:01:00:00
  Active Sessions    : 1
  Configured FKA-ADV : 368640000
  Running FKA-ADV    : 0
    Enode Information
    Enode-MAC: 10:10:94:01:00:01,       Interface: xe-0/0/9
    Configured FKA-ADV : 368640000
    Running FKA-ADV    : 0
      Session Information
      VN-Port MAC: 0e:fc:00:01:0a:01,   FKA-ADV : 0
```

show fip snooping statistics

Syntax

show fip snooping statistics
<**vlan** *vlan-name*>

Release Information

Command introduced in Junos OS Release 10.4 for EX Series switches.
Command introduced in Junos OS Release 11.1 for the QFX Series.

Description

Display FIP snooping statistics.

Options

vlan ***vlan-name***—(Optional) Display FIP snooping statistics for the specified VLAN.

Required Privilege Level

view

Related Documentation

- *Example: Configuring an FCoE Transit Switch*
- Configuring VN2VF_Port FIP Snooping and FCoE Trusted Interfaces on an FCoE Transit Switch on page 294
- show fip snooping on page 492
- show fip snooping enode on page 497
- show fip snooping fcf on page 501
- show fip snooping interface on page 504
- show fip snooping vlan on page 510

List of Sample Output

show fip snooping statistics (FIP Snooping) on page 509
show fip snooping statistics (VN2VN_Port Snooping) on page 509

Output Fields

Table 60 on page 507 lists the output fields for the **show fip snooping statistics** command. Output fields are listed in the approximate order in which they appear.

Table 60: show fip snooping statistics Output Fields

Field Name	Field Description
VLAN	Name of the VLAN for which a set of statistics is displayed.
Mode	(QFX Series only) Snooping mode enabled on the FCoE VLAN: • VN2VF Snooping—The FCoE VLAN is configured for FIP snooping between an ENode VN_Port and a switch VF_Port. • VN2VN Snooping—The FCoE VLAN is configured for VN_Port to VN_Port FIP snooping between ENode VN_Ports.
Number of MDS	Number of multicast discovery solicitation messages sent on the VLAN.

Table 60: show fip snooping statistics Output Fields *(continued)*

Field Name	Field Description
Number of UDS	Number of unicast discovery solicitation messages sent on the VLAN.
Number of FLOGI	Number of fabric logins on the VLAN.
Number of FDISC	Number of fabric discovery logins on the VLAN.
Number of LOGO	Number of fabric logouts on the VLAN.
Number of ENode-keep-alive	Number of ENode keepalive messages sent on the VLAN.
Number of VNPort-keep-alive	Number of VN_Port keepalive messages sent on the VLAN.
Number of MDA	Number of multicast discovery advertisement messages sent on the VLAN.
Number of UDA	Number of unicast discovery advertisement messages sent on the VLAN.
Number of FLOGI_ACC	Number of fabric logins accepted on the VLAN.
Number of FLOGI_RJT	Number of fabric logins rejected on the VLAN.
Number of FDISC_ACC	Number of fabric discoveries accepted on the VLAN.
Number of FDISC_RJT	Number of fabric discoveries rejected on the VLAN.
Number of LOGO_ACC	Number of fabric logouts accepted on the VLAN.
Number of LOGO_RJT	Number of fabric logouts rejected on the VLAN.
Number of CVL	Number of clear virtual links (CVL) actions on the VLAN.
Number of VN_Port Probes Req	(QFX Series only) Number of multicast N_Port_ID probes sent to the ALL-VN2VN-ENode-MACs multicast address on the VLAN.
Number of VN_Port Claim Notif	(QFX Series only) Number of multicast N_Port_ID claim notifications sent on the VLAN.
Number of VN_Port Beacons	(QFX Series only) Number of multicast beacons sent on the VLAN.
Number of VN_Port Probes Reply	(QFX Series only) Number of replies to N_Port_ID probes sent on the VLAN. Replies are unicast to the ENode MAC address of the probe requester.

Table 60: show fip snooping statistics Output Fields *(continued)*

Field Name	Field Description
Number of VN_Port Claim Reply	(QFX Series only) Number of replies to N_Port_ID claim notifications sent on the VLAN. Replies are unicast to the ENode MAC address of the claim notifier.

Sample Output

show fip snooping statistics (FIP Snooping)

