

Configuring FCoE NPV

- FCoE NPV Overview, on page 1
- VNP Ports, on page 3
- Licensing Requirements for FCoE NPV, on page 3
- Information About Virtual Interfaces, on page 4
- Guidelines and Limitations for Configuring FCoE NPV, on page 7
- Configuring FC/FCoE, on page 8
- Configuring QoS, on page 10
- Configuring FCoE NPV, on page 16
- Verifying the FCoE NPV Configuration, on page 28
- FCoE NPV Core Switch and FCoE NPV Edge Switch Configuration Example, on page 30
- FCoE NPV Core Switch and FCoE NPV Edge Switch with Implicit vFC Configuration Example, on page 32
- Verifying the Virtual Interface , on page 34
- Mapping VSANs to VLANs Example Configuration , on page 36
- SAN Boot with vPC, on page 37

FCoE NPV Overview

Fiber Channel over Ethernet (FCoE) N-port Virtulization (NPV) is an enhanced form of FCoE Initialization Protocol (FIP) snooping that provides a secure method to connect FCoE-capable hosts to an FCoE-capable FCoE forwarder (FCF) device.

FCoE NPV enables:

- The switch to act as an N-port virtualizer (NPV) connected to the core switch (FCF).
- The core switch (FCF) to view the NPV switch as another host.
- The multiple hosts connected to the NPV switch are presented as virtualized N-ports on the core switch (FCF).

FCoE NPV Benefits

FCoE NPV provides the following:

- FCoE NPV provides the advantages of NPV to FCoE deployments (such as preventing domain ID sprawl and reducing Fiber-Channel Forwarder (FCF) table size).
- FCoE NPV provides a secure connect between FCoE hosts and the FCoE FCF.
- FCoE NPV does not have the management and troubleshooting issues that are inherent to managing hosts remotely at the FCF.
- FCoE NPV implements FIP snooping as an extension to the NPV function while retaining the traffic-engineering, VSAN-management, administration, and trouble shooting aspects of NPV.

FCoE NPV Features

The following are the FCoE NPV features:

- Automatic load balance of server logins
  - The server interfaces (Host logins) are distributed in a round robin fashion among the available multiple uplinks (NP ports or external-interfaces).
  - You can enable disruptive automatic load balancing to load balance the existing server interfaces (hosts) to newly added NP uplink interfaces.

    Example:

    ```
    switch(config)# npv auto-load-balance disruptive
    ```

- Traffic mapping
  - You can specify the NP uplinks that a server interface can use to connect to core switches.
  - If the current mapped uplink goes down, the server does not log in through other available uplinks.

    Example:

    ```
    switch(config)# npv traffic-map server-interface vfc2/1 external-interface vfc2/1
    ```

- FCoE forwarding in the FCoE NPV bridge.
- FCoE NPV supports the Data Center Bridging Exchange Protocol (DCBX).
- FCoE frames received over VNP ports are forwarded only if the L2_DA matches one of the FCoE MAC addresses assigned to hosts on the VF ports.



Note FCoE NPV over port channel VNP ports use automatic traffic mapping only for FIP negotiations. FCoE traffic distribution over port channel VNP ports is based on the computed hash value.

Fibre Channel Slow Drain Device Detection and Congestion Avoidance

The data traffic between the end devices in Fibre Channel over Ethernet (FCoE) uses link level and per-hop based flow control. When the slow devices are attached to the fabric, the end devices do not accept the frames at a configured rate. The presence of the slow devices leads to traffic congestion on the links. The traffic

congestion affects the unrelated flows in the fabric that use the same inter-switch links (ISLs) for its traffic, even though the destination devices do not experience the slow drain.

Slow drain device detection and congestion avoidance is supported on below platform switches:

- N9K-C93360YC-FX2
- N9K-C9336C-FX2-E
- N9K-C93180YC-EX
- N9K-X9732C-EX Line Card
- N9K-C93180LC-EX
- N9K-C93180YC-FX
- N9K-X9736C-FX line card

Note Slow drain device detection and congestion avoidance is not supported on FEX ports.

VNP Ports

Connectivity from an FCoE NPV bridge to the FCF is supported only over point-to-point links. These links can be individual Ethernet interfaces or port channel interfaces. For each FCF connected to an Ethernet/port-channel interface, a vFC interface must be created and bound to it. These vFC interfaces must be configured as VNP ports.

On the VNP port, the FCoE NPV bridge emulates an FCoE-capable host with multiple enodes, each with a unique enode MAC address. By default, the VNP port is enabled in trunk mode.

Multiple VSANs can be configured on the VNP port. The FCoE VLANs that correspond to the VNP port VSANs must be configured on the bound Ethernet interface.

Note VNP ports on the Cisco Nexus 9000 Series device emulate an FCoE capable host with multiple Ethernet nodes, each with unique Fabric Provided MAC-Addresses (FPMA).

Licensing Requirements for FCoE NPV

The following table shows the licensing requirements for FCoE NPV:

Product	License Requirement
Cisco NX-OS	FCoE NPV requires the FCoE NPV license (FCOE_NPV_PKG). The PIDs N93-16Y-SSK9 or N93-48Y-SSK9 or ACI-STRG can also be used to enable FCoE NPV along with FC NPV on the supported platforms. For a complete explanation of the Cisco NX-OS licensing scheme and how to obtain and apply the licenses, see the Cisco NX-OS Licensing Guide. **Note** ACI-STRG will license only 48 ports of native Fiber Channel ports. Syslog will not be generated if you use this license on more than 48 ports on Cisco Nexus N9K-C93360YC-FX2 and N9K-C9336C-FX2-E platform switches.

Information About Virtual Interfaces

Cisco Nexus devices support Fibre Channel over Ethernet (FCoE), which allows Fibre Channel and Ethernet traffic to be carried on the same physical Ethernet connection between the switch and the servers.

The Fibre Channel portion of FCoE is configured as a virtual Fibre Channel interface. Logical Fibre Channel features (such as interface mode) can be configured on virtual Fibre Channel interfaces.

A virtual Fibre Channel interface must be bound to an interface before it can be used. The binding is to a physical Ethernet interface (when the converged network adapter (CNA) is directly connected to the Cisco Nexus device), a MAC address (when the CNA is remotely connected over a Layer 2 bridge), or an EtherChannel when the CNA connects to the Fibre Channel Forwarder (FCF) over a virtual port channel (vPC).

Information About Shutting Down LAN Traffic

Converged Network Adapters (CNA) enable both FCoE and LAN traffic (Unified I/O) to co-exist over a physical link.

In vPC configurations with CNAs, network parameters need to be consistent across peer switches. If the system detects an inconsistency, the secondary vPC leg goes down. Since vPC legs carry both FCOE and LAN traffic, the FCoE link goes down also.

To avoid having the FCoE link go down in this situation, you can use the **shutdown lan** command to shutdown only the LAN traffic on port-channels and individual Ethernet ports.

Note When vPC triggers the vPC secondary leg to be brought down, only the Ethernet VLANs are brought down for the secondary vPC leg. FCoE/storage VLANs of the secondary vPC leg remain up.

Notes About the shutdown lan Command

- The **shutdown lan** command is only configurable on port-channel interfaces, FEX HIF ports, or on individual Ethernet interfaces that vFC interfaces are bound upon.
- The **shutdown lan** command is only configurable on port-channel interfaces or on individual Ethernet interfaces that are in an operational trunking state.
- The **shutdown lan** command cannot be enabled on the secondary vPC leg, if the vPC enabled shutdown lan is applied on the secondary vPC leg.
- A vPC enabled shutdown LAN is not operable if the **shutdown lan** command is applied on the secondary vPC leg.
- The **shutdown lan** command is not configurable on port-channel members.
- The **shutdown lan** command default is **no shutdown lan** (**shutdown lan** is disabled).
- The **shutdown lan** command has a prerequisite that the Link Layer Discovery Protocol (LLDP) feature be enabled.
- A port with a shutdown LAN configuration enabled cannot be added to a port channel.
- The shutdown LAN enable/disable configuration is on a per interface basis.
- If a shutdown lan is configured on an interface, a **no shut** command on the interface does not bring up LAN VLANs.
- A shutdown LAN is triggered when a Type-1 inconsistency occurs in a VPC network.

