JUNIPER NETWORKS | Engineering Simplicity

Converged Networks (LAN and SAN) User Guide for EX Series Switches

Published
2024-12-10

JUNOS

Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Converged Networks (LAN and SAN) User Guide for EX Series Switches

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 https://support.juniper.net/support/eula/. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

About This Guide

Use this guide to configure data center bridging (DCB) functions to support storage area network (SAN) traffic on EX Series switches that do not use the Enhanced Layer 2 Software (ELS) configuration style. Supported features include DCB capabilities exchange (DCBX), Fibre Channel over Ethernet (FCoE) transit functions, FCoE Initialization Protocol (FIP) snooping, and Priority Flow Control (PFC) for managing lossless traffic classes.

NOTE: For configuring DCB functions on QFX Series switches and EX Series switches that support the Enhanced Layer 2 Software (ELS) configuration style, see Storage User Guide.

1

PART

Overview

CHAPTER 1

Converged Networks Overview

IN THIS CHAPTER

Understanding FIP Snooping

IN THIS SECTION

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 the FC network to access the network. You enable FIP snooping on FCoE VLANs when the switch is being used as an FCoE transit switch connecting FC initiators (servers) on the Ethernet network to FCoE forwarders (FCFs) at the FC storage area network (SAN) edge.

Through the FIP process, servers that have a converged network adapter (CNA) present an FCoE Node (ENode) that can log in to the FC network. The login process establishes a dedicated virtual link between the ENode and the FCF to emulate a point-to-point connection that passes transparently through the FCoE transit switch.

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

This topic describes:

FC Network Security

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

FIP snooping firewall filters emulate these security functions by preventing unauthorized access to the FCF through the transit switch and by ensuring the security of the virtual link between each ENode and the FCF. FIP snooping also prevents man-in-the-middle attacks.

FIP Snooping Functions

When you enable FIP snooping, the FCoE transit switch monitors FIP logins, solicitations, and advertisements that pass through it and gathers information about the ENode address and the address of the FCF. 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, the FCoE transit switch snoops the FIP information, constructs a *firewall filter* that permits access for the ENode, and adds the filter on all transit switch access ports associated with the FCoE VLAN.

The firewall filters allow FCoE frames to pass through the transit switch only between the server ENode FCoE port and the FCF FCoE port to which the server ENode has logged in. This ensures that ENodes can only connect to the FCFs they have successfully logged in to and that only valid FCoE traffic is transmitted. FIP snooping maintains the filters by tracking FCoE sessions.

FIP Snooping Firewall Filters

The FIP snooping firewall filters deny any FCoE traffic on the VLAN except for traffic originating from ENodes that have already logged in to the FCF.

FIP snooping performs these actions and checks to ensure that FCoE traffic is valid:

- Denies ENodes that use the FCF media access control (MAC) address as the source address.
- Denies all traffic from the ENode other than traffic addressed to the FCF that the Enode has logged into.
- Restricts the ENode to sending only FCoE protocol traffic on the virtual link.
- Allows the ENode to transmit only FIP and FCoE frames to the FCF address.
- Ensures that the FCoE source address an ENode uses after fabic login and fabric discovery (FDISC) is the address the FCF assigned to that ENode.
- Ensures that the FCoE source address the FCF assigns or accepts is only used for FCoE traffic.
- Ensures that FCoE frames are only addressed to the accepting FCF.

FIP Snooping Implementation

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

FCoE VLANs can include both access ports and trunk ports. Access ports face the hosts (FCoE servers and other FCoE initiators), and trunk ports face the FCF. When FIP snooping is enabled, the FCoE transit switch inspects both FIP frames and FCoE frames.

The FIP snooping implementation includes these considerations:

Server ENode-Facing Interfaces

We recommend that you enable FIP snooping on all FCoE access ports to ensure secure connections to FCFs. After you enable FIP snooping on an FCoE VLAN, the transit switch denies FCoE traffic from any server on that VLAN until the server performs a valid fabric login with an FCF.

FCF-Facing Interfaces

You must configure the interface that you are using to connect to an FCF as FCoE trusted interface, and it must be a 10 Gigabit Ethernet interface.

An FCoE trusted interface receives FCoE traffic only from an FCF. The following conditions apply to FCFs and FCF-facing interfaces:

- By default, FCFs are trusted entities.

- The FCoE transit switch always processes FCF frames because they come from a trusted source.

FCoE Mapped Address Prefix

When you enable FIP snooping on a VLAN, optionally you can specify the FCoE Mapped Address Prefix (FC-MAP) value for that VLAN if the network uses the fabric-provided MAC address (FPMA) addressing scheme. The FC-MAP value is a 24-bit value that identifies the FCF. The FCF combines the FC-MAP value with a unique 24-bit Fibre Channel ID (FCID) value for the server during the fabric login process, creating a unique 48-bit identifier. The FCF assigns the 48-bit value to the server ENode as its MAC address and unique identifier for the session. Each server session the ENode establishes with the FCF receives a unique FCID, so a server can host multiple virtual links to an FCF, each with a unique 48-bit address identifier.

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

T11 FIP Snooping Specification

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

RELATED DOCUMENTATION

Understanding Using an FCoE Transit Switch | 5

Example: Configuring an FCoE Transit Switch | 22

Configuring VN2VF_Port FIP Snooping and FCoE Trusted Interfaces on an FCoE Transit Switch

Understanding Using an FCoE Transit Switch

You can use an EX4500 switch 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 implement FCoE Initialization Protocol (FIP) snooping. The switch can transport 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 decapsulate 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 an FCoE forwarder (FCF), which is in an FC storage area network (SAN). The transit

switch acts as a passthrough switch and is transparent to the FCF, which detects each connection to an FCoE server as a direct point-to-point link.

When the switch acts as a transit switch, the VLANs you configure for FCoE traffic can use any of the switch ingress and egress ports, because the traffic in both directions is Ethernet traffic. FCoE traffic must use a VLAN dedicated only to FCoE traffic that does not carry any other traffic.

When the switch acts as a transit switch, you must enable *priority-based flow control* (PFC, IEEE standard 802.1Qbb) as a link-level flow control mechanism. See *Understanding Priority-Based Flow Control* for additional information. 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 transit switch transparently connects FCoE-capable servers in an Ethernet LAN to an FCF, which has both FCoE and FC interfaces and processes both the FCoE and FC protocol stacks. The transit switch acts as a transparent access layer between FCoE servers and the FCF.

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

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

RELATED DOCUMENTATION

Understanding FIP Snooping | 2

Understanding Priority-Based Flow Control

IN THIS SECTION

Priority-based flow control (PFC), IEEE standard 802.1Qbb, is a link-level flow control mechanism. The flow control mechanism is similar to that used by IEEE 802.3x Ethernet PAUSE, but it operates on individual priorities. Instead of pausing all traffic on a link, PFC allows you to selectively pause traffic according to its class.

This topic describes:

Reliability of Packet Delivery in Standard Ethernet Networks and in Layer 2 Networks

Standard Ethernet does not guarantee that a packet injected into the network will arrive at its intended destination. Reliability is provided by upper-layer protocols. Generally, a network path consists of multiple hops between the source and destination. A problem arises when transmitters send packets faster than receivers can accept them. When receivers run out of available buffer space to hold incoming flows, they silently drop additional incoming packets. This problem is generally resolved by upper-layer protocols that detect the drops and request retransmission.

Applications that require reliability in Layer 2 must have flow control that includes feedback from a receiver to a sender regarding buffer availability. Using IEEE 802.3x Ethernet PAUSE control frames, a receiver can generate a MAC control frame and send a PAUSE request to a sender when a specified threshold of receiver buffer has been filled to prevent buffer overflow. Upon receiving a PAUSE request, the sender stops transmission of any new packets until the receiver notifies the sender that it has sufficient buffer space to accept them again. The disadvantage of using Ethernet PAUSE is that it operates on the entire link, which might be carrying multiple traffic flows. Some traffic flows do not need flow control in Layer 2, because they are carrying applications that rely on upper-layer protocols for reliability. PFC enables you to configure Layer 2 flow control selectively for the traffic that requires it, such as Fibre Channel over Ethernet (FCoE) traffic, without impacting other traffic on the link. You can also enable PFC for other traffic types, such as iSCSI.

Calculations for Buffer Requirements When Using PFC PAUSE

The receive buffer must be large enough to accommodate all data that is received while the system is responding to a PFC PAUSE frame.

