JUNIPER NETWORKS

Network Configuration Example

Configuring DCBX Application Protocol TLV Exchange

Release
NCE 63

Modified: 2016-08-01

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

Network Configuration Example Configuring DCBX Application Protocol TLV Exchange
NCE 63

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

CHAPTER 1

Configuring DCBX Application Protocol TLV Exchange

About This Network Configuration Example

This network configuration example describes the Data Center Bridging Capability Exchange (DCBX) Protocol, identifies the advantages of enabling DCBX protocol, and provides a step-by-step procedure for configuring DCBX application protocol type, length, and value (TLV) exchange.

Using DCBX Protocol to Lower Costs

Data Center Bridging Capability Exchange (DCBX) protocol 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.

The advantage of configuring DCBX application protocol TLV exchanges is that incoming traffic on the interface is mapped to the appropriate application, receives the correct loss priority, and is placed in the output queue associated with the correct forwarding class.

This application-aware capability allows you to properly support any L2 or L4 application including the Internet Small Computer System Interface (iSCSI) and Fibre Channel over Ethernet (FCoE) applications. The convergence of technologies into a single fabric provides several benefits such as: using less space, generating less heat, using less electricity, requiring less knowledge, and performing less maintenance. It also provides greater granularity in control of bandwidth allocation to ensure bandwidth is used more effectively.

Ultimately this agility enables you to achieve greater IT efficiencies and lower your costs.

Related Documentation

- Understanding DCBX Application Protocol TLV Exchange on page 6
- Example: Configuring DCBX Application Protocol TLV Exchange on page 10
- Configuring DCBX Autonegotiation on page 21

Understanding DCBX Application Protocol TLV Exchange

Data Center Bridging Capability Exchange protocol (DCBX) discovers the data center bridging (DCB) capabilities of connected peers. DCBX also advertises the capabilities of applications on interfaces by exchanging application protocol information through application type, length, and value (TLV) elements. DCBX is an extension of Link Layer Discovery Protocol (LLDP). LLDP must remain enabled on every interface on which you want to use DCBX.

NOTE: **LLDP and DCBX are enabled by default on all interfaces.**

Setting up application protocol exchange consists of:

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

You need to explicitly define the applications that you want an interface to advertise. The FCoE application is a special case (see "Applications" on page 7) and only needs to be defined on an interface if you want DCBX to exchange application protocol TLVs for other applications in addition to FCoE on that interface.

You also need to explicitly map all of the defined applications that you want an interface to advertise to IEEE 802.1p code points in an application map. The FCoE application is a special case that only requires inclusion in an application map when you want an interface to use DCBX for other applications in addition to FCoE, as described later in this topic (see "Application Maps" on page 7).

This topic describes:

Applications

Before an interface can exchange application protocol information, you need to define the applications that you want to advertise. The exception is the FCoE application. If FCoE is the only application that you want the interface to advertise, then you do not need to define the FCoE application. You need to define the FCoE application only if you want interfaces to advertise other applications in addition to FCoE.

NOTE: If FCoE is the only application that you want DCBX to advertise on an interface, DCBX exchanges FCoE application protocol TLVs by default if the interface:

- Carries FCoE traffic (traffic mapped by CoS configuration to the FCoE forwarding class and applied to the interface)
- Has a congestion notification profile with PFC enabled on the FCoE priority (IEEE 802.1p code point)
- Does *not* have an application map

If you apply an application map to an interface, then all applications that you want DCBX to advertise must be defined and configured in the application map, including the FCoE application.

If no CoS configuration for FCoE is mapped to an interface, that interface does not exchange FCoE application protocol TLVs.

You can define:

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

The EtherType is a two-octet field in the Ethernet frame that denotes the protocol encapsulated in the frame. For a list of common EtherTypes, see http://standards.ieee.org/develop/regauth/ethertype/eth.txt on the IEEE standards organization website. For a list of port numbers and protocols, see the *Service Name and Transport Protocol Port Number Registry* at http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml on the Internet Assigned Numbers Authority (IANA) website.

You must explicitly define each application that you want to advertise, except FCoE. The FCoE application is defined by default (EtherType 0x8906).