Examples of Shutdown LAN Traffic

- Shutdown the LAN traffic on port-channel.

```
switch(config)#interface port-channel 955
switch(config-if)# shutdown lan
```

- Shutdown the LAN traffic on individual Ethernet port.

```
switch(config)#interface Ethernet 2/5
switch(config-if)# shutdown lan
```

Examples of Verifying Shutdown LAN Traffic

- Verifying when the **shutdown lan** command is issued on port-channel 955 with Ethernet interface 2/5 as member.

```
switch# sh interface port-channel 955 | grep LAN
All LAN VLANs are administratively shut

switch# sh interface ethernet 2/5 | grep LAN
All LAN VLANs are administratively shut

switch# sh run interface port-channel 955 | grep shut
shutdown lan

switch# sh run interface e2/5 | grep shut
```

```
shutdown lan
```

- Verifying when the vPC triggers shutdown LAN on the secondary vPC leg (port-channel 231 with Ethernet 2/31 as member).

```
switch# sh interface port-channel 231 | grep LAN
All LAN VLANs are administratively shut
```

Guidelines and Limitations for FCoE VLANs and Virtual Interfaces

FCoE VLANs and Virtual Fiber Channel (vFC) interfaces have these guidelines and limitations:

- Each vFC interface must be bound to an FCoE-enabled Ethernet or EtherChannel interface or to the MAC address of a remotely connected adapter. FCoE is supported on 10-Gigabit,25-Gigabit,40-Gigabit and 100-Gigabit Ethernet interfaces. 10-Gigabit and 25-Gigabit breakout is supported on FCoE interfaces.

  The Ethernet or EtherChannel interface that you bind to the vFC interface must be configured as follows:

  - The Ethernet or EtherChannel interface must be a trunk port (use the **switchport mode trunk** command).
  - The FCoE VLAN that corresponds to a vFC's VSAN must be in the allowed VLAN list.
  - You must not configure an FCoE VLAN as the native VLAN of the trunk port.

Note The native VLAN is the default VLAN on a trunk. Any untagged frames transit the trunk as native VLAN traffic.

  - You should use an FCoE VLAN only for FCoE.
  - Do not use the default VLAN, VLAN1, as an FCoE VLAN.
  - You must configure the Ethernet interface as PortFast (use the **spanning-tree port type edge trunk** command).
  - You must configure MTU as 9216 or maximum allowed MTU size.

- The vFC interface cannot be bound to Ethernet port channel with multiple member ports connected to FCoE Initialization Protocol (FIP) snooping bridges. It is recommended to use MAC bound vFC when hosts are connected via snooping bridges.
- For VF mode, each vFC interface is associated with only one VSAN.

  For VNP mode, each vFC interface is associated with multiple VSANs.
- You must map any VSAN with associated vFC interfaces to a dedicated FCoE-enabled VLAN.
- FCoE is not supported on private VLANs.
- If the converged access switches (in the same SAN fabric or in another) need to be connected to each other over Ethernet links for a LAN alternate path, then you must explicitly configure such links to exclude all FCoE VLANs from membership.

- You must use different FCoE VLANs for FCoE in SAN-A and SAN-B fabrics.
- FCoE connectivity to pre-FIP CNAs over virtual port channels (vPCs) is not supported.
- FCoE VLANs do not support Multiple Spanning Trees (MST). Creating an MST instance for an FCoE VLAN might cause SAN traffic disruption.



Note Virtual interfaces are created with the administrative state set to down. You must explicitly configure the administrative state to bring the virtual interface into operation.

Guidelines and Limitations for Configuring FCoE NPV

Configuring FCoE NPV has the following configuration guidelines and limitations:

- The FCoE NPV on N9K-X9732C-EX and N9K-X9736C-FX line cards is supported only with fabric modules N9K-C9508-FM-E or N9K-C9504-FM-E.
- Enabling FCoE NPV requires:
  - Enabling the LLDP feature using **feature lldp**. LLDP is not enabled by default.
  - Downloading and installing any of the FCOE_NPV licenses.
  - Installing the FCoE-NPV feature set using the **install feature-set fcoe-npv** command.
  - Enabling the FCoE-NPV feature set using the **feature-set fcoe-npv** command. You may have to reload the switch if an existing FCoE feature is enabled.
- Fibre Channel N-port Virtualization (NPV) can co-exist with VXLAN on different fabric uplinks but on same or different front panel ports on the Cisco Nexus 93180YC-FX, N9K-C9336C-FX2-E, and N9k-C93360YC-FX2 switches. If FCOE NPV is installed as a RPM, see the modularity section in the Cisco Nexus 9000 Series NX-OS Software Upgrade and Downgrade Guide for more information.
- Beginning with Cisco NX-OS Release 10.2(2)F, FCoE NPV supports Cisco N9K-C9336C-FX2-E platform switches.
- The first operational port of the port-channel (non-lacp) must be shut down before being removed. Otherwise, the vfc-po binding of that port-channel may go down.
- It is mandatory to do a TCAM Reservation (as mentioned here: Configuring QoS for no-drop Support, on page 13) for FCoE NPV to work.
- The **show** commands with the **internal** keyword are not supported.
- FCoE NPV does not convert server FLOGI to FDISC.
- FCoE NPV supports a VFC port bound to an Ethernet interface, port-channel, or break-out interface.
- FCoE NPV does not support nested NPV.
- FCoE NPV supports FLOGI/FDISC (nested NPIV).
- FCOE is not supported with Copper SFPs.

- To support multiple FLOGI from a single port, you must enable the NPIV feature to accommodate hosts or servers that send FDISC followed by FLOGI.

  Examples of commands to enable/disable and display status of the NPIV feature:

  -
  ```
  switch(config)# feature npiv
  switch# show feature | include npiv
  npiv                  1          enabled
  switch#
  ```

  -
  ```
  switch# show npv status

  npiv is enabled

  disruptive load balancing is disabled

  External Interfaces:
  ====================
    Interface: vfc-po100, State: Trunking
          VSAN:    1, State: Waiting For VSAN Up
          VSAN:    2, State: Up
          VSAN:    3, State: Up, FCID: 0x040000
    Interface: vfc1/49, State: Down

    Number of External Interfaces: 2
  ```

  -
  ```
  switch(config)# no feature npiv
  switch# show feature | include npiv
  npiv                  1          disabled
  switch#
  ```

- MST is not supported in T2 platforms.
- When a switch is configured to run non-CoS3 FC/FCoE traffic using User-defined QoS policies, all FC/FCoE interfaces must be configured using the same User-defined QoS input policy.
- FC/FCoE configuration does not support rollback. If FC/FCoE configurations are present, use the best-effort option. All other configurations will be successful, however, error message will be displayed for the FC/FCoE configuration.

Configuring FC/FCoE

Perform TCAM Carving

This section explains how to perform TCAM carving.

```
switch(config)# feature-set fcoe-npv
```

Configure the following (if not configured already) for fcoe-npv to be fully functional:

- hardware access-list tcam region ing-ifacl 256
- 256 is the minimum tcam space required in ing-ifacl regions

If the required tcam space is not available then ing-racl region can be reduced using the following command:

- hardware access-list tcam region ing-racl 1536



Note 'show hardware access-list tcam region' - Use this command to verify the current tcam configuration.

SUMMARY STEPS

1. Perform TCAM carving.
2. Use the command **show hardware access-list tcam region** to view the configured TCAM region size.
3. Save the configuration and use the command **reload** to reload the switch.

DETAILED STEPS

Procedure

Step 1 Perform TCAM carving.

Example:

```
Switch(config)# hardware access-list tcam region ing-racl 1536
Switch(config)# hardware access-list tcam region ing-ifacl 256
Switch(config)# hardware access-list tcam region ing-redirect 256
```

Step 2 Use the command **show hardware access-list tcam region** to view the configured TCAM region size.

Example:

```
Switch(config)# show hardware access-list tcam region
Switch(config)#
```

Step 3 Save the configuration and use the command **reload** to reload the switch.

Example:

```
Switch(config)# reload
Switch(config)#
```

What to do next

You must reload the switch after carving TCAM

Configuring LLDP

This section explains how to configure LLDP.

SUMMARY STEPS

1. **configure terminal**
2. [**no**]**feature lldp**

DETAILED STEPS

Procedure

Step 1 **configure terminal**

Enters global configuration mode.

Step 2 [**no**]**feature lldp**

Enables or disables LLDP on the device. LLDP is disabled by default.

Configuring QoS

Configuring Default QoS

There are four types of FCoE default policies: network QoS, output queuing, input queuing, and QoS. You can enable the FCoE default policies by enabling the FCoE NPV feature using the **feature-set fcoe-npv** command. The default QoS ingress policy, **default-fcoe-in-policy**, is implicitly attached to all FC and SAN-port-channel interfaces to enable FC to FCoE traffic; this can be verified by using **show interface** {*fc slot/port* | *san-port-channel <no>*} **all** command. The default QoS policy uses CoS3 and Q1 for all FC and FCoE traffic.

Configuring User Defined QoS

To use a different queue or CoS value for FCoE traffic, create user-defined policies. The user-defined QoS ingress policy has to be created and attached explicitly to both FC and FCoE interfaces to enable traffic to use a different queue or CoS. User-defined QoS policies must be created and activated for system-wide QoS.