When you calculate buffer requirements, consider the following factors:

- Processing and queuing delay of the PFC PAUSE—In general, the time to detect the lack of sufficient buffer space and to transmit the PFC PAUSE is negligible. However, delays can occur if the switch detects a reduction in buffer space just as the transmitter is beginning to transmit a maximum length frame.
- Propagation delay across the media—The delay amount depends on the length and speed of the physical link.
- Response time to the PFC PAUSE frame

- Propagation delay across the media on the return path

> **NOTE**: We recommend that you configure at least 20 percent of the buffer size for the queue that is using PFC and that you do not specify the **exact** option.
>
> Because it is mandatory to explicitly configure a certain percentage of buffer size for PFC, you must also explicity configure some buffer size for any other forwarding classes that you are planning to use (including the default forwarding classes and the user-defined forwarding classes). The percentage that you allocate depends on the usage of the respective classes.

How PFC and Congestion Notification Profiles Work With or Without DCBX

PFC can be applied to an interface regardless of whether the Data Center Bridging Capability Exchange protocol (DCBX) is enabled.

However, automatic control and advertisement of PFC requires DCBX:

- When DCBX is enabled—DCBX detects the data center bridging (DCB) neighbor's PFC configuration, uses autonegotiation to advertise local and peer PFC configuration, and then enables or disables PFC depending on whether the configurations are compatible or not. When PFC is enabled, it uses the congestion notification profile, which you have configured and applied to the interface.
- When DCBX is not enabled—*Class of service* (CoS) triggers PFC when the incoming frame has a User Priority (UP) field that matches the three-bit pattern specified for the congestion notification profile.

To manually control the use of PFC on the interface regardless of the configuration of the peer data center devices, you can explicitly change the configuration of DCBX on the interface to disable PFC autonegotiation. See "Disabling DCBX to Disable PFC Autonegotiation on EX Series Switches (CLI Procedure)" on page 55. When PFC autonegotiation is disabled, PFC is triggered by the congestion notification profile for PFC regardless of the configuration of the DCB peer.

> **NOTE**: PFC functions effectively only when the peer devices connected to the local interface are also using PFC and are configured compatibly with the local interface. PFC must be symmetrical—if PFC is not configured to use the same traffic class (code point) on both the local and the peer interface, it does not have any impact on the traffic.

Table 1 on page 9 shows the one-to-one mapping between the UP field of an IEEE 802.1Q tagged frame, the traffic class, and the egress queue. In addition to setting a PFC congestion notification profile on an ingress port, you must set a forwarding class to match the priority specified in the PFC congestion notification profile and to forward the frame to the appropriate queue.

Juniper Networks EX Series Ethernet Switches support up to six traffic classes and allow you to associate those classes with six different congestion notification profiles. (The switches support up to 16 forwarding classes.)

Table 1: Input for PFC Congestion Notification Profile and Mapping to Traffic Class and Egress Queue

UP Field of IEEE-802.1Q Tagged Frame	Traffic Class	Egress Queue
000	TC 0	queue 0
001	TC 1	queue 1
010	TC 2	queue 2
011	TC 3	queue 3
100	TC4	queue 4
101	TC 5	queue 5

RELATED DOCUMENTATION

Understanding Data Center Bridging Capability Exchange Protocol for EX Series Switches | 12

Example: Configuring an FCoE Transit Switch | 22

Configuring Priority-Based Flow Control for an EX Series Switch (CLI Procedure)

schedulers

congestion-notification-profile

Understanding DCB Features and Requirements on EX Series Switches

IN THIS SECTION

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.

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.

You can use DCB features on CEE-enabled switches to transport converged Ethernet and FC traffic while providing the class-of-service (CoS) characteristics and other characteristics FC requires for transmitting storage traffic.

NOTE: This topic only applies to DCB features on EX Series switches that do not support the Enhanced Layer 2 Software (ELS) configuration style. EX4500 and EX4550 switches are the only non-ELS EX Series switches that support DCB features.

DCB features on ELS EX Series switches and QFX Series switches are described in *Understanding DCB Features and Requirements*.

This topic describes:

EX Series Switch DCB Features Overview

To accommodate FC traffic, DCB specifications provide:

- High-bandwidth interface
- A discovery and exchange protocol for communicating 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 Discovery Protocol (LLDP, described in IEEE 802.1AB).

- A flow control mechanism called *priority-based flow control* (PFC, described in IEEE 802.1Qbb) to help provide lossless transport.

NOTE: The switches support the DCBX standards and PFC, but do not support enhanced transmission selection (ETS) and quantized congestion notification (QCN).

Physical Interfaces

The switches provide the high-bandwidth interfaces (10-Gigabit Ethernet interfaces) required to support DCB and converged traffic. Your switch can have both 1-gigabit and 10-gigabit interfaces, depending on the configuration. DCBX works only on 10-gigabit, full-duplex interfaces. However, LLDP and DCBX are enabled by default on all the interfaces.

DCBX

DCB devices use DCBX to exchange configuration information with directly connected peers (switches and data center devices such as servers). DCBX is an extension of LLDP. If you attempt to enable DCBX on an interface on which LLDP is disabled, the configuration commit fails. See "Understanding Data Center Bridging Capability Exchange Protocol for EX Series Switches" on page 12 for details.

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 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

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 specified period of time. PFC allows you to assign special priority to a specific traffic class for a specified period of time without stopping the traffic assigned to other priorities on the link. You assign this priority by using a congestion notification profile.

The switches support up to six traffic classes and allow you to associate those classes with six different congestion notification profiles.

PFC enables you to provide lossless transport for traffic assigned to use the PFC congestion notification profile and to use standard 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 across the cable between devices
- Store frames that are already on the wire when the sender receives the PFC PAUSE

The amount of buffer space needed to prevent frame loss due to congestion depends on the cable length, cable speed, and processing speed.

The switch automatically sets the threshold for sending a PFC PAUSE frame to accommodate delay from cables as long as 984 feet (300 meters) and to accommodate large frames that might be on the wire when the switch sends the PAUSE. This ensures that the switch sends PAUSEframes early enough to allow the sender to stop transmitting before the receive buffers on the switch overflow.

RELATED DOCUMENTATION

Understanding Data Center Bridging Capability Exchange Protocol for EX Series Switches | 12

Example: Configuring an FCoE Transit Switch | 22

Understanding Data Center Bridging Capability Exchange Protocol for EX Series Switches

IN THIS SECTION

Data Center Bridging Capability Exchange protocol (DCBX) is a discovery and exchange protocol for communicating configuration and capabilities among neighbors to ensure consistent configuration across the data center bridging network. It is an extension of Link Layer Discovery Protocol (LLDP). If you attempt to enable DCBX on an interface on which LLDP is disabled, the configuration commit fails. Data center bridging devices use DCBX to exchange configuration information with directly connected peers (devices such as switches and servers in a data center bridging network).

NOTE: This topic applies only to DCBX on EX Series switches that do not support the Enhanced Layer 2 Software (ELS) configuration style. EX4500 and EX4550 switches are the only non-ELS EX Series switches that support DCBX.

DCBX support on ELS EX Series switches and QFX Series switches is described in *Understanding DCBX*.

You can use DCBX to:

- Discover the data center bridging capabilities of peers
- Detect data center bridging feature misconfiguration or mismatches between peers
- Automatically enable or disable *priority-based flow control* (PFC) on an interface depending on whether the PFC configuration of the local interface is the same as the PFC configuration of the DCB peer

This topic describes:

Basic DCBX Functioning

DCBX features support PFC, the Fibre Channel over Ethernet (FCoE) application, and other Layer 2 or Layer 4 applications (such as iSCSI). DCBX is enabled or disabled on a per-interface basis. The default autonegotiation behavior is: DCBX is enabled if the peer device connected to the interface also supports DCBX.

If the peer device connected to the interface does not support DCBX, DCBX remains enabled on the switch, but the switch detects that DCBX is not enabled on the peer and reports a misconfiguration for that interface when you issue the `show dcbx neighbors` command.

During negotiation of capabilities, the switch pushes the PFC configuration to an attached peer if the peer is configured as *willing* to learn the PFC configuration from other peers. The switch does not

support autoprovisioning and does not change its own configuration during autonegotiation to match the peer configuration—that is, the switch is not *willing* to learn the PFC configuration from peers.

DCBX and PFC

After you enable PFC on a switch interface, DCBX uses autonegotiation to control the operational state of PFC functionality.

DCB devices must use the same traffic class (code point) on both the local and peer device. If the peer device connected to the interface supports PFC and is provisioned for the same traffic class as the switch interface, DCBX sets the PFC operational state to enabled. If the peer device connected to the interface does not support PFC or is not provisioned for the same traffic class, DCBX sets the operational state to disabled.