```
user@switch> show fip snooping statistics
VLAN: fcoevlan1      Mode: VN2VF Snooping

    Number of MDS:              2
    Number of UDS:              2
    Number of FLOGI:            2
    Number of FDISC:            2
    Number of LOGO:             0
    Number of Enode-keep-alive:  200
    Number of VNPort-keep-alive: 200

    Number of MDA:              25
    Number of UDA:              2
    Number of FLOGI_ACC:        2
    Number of FLOGI_RJT:        0
    Number of FDISC_ACC:        2
    Number of FDISC_RJT:        0
    Number of LOGO_ACC:         0
    Number of LOGO_RJT:         0
    Number of CVL:              0
```

show fip snooping statistics (VN2VN_Port Snooping)

```
user@switch> show fip snooping statistics
VLAN: vlan101   Mode: VN2VN Snooping

Number of VN_Port Probes Req:       3
Number of VN_Port Claim Notif:      3
Number of VN_Port Beacons:          0

Number of VN_Port Probes Reply:     3
Number of VN_Port Claim Reply:      3
Number of FLOGI:                    0
Number of FLOGI_ACC:                0
Number of FLOGI_RJT:                0
Number of FDISC:                    0
Number of FDISC_ACC:                0
Number of FDISC_RJT:                0
Number of LOGO:                     0
Number of LOGO_ACC:                 0
Number of LOGO_RJT:                 0
```


show fip snooping vlan

Syntax

show fip snooping vlan ***vlan-name***
<brief | detail>

Release Information

Command introduced in Junos OS Release 10.4 for EX Series switches.
Command introduced in Junos OS Release 11.1 for the QFX Series.

Description

Display FIP snooping VLAN information.

Options

brief | detail—(Optional) Display the specified level of output.

vlan-name—Display information for the specified VLAN.

Required Privilege Level

view

Related Documentation

- Configuring VN2VF_Port FIP Snooping and FCoE Trusted Interfaces on an FCoE Transit Switch on page 294
- *Example: Configuring an FCoE Transit Switch*
- show fip snooping on page 492
- show fip snooping enode on page 497
- show fip snooping fcf on page 501
- show fip snooping interface on page 504
- show fip snooping statistics on page 507

List of Sample Output

show fip snooping vlan on page 512
show fip snooping vlan (QFX Series, VN2VF_Port FIP Snooping) on page 512
show fip snooping vlan (QFX Series, VN2VN_Port FIP Snooping) on page 512
show fip snooping vlan detail (QFX Series, VN2VN_Port FIP Snooping) on page 513
show fip snooping vlan detail on page 513

Output Fields

Table 61 on page 510 lists the output fields for the **show fip snooping vlan** command. Output fields are listed in the approximate order in which they appear.

Table 61: show fip snooping vlan Output Fields

Field Name	Field Description	Level of Output
VLAN	Name of the VLAN.	All

Table 61: show fip snooping vlan Output Fields *(continued)*

Field Name	Field Description	Level of Output
Mode	(QFX Series only) Snooping mode enabled on the FCoE VLAN: • VN2VF Snooping—The FCoE VLAN is configured for FIP snooping between an ENode VN_Port and a switch VF_Port. • VN2VN Snooping—The FCoE VLAN is configured for VN_Port to VN_Port FIP snooping between ENode VN_Ports.	All
VN_Port count	(QFX Series only) Number of VN_Ports active on an ENode when the mode is VN2VN_Port FIP snooping.	
FC-MAP	FCoE mapped address prefix of the FCoE forwarder for the VLAN.	All
Beacon_Period	(QFX Series only) Beacon period interval in milliseconds.	**detail**
VN2VN Mode	(QFX Series only) Mode of VN2VN_Port snooping: • Multi-Point—Multiple ENodes are connected to the network and form multiple virtual links. Each virtual link is created between one pair of VN_Ports. This is analogous to the loop mode in traditional FC networks. • Point-to-Point—Two ENodes are connected to the network and form a single VN_Port to VN_Port virtual link. This is analogous to the point-to-point FC link between an FC initiator and an FC target.	**detail**
FCF or **FCF-MAC**	MAC address of the FCF.	All
Session Count or **Active Sessions**	Current number of virtual link sessions with VN_Ports.	All
Configured FKA-ADV	FIP keepalive interval in seconds configured on the FCF multiplied by three. For example, if the FKA_ADV period configured on the FCF is 86 seconds, the value of this field is 258.	**detail**
Running FKA-ADV	Runtime interval in seconds of the last FIP keepalive advertisement the FCF received. This value changes every time the FCF receives an FKA_ADV.	**detail**

Table 61: show fip snooping vlan Output Fields *(continued)*

Field Name		Field Description	Level of Output
ENode-MAC		MAC address of the connected ENode.	All
	• **Interface**	Interface connected to the ENode.	**detail**
	• **Configured FKA-ADV**	FIP keepalive interval in seconds configured on the FCF multiplied by three. For example, if the FKA_ADV period configured on the FCF is 86 seconds, the value of this field is 258. This value remains constant.	**detail**
	• **Running FKA-ADV**	Runtime interval in seconds of the last FIP keepalive advertisement the ENode sent to the FCF. This value changes every time the ENode sends an FKA_ADV to the FCF.	**detail**
	• **VN-Port MAC**	MAC address of a VN_Port on the ENode.	All
	• **FKA-ADV**	Runtime interval in seconds of the last FIP keepalive advertisement the ENode sent to the FCF on behalf of the VN_Port (VN_Port FKA_ADV). This value changes every time the ENode sends a VN_Port FKA_ADV to the FCF.	**detail**
	• **Active VN_Ports**	(QFX Series only) Number of VN_Ports active on an ENode.	**detail**
	• **Vlink far-end VN-Port-MAC**	(QFX Series only) Media access control (MAC) address of the VN_Port at the other end of the virtual link.	**detail**

Sample Output

show fip snooping vlan