The following example demonstrates how to configure and activate user-defined QoS policies that use CoS3 and Q2 for all FC and FCoE traffic.

- Creating a user-defined network QOS policy:

```
switch(config)# policy-map type network-qos fcoe_nq
switch(config-pmap-nqos)# class type network-qos c-nq1
switch(config-pmap-nqos-c)# mtu 1500
switch(config-pmap-nqos-c)# class type network-qos c-nq2
switch(config-pmap-nqos-c)# mtu 9216
switch(config-pmap-nqos-c)# pause pfc-cos 3
switch(config-pmap-nqos-c)# class type network-qos c-nq3
switch(config-pmap-nqos-c)# mtu 1500
switch(config-pmap-nqos-c)# class type network-qos c-nq-default
```

```
switch(config-pmap-nqos-c)# mtu 1500
switch(config-pmap-nqos-c)# exit
switch(config-pmap-nqos)# exit
switch(config)#
```

- Creating a user-defined input queuing policy:

```
switch(config)# policy-map type queuing fcoe-in-policy
switch(config-pmap-que)# class type queuing c-in-q2
switch(config-pmap-c-que)# bandwidth percent 50
switch(config-pmap-c-que)# class type queuing c-in-q-default
switch(config-pmap-c-que)# bandwidth percent 50
switch(config-pmap-c-que)# exit
switch(config-pmap-que)# exit
switch(config)
```

- Creating a user-defined output queuing policy:

```
switch(config)# policy-map type queuing fcoe-out-policy
switch(config-pmap-que)# class type queuing c-out-q3
switch(config-pmap-c-que)# priority level 1
switch(config-pmap-c-que)# class type queuing c-out-q-default
switch(config-pmap-c-que)# bandwidth remaining percent 50
switch(config-pmap-c-que)# class type queuing c-out-q1
switch(config-pmap-c-que)# bandwidth remaining percent 0
switch(config-pmap-c-que)# class type queuing c-out-q2
switch(config-pmap-c-que)# bandwidth remaining percent 50
switch(config-pmap-c-que)# exit
switch(config-pmap-que)# exit
switch(config)#
```

- Creating a user-defined QoS input policy:

```
switch(config)# class-map type qos match-any fcoe
switch(config-cmap-qos)# match protocol fcoe
switch(config-cmap-qos)# match cos 3
switch(config-cmap-qos)# exit
switch(config)#
switch(config)# policy-map type qos fcoe_qos_policy
switch(config-pmap-qos)# class fcoe
switch(config-pmap-c-qos)# set cos 3
switch(config-pmap-c-qos)# set qos-group 2
switch(config-pmap-c-qos)# exit
switch(config-pmap-qos)# exit
switch(config)#
```

- Activating a user-defined system QoS policy:

```
switch(config)# system qos
switch(config-sys-qos)# service-policy type queuing input fcoe-in-policy
switch(config-sys-qos)# service-policy type queuing output fcoe-out-policy
switch(config-sys-qos)# service-policy type network-qos fcoe_nq
switch(config-sys-qos)# exit
switch(config)#
```

- Applying a QoS input policy to an FC or FCoE interface:

```
switch# conf
switch(config)# interface fc <slot>/<port> | ethernet <slot>/<port> | san-port-channel
 <no> | port-channel <no>
switch(config-if)# service-policy type qos input fcoe_qos_policy
```

- Removing a QoS input policy from an FC or FCoE interface:

```
switch# conf
switch(config)# interface fc <slot>/<port> | ethernet <slot>/<port> | san-port-channel
 <no> | port-channel <no>
switch(config-if)# no service-policy type qos input fcoe_qos_policy
```

- Verifying a QoS input policy applied to an FC or FCoE interface:

```
switch# show running-config interface fc <slot>/<port> | interface <slot>/<port> |
san-port-channel <no> | port-channel <no> all
```

Note

- When a user-defined QoS policy is used, the same QoS input policy must be applied to all FC and FCoE interfaces in the switch.
- Do not configure **match protocol fcoe** under more than one QoS class map, as FCoE traffic is supported only on a single CoS.

Configuring Traffic Shaping

Traffic shaping is used to control access to available bandwidth and to regulate the flow of traffic in order to avoid congestion that can occur when the sent traffic exceeds the access speed. Because traffic shaping limits the rate of transmission of data, you may use this command only when necessary.

The following example demonstrates how to configure traffic shaper:

- The following command displays the default system level settings for all FC interfaces:

```
switch(config)# show running-config all | i i rate
hardware qos fc rate-shaper
switch(config)#
```

- The following example shows how to configure rate shaper. This command is applied on all FC interfaces:

Note Rarely, you may see input discards on any of the 4G, 8G, 16G, or 32G interfaces. Use the command *hardware qos fc rate-shaper [low]*, to configure the rate shape. Because this is a system level configuration, it will apply to all the FC ports and will reduce the rates for all FC ports. The default option of the command *hardware qos fc rate-shaper* is applicable to all FC interfaces.

```
switch(config)# hardware qos fc rate-shaper low
switch(config)#
switch(config)#end
```

Configuring QoS for no-drop Support

A qos ingress policy is used to mark ingress FCoE frames. The qos ingress policy must be applied to the interfaces that handle FCoE traffic (such as, all ethernet/port-channel interfaces bound to vFCs).



Note Check to ensure that the port qos region has hardware TCAM space reserved.

This step is mandatory for FCoE NPV to work.

- Reserve TCAM space for the QoS region.

  You may need to acquire TCAM space reserved for other regions (such as the l3qos region).
- Save the configuration.
- Reload the line cards or switch.
- Confirm the port qos region TCAM space.
- Example for TCAM carving on N9K-C93180YC-EX, N9K-C93180YC-FX, N9K-C93360YC-FX2, or N9K-C9336C-FX2-E:

  ```
  hardware access-list tcam region ing-racl 1536
  hardware access-list tcam region ing-ifacl 256
  ```

Example:

```
switch# show hardware access-list tcam region | i "IPV4 Port QoS \[qos\] size"
IPV4 Port QoS [qos] size = 0 /*** Value is 0; No reserved TCAM space.***/

switch# config
switch(config)# hardware access-list tcam region qos 256

Warning: Please reload all linecards for the configuration to take effect

switch# copy running-config startup-config

switch# reload

switch# show hardware access-list tcam region | i "IPV4 Port QoS \[qos\] size"
IPV4 Port QoS [qos] size = 256
```

Configuring FCoE QoS policies

- There are four types of FCoE default policies: network-qos, output queuing, input queuing, and qos.
- You can activate the FCoE default policies by enabling the FCoE-NPV feature using the **feature-set fcoe-npv** command and remove the FCoE default policies by executing the **no feature-set fcoe-npv** command.
- Before entering **no feature-set fcoe-npv**, remove all FCoE policies from the interface and system level. The **no feature-set fcoe-npv** command is allowed only when there are no FC ports configured.

Note Cisco recommends using the FCoE default policies. All policies applied must be of the same type, either 4q or 8q mode, and must be explicitly applied or removed at the system and interface level.

- When configuring QoS policies for an active-active FEX topology that is enabled for FCoE, you must configure the QoS policies on the FEX HIF port on both VPC peers to avoid unpredictable results.
- To use a different queue or cos value for FCoE traffic, create user-defined policies.

Configuring QoS Policies for FCoE

- You can configure a QoS policy by following one of these methods:
  - Predefined policies—You can apply a predefined QoS policy: **default-fcoe-in-policy**.

Note No policy will be applied by default for FCoE.

  - User-defined policy—You can create a QoS policy that conforms to one of the system-defined policies.

