Configuring Layer 2 Interfaces

Licensing Requirements

For a complete explanation of Cisco NX-OS licensing recommendations and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide* and the *Cisco NX-OS Licensing Options Guide*.

Information About Ethernet Interfaces

The Ethernet ports can operate as standard Ethernet interfaces connected to servers or to a LAN.

The Ethernet interfaces are enabled by default.

Interface Command

You can enable the various capabilities of the Ethernet interfaces on a per-interface basis using the **interface** command. When you enter the **interface** command, you specify the following information:

The interface numbering convention is extended to support use with a Cisco Nexus Fabric Extender as follows:

switch(config)# **interface ethernet** [*chassis*/]*slot*/*port*

- The chassis ID is an optional entry that you can use to address the ports of a connected Fabric Extender. The chassis ID is configured on a physical Ethernet or EtherChannel interface on the switch to identify the Fabric Extender discovered through the interface. The chassis ID ranges from 100 to 199.

Unidirectional Link Detection Parameter

The Cisco-proprietary Unidirectional Link Detection (UDLD) protocol allows ports that are connected through fiber optics or copper (for example, Category 5 cabling) Ethernet cables to monitor the physical configuration of the cables and detect when a unidirectional link exists. When the switch detects a unidirectional link, UDLD shuts down the affected LAN port and alerts the user. Unidirectional links can cause a variety of problems, including spanning tree topology loops.

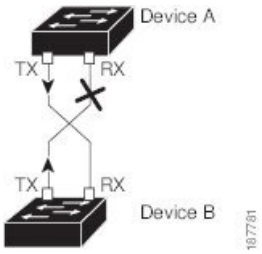
UDLD is a Layer 2 protocol that works with the Layer 1 protocols to determine the physical status of a link. At Layer 1, autonegotiation takes care of physical signaling and fault detection. UDLD performs tasks that autonegotiation cannot perform, such as detecting the identities of neighbors and shutting down misconnected LAN ports. When you enable both autonegotiation and UDLD, Layer 1 and Layer 2 detections work together to prevent physical and logical unidirectional connections and the malfunctioning of other protocols.

A unidirectional link occurs whenever traffic transmitted by the local device over a link is received by the neighbor but traffic transmitted from the neighbor is not received by the local device. If one of the fiber strands in a pair is disconnected, and if autonegotiation is active, the link does not stay up. In this case, the logical link is undetermined, and UDLD does not take any action. If both fibers are working normally at Layer 1, then UDLD at Layer 2 determines whether those fibers are connected correctly and whether traffic is flowing bidirectionally between the correct neighbors. This check cannot be performed by autonegotiation, because autonegotiation operates at Layer 1.

A Cisco Nexus device periodically transmits UDLD frames to neighbor devices on LAN ports with UDLD enabled. If the frames are echoed back within a specific time frame and they lack a specific acknowledgment (echo), the link is flagged as unidirectional and the LAN port is shut down. Devices on both ends of the link must support UDLD in order for the protocol to successfully identify and disable unidirectional links.

The following figure shows an example of a unidirectional link condition. Device B successfully receives traffic from Device A on the port. However, Device A does not receive traffic from Device B on the same port. UDLD detects the problem and disables the port.

Figure 1: Unidirectional Link



Default UDLD Configuration

The following table shows the default UDLD configuration.

Table 1: UDLD Default Configuration

Feature	Default Value
UDLD global enable state	Globally disabled
UDLD aggressive mode	Disabled
UDLD per-port enable state for fiber-optic media	Enabled on all Ethernet fiber-optic LAN ports
UDLD per-port enable state for twisted-pair (copper) media	Disabled on all Ethernet 10/100 and 1000BASE-TX LAN ports

UDLD Aggressive and Nonaggressive Modes

UDLD aggressive mode is disabled by default. You can configure UDLD aggressive mode only on point-to-point links between network devices that support UDLD aggressive mode. If UDLD aggressive mode is enabled, when a port on a bidirectional link that has a UDLD neighbor relationship established stops receiving UDLD frames, UDLD tries to reestablish the connection with the neighbor. After eight failed retries, the port is disabled.