Application Maps

An application map maps defined applications to one or more IEEE 802.1p code points. Each application map contains one or more applications. DCBX includes the configured application code points in the protocol TLVs exchanged with the connected peer.

To exchange protocol TLVs for an application, you must include the application in an application map. The FCoE application is a special case:

- If you want DCBX to exchange application protocol TLVs for more than one application on a particular interface, you must configure the applications, define an application map to map the applications to code points, and apply the application map to the interface. In this case, you must also define the FCoE application and add it to the application map.

  This is the same process and treatment required for all other applications. In addition, for DCBX to exchange FCoE application TLVs, you must enable priority-based flow control (PFC) on the FCoE priority (the FCoE IEEE 802.1p code point) on the interface.

- If FCoE is the only application that you want DCBX to advertise on an interface, then you do not need to configure an application map and apply it to the interface. By default, when an interface has no application map, and the interface carries traffic mapped to the FCoE forwarding class, and PFC is enabled on the FCoE priority, the interface advertises FCoE TLVs (autonegotiation mode). DCBX exchanges FCoE application protocol TLVs by default until you apply an application map to the interface, remove the FCoE traffic from the interface (you can do this by removing the or editing the classifier for FCoE traffic), or disable PFC on the FCoE priority.

  If you apply an application map to an interface that did not have an application map and was exchanging FCoE application TLVs, and you do not include the FCoE application in the application map, the interface stops exchanging FCoE TLVs. Every interface that has an application map must have FCoE included in the application map (and PFC enabled on the FCoE priority) in order for DCBX to exchange FCoE TLVs.

Mapping an application to code points does two things:

- Maps incoming traffic with the same code points to that application
- Allows you to configure classifiers that map incoming application traffic, by code point, to a forwarding class and a loss priority, in order to apply class of service (CoS) to application traffic and prioritize application traffic

You apply an application map to an interface to enable DCBX application protocol exchange on that interface for each application specified in the application map. All of the applications that you want an interface to advertise must be configured in the application map that you apply to the interface, with the previously noted exception for the FCoE application when FCoE is the only application for which you want DCBX to exchange protocol TLVs on an interface.

Classifying and Prioritizing Application Traffic

When traffic arrives at an interface, the interface classifies the incoming traffic based on its code points. Classifiers map code points to loss priorities and forwarding classes. The loss priority prioritizes the traffic. The forwarding class determines the traffic output queue and CoS service level.

When you map an application to an IEEE 802.1p code point in an application map and apply the application map to an interface, incoming traffic on the interface that matches the application code points is mapped to the appropriate application. The application receives the loss priority and the CoS associated with the forwarding class for those code points, and is placed in the output queue associated with the forwarding class.

You can use the default classifier or you can configure a classifier to map the application code points defined in the application map to forwarding classes and loss priorities.

Enabling Interfaces to Exchange Application Protocol Information

Each interface with the **fcoe** forwarding class and PFC enabled on the FCoE code point is enabled for FCoE application protocol exchange by default until you apply an application map to the interface. If you apply an application map to an interface and you want that interface to exchange FCoE application protocol TLVs, you must include the FCoE application in the application map. (In all cases, to achieve lossless transport, you must also enable PFC on the FCoE code point or code points.)

Except when FCoE is the only protocol you want DCBX to advertise on an interface, interfaces on which you want to exchange application protocol TLVs must include the following two items:

- The application map that contains the application(s)
- A classifier

NOTE: **You must also enable PFC on the code point of any traffic for which you want to achieve lossless transport.**

Disabling DCBX Application Protocol Exchange

To disable DCBX application protocol exchange for all applications on an interface, issue the **set protocols dcbx interface *interface-name* applications no-auto-negotiation** command.

You can also disable DCBX application protocol exchange for applications on an interface by deleting the application map from the interface, or by deleting a particular application from the application map. However, when you delete an application from an application map, the application protocol is no longer exchanged on any interface which uses that application map.

On interfaces that use IEEE DCBX mode to exchange DCBX parameters, you can disable sending the enhanced transmission selection (ETS) Recommendation TLV to the peer if you want an asymmetric ETS configuration between the peers.