Configuring System-wide QoS Policy

Note The network-qos policy and output/input queuing policies should be applied at the system level and the qos policy should be applied at the interface level, for every interface that carries the FCoE traffic.

```
switch(config)# system qos
switch(config-sys-qos)# service-policy type queuing input default-fcoe-in-que-policy
switch(config-sys-qos)# service-policy type queuing output { default-fcoe-8q-out-policy |
default-fcoe-out-policy }
switch(config-sys-qos)# service-policy type network-qos { default-fcoe-8q-nq-policy |
default-fcoe-nq-policy }
```

Configuration Example for user-defined policies

```
switch(config)# policy-map type network-qos fcoe_nq
switch(config-pmap-nqos)# class type network-qos c-nq1
switch(config-pmap-nqos-c)# pause pfc-cos 3
switch(config-pmap-nqos-c)# mtu 9216
switch(config-pmap-nqos-c)# class type network-qos c-nq2
switch(config-pmap-nqos-c)# mtu 1500
switch(config-pmap-nqos-c)# class type network-qos c-nq3
switch(config-pmap-nqos-c)# mtu 1500
switch(config-pmap-nqos-c)# class type network-qos c-nq-default
switch(config-pmap-nqos-c)# mtu 1500
switch(config-pmap-nqos-c)# exit
switch(config-pmap-nqos)# exit
switch(config)#
```

```
switch(config)# policy-map type queuing fcoe-in-policy
switch(config-pmap-que)# class type queuing c-in-q1
switch(config-pmap-c-que)# bandwidth percent 50
switch(config-pmap-c-que)# class type queuing c-in-q-default
switch(config-pmap-c-que)# bandwidth percent 50
switch(config-pmap-c-que)# exit
switch(config)
switch(config)# policy-map type queuing fcoe-out-policy
switch(config-pmap-que)# class type queuing c-out-q3
switch(config-pmap-c-que)# priority level 1
switch(config-pmap-c-que)# class type queuing c-out-q-default
switch(config-pmap-c-que)# bandwidth remaining percent 50
switch(config-pmap-c-que)# class type queuing c-out-q1
switch(config-pmap-c-que)# bandwidth remaining percent 50
switch(config-pmap-c-que)# class type queuing c-out-q2
switch(config-pmap-c-que)# bandwidth remaining percent 0
switch(config-pmap-c-que)# exit
switch(config)#
switch(config)# class-map type qos match-any fcoe
switch(config-cmap-qos)# match protocol fcoe
switch(config-cmap-qos)# match cos 3
switch(config-cmap-qos)# exit
switch(config)#
switch(config)# policy-map type qos fcoe_qos_policy
switch(config-pmap-qos)# class fcoe
switch(config-pmap-c-qos)# set cos 3
switch(config-pmap-c-qos)# set qos-group 1
switch(config-pmap-c-qos)# exit
switch(config-pmap-qos)# exit
switch(config)#
switch(config)# system qos
switch(config-sys-qos)# service-policy type queuing input fcoe-in-policy
switch(config-sys-qos)# service-policy type queuing output fcoe-out-policy
switch(config-sys-qos)# service-policy type network-qos fcoe_nq
```

Note The **set cos 3** command under the QOS policy is mandatory only when there are native fiber channel ports and the command is applicable only for N9K-C93180YC-FX, N9K-C9336C-FX2-E , and N9k-C93360YC-FX2 platforms. For all the other Cisco Nexus 9000 Platform switches, this step is optional.

Note When FEX is connected:

- Apply the QoS policy to the system level and to the HIF port to honor the pause frames in the FCoE traffic.
- 8q policies are not supported when FEX is online.

```
switch(config)# system qos
switch(config-sys-qos)# service-policy type queuing input policy-name
switch(config-sys-qos)# service-policy type queuing output policy-name
switch(config-sys-qos)# service-policy type network-qos policy-name
switch(config-sys-qos)# service-policy type qos input policy-name
```

Applying the ingress QoS policy to each Ethernet/port-channel interface that is bound to vFC interface for FCoE.

```
switch(config)# interface ethernet 2/1
switch(config-if)# switchport mode trunk
switch(config-if)# mtu 9216 /* Or maximum allowed value */
switch(config-if)# service-policy type qos input { default-fcoe-in-policy | fcoe_qos_policy
 }
switch(config-if)# exit
switch(config)#
```



Note The QoS policy needs to be attached to an HIF interface or the port-channel of an HIF interface:

- HIF interface

```
interface "HIF port"
service-policy type qos input policy-name
```

- Port-channel of an HIF interface

```
interface port-channel
service-policy type qos input policy-name
```

Configuring FCoE NPV

Configuring VLAN-VSAN Mapping

VSANs and VLANs are required and the VSANs need to be mapped to the VLANs.

One VLAN can be mapped to only one VSAN and vice versa. The VSANs can then be added to F and NP vFC interfaces (described in a subsequent section).

- Example of VSAN creation:

```
switch(config)#
switch(config)# vsan database
switch(config-vsan-db)# vsan 10
switch(config-vsan-db)#
```

- Example VLAN configuration and binding to FCoE VSAN:

```
switch(config)# vlan 10
switch(config-vlan)# fcoe vsan 10
switch(config-vlan)# exit
switch(config)#
```

Binding vFC to MAC Address

A MAC address bound vFC can also be created on the device interface.

Note A MAC bound vFC can be configured to a host sitting behind a FIP Snooping Bridge (FSB).

When both MAC bound vFC and port-bound vFC are configured for the same interface, the port-bound vFC takes precedence.

As a best practice, you should have either a MAC bound vFC or a port- bound vFC for a physical Ethernet port or a port-channel. However, you cannot have both.

SUMMARY STEPS

1. **configure terminal**
2. **interface vfc** <*number*>
3. **bind mac-address** <*mac-address*>

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface vfc** <*number*>	Creates a virtual fibre Channel Interface.
Step 3	**bind mac-address** <*mac-address*>	Binds the MAC address.

Example

The following example shows how to bind a virtual Fibre Channel interface to a MAC address:

```
switch# configure terminal
switch(config)# interface vfc 2
switch(config-if)# bind mac-address 00:0a:00:00:00:36
```

Explicit vFC Configuration

An explicit vFC interface is a vFC interface where the bound ethernet/port-channel interface is explicitly configured. (The interface ID range is 1-8192.)

Note The port VSAN of the vFC and the native VLAN of the ethernet port should not be mapped to each other in a VLAN-VSAN mapping; this will break the FCoE path completely.

- Example of explicit vFC bound to interface Ethernet:

```
switch# configure terminal
switch(config)# interface vfc 21
switch(config-if)# bind interface ethernet 2/1
```

- Example of explicit vFC bound to interface port-channel:

```
switch# configure terminal
switch(config)# interface vfc 100
switch(config-if)# bind interface port-channel 100
```

- Example of explicit vFC bound to break-out port:

```
switch# configure terminal
switch(config)# interface vfc 111
switch(config-if)# bind interface ethernet 1/1/1
```

- Example of NP interface configuration using explicit vFC:

```
switch# configure terminal
switch(config)# interface vfc21
switch(config-if)# switchport mode NP
switch(config-if)# switchport trunk allowed vsan 10 /* optional; for restricting VSANs
 */
```

- Example of NP interface configuration using explicit bound port-channel interface:

```
switch# configure terminal
switch(config)# interface vfc152
switch(config-if)# bind interface port-channel152
switch(config-if)# switchport mode NP
switch(config-if)# switchport trunk allowed vsan 2
switch(config-if)# switchport trunk mode on
switch(config-if)# no shutdown
```

- Example of F interface configuration using explicit vFC:

```
switch# configure terminal
switch(config)# interface vfc15
switch(config-if)# bind interface ethernet 1/5
switch(config-if)# switchport mode F /* Default mode is F */
switch(config-if)# switchport trunk allowed vsan 10
switch (config-if)# exit
switch (config)# vsan database
switch(config-vsan-db)# vsan 10 interface vfc15
switch(config-vsan-db)# exit
```

Implicit vFC Configuration

An implicit vFC interface is a vFC interface that has an ID with the format *slot*/*port* or *unit*/*slot*/*port* or **port-channel** *id*. When this vFC is created, the Ethernet interface *slot*/*port* or *unit*/*slot*/*port* or **port-channel** *id* is automatically (implicitly) bound to the interface. The running configuration displays the bound Ethernet/port-channel interface. If the Ethernet /port-channel interface does not exist or it is bound to another explicit vFC interface, the vFC creation fails with an error.



Note

- The port VSAN of the vFC and the native VLAN of the Ethernet port should not be mapped to each other in a VLAN-VSAN mapping. It breaks the FCoE path completely.
- When a vFC is created through the Cisco DCNM (Data Center Network Manager), the vFC interface goes to VSAN 4094 (isolated), whereas when a vFC is created through the CLI, the vFC interface goes to VSAN 1. The ethernet interface should be up before configuring implicit vFC through the Cisco DCNM because once the vFC goes to VSAN 4094, it cannot be brought up.

- Example of implicit vFC bound to interface Ethernet:

```
switch# configure terminal
switch(config)# interface vfc 2/1
```

- Example of implicit vFC bound to interface port-channel:

```
switch# configure terminal
switch(config)# interface vfc-port-channel 100
```

- Example of implicit vFC bound to break-out port:

```
switch# configure terminal
switch(config)# interface vfc 1/1/1
```

- Example of NP interface configuration using implicit vFC:

```
switch# configure terminal
switch(config)# interface vfc1/1/1
switch(config-if)# switchport mode NP
switch(config-if)# switchport trunk allowed vsan 10 /* optional; for restricting VSANs
 */
```

- Example of F interface configuration using implicit vFC:

```
switch# configure terminal
switch(config)# interface vfc1/1/1
switch(config-if)# switchport mode F /* Default mode is F */
switch(config-if)# switchport trunk allowed vsan 10
switch (config-if)# exit
switch (config)# vsan database
switch(config-vsan-db)# vsan 10 interface vfc1/1/1
switch(config-vsan-db)# exit
```

Configuring the FCoE NPV Core Switch

Perform the following steps to configure an FCoE NPV core switch.