```
user@switch> show fip snooping vlan fcoevlan1
VLAN : fcoevlan1     FC-MAP : 0e:fc:00
  FCF : 00:10:94:00:00:01   Session Count : 2
    Enode-MAC : 00:10:94:00:00:02
      VN-Port-MAC : 0E:FC:00:00:00:05
      VN-Port-MAC : 0E:FC:00:00:00:01
```

show fip snooping vlan (QFX Series, VN2VF_Port FIP Snooping)

```
user@switch> show fip snooping vlan fcoevlan1
VLAN : fcoevlan1     Mode: VN2VF Snooping
  FC-MAP : 0e:fc:00
  FCF : 00:10:94:00:00:01   Session Count : 2
    Enode-MAC : 00:10:94:00:00:02
      VN-Port-MAC : 0E:FC:00:00:00:05
      VN-Port-MAC : 0E:FC:00:00:00:01
```

show fip snooping vlan (QFX Series, VN2VN_Port FIP Snooping)

```
user@switch> show fip snooping vlan vlan101
```



```
VLAN: vlan101,    Mode: VN2VN Snooping
  FC-MAP: 0e:fd:00
    Enode-MAC: 10:10:94:01:00:02 VN_Port count: 1
      VN-Port-MAC: 0e:fd:00:00:0a:01  Session Count: 2
    Enode-MAC: 10:10:94:01:00:03 VN_Port count: 0
```

show fip snooping vlan detail (QFX Series, VN2VN_Port FIP Snooping)

```
user@switch> show fip snooping vlan vlan101 detail
VLAN: vlan101,  Mode: VN2VN Snooping
  FC-MAP: 0e:fd:00
  Beacon_Period:  90000
  VN2VN Mode: Multi-Point
    Enode Information
    Enode-MAC: 10:10:94:01:00:02,       Interface: xe-0/0/10
      Active VN_Ports : 1
      VN_Port Information
      VN-Port MAC: 0e:fd:00:00:0a:01
         Active Sessions    : 2
         Session Information
     Vlink far-end VN-Port-MAC: 0e:fd:00:00:0b:01
     Vlink far-end VN-Port-MAC: 0e:fd:00:00:0c:01
    Enode-MAC: 10:10:94:01:00:02,       Interface: xe-0/0/11
      Active VN_Ports : 0
```

show fip snooping vlan detail