Related Documentation

- *Understanding DCBX*
- Configuring DCBX Autonegotiation on page 21
- *Disabling the ETS Recommendation TLV*
- *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*

Example: Configuring DCBX Application Protocol TLV Exchange

Data Center Bridging Capability Exchange protocol (DCBX) discovers the data center bridging (DCB) capabilities of connected peers by exchanging application configuration information. DCBX detects feature misconfiguration and mismatches and can configure DCB on peers. DCBX is an extension of the Link Layer Discovery Protocol (LLDP). LLDP must remain enabled on every interface on which you want to use DCBX.

NOTE: **LLDP and DCBX are enabled by default on all interfaces.**

The switch supports DCBX application protocol exchange for Layer 2 and Layer 4 applications such as the Internet Small Computer System Interface (iSCSI). You specify applications by EtherType (for Layer 2 applications) or by the destination port and protocol (for Layer 4 applications; the protocol can be either TCP or UDP).

The switch handles Fibre Channel over Ethernet (FCoE) application protocol exchange differently than other protocols in some cases:

- If FCoE is the only application for which you want to enable DCBX application protocol TLV exchange on an interface, you do not have to explicitly configure the FCoE application or an application map. By default, the switch exchanges FCoE application protocol TLVs on all interfaces that carry FCoE traffic (traffic mapped to the **fcoe** forwarding class) and have priority-based flow control (PFC) enabled on the FCoE priority (the FCoE IEEE 802.1p code point). The default priority mapping for the FCoE application is IEEE 802.1p code point 011 (the default **fcoe** forwarding class code point).
- If you want an interface to use DCBX to exchange application protocol TLVs for any other applications in addition to FCoE, you must configure the applications (including FCoE), define an application map (including FCoE), and apply the application map to the interface. If you apply an application map to an interface, you must explicitly configure the FCoE application, or the interface does not exchange FCoE application protocol TLVs.

This example shows how to configure interfaces to exchange both Layer 2 and Layer 4 applications by configuring one interface to exchange iSCSI and FCoE application protocol information and configuring another interface to exchange iSCSI and Precision Time Protocol (PTP) application protocol information.

Requirements

This example uses the following hardware and software components:

- Juniper Networks QFX Series device
- Junos OS Release 12.1 or later for the QFX Series

Overview

The switch supports DCBX application protocol exchange for:

- Layer 2 applications, defined by EtherType
- Layer 4 applications, defined by destination port and protocol

NOTE: DCBX also advertises PFC and enhanced transmission selection (ETS) information. See "Configuring DCBX Autonegotiation" on page 21 for how DCBX negotiates and advertises configuration information for these features and for the applications.

DCBX is configured on a per-interface basis for each supported feature or application. For applications that you want to enable for DCBX application protocol exchange, you must:

- Define the application name and configure the EtherType or the destination port and protocol (TCP or UDP) of the application. Use the EtherType for Layer 2 applications, and use the destination port and protocol for Layer 4 protocols.
- Map the application to an IEEE 802.1p code point in an application map.
- Add the application map to DCBX interface.

In addition, for all applications (including FCoE, even when you do not use an application map), you either must create an IEEE 802.1p classifier and apply it to the appropriate ingress interfaces or use the default classifier. A classifier maps the code points of incoming traffic to a forwarding class and a loss priority so that ingress traffic is assigned to the correct class of service (CoS). The forwarding class determines the output queue on the egress interface.

If you do not create classifiers, trunk and tagged-access ports use the unicast IEEE 802.1 default trusted classifier. Table 1 on page 12 shows the default mapping of IEEE 802.1 code-point values to unicast forwarding classes and loss priorities for ports in trunk mode or tagged-access mode. Table 2 on page 12 shows the default untrusted classifier IEEE 802.1 code-point values to unicast forwarding class mapping for ports in access mode.

Table 1: Default IEEE 802.1 Classifiers for Trunk Ports and Tagged-Access Ports (Default Trusted Classifier)

Code Point	Forwarding Class	Loss Priority
be (000)	best-effort	low
be1 (001)	best-effort	low
ef (010)	best-effort	low
ef1 (011)	fcoe	low
af11 (100)	no-loss	low
af12 (101)	best-effort	low
nc1 (110)	network-control	low
nc2 (111)	network-control	low

Table 2: Default IEEE 802.1 Unicast Classifiers for Access Ports (Default Untrusted Classifier)

Code Point	Forwarding Class	Loss Priority
000	best-effort	low
001	best-effort	low
010	best-effort	low
011	best-effort	low
100	best-effort	low
101	best-effort	low
110	best-effort	low
111	best-effort	low

Topology

This example shows how to configure DCBX application protocol exchange for three protocols (iSCSI, PTP, and FCoE) on two interfaces. One interface exchanges iSCSI and FCoE application protocol information, and the other interface exchanges iSCSI and PTP application protocol information.