SUMMARY STEPS

1. **configure terminal**
2. (Optional) **switchto vdc** *vdc-name*
3. **feature npiv**
4. (Optional) **feature fport-channel-trunk**
5. **interface ethernet** *slot/port*
6. **switchport**
7. **no switchport**
8. **switchport mode trunk**
9. **mtu 9216**
10. **service-policy type** {**network-qos** | **qos** | **queuing**} [**input** | **output**] *fcoe default policy-name*
11. **exit**
12. **interface vfc** *vfc-id*
13. **switchport mode f**
14. **bind interface ethernet** *slot/port*
15. **exit**
16. **vsan database**
17. **vsan** *vsan-id*
18. **vsan** *vsan-id* **interface vfc** *vfc-id*
19. **exit**
20. **vlan** *vlan-id*
21. **fcoe vsan** *vsan-id*
22. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters configuration mode.
Step 2	(Optional) **switchto vdc** *vdc-name*	Switch to storage VDC. **Note** This step is required only when a Cisco Nexus 7000 Series switch is used as the core switch.
Step 3	**feature npiv**	Enable NPIV.
Step 4	(Optional) **feature fport-channel-trunk**	Enables F port channel trunking.
Step 5	**interface ethernet** *slot/port*	Enters interface configuration mode.

	Command or Action	Purpose
Step 6	**switchport**	Configures the interface as a Layer 2 interface and deletes any configuration specific to Layer 3 on this interface.
Step 7	**no switchport**	Configures the interface as a Layer 3 interface and deletes any configuration specific to Layer 2 on this interface.
Step 8	**switchport mode trunk**	Set physical interface mode to trunk.
Step 9	**mtu 9216**	Configure MTU as 9216. You must configure MTU as 9216 or maximum allowed MTU size **Note** This step is required only when a Cisco Nexus N9K-C93180YC-FX, N9K-C9336C-FX2-E or N9K-C93360YC-FX2 switch is used as the core switch.
Step 10	**service-policy type** {**network-qos** | **qos** | **queuing**} [**input** | **output**] *fcoe default policy-name*	Specifies the QoS policy on the port to a no drop policy. **Note** This step is required only when a Cisco Nexus N9K-C93180YC-FX, N9K-C9336C-FX2-E or N9K-C93360YC-FX2 switch is used as the core switch.
Step 11	**exit**	Exits the interface mode.
Step 12	**interface vfc** *vfc-id*	Enters interface configuration mode.
Step 13	**switchport mode f**	Set vFC port mode to VF.
Step 14	**bind interface ethernet** *slot/port*	Binds a ethernet interface to a vFC. **Important** The **bind interface ethernet** command is not required for an implicit vFC configuration.
Step 15	**exit**	Exits the interface configuration mode.
Step 16	**vsan database**	Enters VSAN configuration mode.
Step 17	**vsan** *vsan-id*	Create VSAN
Step 18	**vsan** *vsan-id* **interface vfc** *vfc-id*	Add vFC into VSAN.
Step 19	**exit**	Exits the VSAN configuration mode.
Step 20	**vlan** *vlan-id*	Enters VLAN configuration mode.
Step 21	**fcoe vsan** *vsan-id*	Creates FCoE VLAN and map FCoE VLAN to VSAN.
Step 22	**exit**	Exits the VLAN configuration mode.

Configuring the FCoE NPV Edge Switch

Perform the following steps to configure an FCoE NPV edge switch.

SUMMARY STEPS

1. **install feature-set fcoe-npv**
2. **feature-set fcoe-npv**
3. **[no] feature lldp**
4. **vsan database**
5. **vsan** *vsan-id*
6. **exit**
7. **vlan** *vlan-id*
8. **fcoe vsan** *vsan-id*
9. **exit**
10. **interface ethernet** *slot/port*
11. **switchport**
12. **switchport mode trunk**
13. **mtu 9216**
14. **service-policy type** {**network-qos** | **qos** | **queuing**} [**input** | **output**] *fcoe default policy-name*
15. **exit**
16. **interface vfc** *vfc-id*
17. **switchport mode NP**
18. **bind interface ethernet** *slot/port*
19. **exit**
20. **interface ethernet** *slot/port*
21. **switchport**
22. **switchport mode trunk**
23. **mtu 9216**
24. **service-policy type** {**network-qos** | **qos** | **queuing**} [**input** | **output**] *fcoe default policy-name*
25. **exit**
26. **interface vfc** *vfc-id*
27. **switchport mode f**
28. **switchport trunk mode on**
29. **switchport trunk allowed vsan** *vsan-id*
30. **bind interface ethernet** *slot/port*
31. **no shutdown**
32. **exit**
33. **vsan database**
34. **vsan** *vsan-id* **interface vfc** *vfc-id*
35. **vsan** *vsan-id* **interface vfc** *vfc-id*
36. **exit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**install feature-set fcoe-npv**	Installs FCoE NPV.

Step 2	**feature-set fcoe-npv**	Enables FCoE NPV.
		Note
		When enabling FCoE NPV for Cisco NX-OS 7.0(3)I4(1) and later releases, the following BCM settings are required per FCoE VLAN:
		`LEARN_DISABLE=1`
		`L2_NON_UCAST_DROP=1`
		`L2_MISS_DROP=1`
		• Ethernet VLANs do not require these BCM settings.

Step 3	**[no] feature lldp**	Enables or disables LLDP on the device. LLDP is disabled by default.
Step 4	**vsan database**	Enters VSAN configuration mode.
Step 5	**vsan** *vsan-id*	Creates VSAN.
Step 6	**exit**	Exits the VSAN configuration mode.
Step 7	**vlan** *vlan-id*	Enters VLAN configuration mode and creates the VLAN.
Step 8	**fcoe vsan** *vsan-id*	Maps the FCoE VLAN to VSAN.
Step 9	**exit**	Exits the VLAN configuration mode.
Step 10	**interface ethernet** *slot/port*	Enters interface configuration mode.
Step 11	**switchport**	To put an interface that is in Layer 3 mode into Layer 2 mode for Layer 2 configuration, use the **switchport** command in interface configuration mode. To put an interface into Layer 3 mode, use the **no** form of this command.
Step 12	**switchport mode trunk**	Sets switch side physical interface to trunk mode.
Step 13	**mtu 9216**	Configure MTU as 9216. You must configure MTU as 9216 or maximum allowed MTU size
Step 14	**service-policy type** {**network-qos** | **qos** | **queuing**} [**input** | **output**] *fcoe default policy-name*	Specifies the QoS policy on the port to a no drop policy.
Step 15	**exit**	Exits the interface configuration mode.
Step 16	**interface vfc** *vfc-id*	Enters interface configuration mode.

	Command or Action	Purpose
Step 17	**switchport mode NP**	Sets vFC port mode to VNP.
Step 18	**bind interface ethernet** *slot/port*	Binds the ethernet interface to vFC. **Important** The **bind interface ethernet** command is not required for an implicit vFC configuration.
Step 19	**exit**	Exits the interface configuration mode.
Step 20	**interface ethernet** *slot/port*	Enters interface configuration mode.
Step 21	**switchport**	To put an interface that is in Layer 3 mode into Layer 2 mode for Layer 2 configuration, use the **switchport** command in interface configuration mode. To put an interface into Layer 3 mode, use the **no** form of this command.
Step 22	**switchport mode trunk**	Sets server side physical interface to trunk mode.
Step 23	**mtu 9216**	Configure MTU as 9216.
Step 24	**service-policy type** {**network-qos** | **qos** | **queuing**} [**input** | **output**] *fcoe default policy-name*	Specifies the default FCoE policy map to use as the service policy for the system.
Step 25	**exit**	Exits the interface configuration mode.
Step 26	**interface vfc** *vfc-id*	Enters interface configuration mode.
Step 27	**switchport mode f**	Sets the mode to F on a Fibre Channel interface.
Step 28	**switchport trunk mode on**	Sets server side physical interface to trunk mode.
Step 29	**switchport trunk allowed vsan** *vsan-id*	Configure vFC port to allow VSAN vsan-id.
Step 30	**bind interface ethernet** *slot/port*	Binds the ethernet interface to vFC. **Important** The **bind interface ethernet** command is not required for an implicit vFC configuration.
Step 31	**no shutdown**	Keeps the Fibre Channel interface active
Step 32	**exit**	Exits the interface configuration mode.
Step 33	**vsan database**	Enters VSAN configuration mode.
Step 34	**vsan** *vsan-id* **interface vfc** *vfc-id*	Adds port VSAN vsan-id to VF port.
Step 35	**vsan** *vsan-id* **interface vfc** *vfc-id*	Add VNP port to VSAN vsan-id. **Note** This step is optional. The default port VSAN is 1 and is preferable for the VNP port.