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 remains enabled on that interface regardless of the peer configuration. To disable PFC on an interface, delete any PFC configuration on the interface.

DCBX and FCoE

DCBX is mandatory for running FCoE applications, because FCoE traffic requires PFC to ensure lossless transport and PFC is a component of DCBX.

The FCoE application is configured by default on DCBX interfaces. Because of the FCoE requirement for lossless transport, we recommend that you configure the interfaces that carry FCoE traffic for PFC. See *Configuring Priority-Based Flow Control for an EX Series Switch (CLI Procedure)*.

DCBX advertisement of the FCoE application functions as follows:

- If you configure the `fcoe` forwarding class and PFC congestion notification profile and assign these components to the interfaces that carry FCoE traffic, DCBX advertises their FCoE capability and assigned 802.1p code points to the DCB peer, and DCBX reports the FCoE capability and assigned 802.1p code points of the DCB peer to the switch.

DCBX and iSCSI

DCBX is not essential for iSCSI applications. These applications provide a method for linking data storage facilities over IP networks. Unlike Fibre Channel (FC) communications, which require special-purpose cabling, iSCSI can be run over long distances by using existing network infrastructure.

You might want to use iSCSI over DCB to reduce latency in a network that is oversubscribed. You might also want to use it to provide predictable and certain application responsiveness, eliminating Ethernet's dependence on TCP/IP for the retransmission of dropped Ethernet frames.

DCBX advertises switch interfaces that are configured to support the iSCSI application, their PFC capability, and their assigned 802.1p code points.

How DCBX Is Implemented on the Switches

On the switches, the implementation of DCBX is:

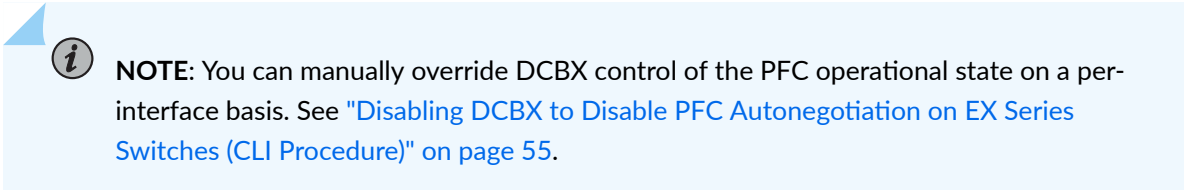
- Supported on aggregated Ethernet interfaces composed of 10-Gigabit Ethernet interfaces
- Enabled by default on all 10-Gigabit Ethernet interfaces

On the switches, DCBX supports the application type-length-value (TLV) — thus, DCBX interfaces on the switch can exchange information with their DCB peers about application capability, PFC capability, and 802.1p code-point settings. This implementation includes the following:

- The FCoE application is enabled by default on DCBX interfaces on the switch. Therefore, you do not configure an application map for the default FCoE application.

  The switches do not have a default FCoE forwarding class—therefore, you must explicitly configure a forwarding class with the name `fcoe` and associate that class with the interfaces carrying FCoE traffic. If PFC is enabled, the 802.1p code points are assigned, and the interfaces are associated with a forwarding class, the switch negotiates FCoE application capability on the DCBX interface.

- Do not explicitly configure an FCoE application map, because that generates a commit error.
- You can configure additional Layer 2 or Layer 4 applications to be supported by the DCBX application TLV feature. To do this, explicitly configure an application map and associate the application map with one of the DCBX interfaces. DCBX then advertises the application capabilities of the associated interface and checks the capabilities of the connected peer device.
- If the peer device connected to the local interface does not support PFC or the peer's PFC configuration is not the same as the local interface's PFC configuration, DCBX automatically disables PFC for the local interface.



NOTE: You can manually override DCBX control of the PFC operational state on a per-interface basis. See "Disabling DCBX to Disable PFC Autonegotiation on EX Series Switches (CLI Procedure)" on page 55.

Features That Are Not Fully Supported by DCBX on EX Series Switches

The implementation of DCBX on EX Series switches does not fully support the following features:

- Enhanced transmission selection (ETS) (IEEE 802.1Qaz)—ETS is a bandwidth management mechanism to support dynamic allocation of bandwidth for DCBX traffic classes.
  - EX Series switches do not support using ETS to dynamically allocate bandwidth to specified traffic classes. Instead, the switches handle all DCBX traffic as a single default traffic class, group 7.
  - However, the switches do support the ETS Recommendation TLV. The ETS Recommendation TLV communicates the ETS settings that the switch wants the connected DCBX peer interface to use.
  - If the peer interface is *willing*, it changes its configuration to match the configuration in the ETS Recommendation TLV sent by the switch (group 7).
  - The switch also advertises that it is not *willing* to change its ETS settings.
  - The advertisement of ETS TLV is enabled by default for DCBX interfaces. If you want, you can disable this advertisement. See *Disabling the ETS Recommendation TLV*.
- A default FCoE forwarding class—The switch does not have a default FCoE forwarding class with default mapping to a priority queue for FCoE traffic.

NOTE: Because the switches do not support a default FCoE forwarding class, you must explicitly configure a forwarding class and name it `fcoe`.

RELATED DOCUMENTATION

Understanding DCB Features and Requirements on EX Series Switches | 10

Understanding Using an FCoE Transit Switch | 5

Example: Configuring an FCoE Transit Switch | 22

Understanding DCBX Application Protocol TLV Exchange on EX Series Switches

IN THIS SECTION

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.

LLDP and DCBX are enabled by default on all 10-Gigabit Ethernet interfaces of EX4500 CEE-enabled switches.

NOTE: This topic applies only to DCBX on EX Series switches that do not support the Enhanced Layer 2 Software (ELS) configuration style. EX4500 and EX4550 switches are the only non-ELS EX Series switches that support DCBX.

DCBX TLV exchange on ELS EX Series switches and QFX Series switches is described in *Understanding DCBX Application Protocol TLV Exchange*.

This topic describes:

Basic Steps for Setting Up Application Protocol TLV Exchange

Setting up application protocol exchange for FCoE applications consists of:

- Configuring the **fcoe** forwarding class for IEEE 802.1p code point **011**
- Configuring PFC for IEEE 802.1p code point **011**

We recommend that you use code point **011** for the **fcoe** forwarding class, because this is the conventional IEEE 802.1p code point for FCoE traffic. We recommend that you configure PFC to use the same code point. See "Example: Configuring an FCoE Transit Switch" on page 22.

Setting up application protocol exchange for non-FCoE applications consists of:

- Defining applications
- Mapping the applications to IEEE 802.1p code points
- Configuring classifiers to prioritize incoming traffic map and map the incoming traffic to the application by the traffic code points
- Applying the application maps and classifiers to interfaces

Except for FCoE applications, you must explicitly define and map all applications that you want an interface to advertise.

NOTE: Do not explicitly configure an FCoE application map, because doing that generates a commit error.

Applications

Before an interface can exchange application protocol information, you must define the applications that you want to advertise, except for the FCoE application, which is defined by default. You can define:

- Layer 2 applications by EtherType
- Layer 4 applications (such as iSCSI applications) by a combination of protocol (TCP or UDP) and destination port

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.

The switch automatically defines the FCoE application as 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 (with the exception of the FCoE application).

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 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. Applications that you want an interface to advertise must be configured in the application map that you apply to the interface (except the FCoE application). Do not explicitly configure an FCoE application map, because doing that generates a commit error.

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 its trafffic 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.

Traffic for the FCoE application is classified and prioritized by your configuration of the **fcoe** forwarding class.

Requirements for Interfaces in Non-FCoE Applications to Exchange Application Protocol Information

For non-FCoE applications, interfaces on which you want to exchange application protocol TLVs must include the following two items:

- The application map that contains the application
- A classifier

See *Defining an Application for DCBX Application Protocol TLV Exchange* and *Configuring an Application Map for DCBX Application Protocol TLV Exchange*.

RELATED DOCUMENTATION

- Understanding Data Center Bridging Capability Exchange Protocol for EX Series Switches | **12**
- Understanding DCB Features and Requirements on EX Series Switches | **10**
- *Understanding Priority-Based Flow Control*
- Disabling DCBX Application Protocol Exchange on EX Series Switches (CLI Procedure) | **56**

2

PART

Configuration

CHAPTER 2

Configuration Examples

IN THIS CHAPTER

Example: Configuring an FCoE Transit Switch

IN THIS SECTION

You can use an EX4500 CEE-enabled switch as a Fibre Channel over Ethernet (FCoE) transit switch, enabling it to transport both FCoE and Ethernet LAN traffic. Using the same switch to support both your storage network and traditional IP-based data communications reduces the costs of powering, cooling, provisioning, maintaining, and managing your network.