```
user@switch> show fip snooping vlan fcoevlan1 detail
VLAN : fcoevlan1    FC-MAP : 0e:fc:00
  FCF Information
  FCF-MAC            : 00:10:94:00:00:01
  Active Sessions    : 2
  Configured FKA-ADV : 258
  Running FKA-ADV    : 235
    Enode Information
    Enode-MAC : 00:10:94:00:00:02       Interface : xe-0/0/1
    Configured FKA-ADV : 258
    Running FKA-ADV    : 239
      Session Information
      VN-Port MAC : 0E:FC:00:00:00:05   FKA-ADV : 255
      VN-Port MAC : 0E:FC:00:00:00:01   FKA-ADV : 251
```


show fip vlan-discovery

Syntax **show fip vlan-discovery (enodes | statistics)**

Release Information Command introduced in Junos OS Release 12.1 for the QFX Series.

Description Display FCoE VLAN information from the Fibre Channel switch or FCoE forwarder (FCF).

Options **enodes**—Display VLAN discovery information for each ENode.

statistics—Display VLAN discovery information statistics.

Required Privilege Level view

Related Documentation

- **clear fip vlan-discovery statistics on page 400**
- Understanding FIP Functions on page 38
- Understanding FIP Implementation on page 42

List of Sample Output **show fip vlan-discovery enodes on page 515**
show fip vlan-discovery statistics (QFX3500) on page 515
show fip vlan-discovery statistics (QFabric Systems) on page 515

Output Fields Table 62 on page 514 lists the output fields for the **show fip vlan-discovery** command. Output fields are listed in the approximate order in which they appear.

Table 62: show fip vlan-discovery Output Fields

Field Name	Field Description	Level of Output
Enode-MAC	Media access control (MAC) address of the ENode.	**enodes**
Interface	Name of the interface.	**enodes**
Unsolicited notification count	Number of unsolicited VLAN discovery notifications.	All
Solicited notification count	Number of solicited VLAN discovery notifications.	**statistics**
Node Group Name	Displays the name of the Node group on QFabric systems.	**statistics**
Request count	Number of VLAN discovery requests sent by the ENode. This number should match the **Solicited notification count** number.	**statistics**
VLAN tags	Tags of the FIP-enabled VLANs.	**enodes**

Sample Output

show fip vlan-discovery enodes

```
user@switch> show fip vlan-discovery enodes

Enode-MAC           Interface      Unsolicited     Vlan Tags
                                   Notification
                                   Count
00:10:94:00:00:02   xe-0/0/9.0        0            400
```

show fip vlan-discovery statistics (QFX3500)

```
user@switch> show fip vlan-discovery statistics
Request count: 0
Solicited notification count: 0
Unsolicited notification count: 1
```

show fip vlan-discovery statistics (QFabric Systems)

```
user@switch> show fip vlan-discovery statistics
NW-NG-0:
--------------------------------------------
Request count: 0
Solicited notification count: 0
Unsolicited notification count: 1

BBAK0399:
--------------------------------------------
Request count: 0
Solicited notification count: 0
Unsolicited notification count: 1

FCG001:
--------------------------------------------
Request count: 0
Solicited notification count: 0
Unsolicited notification count: 1
```


show route forwarding-table family fibre-channel

Syntax

show route forwarding-table family fibre-channel
<brief | detail | extensive>
<all>
<destination *destination-prefix*>
<interface-name *interface-name*>
<label *label*>
<matching *ip-prefix*>
<multicast>
<summary>
<table *routing-table-name*>
<vlan *vlan-name*>
<vpn *vpn-instance-name*>

Release Information Command introduced in Junos OS Release 11.1 for the QFX Series.

Description Display Fibre Channel family forwarding table route information.

Options

brief | detail | extensive—(Optional) Display the specified level of output.

all—Display all routing forwarding tables.

destination *destination-prefix*—Destination prefix.

interface-name *interface-name*—Name of the interface.

label *label*—Display route entries for the specified label name.

matching *ip-prefix*—Display route entries for the specified IP prefix or length.

multicast—Display multicast routes.

summary—Display route count instead of details.

table *routing-table-name*—Name of the routing table.

vlan *vlan-name*—Name of the VLAN.

vpn *vpn-instance-name*—Name of the VPN instance.

Required Privilege Level view

Related Documentation

- show fibre-channel next-hops on page 475
- show fibre-channel routes on page 477

List of Sample Output show route forwarding-table family fibre-channel on page 517

Output Fields Table 63 on page 517 lists the output fields for the **show route forwarding-table family fibre-channel** command. Output fields are listed in the approximate order in which they appear.

Table 63: show route forwarding-table family fibre-channel Output Fields

Field Name	Field Description
Routing table	Name of the routing table.
Destination	Route destination.
Type	Type of route internal to Junos OS.
RtRef	Route reference count internal to Junos OS.
Next hop Type	Type of next hop internal to Junos OS.
Index	Next-hop index identifier.
NhRef	Number of routes that refer to the next hop.
Netif	Interface used to reach the next hop.

Sample Output

show route forwarding-table family fibre-channel