	Command or Action	Purpose
Step 36	**exit**	Exits the VSAN configuration mode.

Configuring a Pause Frame Timeout Value

You can enable or disable a pause frame timeout value on a port. The system periodically checks the ports for a pause condition and enables a pause frame timeout on a port if it is in a continuous pause condition for a period of time. This situation results in all frames that come to that port getting dropped in the egress. This function empties the buffer space in the ISL link and helps to reduce the fabric slowdown and the congestion on the other unrelated flows using the same link.

Note Configuring a pause frame timeout value is supported on the following switches and line cards:

- N9K-C93360YC-FX2
- N9K-C93180YC-EX
- N9K-C93180YC-FX
- N9K-C93180LC-EX
- N9K-X9732C-EX Line Card
- N9K-X9736C-FX Line Card
- N9K-C9336C-FX2-E

When a pause condition is cleared on a port or when a port flaps, the system disables the pause frame timeout on that particular port.

The pause frame timeout is disabled by default. We recommend that you retain the default configuration for the ISLs and configure a value that does not exceed the default value for the edge ports.

For a faster recovery from the slow drain device behavior, you should configure a pause frame timeout value because it drops all the frames in the edge port that face the slow drain whether the frame is in the switch for a congested timeout or not. This process instantly clears the congestion in the ISL.

Use the **no system default interface pause mode edge** command to disable the pause frame timeout value on the edge ports. The default pause timeout value is 500 milliseconds.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch# **system default interface pause timeout** *milliseconds* **mode edge**
3. switch# **system default interface pause mode edge**
4. switch# **no system default interface pause timeout** *milliseconds* **mode edge**
5. switch# **no system default interface pause mode edge**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch# **system default interface pause timeout** *milliseconds* **mode edge**	Configures a new pause frame timeout value in milliseconds and the port mode for the device. **Note** Timeout value is specified in multiples of 100 (range is 100-500). **Note** The **system default interface pause timeout** *milliseconds* **mode core** command is not supported.
Step 3	switch# **system default interface pause mode edge**	Configures the default pause frame timeout value in milliseconds and the port mode for the device. **Note** Only the **system default interface pause** *milliseconds* **mode edge** command is supported. The **system default interface pause** *milliseconds* **mode core** command is not supported.
Step 4	switch# **no system default interface pause timeout** *milliseconds* **mode edge**	Disables the pause frame timeout for the device.
Step 5	switch# **no system default interface pause mode edge**	Disables the default pause frame timeout for the device.

Example

This example shows how to configure a pause frame timeout value:

```
switch# configure terminal
switch(config)# system default interface pause timeout 500 mode edge
switch(config)# system default interface pause mode edge
switch(config)# no system default interface pause timeout 500 mode edge
switch(config)# no system default interface pause mode edge
switch(config)# end
```

This example shows how to display pause frame timeout information:

```
switch#(config-if)# attach module 1
module-1# sh creditmon interface ethernet 1/35

Ethernet1/35: PORT is EDGE, xoff_hits=2
         flush-status        : OFF
         total_xoff_hits     : 2
         (cntr) pause frames : 832502
```

```
(cntr) pause quanta : 1962909 milli-seconds
(cntr) force drops  : 94320764
(cntr-pg) to_drops  : 0
DBG_xoff_hit_cnt    : 0
DBG_xoff_hit_time   : 274
DBG_port_fc_mode    : 2
DBG_force_tmo_val   : 300 milli-seconds
CFG_congestion_tmo  : 0 milli-seconds
```

This example shows how to display pause frame timeout information:

```
switch(config-if)# attach module 1
module-1#
module-1# sh creditmon interface all
Ethernet1/1: PORT is NONE, xoff_hits=0
Ethernet1/2: PORT is NONE, xoff_hits=0
Ethernet1/3: PORT is NONE, xoff_hits=0
Ethernet1/4: PORT is NONE, xoff_hits=0
Ethernet1/5: PORT is NONE, xoff_hits=0
Ethernet1/6: PORT is NONE, xoff_hits=0
Ethernet1/7: PORT is NONE, xoff_hits=0
Ethernet1/8: PORT is NONE, xoff_hits=0
Ethernet1/9: PORT is NONE, xoff_hits=0
Ethernet1/10: PORT is NONE, xoff_hits=0
Ethernet1/11: PORT is NONE, xoff_hits=0
Ethernet1/12: PORT is NONE, xoff_hits=0
Ethernet1/13: PORT is NONE, xoff_hits=0
Ethernet1/14: PORT is NONE, xoff_hits=0
Ethernet1/15: PORT is NONE, xoff_hits=0
Ethernet1/16: PORT is NONE, xoff_hits=0
Ethernet1/17: PORT is NONE, xoff_hits=0
Ethernet1/18: PORT is NONE, xoff_hits=0
Ethernet1/19: PORT is NONE, xoff_hits=0
Ethernet1/20: PORT is NONE, xoff_hits=0
Ethernet1/21: PORT is NONE, xoff_hits=0
Ethernet1/22: PORT is NONE, xoff_hits=0
Ethernet1/23: PORT is NONE, xoff_hits=0
Ethernet1/24: PORT is NONE, xoff_hits=0
Ethernet1/25: PORT is NONE, xoff_hits=0
Ethernet1/26: PORT is NONE, xoff_hits=0
Ethernet1/27: PORT is NONE, xoff_hits=0
Ethernet1/28: PORT is NONE, xoff_hits=0
Ethernet1/29: PORT is NONE, xoff_hits=0
Ethernet1/30: PORT is NONE, xoff_hits=0
Ethernet1/31: PORT is NONE, xoff_hits=0
Ethernet1/32: PORT is NONE, xoff_hits=0
Ethernet1/33: PORT is NONE, xoff_hits=0
Ethernet1/34: PORT is NONE, xoff_hits=0
Ethernet1/35: PORT is NONE, xoff_hits=0
Ethernet1/36: PORT is NONE, xoff_hits=0
Ethernet1/37: PORT is NONE, xoff_hits=0
Ethernet1/38: PORT is NONE, xoff_hits=0
Ethernet1/39: PORT is NONE, xoff_hits=0
Ethernet1/40: PORT is NONE, xoff_hits=0
Ethernet1/41: PORT is NONE, xoff_hits=0
Ethernet1/42: PORT is NONE, xoff_hits=0
Ethernet1/43: PORT is NONE, xoff_hits=0
Ethernet1/44: PORT is NONE, xoff_hits=0
Ethernet1/45: PORT is NONE, xoff_hits=0
Ethernet1/46: PORT is NONE, xoff_hits=0
Ethernet1/47: PORT is NONE, xoff_hits=0
Ethernet1/48: PORT is NONE, xoff_hits=0
Ethernet1/49: PORT is NONE, xoff_hits=0
```

```
Ethernet1/49/2: PORT is NONE, xoff_hits=0
Ethernet1/49/3: PORT is NONE, xoff_hits=0
Ethernet1/49/4: PORT is NONE, xoff_hits=0
Ethernet1/50: PORT is NONE, xoff_hits=0
Ethernet1/50/2: PORT is NONE, xoff_hits=0
Ethernet1/50/3: PORT is NONE, xoff_hits=0
Ethernet1/50/4: PORT is NONE, xoff_hits=0
Ethernet1/51: PORT is NONE, xoff_hits=0
Ethernet1/51/2: PORT is NONE, xoff_hits=0
Ethernet1/51/3: PORT is NONE, xoff_hits=0
Ethernet1/51/4: PORT is NONE, xoff_hits=0
Ethernet1/52: PORT is NONE, xoff_hits=0
Ethernet1/52/2: PORT is NONE, xoff_hits=0
Ethernet1/52/3: PORT is NONE, xoff_hits=0
Ethernet1/52/4: PORT is NONE, xoff_hits=0
Ethernet1/53: PORT is NONE, xoff_hits=0
Ethernet1/53/2: PORT is NONE, xoff_hits=0
Ethernet1/53/3: PORT is NONE, xoff_hits=0
Ethernet1/53/4: PORT is NONE, xoff_hits=0
Ethernet1/54: PORT is NONE, xoff_hits=0
Ethernet1/54/2: PORT is NONE, xoff_hits=0
Ethernet1/54/3: PORT is NONE, xoff_hits=0
Ethernet1/54/4: PORT is NONE, xoff_hits=0

module-1#
```