This example includes:

- FIP snooping for security
- Priority-based flow control (PFC) for lossless transport
- The FCoE forwarding class for the DCBX application protocol type, length, value (TLV) exchange
- A trusted port connecting to the FCoE forwarder (FCF)
- Enlarged maximum transmission unit (MTU) size for handling FCoE traffic

This example shows how to configure an FCoE transit switch:

Requirements

This example uses the following hardware and software components:

- One EX4500 switch (CEE-capable model)
- Junos OS Release 12.1 or later for EX Series switches
- One FCoE Node (ENode)
- One FCoE forwarder (FCF)

Before you begin, be sure you have:

- Configured the VLAN `fcoe-vlan` on the switch. See Configuring VLANs for EX Series Switches.

Overview and Topology

IN THIS SECTION

FCoE transmissions are vulnerable to address spoofing and man-in-the-middle attacks, because they are not actually sent through point-to-point links. This example describes how to configure the switch so that it provides security similar to that provided by traditional Fibre Channel (FC) networks. The switch is transparent to the ENode and the FCF, so the ENode and FCF communicate just as they would for a point-to-point link.

FIP snooping is disabled by default. You enable 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 FIP snooping denies access for all other Ethernet traffic.

This example shows how to configure FIP snooping on a VLAN of the EX4500 switch that is connected with one ENode, that is, a server equipped with converged network adapters (CNAs). The setup for this example includes the VLAN `fcoe-vlan` on the switch.

This example also shows how to configure PFC on the interfaces that are being used for FCoE traffic and how to configure an FCoE trusted port to handle traffic between the switch and the FCF gateway to the storage area network (SAN).

You must configure PFC properties for the interfaces that are carrying FCoE traffic, because flow control must be implemented on the link level for this type of traffic.

NOTE: Data Center Bridging Capability Exchange protocol (DCBX) is enabled by default on all 10-Gigabit Ethernet interfaces on EX4500 switches. DCBX automatically controls whether PFC is enabled or disabled on the interface. However, you must configure the PFC properties selecting the traffic class and queue. See *Configuring Priority-Based Flow Control for an EX Series Switch (CLI Procedure)*.

You configure trunk interfaces that connect to the FCF as trusted interfaces. The switch must use the same FCoE MAC Address Prefix (FC-MAP) value that is being used by the FCF. Therefore, if the FCF is using a nondefault FC-MAP value, you must configure the FC-MAP value on the switch to match that value.

You must also enlarge the MTU size for all interfaces (both access and trunk) that are handling FCoE traffic to accommodate the maximum FC frame and Ethernet header sizes.

This example also includes configuring the `fcoe` forwarding class to be used for the FCoE traffic, so that it can take advantage of DCBX support for the Application Protocol TLV Exchange. See "Understanding Data Center Bridging Capability Exchange Protocol for EX Series Switches" on page 12 for additional information.

NOTE: Configuring and applying PFC and a forwarding class `fcoe` on the DCBX interfaces automatically enables the DCBX FCoE application protocol exchange on those interfaces. Do not explicitly configure an FCoE application map, because doing that generates a commit error. See "Understanding Data Center Bridging Capability Exchange Protocol for EX Series Switches" on page 12 for additional information.

NOTE: PFC is supported only on 10-Gigabit Ethernet interfaces.

NOTE: We recommend that you also:

- Configure the PFC congestion notification profile for the same 802.1p code points that you are using for the `fcoe` forwarding class. We recommend code point `011`, because this is the conventional IEEE 802.1p code point for FCoE traffic.
- Configure at least 20 percent of the buffer for the queue that is using PFC.
- Do not specify the `exact` option when configuring the buffer for the queue that is using PFC.

- Configure the `loss-priority` statement to `low` for a traffic class that is using PFC.
- Configure an appropriate percent of the buffer for any other forwarding classes (default forwarding classes and the user-defined forwarding classes) that you are using

Topology

The components of the topology for this example are shown in Table 2 on page 25.

Table 2: Components of the FCoE Security Topology

Properties	Settings
Switch hardware	One EX4500 CEE-enabled switch
VLAN name and ID	`fcoe-vlan`, tag `20`
Forwarding class for FCoE traffic	`fcoe`, code point `011`

Interfaces in `fcoe-vlan`	`xe-0/0/1`
	`xe-0/0/2`
	`xe-0/0/3`
	`xe-0/0/30`

FCoE trusted port to the FCF	`xe-0/0/30`

PFC interfaces	`xe-0/0/1`
	`xe-0/0/2`
	`xe-0/0/3`
	`xe-0/0/30`

CoS forwarding-class interface	`xe-0/0/30`
CoS scheduler-map interface	`xe-0/0/30`

Table 2: Components of the FCoE Security Topology *(Continued)*

Properties	Settings
Interfaces configured with MTU of 2500	xe-0/0/1 xe-0/0/2 xe-0/0/3 xe-0/0/30

In this example, the switch has already been configured as follows:

- All access ports are untrusted, which is the default setting.
- DCBX is enabled by default on all 10-Gigabit Ethernet interfaces.
- The port connecting the switch to the FCF is configured as a trunk port.

Configuration

IN THIS SECTION

To configure an FCoE transit switch, perform these tasks:

Procedure

CLI Quick Configuration

To quickly configure an FCoE transit switch, copy the following commands and paste them into the switch terminal window:

```
[edit]
set ethernet-switching-options secure-access-port vlan fcoe-vlan examine-fip fc-map 0x0EFC03
set ethernet-switching-options secure-access-port  interface xe-0/0/30 fcoe-trusted
set interfaces xe-0/0/1 ether-options no-flow-control
set interfaces xe-0/0/2 ether-options no-flow-control
set interfaces xe-0/0/3 ether-options no-flow-control
```

```
set interfaces xe-0/0/30 ether-options no-flow-control
set class-of-service congestion-notification-profile cn-profile input ieee-802.1 code-point 011
pfc
set class-of-service interfaces xe-0/0/1 congestion-notification-profile cn-profile
set class-of-service interfaces xe-0/0/2 congestion-notification-profile cn-profile
set class-of-service interfaces xe-0/0/3 congestion-notification-profile cn-profile
set class-of-service interfaces xe-0/0/30 congestion-notification-profile cn-profile
set class-of-service classifiers ieee-802.1 pfc-class import default
set class-of-service classifiers ieee-802.1 pfc-class forwarding-class fcoe loss-priority low
code-points 011
set class-of-service interfaces xe-0/0/1 unit 0 classifiers ieee-802.1 pfc-class
set class-of-service interfaces xe-0/0/2 unit 0 classifiers ieee-802.1 pfc-class
set class-of-service interfaces xe-0/0/3 unit 0 classifiers ieee-802.1 pfc-class
set class-of-service interfaces xe-0/0/30 unit 0 classifiers ieee-802.1 pfc-class
set class-of-service forwarding-classes class fcoe queue-num 3
set class-of-service schedulers pfc-sched buffer-size percent 25
set class-of-service schedulers default-sched buffer-size percent 17
set class-of-service scheduler-maps pfc-map forwarding-class fcoe scheduler pfc-sched
set class-of-service scheduler-maps pfc-map forwarding-class assured-forwarding scheduler
default-sched
set class-of-service scheduler-maps pfc-map forwarding-class best-effort scheduler default-sched
set class-of-service scheduler-maps pfc-map forwarding-class network-control scheduler default-
sched
set class-of-service scheduler-maps pfc-map forwarding-class expedited-forwarding scheduler
default-sched
set class-of-service interfaces xe-0/0/30 scheduler-map pfc-map
set interfaces xe-0/0/1 mtu 2500
set interfaces xe-0/0/2 mtu 2500
set interfaces xe-0/0/3 mtu 2500
set interfaces xe-0/0/30 mtu 2500
```

Step-by-Step Procedure

To configure an FCoE transit switch:

1. Enable FIP snooping on the VLAN and modify the FC-MAP value to match the FC-MAP value being used by the FCF:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set vlan fcoe-vlan examine-fip fc-map 0x0EFC03
```

2. Set the FCF-facing interface (`xe-0/0/30`) as FCoE-trusted:

```
[edit ethernet-switching-options secure-access-port]
user@switch# set interface xe-0/0/30 fcoe-trusted
```

3. Configure a congestion notification profile, specifying the name of the profile and applying it to the traffic class that is indicated by the User Priority bits in the 802.1Q tagged frame of an incoming packet:

NOTE: The ENode and the switch must use the same traffic class for the FCoE traffic. DCBX advertises the traffic class being used by the switch and detects the traffic class being used by the ENode. If there is a mismatch, the switch disables the PFC capability of the switch interface.