To prevent spanning tree loops, nonaggressive UDLD with the default interval of 15 seconds is fast enough to shut down a unidirectional link before a blocking port transitions to the forwarding state (with default spanning tree parameters).

When you enable the UDLD aggressive mode, the following occurs:

- One side of a link has a port stuck (both transmission and receive)
- One side of a link remains up while the other side of the link is down

In these cases, the UDLD aggressive mode disables one of the ports on the link, which prevents traffic from being discarded.

Guidelines and Limitations for Layer 2 Interfaces

Layer 2 interfaces have the following configuration guidelines and limitations:

- Auto-negotiation is not supported.
- 1G autonegotiation not supported on N3K-C36180YC-R and N9K-X96136YC-R switches. To work around this issue, you must manually set speed to 1000. If autonegotiation is enabled on the neighbors, you must disable autonegotiation on those neighbors.
- On Cisco Nexus N3K-C3636C-R and N3K-C36180YC-R switches, auto-negotiation may not work on ports 49-64 when bringing up 100G links using QSFP-100G-CR4 cable. To work around this issue, you must hard-code the speed on ports 49-64 and disable auto-negotiation

Interface Speed

Cisco Nexus 36180YC-R switches have 48 small form-factor pluggable (SFP) ports with a default speed of 10 G and 6 quad small form-factor pluggable (QSFP) ports with a default speed of 100 G. 48 SFP interface ports can support 25 G, 10 G, 1 G speeds. 6 QSFP interface ports can support 100 G and 40 G speeds.

In the first 48 ports, each 4 ports in the port group must have the same speed configured. You cannot configure one port at a time which might result in an error. For more information, see CSCve80686.

Table 2: Breakout Modes Support Matrix

Switches	4x10G	4x25G	2x50G
N3K-C3636C-R	Yes	Yes	Yes
N3K-C36180YC-R	Yes	Yes	Yes

40-Gigabit Ethernet Interface Speed

You can operate QSFP ports as either 40-Gigabit Ethernet or 4x10-Gigabit Ethernet modes on Cisco Nexus 3600 platform switches. By default, there are 6 QSFP interface ports numbered 49 to 54 which can operated in 40-Gigabit Ethernet mode. These 40-Gigabit Ethernet ports are numbered in a 2-tuple naming convention. For example, the second 40-Gigabit Ethernet port is numbered as 1/50. The process of changing the configuration from 40-Gigabit Ethernet to 10-Gigabit Ethernet is called breakout and the process of changing the configuration from 10-Gigabit Ethernet to Gigabit Ethernet is called breakin. When you break out a 40-Gigabit Ethernet port into 10-Gigabit Ethernet ports, the resulting ports are numbered using a 3-tuple naming convention. For example, the break-out ports of the second 40-Gigabit Ethernet port are numbered as 1/49/1, 1/49/2, 1/49/3, 1/49/4.

Note The breakout ports are in administratively enabled state after the breakout of the 40G ports into 4x10G mode or the breaking of the 100G ports into 4x25G mode. On upgrade from the earlier releases, the configuration restored takes care of restoring the appropriate administrative state of the ports.

Note When you break out from 40-Gigabit Ethernet to 10-Gigabit Ethernet, or break in from 10-Gigabit Ethernet to 40-Gigabit Ethernet, all interface configurations are reset, and the affected ports are administratively unavailable. To make these ports available, use the **no shut** command.

Note A new QSFP+ 40-Gb transceiver is supported on the Cisco Nexus 3600 platform switches. The new QSFP+ (40-Gb) transceiver has a cable that splits into four 10Gb SFP-10G-LR transceivers. To use it, you need the port to be in 4x10G mode. If you are using the breakout cable, you need to run that 40G port in 4x10G mode.