This example shows syslog messages that are displayed when a pause frame timeout occurs:

```
2021 Jun 25 10:07:41 StArcher-Peer1 %TAHUSD-SLOT1-2-TAHUSD_SYSLOG_CRIT:
   PAUSE-TIMEOUT_BEGIN: Ethernet1/23, PFC pause timeout of 500ms reached for qos_group 1
cos 3 occurrences 1,
   setting port to drop class traffic
2021 Jun 25 10:08:23 StArcher-Peer1 %TAHUSD-SLOT1-2-TAHUSD_SYSLOG_CRIT:
   PAUSE-TIMEOUT_END: Ethernet1/23, PFC pause timeout ended for qos_group 1 cos 3 duration
 40 seconds,
   setting port to transmit class traffic
```

Verifying the FCoE NPV Configuration

To display FCoE/VPC configuration information, perform one of the following:

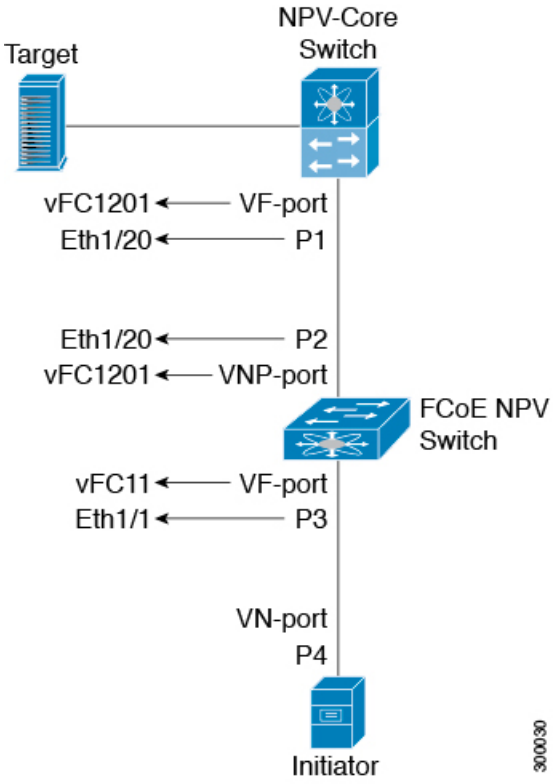
Command	Purpose
show fcoe	Displays status of Fibre Channel over Ethernet (FCoE) parameters on the switch.
show fcoe database	Displays content of the Fibre Channel over Ethernet (FCoE) database.
show int vfc *vfc-id*	Displays vFC interface information.

To display NPV configuration information, perform one of the following:

Command	Purpose
show npv status	Displays N Port Virtualization (NPV) current status
show npv traffic-map	Displays N Port Virtualization (NPV) traffic map.
show npv external-interface-usage server-interface *if*	Displays external vFC interfaces (NP interfaces) designated/allocated to the server vFC interface *if* through automatic or static allocation.
show npv external-interface-usage	Displays external vFC interfaces (NP interfaces) designated/allocated to all available server vFC interfaces through automatic or static allocation.
show npv flogi-table interface *if*	Displays host FLOGI table that lists server interface; VSAN; fcid allocated to the initiator connected to the server interface; PWWN and NWWN of initiator; and external interface/gateway on NPV switch designated to the server interface.
show npv flogi-table vsan *vsan*	Displays information about N Port Virtualization (NPV) FLOGI session specific to the VSAN.
show npv flogi-table	Displays information about N Port Virtualization (NPV) FLOGI session.
show fcoe-npv issu-impact	Displays information about VNP ports where FKA is disabled.

FCoE NPV Core Switch and FCoE NPV Edge Switch Configuration Example

Figure 1: Configuring FCoE NPV Core Switch and FCoE NPV Edge Switch



- Configure NPV core switch:
  - Enable NPIV

```
npv-core(config)# feature npiv
```

  - Set physical interface mode to trunk

```
npv-core(config)# interface Eth 1/20
npv-core(config)# switchport
npv-core(config)# switchport mode trunk
npv-core(config)# mtu 9216
npv-core(config)# service-policy type qos input default-fcoe-in-policy
```

Note The steps *switchport, MTU and service-policy* are required only when a Cisco Nexus C93180YC-FX, N9K-C9336C-FX2-E, or N9K-C93360YC-FX2 switches are used as a core switch.

  - Set vFC port mode of P1 to VF

```
npv-core(config)# interface vfc1201
npv-core(config)# bind interface Eth1/20
npv-core(config)# switchport mode F
```

- Create VSAN and add vFC into VSAN

```
npv-core(config)# vsan database
npv-core(config-vsan-db)# vsan 100
npv-core(config-vsan-db)# vsan 100 interface vfc1201
```

- Create FCoE VLAN & map it to VSAN

```
npv-core(config)# vlan 100
npv-core(config-vlan)# fcoe vsan 100
```

- Configure FCoE NPV switch:
  - Install FCoE NPV

```
npv(config)# install feature-set fcoe-npv
```

  - Enable FCoE NPV

```
npv(config)# feature-set fcoe-npv
```

  - Create VSAN

```
npv(config)# vsan database
npv(config-vsan-db)# vsan 100
```

  - Create FCoE VLAN and map it to VSAN

```
npv(config)# vlan 100
npv(config-vlan)# fcoe vsan 100
```

  - Set switch side physical interface to trunk mode

```
npv(config)# interface Eth 1/20
npv(config-if)# switchport mode trunk
npv(config-if)# mtu 9216
npv(config-if)# service-policy type qos input default-fcoe-in-policy
```

  - Set vFC port mode of P2 to VNP

```
npv(config)# interface vfc1201
npv(config-if)# switchport mode NP
npv(config-if)# bind interface Eth1/20
```

  - Set server side physical interface to trunk mode

  -
```
npv(config)# interface Eth 1/1
npv(config-if)# switchport mode trunk
npv(config-if)# mtu 9216
npv(config-if)# service-policy type qos input default-fcoe-in-policy
```

  - Configure vFC port P3 to allow VSAN 100

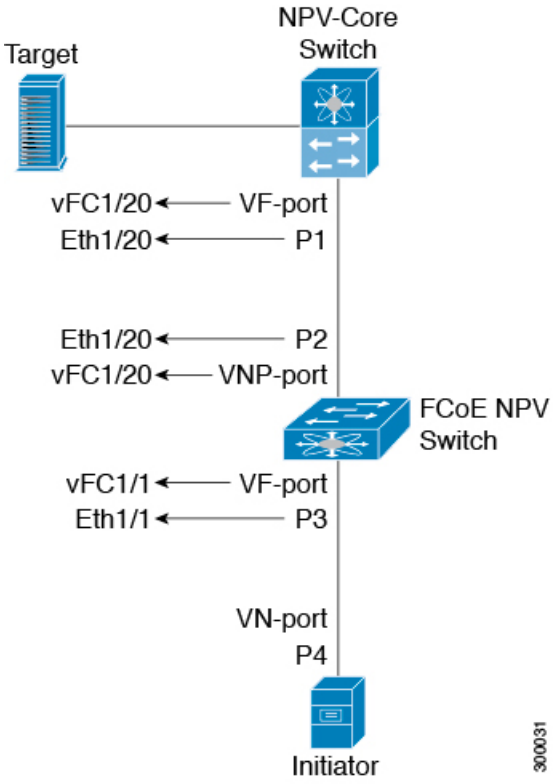
```
npv(config)# interface vfc11
npv(config-if)# switchport trunk allowed vsan 100
npv(config-if)# bind interface Eth1/1
```

  - Add both VNP and VF ports into VSAN 100

```
npv(config)# vsan database
npv(config-vsan-db)# vsan 100 interface vfc1201
npv(config-vsan-db)# vsan 100 interface vfc11
```

FCoE NPV Core Switch and FCoE NPV Edge Switch with Implicit vFC Configuration Example

Figure 2: Configuring FCoE NPV Core Switch and FCoE NPV Edge Switch with Implicit vFC



- Configure NPV core switch:
  - Enable NPIV

```
npv-core(config)# feature npiv
```

  - Set physical interface mode to trunk

```
npv-core(config)# interface Eth 1/20
npv-core(config)# switchport
npv-core(config)# switchport mode trunk
npv-core(config)# mtu 9216
npv-core(config)# service-policy type qos input default-fcoe-in-policy
```

Note The steps *switchport, MTU and service-policy* are required only when a Cisco Nexus C93180YC-FX, N9K-C9336C-FX2-E, or N9K-C93360YC-FX2 switches are used as a core switch.