```
[edit class-of-service]
user@switch# set congestion-notification-profile cn-profile input ieee-802.1 code-point 011
pfc
```

NOTE: The configuration of PFC includes two different `ieee-802.1` configuration statements:

- *ieee-802.1 (Congestion Notification)*—Use to configure the congestion notification profile.
- ieee-802.1—Use to configure the CoS classifier.

4. Disable standard flow control on the interfaces that you want to use for the FCoE VLAN.

NOTE: PFC and standard flow control cannot be enabled on the same interface, and you must use PFC for FCoE traffic.

```
[edit interfaces]
user@switch# set xe-0/0/1 ether-options no-flow-control
user@switch# set xe-0/0/2 ether-options no-flow-control
user@switch# set xe-0/0/3 ether-options no-flow-control
user@switch# set xe-0/0/30 ether-options no-flow-control
```

5. Bind the congestion notification profile to all interfaces of the FCoE VLAN:

```
[edit class-of-service]
user@switch# set interface xe-0/0/1 congestion-notification-profile cn-profile
user@switch# set interface xe-0/0/2 congestion-notification-profile cn-profile
user@switch# set interface xe-0/0/3 congestion-notification-profile cn-profile
user@switch# set interface xe-0/0/30 congestion-notification-profile cn-profile
```

6. Create a CoS classifier for the fcoe forwarding class:

```
[edit class-of-service]
user@switch# set forwarding-classes fcoe queue-num 3
```

7. Configure this forwarding class (**fcoe**) to use a low loss priority value and to use the same code point that is used for PFC:

NOTE: We recommend that you use code point 011, because this is the conventional IEEE 802.1p code point for FCoE traffic.

```
[edit class-of-service]
user@switch# set classifiers ieee-802.1 pfc-class forwarding-class fcoe loss-priority low
code-points 011
```

8. Bind the `pfc-class` classifier to all interfaces of the FCoE VLAN:

```
[edit class-of-service]
user@switch# set interfaces xe-0/0/1 unit 0 classifiers ieee-802.1 pfc-class
user@switch# set interfaces xe-0/0/2 unit 0 classifiers ieee-802.1 pfc-class
user@switch# set interfaces xe-0/0/3 unit 0 classifiers ieee-802.1 pfc-class
user@switch# set interfaces xe-0/0/30 unit 0 classifiers ieee-802.1 pfc-class
```

9. Assign forwarding-class `fcoe` to an egress queue:

```
[edit class-of-service]
user@switch# set forwarding-classes fcoe queue-num 3
```

10. Set a scheduler for this queue, allocating at least 20 percent of the buffer to `pfc-sched`:

```
[edit class-of-service]
user@switch# set schedulers pfc-sched buffer-size percent 25
```

11. Set a scheduler for the default queue, allocating 17 percent of the buffer to that queue:

```
[edit class-of-service]
uuser@switch# set schedulers default-sched buffer-size percent 17
```

12. Configure a scheduler map (`pfc-map`) that associates the scheduler (`pfc-sched`) with the `fcoe` forwarding class and associates the default forwarding classes (assured-forwarding, best-effort and network-control) with the default schedule:

```
[edit class-of-service]
user@switch# set scheduler-maps pfc-map forwarding-class fcoe scheduler pfc-sched
user@switch# set scheduler-maps pfc-map forwarding-class assured-forwarding
schedulerdefault-sched
user@switch# set scheduler-maps pfc-map forwarding-class best-effort scheduler default-
sched
user@switch# set scheduler-maps pfc-map forwarding-class network-control scheduler default-
sched
user@switch# set scheduler-maps pfc-map forwarding-class expedited-forwarding scheduler
default-sched
```

13. Assign the scheduler map (`pfc-map`) to the FCF-facing interface (xe-0/0/30):

```
[edit class-of-service]
user@switch# set interfaces xe-0/0/30 scheduler-map pfc-map
```

14. Enlarge the MTU size to 2500 bytes for all the interfaces (both access and trunk) that are handling FCoE traffic:

```
[edit interfaces]
user@switch# set xe-0/0/1 mtu 2500
user@switch# set xe-0/0/2 mtu 2500
```

```
user@switch# set xe-0/0/3 mtu 2500
user@switch# set xe-0/0/30 mtu 2500
```

Results

Display the results of the configuration:

```
[edit]
user@switch#show
```

```
interfaces {
        xe-0/0/1 {
            mtu 2500;
            ether-options {
                no-flow-control;
            }
            unit 0 {
                family ethernet-switching {
                    vlan {
                        members fcoe-vlan;
                    }
                }
            }
        }
        xe-0/0/2 {
            mtu 2500;
            ether-options {
                no-flow-control;
            }
            unit 0 {
                family ethernet-switching {
                    vlan {
                        members fcoe-vlan;
                    }
                }
            }
        }
        xe-0/0/3 {
            mtu 2500;
            ether-options {
```

```
                no-flow-control;
            }
            unit 0 {
                family ethernet-switching {
                    vlan {
                        members fcoe-vlan;
                    }
                }
            }
        }
        xe-0/0/30 {
            mtu 2500;
            ether-options {
                no-flow-control;
            }
            unit 0 {
                family ethernet-switching {
                    port-mode trunk;
                    vlan {
                        members fcoe-vlan;
                    }
                }
            }
        }
    }
class-of-service {
    classifiers {
        ieee-802.1 pfc-class {
        import default;
        forwarding-class fcoe {
        loss-priority low code-points 011;
    }
    forwarding-classes {
        class fcoe queue-num 3;
    }
    congestion-notification-profile {
        cn-profile {
            input {
                ieee-802.1 {
                    code-point 011 {
                        pfc;
                    }
                }
```

```
            }
        }
    }
    interfaces {
        xe-0/0/1 {
            congestion-notification-profile cn-profile;
            unit 0 {
                classifiers {
                    ieee-802.1 pfc-class;
                }
            }
        }
        xe-0/0/2 {
        congestion-notification-profile cn-profile;
            unit 0 {
                classifiers {
                    ieee-802.1 pfc-class;
                }
            }
        xe-0/0/3 {
            congestion-notification-profile cn-profile;
            unit 0 {
                classifiers {
                    ieee-802.1 pfc-class;
                }
            }
        }
        xe-0/0/30 {
            congestion-notification-profile cn-profile;
            scheduler-map pfc-map;
            unit 0 {
                classifiers {
                    ieee-802.1 pfc-class;
                }
            }
        }
            scheduler-maps {
                pfc-map {
                    forwarding-class fcoe scheduler pfc-sched;
                    forwarding-class assured-forwarding scheduler default-sched;
                    forwarding-class best-effort scheduler default-sched;
                    forwarding-class network-control scheduler default-sched;
                    forwarding-class expedited-forwarding scheduler default-sched;
```

```
            }
        }
        schedulers {
            pfc-sched {
                buffer-size percent 25;
            }
            default-sched {
                buffer-size percent 17;
            }
        }
    }
}
ethernet-switching-options {
    secure-access-port {
        interface xe-0/0/30.0 {
            fcoe-trusted;
        }
        vlan fcoe-vlan {
            examine-fip {
                fc-map 0x0EFC03;
            }
        }
    }
}
```

Verification

IN THIS SECTION

Confirm that the configuration of the FCoE transit switch is working properly:

Verifying That FIP Snooping Is Working Correctly on the Switch

Purpose

Verify that FIP snooping is being implemented on the appropriate VLAN.

Action

Send some requests from ENodes to the switch.

Display the FIP snooping information :