```
user@switch> show route forwarding-table family fibre-channel
Routing table: default.fibre-channel
Fibre Channel:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            perm     0                    dscd   126     1
0x30000/24         user     0                    ucst   674     2 fc-0/0/0.0
0x30001/24         user     0                    ucst   675     2 vlan.100
0x30002/24         user     0                    ucst   676     2 vlan.100
0x30003/24         user     0                    ucst   677     2 vlan.100
0x30004/24         user     0                    ucst   678     2 vlan.100
```


PART 4

Troubleshooting

CHAPTER 10

Troubleshooting Procedures

Troubleshooting Dropped FCoE Traffic

Problem **Description:** Fibre Channel over Ethernet (FCoE) traffic for which you want guaranteed delivery is dropped.

Cause There are several possible causes of dropped FCoE traffic (the list numbers of the possible causes correspond to the list numbers of the solutions in the *Solution* section.):

1. Priority-based flow control (PFC) is not enabled on the FCoE priority (IEEE 802.1p code point) in both the input and output stanzas of the congestion notification profile.
2. The FCoE traffic is not classified correctly at the ingress interface. FCoE traffic should either use the default **fcoe** forwarding class and classifier configuration (maps the **fcoe** forwarding class to IEEE 802.1p code point 011) or be mapped to a lossless forwarding class and to the code point enabled for PFC on the input and output interfaces.
3. The congestion notification profile that enables PFC on the FCoE priority is not attached to the interface.
4. The forwarding class set (priority group) used for guaranteed delivery traffic does not include the forwarding class used for FCoE traffic.
5. Insufficient bandwidth has been allocated for the FCoE queue or for the forwarding class set to which the FCoE queue belongs.
6. If you are using Junos OS Release 12.2, the **fcoe** forwarding class has been explicitly configured instead of using the default **fcoe** forwarding class configuration (forwarding-class-to-queue mapping).

NOTE: If you are using Junos OS Release 12.2, use the default forwarding-class-to-queue mapping for the lossless **fcoe** and **no-loss** forwarding classes. If you explicitly configure the lossless forwarding classes, the traffic mapped to those forwarding classes is treated as lossy (best effort) traffic and does *not* receive lossless treatment.

7. If you are using Junos OS Release 12.3 or later and you are not using the default **fcoe** forwarding class configuration, the forwarding class used for FCoE is not configured with the **no-loss** packet drop attribute. In Junos OS 12.3 or later, explicit forwarding classes configurations must include the **no-loss** packet drop attribute to be treated as lossless forwarding classes.

Solution The list numbers of the possible solutions correspond to the list numbers of the causes in the *Cause* section.

1. Check the congestion notification profile (CNP) to see if PFC is enabled on the FCoE priority (the correct IEEE 802.1p code point) on both input and output interfaces. Use the **show class-of-service congestion-notification** operational command to show the code points that are enabled for PFC in each CNP.

   If you are using the default configuration, FCoE traffic is mapped to code point 011 (priority 3). In this case, the input stanza of the CNP should show that PFC is enabled on code point 011, and the output stanza should show that priority 011 is mapped to flow control queue 3.

   If you explicitly configured a forwarding class for FCoE traffic, ensure that:

   - You specified the **no-loss** packet drop attribute in the forwarding class configuration
   - The code point mapped to the FCoE forwarding class in the ingress classifier is the code point enabled for PFC in the CNP input stanza
   - The code point and output queue used for FCoE traffic are mapped to each other in the CNP output stanza (if you are not using the default priority and queue, you must explicitly configure each output queue that you want to respond to PFC messages)

   For example, if you explicitly configure a forwarding class for FCoE traffic that is mapped to output queue 5 and to code point 101 (priority 5), the output of the **show class-of-service congestion-notification** looks like:

```
Name: fcoe_p5_cnp, Index: 12183
Type: Input
Cable Length: 100 m
  Priority     PFC         MRU
  000          Disabled
  001          Disabled
  010          Disabled
  011          Disabled
  100          Disabled
  101          Enabled     2500
  110          Disabled
  111          Disabled
Type: Output
  Priority     Flow-Control-Queues
  101
               5
```

2. Use the **show class-of-service classifier type ieee-802.1p** operational command to check if the classifier maps the forwarding class used for FCoE traffic to the correct IEEE 802.1p code point.

3. Ensure that the congestion notification profile and classifier are attached to the correct ingress interface. Use the operational command **show configuration class-of-service interfaces *interface-name***.

4. Check that the forwarding class set includes the forwarding class used for FCoE traffic. Use the operational command **show configuration class-of-service forwarding-class-sets** to show the configured priority groups and their forwarding classes.

5. Verify the amount of bandwidth allocated to the queue mapped to the FCoE forwarding class and to the forwarding class set to which the FCoE traffic queue belongs. Use the **show configuration class-of-service schedulers *scheduler-name*** operational command (specify the scheduler for FCoE traffic as the ***scheduler-name***) to see the minimum guaranteed bandwidth (**transmit-rate**) and maximum bandwidth (**shaping-rate**) for the queue.

   Use the **show configuration class-of-service traffic-control-profiles *traffic-control-profile*** operational command (specify the traffic control profile used for FCoE traffic as the ***traffic-control-profile***) to see the minimum guaranteed bandwidth (**guaranteed-rate**) and maximum bandwidth (**shaping-rate**) for the forwarding class set.

6. Delete the explicit FCoE forwarding-class-to-queue mapping so that the system uses the default FCoE forwarding-class-to-queue mapping. Include the **delete forwarding-classes class fcoe queue-num 3** statement at the **[edit class-of-service]** hierarchy level to remove the explicit configuration. The system then uses the default configuration for the FCoE forwarding class and preserves the lossless treatment of FCoE traffic.

7. Use the **show class-of-service forwarding-class** operational command to display the configured forwarding classes. The *No-Loss* column shows whether lossless transport is enabled or disabled for each forwarding class. If the forwarding class used for FCoE traffic is not enabled for lossless transport, include the **no-loss** packet drop attribute in the forwarding class configuration (**set class-of-service forwarding-classes class *fcoe-forwarding-class-name* queue-num *queue-number* no-loss**).

See "Example: Configuring CoS PFC for FCoE Traffic" on page 186 for step-by-step instructions on how to configure PFC for FCoE traffic, including classifier, interface, congestion notification profile, PFC, and bandwidth scheduling configuration.

Related Documentation

- *show class-of-service congestion-notification*
- *show class-of-service forwarding-class-set*
- *Configuring CoS PFC (Congestion Notification Profiles)*
- Example: Configuring CoS PFC for FCoE Traffic on page 186
- *Overview of CoS Changes Introduced in Junos OS Release 12.2*
- Understanding CoS Flow Control (Ethernet PAUSE and PFC) on page 121

Troubleshooting Fibre Channel Interface Deletion

Problem **Description:** You deleted a Fibre Channel (FC) interface at the **[edit interfaces]** hierarchy level, but the commit check fails so the interface is not deleted.

Cause You must first delete the FC interface from the FC fabric on the QFX Series before you can delete the FC interface at the **[edit interfaces]** hierarchy level. You must perform both operations to delete a FC interface.

Solution First delete the interface from the FC fabric and then delete the interface from the QFX Series:

1. Delete the FC interface from the FC fabric to which it belongs:

   ```
   [edit]
   user@switch# delete fc-fabrics fabric-name interface interface-name
   ```

   For example, to delete the FC interface **fc-0/0/3.0** from an FC fabric named **sanfab1**:

   ```
   [edit]
   user@switch# delete fc-fabrics sanfab1 interface fc-0/0/3.0
   ```
2. Delete the FC interface at the **[edit interfaces]** hierarchy level:

   ```
   [edit]
   user@switch: delete interfaces interface-name
   ```

   For example, to delete the interface **fc-0/0/3.0** from the switch:

   ```
   [edit]
   user@switch: delete interfaces fc-0/0/3.0
   ```

Related Documentation

- fc-fabrics on page 338
- interface on page 349
- *interfaces*
- Understanding Interfaces on an FCoE-FC Gateway on page 50

Troubleshooting Dropped FIP Traffic

Problem **Description:** Fibre Channel over Ethernet (FCoE) Initialization Protocol (FIP) traffic such as FIP VLAN discovery and notification frames is dropped on the QFX Series.

Cause The interface on which the FIP traffic is dropped does not have a native VLAN configured. FIP VLAN discovery and notification messages are exchanged as untagged packets on the native VLAN. (After the FCoE session with the Fibre Channel switch is established, FCoE traffic uses the FCoE VLAN.)

Solution Check to ensure that every 10-Gigabit Ethernet interface that connects to an FCoE device includes a native VLAN. Configure a native VLAN on all 10-Gigabit Ethernet interfaces that connect to FCoE devices.

NOTE: **Make sure that the native VLAN you are using on the QFX Series is the same native VLAN that the FCoE devices use for Ethernet traffic.**

The procedure for configuring a native VLAN on an interface is different on switches that use the original CLI than on switches that use the Enhanced Layer 2 Software (ELS) CLI. This topic provides the configuration procedure for each CLI.

Configuring a Native VLAN on Switches Using the Original CLI

To configure a native VLAN on an interface:

1. Set the interface port mode to **tagged-access** if you have not already done so:

   `[edit]`
   `user@switch#` **set interfaces *interface* unit *unit* family ethernet-switching port-mode tagged-access**

   For example, to set the port mode to **tagged-access** for interface **xe-0/0/6.0**:

   `[edit]`
   `user@switch#` **set interfaces xe-0/0/6 unit 0 family ethernet-switching port-mode tagged-access**

2. Configure the native VLAN if it does not already exist:

   `[edit]`
   `user@switch#` **set vlans *vlan-name* vlan-id *vlan-id***

   For example, to name the native VLAN **native** and use the VLAN ID **1**:

   `[edit]`
   `user@switch#` **set vlans native vlan-id 1**

3. Configure the native VLAN on the interface:

   `[edit]`
   `user@switch#` **set interfaces *interface* unit *unit* family ethernet-switching native-vlan-id *vlan-id***

   For example, to configure a native VLAN with the VLAN ID **1** on interface **xe-0/0/6.0**:

   `[edit]`
   `user@switch#` **set interfaces xe-0/0/6 unit 0 family ethernet-switching native-vlan-id 1**

Configuring a Native VLAN on Switches Using the ELS CLI

To configure a native VLAN on an interface:

1. Set the interface mode to **trunk** if you have not already done so:

   `[edit]`
   `user@switch#` **set interfaces *interface* unit *unit* family ethernet-switching interface-mode trunk**

   For example, to set the interface mode to **trunk** for interface **xe-0/0/6.0**:

   `[edit]`
   `user@switch#` **set interfaces xe-0/0/6 unit 0 family ethernet-switching interface-mode trunk**

2. Configure the native VLAN if it does not already exist:

   `[edit]`
   `user@switch#` **set vlans *vlan-name* vlan-id *vlan-id***

   For example, to name the native VLAN **native** and use the VLAN ID **1**:

   `[edit]`
   `user@switch#` **set vlans native vlan-id 1**

3. Configure the native VLAN on the physical Ethernet interface:

```
[edit]
user@switch# set interfaces interface native-vlan-id vlan-id
```

For example, to configure a native VLAN with the VLAN ID **1** on interface **xe-0/0/6.0**:

```
[edit]
user@switch# set interfaces xe-0/0/6 native-vlan-id 1
```

4. Configure the Ethernet interface as a member of the native VLAN:

```
[edit]
user@switch# set interfaces interface unit unit family ethernet-switching vlan members
vlan-name
```

For example, to configure an Ethernet interface as a member of a native VLAN with the VLAN ID **1** on interface **xe-0/0/6.0**:

```
[edit]
user@switch# set interfaces xe-0/0/6 unit 0 family ethernet-switching vlan members native
```

Related Documentation

- *interfaces*
- *vlans*
- Understanding FIP Functions on page 38
- Configuring VLANs for FCoE Traffic on an FCoE Transit Switch on page 289