  - Set vFC port mode of P1 to VF (implicit vFC)

```
npv-core(config)# interface vfc 1/20
npv-core(config)# switchport mode F
```

- Create VSAN and add vFC into VSAN

```
npv-core(config)# vsan database
npv-core(config-vsan-db)# vsan 100
npv-core(config-vsan-db)# vsan 100 interface vfc 1/20
```

- Create FCoE VLAN & map it to VSAN

```
npv-core(config)# vlan 100
npv-core(config-vlan)# fcoe vsan 100
```

- Configure FCoE NPV switch:
  - Install FCoE NPV

```
npv(config)# install feature-set fcoe-npv
```

  - Enable FCoE NPV

```
npv(config)# feature-set fcoe-npv
```

  - Create VSAN

```
npv(config)# vsan database
npv(config-vsan-db)# vsan 100
```

  - Create FCoE VLAN and map it to VSAN

```
npv(config)# vlan 100
npv(config-vlan)# fcoe vsan 100
```

  - Set switch side physical interface to trunk mode

```
npv(config)# interface Eth 1/20
npv(config-if)# switchport mode trunk
npv(config-if)# mtu 9216
npv(config-if)# service-policy type qos input default-fcoe-in-policy
```

  - Set vFC port mode of P2 to VNP (implicit vFC)

```
npv(config)# interface vfc 1/20
npv(config-if)# switchport mode NP
```

  - Set server side physical interface to trunk mode

```
npv(config)# interface Eth 1/1
npv(config-if)# switchport mode trunk
npv(config-if)# mtu 9216
npv(config-if)# service-policy type qos input default-fcoe-in-policy
```

  - Configure vFC port P3 to allow VSAN 100 (implicit vFC)

```
npv(config)# interface vfc 1/1
npv(config-if)# switchport trunk allowed vsan 100
```

  - Add both VNP and VF ports into VSAN 100

```
npv(config)# vsan database
npv(config-vsan-db)# vsan 100 interface vfc 1/20
npv(config-vsan-db)# vsan 100 interface vfc 1/1
```

Verifying the Virtual Interface

To display configuration information about virtual interfaces, perform one of the following tasks:

Command	Purpose
switch# **show interface vfc** *vfc-id*	Displays the detailed configuration of the specified Fibre Channel interface.
switch# **show interface brief**	Displays the status of all interfaces.
switch# **show vlan fcoe**	Displays the mapping of FCoE VLANs to VSANs.

This example shows how to display a virtual Fibre Channel interface bound to an Ethernet interface:

```
switch(config-if)# sh int vfc 172

vfc172 is trunking (Not all VSANs UP on the trunk)
    Bound interface is Ethernet1/72
    Hardware is Ethernet
    Port WWN is 20:ab:e0:0e:da:4a:5d:9d
    Admin port mode is F, trunk mode is on
    snmp link state traps are enabled
    Port mode is TF
    Port vsan is 200
    Speed is auto
    Trunk vsans (admin allowed and active) (1,10,100,200)
    Trunk vsans (up)                       (200)
    Trunk vsans (isolated)                 ()
    Trunk vsans (initializing)             (1,10,100)
    799 fcoe in packets
    80220 fcoe in octets
    2199 fcoe out packets
    2219828 fcoe out octets
    Interface last changed at Thu Sep 15 08:52:51 2016
```

This example shows how to display a virtual Fibre Channel interface bound to a MAC address:

```
switch(config-if)# sh int vfc 132

vfc132 is trunking (Not all VSANs UP on the trunk)
    Bound MAC is 000e.1e1b.c1c9
    Hardware is Ethernet
    Port WWN is 20:83:00:2a:10:7a:89:bf
    Admin port mode is F, trunk mode is on
    snmp link state traps are enabled
    Port mode is TF
    Port vsan is 2101
    Speed is auto
    Trunk vsans (admin allowed and active) (1,2001-2003,2101-2103)
    Trunk vsans (up)                       (2101)
    Trunk vsans (isolated)                 ()
    Trunk vsans (initializing)             (1,2001-2003,2102-2103)
    Interface last changed at Wed Sep 14 12:14:29 2016
```

This example shows how to display the status of all the interfaces on the switch (some output has been removed for brevity):

```
switch# show interface brief
--------------------------------------------------------------------------------
Interface  Vsan   Admin  Admin   Status          SFP    Oper  Oper   Port
                 Mode   Trunk                          Mode  Speed  Channel
                        Mode                                 (Gbps)
--------------------------------------------------------------------------------
fc3/1      1      auto   on      trunking         swl    TE      2    --
fc3/2      1      auto   on      sfpAbsent        --     --           --
...
fc3/8      1      auto   on      sfpAbsent        --     --           --
--------------------------------------------------------------------------------
Interface               Status     IP Address        Speed    MTU   Port
                                                                   Channel
--------------------------------------------------------------------------------
Ethernet1/1             hwFailure  --                --       1500  --
Ethernet1/2             hwFailure  --                --       1500  --
Ethernet1/3             up         --                10000    1500  --
...
Ethernet1/39            sfpIsAbsen --                --       1500  --
Ethernet1/40            sfpIsAbsen --                --       1500  --
--------------------------------------------------------------------------------
Interface               Status     IP Address        Speed    MTU
--------------------------------------------------------------------------------
mgmt0                   up         172.16.24.41      100      1500
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
Interface  Vsan   Admin  Admin   Status          SFP    Oper  Oper   Port
                 Mode   Trunk                          Mode  Speed  Channel
                        Mode                                 (Gbps)
--------------------------------------------------------------------------------
vfc 1        1      F      --      down             --     --           --
...
```

This example shows how to display the mapping between the VLANs and VSANs on the switch:

```
switch# show vlan fcoe
VLAN      VSAN      Status
--------  --------  --------
```

```
15        15        Operational
20        20        Operational
25        25        Operational
30        30        Non-operational
```

Mapping VSANs to VLANs Example Configuration

The following example shows how to configure the FCoE VLAN and a virtual Fibre Channel interface:

SUMMARY STEPS

1. Enable the associated VLAN and map the VLAN to a VSAN.
2. Configure the VLAN on a physical Ethernet interface.
3. Create a virtual Fibre Channel interface and bind it to a physical Ethernet interface.
4. Associate the virtual Fibre Channel interface to the VSAN.
5. (Optional) Display membership information for the VSAN.
6. (Optional) Display the interface information for the virtual Fibre Channel interface.

DETAILED STEPS

Procedure

Step 1 Enable the associated VLAN and map the VLAN to a VSAN.

```
switch(config)# vlan 200
switch(config-vlan)# fcoe vsan 2
switch(config-vlan)# exit
```

Step 2 Configure the VLAN on a physical Ethernet interface.

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree port type edge trunk
switch(config-if)# switchport mode trunk
switch(config-if)# switchport trunk allowed vlan 1,200
switch(config-if)# exit
```

Step 3 Create a virtual Fibre Channel interface and bind it to a physical Ethernet interface.

```
switch(config)# interface vfc 4
switch(config-if)# bind interface ethernet 1/4
switch(config-if)# exit
```

Note

By default, all virtual Fibre Channel interfaces reside on VSAN 1. If the VLAN to VSAN mapping is to a VSAN other than VSAN 1, then proceed to Step 4.

Step 4 Associate the virtual Fibre Channel interface to the VSAN.

```
switch(config)# vsan database
switch(config-vsan)# vsan 2 interface vfc 4
switch(config-vsan)# exit
```

Step 5 (Optional) Display membership information for the VSAN.

```
switch# show vsan 2 membership
vsan 2 interfaces
           vfc 4
```

Step 6 (Optional) Display the interface information for the virtual Fibre Channel interface.

```
switch# show interface vfc 4

vfc4 is up
Bound interface is Ethernet1/4
Hardware is Virtual Fibre Channel
Port WWN is 20:02:00:0d:ec:6d:95:3f
Port WWN is 20:02:00:0d:ec:6d:95:3f
snmp link state traps are enabled
Port WWN is 20:02:00:0d:ec:6d:95:3f
APort WWN is 20:02:00:0d:ec:6d:95:3f
snmp link state traps are enabled
Port mode is F, FCID is 0x490100
Port vsan is 931
1 minute input rate 0 bits/sec, 0 bytes/sec, 0 frames/sec
1 minute output rate 0 bits/sec, 0 bytes/sec, 0 frames/sec
0 frames input, 0 bytes 0 discards, 0 errors
0 frames output, 0 bytes 0 discards, 0 errors
Interface last changed at Thu Mar 11 04:44:42 2010
```

SAN Boot with vPC

Cisco Nexus 9000 Series devices support the SAN boot of initiators on Link Aggregation Control Protocol (LACP) based vPC. This limitation is specific to LACP-based port channels. The host-facing vFC interfaces are bound to port channel members instead of the port channel itself. This binding ensures that the host-side vFC comes up during a SAN boot as soon as the link on the CNA/Host Bus Adapter (HBA) comes up, without relying on the LACP-based port channel to form first.

Note Cisco Nexus 9000 Series devices support the SAN boot of channel mode on also.

Note LACP suspend-individual should be removed from the port-channel, otherwise the physical interface will be suspended when LACP BPDU is not received from the host.