```
user@switch> show fip snooping vlan detail fcoe-vlan

VLAN: fcoe-vlan,   FC-MAP: 0e:fc:03
  FCF Information
  FCF-MAC            : 30:10:94:01:00:00
  Active Sessions    : 2
  Configured FKA-ADV : 195
  Running FKA-ADV    : 73
    Enode Information
    Enode-MAC: 10:10:94:01:00:01,       Interface: xe-0/0/1
    Configured FKA-ADV : 195
    Running FKA-ADV    : 103
      Session Information
      VN-Port MAC: 0E:FC:03:01:0A:01,   FKA-ADV : 178
      VN-Port MAC: 0E:FC:03:01:0B:01,   FKA-ADV : 194
  FCF Information
  FCF-MAC            : 40:10:94:01:00:00
  Active Sessions    : 2
  Configured FKA-ADV : 258
  Running FKA-ADV    : 212
    Enode Information
    Enode-MAC: 20:10:94:01:00:02,       Interface: xe-0/0/0
    Configured FKA-ADV : 258
    Running FKA-ADV    : 242
      Session Information
      VN-Port MAC: 0E:FC:03:02:0C:02,   FKA-ADV : 254
      VN-Port MAC: 0E:FC:03:02:0D:02,   FKA-ADV : 269
```

Meaning

The output for this VLAN (`fcoe-vlan`) includes the FC MAP value that you configured. It shows the MAC addresses of the FCF and the ENode that are transmitting FCoE traffic through the switch.

Verifying That PFC is Enabled, That the FCoE Application Is Advertised, and That the Switch Interface and DCB Peer Are Using the Same 802.1p Code Points

Purpose

Verify that PFC is enabled on the local switch interface and on the peer interface, and that the local interface and the peer interface are using the same code point.

Action

Send some requests from ENodes to the switch.

Display the DCBX information advertised by the configured CoS forwarding class interface (`xe-0/0/30`) and detected by the switch:

```
user@switch> show dcbx neighbors interface xe-0/0/30

Interface : xe-0/0/30.0

    Protocol-State: in-sync

    Local-Advertisement:
        Operational version: 0
        sequence-number: 1, acknowledge-id: 1

    Peer-Advertisement:
        Operational version: 0
        sequence-number: 1, 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                 Disabled
          001                 Disabled
          010                 Disabled
          011                 Disabled
          100                 Disabled
          011                 Enabled
          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                 Disabled
          100                 Disabled
          011                 Enabled
          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


      Peer-Advertisement:

        Enable: Yes, Willing: No, Error: No
```

```
Appl-Name     Ethernet-Type     Socket-Number     Priority-Map     Status

  FCoE          0x8906                              00001000         Enabled
```

Meaning

PFC is a requirement for transmitting FCoE traffic and PFC works only when the local and peer devices are both enabled for PFC and are both using the same traffic class (code point) for transmitting the PFC traffic.

In the output for `Feature: PFC`, check the status of `Local-Advertisement` to verify that PFC is enabled. If DCBX detects a misconfiguration with the DCB peer, it disables the PFC capability. In this example, the PFC `Operational State` is `enabled`, because PFC is configured symmetrically on the switch and the DCB peer. Both devices are using code point `011` for forwarding the traffic.

If the results show that PFC is disabled, you van use the information provided by this command to reconfigure the congestion notification profile to match the code point being used for PFC by the peer device. See *Configuring Priority-Based Flow Control for an EX Series Switch (CLI Procedure)*.

`Appl-Name` shows the default FCoE application. The FCoE application always indicates `Ethernet-Type 0x8906`. The `Priority-Map` for the FCoE application shows the 8-bit format of the code-point setting that was specified for the PFC congestion notification profile. In this case, the three bit code point is 3, `011`. So the `Priority-Map` for the default FCoe application is `00001000`.

The `fcoe` forwarding-class and PFC were configured; and the configuration of the application on the switch and on the DCB are synchronized. Therefore, the `Status` of the FCoE application is `Enabled`.

If the configuration of the FCoE application on the switch did not match the FCoE application of the DCB peer, the status of the application would appear as `Disabled`.

RELATED DOCUMENTATION

Configuring VN2VF_Port FIP Snooping and FCoE Trusted Interfaces on an FCoE Transit Switch

Configuring Priority-Based Flow Control for an EX Series Switch (CLI Procedure)

Understanding Data Center Bridging Capability Exchange Protocol for EX Series Switches | **12**

congestion-notification-profile

Example: Configuring DCBX to Support an iSCSI Application

IN THIS SECTION

Data Center Bridging Capability Exchange protocol (DCBX) support for the application protocol type, length, and value (TLV) enables you to implement DCBX for various Layer 2 and Layer 4 applications. Internet small computer system interface (iSCSI) is a Layer 4 storage application that can benefit from DCBX. Implementing iSCSI over data center bridging (DCB) reduces latency in networks that are oversubscribed and provides a predictable and certain application responsiveness, eliminating Ethernet's dependence on TCP/IP for the retransmission of dropped Ethernet frames. Although DCBX is not a requirement for such applications, it adds the reliability required for enterprise data storage.

NOTE: You can configure and apply priority flow control (PFC) for any DCBX interfaces, but it is not a requirement for applications other than Fiber Channel over Ethernet (FCoE).

This example shows how to configure DCBX to support an iSCSI application:

Requirements

This example uses the following hardware and software components:

- One EX4500 switch (CEE-capable model)
- Junos OS Release 12.1 or later for EX Series switches

Overview and Topology

IN THIS SECTION

You can use the same switch to support your LAN traffic and your storage area network (SAN) traffic—including both FCoE and iSCSI traffic. The DCBX application protocol TLV allows you to associate a specific DCBX interface with a specific application map.

DCBX discovers the DCB capabilities of peers by exchanging feature configuration information, detects feature misconfiguration and mismatches, and can configure DCB on peers. DCBX is an extension of Link Layer Discovery Protocol (LLDP). 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 fails.

The switch supports DCBX information exchange for other applications, such as iSCSI, as specified in your configuration by EtherType or by the destination port and protocol.

To take advantage of this feature for non-FCoE applications, you must configure the application and application map and associate the application map with the interface that is carrying the application's traffic. This configuration includes specifying the 802.1 code points to be used for this application.

When you configure an iSCSI application, you must always designate **destination-port 3260.**

NOTE: DCBX is enabled by default on all 10-Gigabit Ethernet interfaces on EX4500 switches (CEE-capable models).

This example shows how to configure an iSCSI application on a DCBX interface of the EX4500 switch that is connected to an iSCSI storage device.

Topology

The components of the topology for this example are shown in Table 3 on page 40.

Table 3: Components of the DCBX iSCSI Topology

Properties	Settings
Switch hardware	One EX4500 switch (CEE capable model)
Application	**iSCSI**
Application map code points	**101**
Interface for iSCSI application	**xe-0/0/37**

Table 3: Components of the DCBX iSCSI Topology *(Continued)*

Properties	Settings
Destination port	3260

In this example, the switch has already been configured as follows:

- DCBX is enabled by default on all 10-Gigabit Ethernet interfaces.

Configuration

IN THIS SECTION

- Procedure | 41

To configure DCBX to support an iSCSI application, perform these tasks:

Procedure

CLI Quick Configuration

To quickly configure a DCBX interface for an iSCSI application, copy the following commands and paste them into the switch terminal window:

```
[edit]
set applications application iscsi protocol tcp destination-port 3260
set policy-options application-maps iscsi-map application iscsi code-points 101
set protocols dcbx interface xe-0/0/37 application-map iscsi-map
```

Step-by-Step Procedure

Configure a DCBX interface for an iSCSI application:

1. Create the application:

```
[edit]
user@switch# set applications application iscsi protocol tcp destination-port 3260
```

2. Create the application map:

```
[edit policy-options]
user@switch# set application-maps iscsi-map application iscsi code-points 101
```

3. Apply the application map to the DCBX interface that you want to use for iSCSI:

```
[edit protocols]
user@switch# set dcbx interface xe-0/0/37 application-map iscsi-map
```

Results

Check the results of the configuration:

```
user@switch> show configuration
protocols {
    dcbx {
        interface all;
        interface xe-0/0/37.0 {
            application-map iscsi-map;
        }
    }
    lldp {
        interface all;
    }
}
policy-options {
    application-maps {
        iscsi-map {
            application iscsi code-points 101;
        }
    }
}
applications {
```

```
    application iscsi {
        protocol tcp;
        destination-port 3260;
    }
}
```

Verification

IN THIS SECTION

To confirm that the configuration is working properly:

Verifying That the iSCSI Application Is Advertised and That the Switch Interface and DCB Peer Are Using the Same 802.1p Code Points

Purpose

Verify that both the switch and the DCB peer are using a DCBX iSCSI application configured for the same 802.1p code points.

Action

Send some requests from the switch to the DCB peer.

Display the DCBX information advertised by DCBX interface (**xe-0/0/37**) and detected by the switch:

```
user@switch> show dcbx neighbors interface

Interface : xe-0/0/37.0

    Protocol-State: in-sync
    Active-application-map: iscsi-map


    Local-Advertisement:
        Operational version: 0
```

```
        sequence-number: 1, acknowledge-id: 1

    Peer-Advertisement:
        Operational version: 0
        sequence-number: 1, acknowledge-id: 1