The ability to break out a 40-Gigabit Ethernet port into four 10-Gigabit Ethernet ports and break in four 10-Gigabit Ethernet ports into a 40-Gigabit Ethernet port dynamically allows you to use any of the breakout-capable ports to work in the 40-Gigabit Ethernet or 10-Gigabit Ethernet modes without permanently defining them.

SVI Autostate

The Switch Virtual Interface (SVI) represents a logical interface between the bridging function and the routing function of a VLAN in the device. By default, when a VLAN interface has multiple ports in the VLAN, the SVI goes to the down state when all the ports in the VLAN go down.

Autostate behavior is the operational state of an interface that is governed by the state of the various ports in its corresponding VLAN. An SVI interface on a VLAN comes up when there is at least one port in that vlan that is in STP forwarding state. Similarly, this interface goes down when the last STP forwarding port goes down or goes to another STP state.

By default, Autostate calculation is enabled. You can disable Autostate calculation for an SVI interface and change the default value.

Cisco Discovery Protocol

The Cisco Discovery Protocol (CDP) is a device discovery protocol that runs over Layer 2 (the data link layer) on all Cisco-manufactured devices (routers, bridges, access servers, and switches) and allows network management applications to discover Cisco devices that are neighbors of already known devices. With CDP, network management applications can learn the device type and the Simple Network Management Protocol (SNMP) agent address of neighboring devices that are running lower-layer, transparent protocols. This feature enables applications to send SNMP queries to neighboring devices.

CDP runs on all media that support Subnetwork Access Protocol (SNAP). Because CDP runs over the data-link layer only, two systems that support different network-layer protocols can learn about each other.

Each CDP-configured device sends periodic messages to a multicast address, advertising at least one address at which it can receive SNMP messages. The advertisements also contain time-to-live, or holdtime information, which is the length of time a receiving device holds CDP information before discarding it. Each device also listens to the messages sent by other devices to learn about neighboring devices.

The switch supports both CDP Version 1 and Version 2.

Default CDP Configuration

The following table shows the default CDP configuration.

Table 3: Default CDP Configuration

Feature	Default Setting
CDP interface state	Enabled
CDP timer (packet update frequency)	60 seconds

Feature	Default Setting
CDP holdtime (before discarding)	180 seconds
CDP Version-2 advertisements	Enabled

Error-Disabled State

An interface is in the error-disabled (err-disabled) state when the inteface is enabled administratively (using the **no shutdown** command) but disabled at runtime by any process. For example, if UDLD detects a unidirectional link, the interface is shut down at runtime. However, because the interface is administratively enabled, the interface status displays as err-disabled. Once an interface goes into the err-disabled state, you must manually reenable it or you can configure an automatic timeout recovery value. The err-disabled detection is enabled by default for all causes. The automatic recovery is not configured by default.

When an interface is in the err-disabled state, use the **errdisable detect cause** command to find information about the error.

You can configure the automatic err-disabled recovery timeout for a particular err-disabled cause by changing the time variable.

The **errdisable recovery cause** command provides automatic recovery after 300 seconds. To change the recovery period, use the **errdisable recovery interval** command to specify the timeout period. You can specify 30 to 65535 seconds.

To disable recovery of an interface from the err-disabled state, use the **no errdisable recovery cause** command.

The various options for the **errdisable recover cause** command are as follows:

- all—Enables a timer to recover from all causes.
- bpduguard—Enables a timer to recover from the bridge protocol data unit (BPDU) Guard error-disabled state.
- failed-port-state—Enables a timer to recover from a Spanning Tree Protocol (STP) set port state failure.
- link-flap—Enables a timer to recover from linkstate flapping.
- pause-rate-limit—Enables a timer to recover from the pause rate limit error-disabled state.
- udld—Enables a timer to recover from the Unidirectional Link Detection (UDLD) error-disabled state.
- loopback—Enables a timer to recover from the loopback error-disabled state.