NOTE: You must map FCoE traffic to the interfaces on which you want to forward FCoE traffic. You must also enable PFC on the FCoE interfaces and create an ingress classifier for FCoE traffic, or else use the default classifier.

Table 3 on page 13 shows the configuration components for this example.

Table 3: Components of DCBX Application Protocol Exchange Configuration Topology

Component	Settings
Hardware	QFX Series device
LLDP	Enabled by default on Ethernet interfaces
DCBX	Enabled by default on Ethernet interfaces
iSCSI application (Layer 4)	Application name—**iscsi** protocol—**TCP** destination-port—**3260** code-points—**111**
PTP application (Layer 2)	Application name—**ptp** ether-type—**0x88F7** code-points—**001, 101**
FCoE application (Layer 2)	Application name—**fcoe** ether-type—**0x8906** code-points—**011** NOTE: You explicitly configure the FCoE application because you are applying an application map to the interface. When you apply an application map to an interface, all applications must be explicitly configured and included in the application map.
Application maps	**dcbx-iscsi-fcoe-app-map**—Maps the iSCSI and FCoE applications to IEEE 802.1p code points **dcbx-iscsi-ptp-app-map**—Maps iSCSI and PTP applications to IEEE 802.1p code points
Interfaces	**xe-0/0/10**—Configured to exchange FCoE and iSCSI application TLVs (uses application map **dcbx-iscsi-fcoe-app-map**, carries FCoE traffic, and has PFC enabled on the FCoE priority) **xe-0/0/11**—Configured to exchange iSCSI and PTP application TLVs (uses application map **dcbx-iscsi-ptp-app-map**)

Table 3: Components of DCBX Application Protocol Exchange Configuration Topology *(continued)*

Component	Settings
PFC congestion notification profile for FCoE application exchange	**fcoe-cnp**: • Code point—**011** • Interface—**xe-0/0/10**
Behavior aggregate classifiers (map forwarding classes to incoming packets by the packet's IEEE 802.1 code point)	**fcoe-iscsi-cl1**: • Maps the **fcoe** forwarding class to the IEEE 802.1p code point used for the FCoE application (**011**) and a loss priority of **high** • Maps the **network-control** forwarding class to the IEEE 802.1p code point used for the iSCSI application (**111**) and a loss priority of **high** • Applied to interface **xe-0/0/10** **iscsi-ptp-cl2**: • Maps the **network-control** forwarding class to the IEEE 802.1p code point used for the iSCSI application (**111**) and a loss priority of **low** • Maps the **best-effort** forwarding class to the IEEE 802.1p code points used for the PTP application (**001** and **101**) and a loss priority of **low** • Applied to interface **xe-0/0/11**

NOTE: This example does not include scheduling (bandwidth allocation) configuration or lossless configuration for the iSCSI forwarding class.

Configuration

CLI Quick Configuration

To quickly configure DCBX application protocol exchange, copy the following commands, paste them in a text file, remove line breaks, change variables and details to match your network configuration, and then copy and paste the commands into the CLI at the **[edit]** hierarchy level.

```
set applications application iSCSI protocol tcp destination-port 3260
set applications application FCoE ether-type 0x8906
set applications application PTP ether-type 0x88F7
set policy-options application-maps dcbx-iscsi-fcoe-app-map application iSCSI code-points 111
set policy-options application-maps dcbx-iscsi-fcoe-app-map application FCoE code-points 011
set policy-options application-maps dcbx-iscsi-ptp-app-map application iSCSI code-points 111
set policy-options application-maps dcbx-iscsi-ptp-app-map application PTP code-points [001
101]
set protocols dcbx interface xe-0/0/10 application-map dcbx-iscsi-fcoe-app-map
set protocols dcbx interface xe-0/0/11 application-map dcbx-iscsi-ptp-app-map
set class-of-service congestion-notification-profile fcoe-cnp input ieee-802.1 code-point 011 pfc
set class-of-service interfaces xe-0/0/10 congestion-notification-profile fcoe-cnp
set class-of-service classifiers ieee-802.1 fcoe-iscsi-cl1 import default forwarding-class fcoe
loss-priority high code-points 011
set class-of-service classifiers ieee-802.1 fcoe-iscsi-cl1 import default forwarding-class
network-control loss-priority high code-points 111
```