    Feature: PFC, Protocol-State: in-sync

      Operational State: Disabled

      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                 Disabled
          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                 Disabled
          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

        iscsi                         3260             00100000        Enabled



      Peer-Advertisement:

        Enable: Yes, Willing: No, Error: No


        Appl-Name    Ethernet-Type    Socket-Number    Priority-Map    Status


        iscsi                         3260             00100000        Enabled
```

Meaning

Check the status for **Local-Advertisement** in the section **Feature: Application**.

If there is misconfiguration between the switch and the DCB peer, the status displays **Error: Yes**.

In this example, there is no error. The output for **Feature: Application, Protocol-State**, displays a list of DCBX applications under **Appl-Name**.

This field displays information for the user-configured application **iscsi**. When you configure an iSCSI application, you must always designate the destination port as **3260**. The output displays this as the **Socket-Number** .

The **Priority-Map** for the iSCSI application reflects the 802.1p code points that were specified in this example for the **iSCSI-map**. The example specified **101** for the iSCSI application map code points. The **Priority-Map** is an 8-bit code point format of the 802.1p code points; thus, **0010000**.

The **Status** of the iSCSI application is **Enabled**, because the switch and the DCB are using the same code points for the iSCI application.

RELATED DOCUMENTATION

Example: Configuring an FCoE Transit Switch | 22

Defining an Application for DCBX Application Protocol TLV Exchange

Configuring an Application Map for DCBX Application Protocol TLV Exchange

Applying an Application Map to an Interface for DCBX Application Protocol TLV Exchange

Understanding DCBX Application Protocol TLV Exchange on EX Series Switches | 17

CHAPTER 3

Configuration Tasks

IN THIS CHAPTER

Configuring VN2VF_Port FIP Snooping and FCoE Trusted Interfaces on an FCoE Transit Switch

SUMMARY

On a Fibre Channel (FC) over Ethernet (FCoE) transit switch, VN_Port to VF_Port FCoE Initialization Protocol (FIP) snooping sets up firewall filters to prevent unauthorized access through the transit switch to an FC switch or FCoE forwarder (FCF). You configure FIP snooping using different commands on FCoE transit switches that use the Enhanced Layer 2 Software (ELS) configuration style than on switches that don't use ELS.

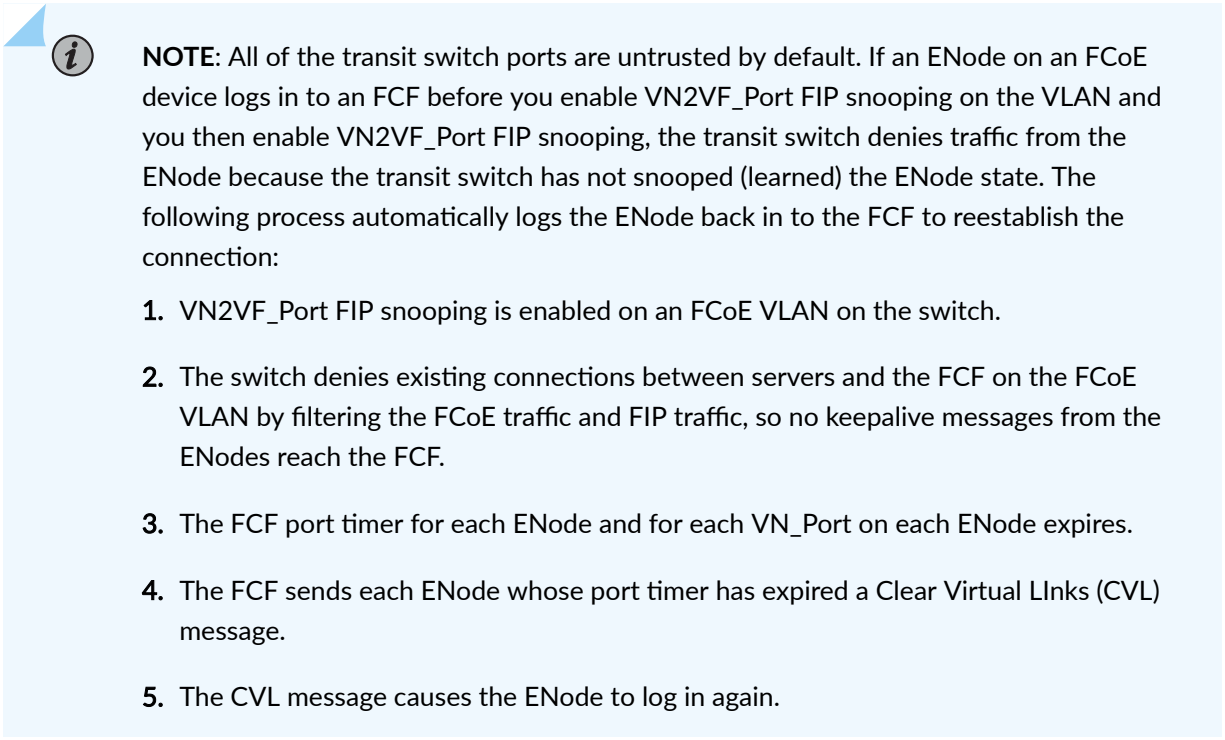
IN THIS SECTION

Considerations When Configuring VN2VF_Port FIP Snooping

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 switch 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.

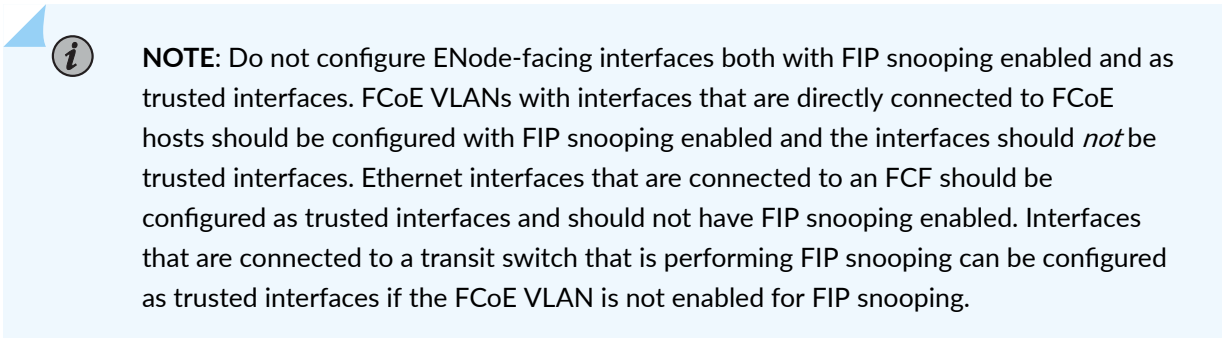
VN2VF_Port FIP snooping is disabled by default. You enable VN2VF_Port FIP snooping on a per-VLAN basis for VLANs that carry FCoE traffic. Ensure that a VLAN that carries FCoE traffic carries only FCoE traffic, because enabling VN2VF_Port FIP snooping denies access for all other Ethernet traffic.



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 some differences in the CLI commands you use to configure FIP snooping and FCoE trusted interfaces on a transit switch depending on whether the switch uses the Enhanced Layer 2 Software (ELS) configuration style or the original non-ELS CLI.

Configure VN2VF_Port FIP Snooping on ELS FCoE Transit Switches

Configure the following to enable VN2VF_Port FIP snooping on FCoE transit switches that run the Enhanced Layer 2 Software (ELS) CLI:

- Enable VN2VF_Port FIP snooping on a VLAN and optionally specify the 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.

- 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
```

Configure VN2VF_Port FIP Snooping on non-ELS FCoE Transit Switches

Configure either of the following to enable VN2VF_Port FIP snooping on FCoE transit switches that don't use ELS, depending on whether you want to specify an FC-MAP value or use the default FC-MAP value:

- 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
```

- 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
```

RELATED DOCUMENTATION

- Example: Configuring an FCoE Transit Switch
- *Configuring an FCoE VLAN Interface on an FCoE-FC Gateway*
- *Configuring VLANs for FCoE Traffic on an FCoE Transit Switch*
- *Configuring an FCoE LAG*
- *Disabling Enhanced FIP Snooping Scaling*
- Understanding FIP Snooping
- *Understanding VN_Port to VF_Port FIP Snooping on an FCoE Transit Switch*
- *Understanding FCoE LAGs*

Configuring Priority-Based Flow Control for an EX Series Switch (CLI Procedure)

You can configure priority-based flow control (PFC) to apply link-level flow control on a specific traffic class so that different types of traffic can efficiently use the same network interface card (NIC). You must configure PFC for all interfaces carrying Fibre Channel over Ethernet (FCoE) traffic. You can also configure PFC on interfaces carrying other traffic types, such as Internet small computer system interface (iSCSI) traffic. Using PFC is optional for traffic types other than FCoE.