If you do not enable the err-disabled recovery for the cause, the interface stays in the err-disabled state until you enter the **shutdown** and **no shutdown** commands. If the recovery is enabled for a cause, the interface is brought out of the err-disabled state and allowed to retry operation once all the causes have timed out. Use the **show interface status err-disabled** command to display the reason behind the error.

Default Interfaces

You can use the default interface feature to clear the configured parameters for both physical and logical interfaces such as the Ethernet, loopback, management, VLAN, and the port-channel interface.

Debounce Timer Parameters

The debounce timer delays notification of a link change, which can decrease traffic loss due to network reconfiguration. You can configure the debounce timer separately for each Ethernet port and specify the delay time in milliseconds. The delay time can range from 0 milliseconds to 5000 milliseconds. By default, this parameter is set for 100 milliseconds, which results in the debounce timer not running. When this parameter is set to 0 milliseconds, the debounce timer is disabled.

Caution Enabling the debounce timer causes the link-down detections to be delayed, which results in a loss of traffic during the debounce period. This situation might affect the convergence and reconvergence of some Layer 2 and Layer 3 protocols.

MTU Configuration

The switch does not fragment frames. As a result, the switch cannot have two ports in the same Layer 2 domain with different maximum transmission units (MTUs). A per-physical Ethernet interface MTU is not supported. Instead, the MTU is set according to the QoS classes. You modify the MTU by setting class and policy maps.

Note When you show the interface settings, a default MTU of 1500 is displayed for physical Ethernet interfaces.

Counter Values

See the following information on the configuration, packet size, incremented counter values, and traffic.

Configuration	Packet Size	Incremented Counters	Traffic
L2 port – without any MTU configuration	6400 and 10000	Jumbo, giant, and input error	Dropped
L2 port – with jumbo MTU 9216 in network-qos configuration	6400	Jumbo	Forwarded
L2 port – with jumbo MTU 9216 in network-qos configuration	10000	Jumbo, giant, and input error	Dropped

Configuration	Packet Size	Incremented Counters	Traffic
Layer 3 port with default Layer 3 MTU and jumbo MTU 9216 in network-qos configuration	6400	Jumbo	Packets are punted to the CPU (subjected to CoPP configs), get fragmented, and then they are forwarded by the software.
Layer 3 port with default Layer 3 MTU and jumbo MTU 9216 in network-qos configuration	6400	Jumbo	Packets are punted to the CPU (subjected to CoPP configs), get fragmented, and then they are forwarded by the software.
Layer 3 port with default Layer 3 MTU and jumbo MTU 9216 in network-qos configuration	10000	Jumbo, giant, and input error	Dropped
Layer 3 port with jumbo Layer 3 MTU and jumbo MTU 9216 in network-qos configuration	6400	Jumbo	Forwarded without any fragmentation.
Layer 3 port with jumbo Layer 3 MTU and jumbo MTU 9216 in network-qos configuration	10000	Jumbo, giant, and input error	Dropped
Layer 3 port with jumbo Layer 3 MTU and default L2 MTU configuration	6400 and 10000	Jumbo, giant, and input error	Dropped

Note

- Under 64 bytes packet with good CRC–The short frame counter increments.
- Under 64 bytes packet with bad CRC–The runts counter increments.
- Greater than 64 bytes packet with bad CRC–The CRC counter increments.

Downlink Delay

You can operationally enable uplink SFP+ ports before downlink RJ-45 ports after a reload on a Cisco Nexus 3048 switch. You must delay enabling the RJ-45 ports in the hardware until the SFP+ ports are enabled.

You can configure a timer that during reload enables the downlink RJ-45 ports in hardware only after the specified timeout. This process allows the uplink SFP+ ports to be operational first. The timer is enabled in the hardware for only those ports that are admin-enable.

Downlink delay is disabled by default and must be explicitly enabled. When enabled, if the delay timer is not specified, it is set for a default delay of 20 seconds.