set class-of-service classifiers ieee-802.1 iscsi-ptp-cl2 import default forwarding-class network-control loss-priority low code-points 111
set class-of-service classifiers ieee-802.1 iscsi-ptp-cl2 import default forwarding-class best-effort loss-priority low code-points [001 101]
set class-of-service interfaces xe-0/0/10 unit 0 classifiers ieee-802.1 fcoe-iscsi-cl1
set class-of-service interfaces xe-0/0/11 unit 0 classifiers ieee-802.1 iscsi-ptp-cl2

Configuring DCBX Application Protocol TLV Exchange

Step-by-Step Procedure

To define the applications, map the applications to IEEE 802.1p code points, apply the applications to interfaces, and create classifiers for DCBX application protocol exchange:

1. Define the iSCSI application by specifying its protocol and destination port, and define the FCoE and PTP applications by specifying their EtherTypes.

   ```
   [edit applications]
   user@switch# set application iSCSI protocol tcp destination-port 3260
   user@switch# set application FCoE ether-type 0x8906
   user@switch# set application PTP ether-type 0x88F7
   ```

2. Define an application map that maps the iSCSI and FCoE applications to IEEE 802.1p code points.

   ```
   [edit policy-options]
   user@switch# set application-maps dcbx-iscsi-fcoe-app-map application iSCSI code-points 111
   user@switch# set application-maps dcbx-iscsi-fcoe-app-map application FCoE code-points 011
   ```

3. Define the application map that maps the iSCSI and PTP applications to IEEE 802.1p code points.

   ```
   [edit policy-options]
   user@switch# set application-maps dcbx-iscsi-ptp-app-map application iSCSI code-points 111
   user@switch# set application-maps dcbx-iscsi-ptp-app-map application PTP code-points [001 101]
   ```

4. Apply the iSCSI and FCoE application map to interface **xe-0/0/10**, and apply the iSCSI and PTP application map to interface **xe-0/0/11**.

   ```
   [edit protocols dcbx]
   user@switch# set interface xe-0/0/10 application-map dcbx-iscsi-fcoe-app-map
   user@switch# set interface xe-0/0/11 application-map dcbx-iscsi-ptp-app-map
   ```

5. Create the congestion notification profile to enable PFC on the FCoE code point (**011**), and apply the congestion notification profile to interface **xe-0/0/10**.

   ```
   [edit class-of-service]
   user@switch# set congestion-notification-profile fcoe-cnp input ieee-802.1 code-point 011 pfc
   user@switch# set interfaces xe-0/0/10 congestion-notification-profile fcoe-cnp
   ```

6. Configure the classifier to apply to the interface that exchanges iSCSI and FCoE application information.

   ```
   [edit class-of-service classifiers]
   user@switch# set ieee-802.1 fcoe-iscsi-cl1 import default forwarding-class fcoe loss-priority high code-points 011
   user@switch# set ieee-802.1 fcoe-iscsi-cl1 import default forwarding-class network-control loss-priority high code-points 111
   ```


7. Configure the classifier to apply to the interface that exchanges iSCSI and PTP application information.

   ```
   [edit class-of-service classifiers]
   user@switch# set ieee-802.1 iscsi-ptp-cl2 import default forwarding-class network-control loss-priority low code-points 111
   user@switch# set ieee-802.1 iscsi-ptp-cl2 import default forwarding-class best-effort loss-priority low code-points [001 101]
   ```

8. Apply the classifiers to the appropriate interfaces.

   ```
   [edit class-of-service]
   user@switch# set interfaces xe-0/0/10 unit 0 classifiers ieee-802.1 fcoe-iscsi-cl1
   user@switch# set interfaces xe-0/0/11 unit 0 classifiers ieee-802.1 iscsi-ptp-cl2
   ```

Verification

To verify that DCBX application protocol exchange configuration has been created and is operating properly, perform these tasks:

- Verifying the Application Configuration on page 16
- Verifying the Application Map Configuration on page 17
- Verifying DCBX Application Protocol Exchange Interface Configuration on page 17
- Verifying the PFC Configuration on page 18
- Verifying the Classifier Configuration on page 18

Verifying the Application Configuration

Purpose Verify that DCBX applications have been configured.