NOTE:

- PFC is supported only on 10-Gigabit Ethernet interfaces.
- If you are using PFC for a non-FCoE DCBX application, use the same 802.1p code points for the PFC congestion notification profile and for the application map that is carrying that application traffic.

Data Center Bridging Capability Exchange protocol (DCBX) is enabled by default on all 10-Gigabit Ethernet interfaces. DCBX enables or disables PFC on the local interface depending on whether the PFC configuration on that interface is the same as the PFC configuration of the connected interface on the data center bridging (DCB) peer.

NOTE: When you configure PFC, we recommend that you:

- Configure at least 20 percent of the buffer for the queue that is using PFC.
- Configure an appropriate percent of the buffer for any other forwarding classes (default forwarding classes and the user-defined forwarding classes) that you are using.
- Do not specify the `exact` option when configuring the buffer for the queue that is using PFC.
- Configure the `loss-priority` statement to `low` for a traffic class that is using PFC.
- Verify that the PFC configurations of the local interfaces are the same as the PFC configurations of the connected interfaces on the DCB peer. See *show dcbx neighbors*.

EX Series switches support up to six congestion notification profiles for PFC.

To configure PFC:

1. Configure a congestion notification profile, specifying the name of the profile and specifying the three-bit pattern of the User Priority bits in an incoming frame that will trigger the priority-based flow control on that traffic class:

```
[edit class-of-service]
user@switch# set congestion-notification-profile profile-name  input ieee-802.1 code-point
up-bits pfc
```

2. Disable standard Ethernet flow control on the interfaces that will be used for the traffic class that you have selected for PFC:

```
[edit interfaces]
user@switch# set interface-name  ether-options no-flow-control
```

NOTE: You cannot apply PFC to interfaces that are using standard Ethernet flow control. You must first disable flow control on those interfaces.

3. Bind the congestion notification profile to the interfaces that will be used for the traffic class that you have selected for PFC:

```
[edit class-of-service]
user@switch# set interfaces interface-name  congestion-notification-profile profile-name
```

4. Create a CoS classifier for a traffic class that will use PFC:

```
[edit class-of-service]
user@switch# set classifiers ieee-802.1 classifier-name import default
```

5. Configure this traffic class (***classifier-name***) to use a user-defined or default forwarding class with a low loss priority value and specify the 802.1p code points::

```
[edit class-of-service]
user@switch# set classifiers ieee-802.1 classifier-name forwarding-class class-name loss-
priority low code-points 3 bit-patterns
```

6. Bind the *classifier-name* classifier to all interfaces that require PFC:

```
[edit class-of-service]
user@switch# set interfaces interface-name unit logical-unit-number classifiers ieee-802.1
classifier-name
```

7. Assign the specified forwarding-class to an egress queue:

```
[edit class-of-service]
user@switch# set forwarding-classes class-name queue-number
```

8. Set a scheduler for this queue, allocating at least 20 percent of the buffer to be used for FCoE traffic:

```
[edit class-of-service]
user@switch# set schedulers scheduler-name  buffer-size percent
```

9. Set a scheduler to allocate buffer space for forwarding classes carrying other traffic:

NOTE: You must explicitly allocate some buffer space for the other forwarding classes. The default allocation of buffer space for forwarding classes is overridden when you manually configure the requisite amount of buffer space for the FCoE traffic.

```
[edit class-of-service]
user@switch# set scheduler-name  buffer-size percent
```

10. Configure a scheduler map that associates the specified scheduler with the specified forwarding class:

```
[edit class-of-service]
user@switch# set scheduler-maps map-name forwarding-class class-name scheduler scheduler-
name
```

For example:

```
[edit class-of-service]
user@switch# set scheduler-maps pfc-map forwarding-class af2 scheduler pfc-sched
user@switch# set scheduler-maps pfc-map forwarding-class best-effort scheduler default-
sched user@switch# set scheduler-maps pfc-map forwarding-class network-control scheduler
default-sched
user@switch# set scheduler-maps pfc-map forwarding-class expedited-forwarding scheduler
default-sched
```

11. Assign the scheduler map to the egress interface:

```
[edit class-of-service]
user@switch# set interfaces interface-name scheduler-map pfc-map
```

RELATED DOCUMENTATION

Example: Configuring an FCoE Transit Switch | 22

Understanding Priority-Based Flow Control

congestion-notification-profile

Disabling DCBX to Disable PFC Autonegotiation on EX Series Switches (CLI Procedure)

As part of its autonegotiation capabilities, the Data Center Bridging Capability Exchange protocol (DCBX) automatically does the following:

- Advertises the priority flow control (PFC) configuration of the local interfaces to directly connected peers (switches and data center devices such as servers)
- Detects the PFC capabilities of the connected peers
- Enables the local interface's PFC capabilities if DCBX detects that the peer interface's PFC configuration is the same as the PFC configuration of the local interface.
- Disables the local interface's PFC capabilities if DCBX detects that the peer interface's PFC configuration is not the same as the PFC configuration of the local interface.

DCBX is enabled by default on all 10-Gigabit Ethernet interfaces on EX4500 switches. You can manually override DCBX control of the PFC operational state on a per-interface basis. You might want to disable autonegotiation if the DCB peer does not support PFC.

To disable the DCBX control of the PFC operational state:

- On an individual interface:

```
[edit protocols]
user@switch# set dcbx interface interface-name priority-flow-control no-auto-negotiation
```

- On all 10-Gigabit Ethernet interfaces:

```
[edit protocols]
user@switch# set dcbx interface all priority-flow-control no-auto-negotiation
```

RELATED DOCUMENTATION

show dcbx neighbors

Understanding DCB Features and Requirements on EX Series Switches | **10**

Disabling DCBX Application Protocol Exchange on EX Series Switches (CLI Procedure)

You can disable the Data Center Bridging Capability Exchange protocol (DCBX) Application Protocol exchange on a specific interface or on all interfaces.

To disable the DCBX application protocol exchange for any DCBX application, do the following:

NOTE: The format of the configuration statement specifies the **fcoe** application, however, this applies to *any* DCBX application (both FCoE applications and non-FCoE applications) on that interface.

- For a specific interface:

```
[edit protocols]
user@switch# set dcbx interface interface-name applications fcoe no-auto-negotiation
```

- For all interfaces:

```
[edit protocols]
user@switch# set dcbx interface all applications fcoe no-auto-negotiation
```

NOTE: If you disable the DCBX application protocol exchange, the *show dcbx neighbors* command displays **Feature: Application, Protocol-State: not-applicable**.

RELATED DOCUMENTATION

show dcbx neighbors

Disabling DCBX to Disable PFC Autonegotiation on EX Series Switches (CLI Procedure) | **55**

Understanding DCB Features and Requirements on EX Series Switches | **10**

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*
- *Applying an Application Map to an Interface for DCBX Application Protocol TLV Exchange*
- *Configuring DCBX Autonegotiation*
- *Example: Configuring DCBX Application Protocol TLV Exchange*
- Example: Configuring DCBX to Support an iSCSI Application
- *Understanding DCBX Application Protocol TLV Exchange*
- *show dcbx neighbors*

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

Applying an Application Map to an Interface for DCBX Application Protocol TLV Exchange

Configuring DCBX Autonegotiation

Example: Configuring DCBX Application Protocol TLV Exchange

Example: Configuring DCBX to Support an iSCSI Application

show dcbx neighbors

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

Configuring an Application Map for DCBX Application Protocol TLV Exchange

Configuring DCBX Autonegotiation

Example: Configuring DCBX Application Protocol TLV Exchange

Example: Configuring DCBX to Support an iSCSI Application

show dcbx neighbors

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

Configuring DCBX Autonegotiation

Understanding DCBX

Understanding Data Center Bridging Capability Exchange Protocol for EX Series Switches
  ```

PART 3

Configuration Statements and Operational Commands

Junos CLI Reference Overview

We've consolidated all Junos CLI commands and configuration statements in one place. Learn about the syntax and options that make up the statements and commands and understand the contexts in which you'll use these CLI elements in your network configurations and operations.

- Junos CLI Reference

Click the links to access Junos OS and Junos OS Evolved configuration statement and command summary topics.

- Configuration Statements
- Operational Commands