Default Physical Ethernet Settings

The following table lists the default settings for all physical Ethernet interfaces:

Parameter	Default Setting
Duplex	Auto (full-duplex)
Encapsulation	ARPA
MTU[1]	1500 bytes
Port Mode	Access
Speed	Auto (10000)

[1] MTU cannot be changed per-physical Ethernet interface. You modify MTU by selecting maps of QoS classes.

Configuring Ethernet Interfaces

Guidelines for Configuring Ethernet Interfaces

There is a behavior change in configuring the interface Ethernet commands on Cisco Nexus 3000 Series switches. For example, the command **sh int ethernet Eth1/1 transceiver** does not work anymore. You have to configure the command as **sh int ethernet 1/1 transceiver**.

Configuring the UDLD Mode

You can configure normal or aggressive unidirectional link detection (UDLD) modes for Ethernet interfaces on devices configured to run UDLD. Before you can enable a UDLD mode for an interface, you must make sure that UDLD is already enabled on the device that includes the interface. UDLD must also be enabled on the other linked interface and its device.

To use the normal UDLD mode, you must configure one of the ports for normal mode and configure the other port for the normal or aggressive mode. To use the aggressive UDLD mode, you must configure both ports for the aggressive mode.

Note Before you begin, UDLD must be enabled for the other linked port and its device.

SUMMARY STEPS

1. switch# **configure terminal**

2. switch(config)# **feature udld**
3. switch(config)# **no feature udld**
4. switch(config)# **show udld global**
5. switch(config)# **interface** *type slot*/*port*
6. switch(config-if)# **udld** {**enable** | **disable** | **aggressive**}
7. switch(config-if)# **show udld** *interface*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature udld**	Enables UDLD for the device.
Step 3	switch(config)# **no feature udld**	Disables UDLD for the device.
Step 4	switch(config)# **show udld global**	Displays the UDLD status for the device.
Step 5	switch(config)# **interface** *type slot*/*port*	Specifies an interface to configure, and enters interface configuration mode.
Step 6	switch(config-if)# **udld** {**enable** | **disable** | **aggressive**}	Enables the normal UDLD mode, disables UDLD, or enables the aggressive UDLD mode.
Step 7	switch(config-if)# **show udld** *interface*	Displays the UDLD status for the interface.

Example

This example shows how to enable UDLD for the switch:

```
switch# configure terminal
switch(config)# feature udld
```

This example shows how to enable the normal UDLD mode for an Ethernet port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# udld enable
```

This example shows how to enable the aggressive UDLD mode for an Ethernet port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# udld aggressive
```

This example shows how to disable UDLD for an Ethernet port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
```

```
switch(config-if)# udld disable
```

This example shows how to disable UDLD for the switch:

```
switch# configure terminal
switch(config)# no feature udld
```

Triggering the Link State Consistency Checker

You can manually trigger the link state consistency checker to compare the hardware and software link status of an interface and display the results. To manually trigger the link state consistency checker and display the results, use the following command in any mode:

SUMMARY STEPS

1. switch# **show consistency-checker link-state module** *slot*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **show consistency-checker link-state module** *slot*	Starts a link state consistency check on the specified module and displays its results.

Example

This example shows how to trigger a Link State consistency check and display its results:

```
switch# show consistency-checker link-state module 1
Link State Checks: Link state only
Consistency Check: FAILED
No inconsistencies found for:
   Ethernet1/1
   Ethernet1/2
   Ethernet1/3
   Ethernet1/4
   Ethernet1/5
   Ethernet1/6
   Ethernet1/7
   Ethernet1/8
   Ethernet1/9
   Ethernet1/10
   Ethernet1/12
   Ethernet1/13
   Ethernet1/14
   Ethernet1/15
Inconsistencies found for following interfaces:
   Ethernet1/11
```

Configuring the Interface Speed

The first 48 ports support 1 G/10 G/25 G and the remaining 6 ports support 40 G/100 G.

In the first 48 ports, each 4 ports in the port group must have the same speed configured. You cannot configure one port at a time which might result in an error. For more information, see CSCve80686.

Table 4:

Port Groups	Ports
Port-Group 1	Ports 1-4
Port-Group 2	Ports 5-8
Port-Group 3	Ports 9-12
Port-Group 4	Ports 13-16
Port-Group 5	Ports 17-20
Port-Group 6	Ports 21-24
Port-Group 7	Ports 25-28
Port-Group 8	Ports 29-32
Port-Group 9	Ports 33-36
Port-Group 10	Ports 37-40
Port-Group 11	Ports 41-44
Port-Group 12	Ports 45-48

Note If the interface and transceiver speed is mismatched, the SFP validation failed message is displayed when you enter the **show interface ethernet** *slot*/*port* command. For example, if you insert a 1-Gigabit SFP transceiver into a port without configuring the **speed 1000** command, you will get this error. By default, all ports are 10 Gbps.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot*/*port*
3. switch(config-if)# **speed** *speed*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Enters interface configuration mode for the specified interface. This interface must have a 1-Gigabit Ethernet SFP transceiver inserted into it.

	Command or Action	Purpose
Step 3	switch(config-if)# **speed** *speed*	Sets the speed on the interface. This command can only be applied to a physical Ethernet interface. The *speed* argument can be set to one of the following: • 10 Mbps • 100 Mbps • 1 Gbps • 10 Gbps • automatic

Example

This example shows how to set the speed for a 1-Gigabit Ethernet port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# speed 1000
```

Configuring Breakout on QSFP 40-Gigabit Ethernet Interfaces

When you break out ports into the 10-GbE mode, you can switch between the first QSFP port and SFP+ ports 1 to 4. Either the first QSFP port or the four SFP+ ports can be active at any time. QSFP is the default port with an interface speed of 40 Gbps.

When the first QSFP port is in the 40-GbE mode, you cannot switch the port to four SFP+ ports and the first QSFP port will be active until you break out the port into the 10-GbE mode. This is because SFP+ ports do not support the 40-GbE mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface breakout module** *module number* **port** *port range***map 10g-4x**
3. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface breakout module** *module number* **port** *port range***map 10g-4x**	Enables you to configure the module in 10g mode. When you are changing the portmode from QSFP to SFP+, the **hardware profile front portmode** command takes effect only after breaking out the first QSFP port as displayed in this command.

	Command or Action	Purpose
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure breakout on 40-Gigabit Ethernet Interface:

```
switch# show int e1/49 tranceiver
Ethernet1/49transceiver is present
type is QSFP-4X10G-AOC1M
name is CISCO-AVAGO
part number is AFBR-7IER01Z-CS2
revision is 01
serial number is AVE20421070
nominal bitrate is 10300 MBit/sec per channel
Link length supported for copper is 1 m
cisco id is 13
cisco extended id number is 16
cisco part number is 10-2932-02
cisco product id is QSFP-4X10G-AOC1M
cisco vendor id is V02

switch# configure terminal
switch(config)#
switch(config)# interface breakout module 1 port 49 map 10g-4x
switch(config)# exit


switch# show interface ethernet 1/49/1-4 br

--------------------------------------------------------------------------------
Ethernet VLAN Type Mode Status Reason Speed Port
Interface Ch #
--------------------------------------------------------------------------------
Eth1/49/1 1 eth access up none 10G(D) --
Eth1/49/2 1 eth access up none 10G(D) --
Eth1/49/3 1 eth access up none 10G(D) --
Eth1/49/4 1 eth access up none 10G(D) --
```

Disabling Link Negotiation

You can disable link negotiation using the **no negotiate auto** command. By default, auto-negotiation is enabled on 1-Gigabit ports and disabled on 10-Gigabit ports. By default, auto-negotiation is enabled on the Cisco Nexus 3064 and 3064-X switches and disabled on the Cisco Nexus 3048 switch. You cannot disable auto-negotiation on 1-Gigabit ports.