Action List the applications by using the configuration mode command **show applications**:

```
user@switch# show applications
application iSCSI {
   protocol tcp;
   destination-port 3260;
}

application fcoe {
   ether-type 0x8906;
}

application ptp {
   ether-type 0x88F7;
}
```

Meaning The **show applications** configuration mode command lists all of the configured applications and either their protocol and destination port (Layer 4 applications) or their EtherType (Layer 2 applications). The command output shows that the iSCSI application is configured with the **tcp** protocol and destination port **3260**, the FCoE application is configured with the EtherType **0x8906**, and that the PTP application is configured with the EtherType **0x88F7**.

Verifying the Application Map Configuration

Purpose Verify that the application maps have been configured.

Action List the application maps by using the configuration mode command **show policy-options application-maps**:

```
user@switch# show policy-options application-maps
dcbx-iscsi-fcoe-app-map {
   application iSCSI code-points 111;
   application FCoE code-points 011;
}

dcbx-iscsi-ptp-app-map {
   application iSCSI code-points 111;
   application PTP code-points [001 101];
}
```

Meaning The **show policy-options application-maps** configuration mode command lists all of the configured application maps and the applications that belong to each application map. The command output shows that there are two application maps, **dcbx-iscsi-fcoe-app-map** and **dcbx-iscsi-ptp-app-map**.

The application map **dcbx-iscsi-fcoe-app-map** consists of the iSCSI application, which is mapped to IEEE 802.1p code point **111**, and the FCoE application, which is mapped to IEEE 802.1p code point **011**.

The application map **dcbx-iscsi-ptp-app-map** consists of the iSCSI application, which is mapped to IEEE 802.1p code point **111**, and the PTP application, which is mapped to IEEE 802.1p code points **001** and **101**.

Verifying DCBX Application Protocol Exchange Interface Configuration

Purpose Verify that the application maps have been applied to the correct interfaces.

Action List the application maps by using the configuration mode command **show protocols dcbx**:

```
user@switch# show protocols dcbx
interface xe-0/0/10.0 {
   application-map dcbx-iscsi-fcoe-app-map;
}

interface xe-0/0/11.0 {
   application-map dcbx-iscsi-ptp-app-map;
}
```

Meaning The **show protocols dcbx** configuration mode command lists whether the interfaces are enabled for DCBX and lists the application map applied to each interface. The command output shows that interfaces **xe-0/0/10.0** and **xe-0/0/11.0** are enabled for DCBX, and that interface **xe-0/0/10.0** uses application map **dcbx-iscsi-fcoe-app-map**, and interface **xe-0/0/11.0** uses application map **dcbx-iscsi-ptp-app-map**.

Verifying the PFC Configuration

Purpose Verify that PFC has been enabled on the FCoE code point and applied to the correct interface.

Action Display the PFC configuration to verify that PFC is enabled on the FCoE code point (**011**) in the congestion notification profile **fcoe-cnp** by using the configuration mode command **show class-of-service congestion-notification-profile**:

```
user@switch# show class-of-service congestion-notification-profile
fcoe-cnp {
    input {
        ieee-802.1 {
            code-point 011 {
                pfc;
            }
        }
    }
}
```

Display the class-of-service (CoS) interface information to verify that the correct interface has PFC enabled for the FCoE application by using the configuration mode command **show class-of-service interfaces**:

```
user@switch# show class-of-service interfaces
xe-0/0/10 {
    congestion-notification-profile fcoe-cnp;
}
```

NOTE: The sample output does not include all of the information this command can show. The output is abbreviated to focus on verifying the PFC configuration.

Meaning The **show class-of-service congestion-notification-profile** configuration mode command lists the configured congestion notification profiles. The command output shows that the congestion notification profile **fcoe-cnp** has been configured and has enabled PFC on the IEEE 802.1p code point **011** (the default FCoE code point).

The **show class-of-service interfaces** configuration mode command shows the interface CoS configuration. The command output shows that the congestion notification profile **fcoe-cnp**, which enables PFC on the FCoE code point, is applied to interface **xe-0/0/10**.

Verifying the Classifier Configuration

Purpose Verify that the classifiers have been configured and applied to the correct interfaces.

Action Display the classifier configuration by using the configuration mode command **show class-of-service**:

```
user@switch# show class-of-service
```



```
classifiers {
    ieee-802.1 fcoe-iscsi-cl1 {
        import default;
        forwarding-class network-control {
            loss-priority high code-points 111;
        }
        forwarding-class fcoe {
            loss-priority high code-points 011;
        }
    }
    ieee-802.1 iscsi-ptp-cl2 {
        import default;
        forwarding-class network-control {
            loss-priority low code-points 111;
        }
        forwarding-class best-effort {
            loss-priority low code-points [ 001 101 ];
        }
    }
}
interfaces {
    xe-0/0/10 {
        congestion-notification-profile fcoe-cnp;
        unit 0 {
            classifiers {
                ieee-802.1 fcoe-iscsi-cl1;
            }
        }
    }
    xe-0/0/11 {
        unit 0 {
            classifiers {
                ieee-802.1 iscsi-ptp-cl2;
            }
        }
    }
}
```

NOTE: The sample output does not include all of the information this command can show. The output is abbreviated to focus on verifying the classifier configuration.

Meaning The **show class-of-service** configuration mode command lists the classifier and CoS interface configuration, as well as other information not shown in this example. The command output shows that there are two classifiers configured, **fcoe-iscsi-cl1** and **iscsi-ptp-cl2**.

Classifier **fcoe-iscsi-cl1** uses the **default** classifier as a template and edits the template as follows:

- The forwarding class **network-control** is set to a loss priority of **high** and is mapped to code point **111** (the code point mapped to the iSCSI application).

- The forwarding class **fcoe** is set to a loss priority of **high** and is mapped to code point **011** (the code point mapped by default to the FCoE application).

Classifier **iscsi-ptp-cl2** uses the **default** classifier as a template and edits the template as follows:

- The forwarding class **network-control** is set to a loss priority of **low** and is mapped to IEEE 802.1p code point **111** (the code point mapped to the iSCSI application).
- The forwarding class **best-effort** is set to a loss priority of **low** and is mapped to IEEE 802.1p code points **001** and **101** (the code points mapped by default to the PTP application).

The command output also shows that classifier **fcoe-iscsi-cl1** is mapped to interface **xe-0/0/10.0** and that classifier **iscsi-ptp-cl2** is mapped to interface **xe-0/0/11.0**.

Related Documentation

- *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*
- Configuring DCBX Autonegotiation on page 21
- *show dcbx*
- *show dcbx neighbors*
- Understanding DCBX Application Protocol TLV Exchange on page 6

Configuring DCBX Autonegotiation

Data Center Bridging Capability Exchange protocol (DCBX) discovers the data center bridging (DCB) capabilities of peers by exchanging feature configuration information. DCBX also detects feature misconfiguration and mismatches, and can configure DCB on peers. DCBX is an extension of the Link Layer Discovery Protocol (LLDP), and LLDP must remain enabled on every interface for which you want to use DCBX. If you attempt to enable DCBX on an interface on which LLDP is disabled, the configuration commit operation fails.

NOTE: LLDP and DCBX are enabled by default on all interfaces.

The switch supports DCBX autonegotiation for:

- Priority-based flow control (PFC) configuration
- Layer 2 and Layer 4 applications such as Fibre Channel over Ethernet (FCoE) and Internet Small Computer System Interface (iSCSI)
- Enhanced transmission selection (ETS) advertisement

DCBX autonegotiation is configured on a per-interface basis for each supported feature or application. The PFC and application DCBX exchanges use autonegotiation by default. The default autonegotiation behavior is:

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

You can override the default behavior for each feature by turning off autonegotiation to force an interface to enable or disable the feature.