By default, auto-negotiation is enabled on all 1G SFP+ and 40G QSFP ports and it is disabled on 10G SFP+ ports. Auto-negotiation is by default enabled on all 1G and 10G Base-T ports. It cannot be disabled on 1G and 10G Base-T ports.

This command is equivalent to the Cisco IOS **speed non-negotiate** command.

Starting with Release 6.0(2)U5(1), you can disable auto-negotiation on all 40G interfaces. A new CLI command **no system default interface 40g auto-negotation** is introduced to disable auto-negotiation across all the 40G

interfaces. The new CLI command is only effective on the 40G interfaces and it does not have any effect on 1G or10G interfaces. For CR4 cables, the auto-negotiation configuration has to be identical at both the end devices for the link to come up.

Note The auto-negotiation configuration is not applicable on 10-Gigabit Ethernet ports. When auto-negotiation is configured on a 10-Gigabit port, the following error message is displayed:

```
ERROR: Ethernet1/40: Configuration does not match the port capability
```

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface ethernet** *slot*/*port*
3. switch(config-if)# **no negotiate auto**
4. (Optional) switch(config-if)# **negotiate auto**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot*/*port*	Selects the interface and enters interface mode.
Step 3	switch(config-if)# **no negotiate auto**	Disables link negotiation on the selected Ethernet interface (1-Gigabit port).
Step 4	(Optional) switch(config-if)# **negotiate auto**	Enables link negotiation on the selected Ethernet interface. The default for 1-Gigabit Ethernet ports is enabled. **Note** This command is not applicable for 10GBASE-T ports. It should not be used on 10-GBASE-T ports.

Example

This example shows how to disable auto-negotiation on a specified Ethernet interface (1-Gigabit port):

```
switch# configure terminal
switch(config)# interface ethernet 1/1
switch(config-if)# no negotiate auto
switch(config-if)#
```

This example shows how to enable auto-negotiation on a specified Ethernet interface (1-Gigabit port):

```
switch# configure terminal
switch(config)# interface ethernet 1/5
switch(config-if)# negotiate auto
switch(config-if)#
```

Disabling SVI Autostate

You can configure a SVI to remain active even if no interfaces are up in the corresponding VLAN. This enhancement is called Autostate Disable.

When you enable or disable autostate behavior, it is applied to all the SVIs in the switch unless you configure autostate per SVI .

Note Autostate behavior is enabled by default.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **feature interface-vlan**
3. switch(config)# **[no]system default interface-vlan autostate**
4. (Optional) switch(config)# **interface vlan** *interface-vlan-number*
5. (Optional) switch(config-if)# [**no**] **autostate**
6. (Optional) switch(config)# **show interface-vlan** *interface-vlan*
7. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature interface-vlan**	Enables the interface-vlan feature.
Step 3	Required: switch(config)# **[no]system default interface-vlan autostate**	Configures the system to enable or disable the Autostate default behavior.
Step 4	(Optional) switch(config)# **interface vlan** *interface-vlan-number*	Creates a VLAN interface. The number range is from 1 to 4094.
Step 5	(Optional) switch(config-if)# [**no**] **autostate**	Enables or disables Autostate behavior per SVI.
Step 6	(Optional) switch(config)# **show interface-vlan** *interface-vlan*	Displays the enabled or disabled Autostate behavior of the SVI.
Step 7	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to disable the systems Autostate default for all the SVIs on the switch:

```
switch# configure terminal
switch(config)# feature interface-vlan
switch(config)# system default interface-vlan no autostate
```

```
switch(config)# interface vlan 50
switch(config-if)# no autostate
switch(config)# copy running-config startup-config
```

This example shows how to enable the systems autostate configuration:

```
switch(config)# show interface-vlan 2
Vlan2 is down, line protocol is down, autostate enabled
Hardware is EtherSVI, address is 547f.ee40.a17c
MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec
```

Configuring a Default Interface

The default interface feature allows you to clear the existing configuration of multiple interfaces such as Ethernet, loopback, management, VLAN, and port-channel interfaces. All user configuration under a specified interface will be deleted.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **default interface** *type interface number*
3. switch(config)# **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **default interface** *type interface number*	Deletes the configuration of the interface and restores the default configuration. The following are the supported interfaces: • **ethernet** • **loopback** • **mgmt** • **port-channel** • **vlan**
Step 3	switch(config)# **exit**	Exits global configuration mode.