Autonegotiation of ETS means that when ETS is enabled on an interface (priority groups are configured), the interface advertises its ETS configuration to the peer device. In this case, priorities (forwarding classes) that are not part of a priority group (forwarding class set) receive no bandwidth and are advertised in an automatically generated default forwarding class. If ETS is not enabled on an interface (no priority groups are configured), all of the priorities are advertised in one automatically generated default priority group that receives 100 percent of the port bandwidth.

Disabling ETS autonegotiation prevents the interface from sending the Recommendation TLV or the Configuration TLV to the connected peer.

On interfaces that use IEEE DCBX mode to exchange DCBX parameters, you can disable autonegotiation of the ETS Recommendation TLV to the peer if you want an asymmetric ETS configuration between the peers. DCBX still exchanges the ETS Configuration TLV if you disable the ETS Recommendation TLV.

Autonegotiation of PFC means that when PFC is enabled on an interface, if the peer device connected to the interface supports PFC and is provisioned compatibly with the switch, DCBX sets the PFC operational state to enabled. If the peer device connected to the interface does not support PFC or is not provisioned compatibly with the switch, DCBX sets the operational state to disabled.

In addition, if the peer advertises that it is "willing" to learn its PFC configuration from the switch, DCBX pushes the switch's PFC configuration to the peer and does not check the peer's administrative state. The switch does not learn PFC configuration from peers (the switch does not advertise its state as "willing").

Disabling PFC autonegotiation prevents the interface from exchanging PFC configuration information with the peer. It forces the interface to enable PFC if PFC is configured on the interface or to disable PFC if PFC is not configured on the interface. If you disable PFC autonegotiation, the assumption is that the peer is also configured manually.

Autonegotiation of applications depends on whether or not you apply an application map to an interface. If you apply an application map to an interface, the interface autonegotiates DCBX for each application in the application map. PFC must be enabled on the FCoE priority (the FCoE IEEE 802.1p code point) for the interface to advertise the FCoE application. The interface only advertises applications that are included in the application map.

For example, if you apply an application map to an interface and the application map does not include the FCoE application, then that interface does not perform DCBX advertisement of FCoE.

If you do not apply an application map to an interface, DCBX does not advertise applications on that interface, with the exception of FCoE, which is handled differently than other applications.

NOTE: **If you do not apply an application map to an interface, the interface performs autonegotiation of FCoE if the interface carries traffic in the FCoE forwarding class and also has PFC enabled on the FCoE priority. On such interfaces, if DCBX detects that the peer device connected to the interface supports FCoE, the switch advertises its FCoE capability and IEEE 802.1p code point on that interface. If DCBX detects that the peer device connected to the interface does not support FCoE, DCBX marks that interface as "FCoE down" and disables FCoE on the interface.**

When DCBX marks an interface as "FCoE down," the behavior of the switch depends on how you use it in the network:

- When the switch acts as an FCoE transit switch, the interface drops all of the FIP packets it receives. In addition, FIP packets received from an FCoE forwarder (FCF) are not forwarded to interfaces marked as "FCoE down."
- When the switch acts as an FCoE-FC gateway (only switches that support native Fibre Channel interfaces), it does not send or receive FCoE Initialization Protocol (FIP) packets.

Disabling autonegotiation prevents the interface from exchanging application information with the peer. In this case, the assumption is that the peer is also configured manually.

To disable DCBX autonegotiation of PFC, applications (including FCoE), and ETS using the CLI:

1. Turn off autonegotiation for PFC.

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

2. Turn off autonegotiation for applications.

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

3. Turn off autonegotiation for ETS.

   `[edit]`
   `user@switch#` **set protocols dcbx interface *interface-name* enhanced-transmission-selection no-auto-negotiation**

To disable autonegotiation of the ETS Recommendation TLV so that DCBX exchanges only the ETS Configuration TLV:

- `[edit protocols dcbx interface` *`interface-name`*`]`
  `user@switch#` **set enhanced-transmission-selection no-recommendation-tlv**

Related Documentation

- Example: Configuring DCBX Application Protocol TLV Exchange on page 10
- *Example: Configuring CoS PFC for FCoE Traffic*
- *Disabling the ETS Recommendation TLV*
- Understanding DCBX Application Protocol TLV Exchange on page 6