Example

This example shows how to delete the configuration of an Ethernet interface and revert it to its default configuration:

```
switch# configure terminal
switch(config)# default interface ethernet 1/3
.......Done
switch(config)# exit
```

Configuring the CDP Characteristics

You can configure the frequency of Cisco Discovery Protocol (CDP) updates, the amount of time to hold the information before discarding it, and whether or not to send Version-2 advertisements.

SUMMARY STEPS

1. switch# **configure terminal**
2. (Optional) switch(config)# [**no**] **cdp advertise** {**v1** | **v2** }
3. (Optional) switch(config)# [**no**] **cdp format device-id** {**mac-address** | **serial-number** | **system-name**}
4. (Optional) switch(config)# [**no**] **cdp holdtime** *seconds*
5. (Optional) switch(config)# [**no**] **cdp timer** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	(Optional) switch(config)# [**no**] **cdp advertise** {**v1**	**v2** }	Configures the version to use to send CDP advertisements. Version-2 is the default state. Use the **no** form of the command to return to its default setting.
Step 3	(Optional) switch(config)# [**no**] **cdp format device-id** {**mac-address**	**serial-number**	**system-name**}	Configures the format of the CDP device ID. The default is the system name, which can be expressed as a fully qualified domain name. Use the **no** form of the command to return to its default setting.
Step 4	(Optional) switch(config)# [**no**] **cdp holdtime** *seconds*	Specifies the amount of time a receiving device should hold the information sent by your device before discarding it. The range is 10 to 255 seconds; the default is 180 seconds. Use the **no** form of the command to return to its default setting.
Step 5	(Optional) switch(config)# [**no**] **cdp timer** *seconds*	Sets the transmission frequency of CDP updates in seconds. The range is 5 to 254; the default is 60 seconds. Use the **no** form of the command to return to its default setting.

Example

This example shows how to configure CDP characteristics:

```
switch# configure terminal
switch(config)# cdp timer 50
switch(config)# cdp holdtime 120
switch(config)# cdp advertise v2
```

Enabling or Disabling CDP

You can enable or disable CDP for Ethernet interfaces. This protocol works only when you have it enabled on both interfaces on the same link.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot/port*
3. switch(config-if)# **cdp enable**
4. switch(config-if)# **no cdp enable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot/port*	Enters interface configuration mode for the specified interface.
Step 3	switch(config-if)# **cdp enable**	Enables CDP for the interface. To work correctly, this parameter must be enabled for both interfaces on the same link.
Step 4	switch(config-if)# **no cdp enable**	Disables CDP for the interface.

Example

This example shows how to enable CDP for an Ethernet port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# cdp enable
```

This command can only be applied to a physical Ethernet interface.

Enabling the Error-Disabled Detection

You can enable error-disable (err-disabled) detection in an application. As a result, when a cause is detected on an interface, the interface is placed in an err-disabled state, which is an operational state that is similar to the link-down state.

Note Base ports in Cisco Nexus 5500 never get error disabled due to pause rate-limit like in the Cisco Nexus 5020 or 5010 switch.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **errdisable detect cause** {*all* | *link-flap* | *loopback*}
3. switch(config)# **shutdown**
4. switch(config)# **no shutdown**
5. switch(config)# **show interface status err-disabled**
6. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **errdisable detect cause** {*all* | *link-flap* | *loopback*}	Specifies a condition under which to place the interface in an err-disabled state. The default is enabled.
Step 3	switch(config)# **shutdown**	Brings the interface down administratively. To manually recover the interface from the err-disabled state, enter this command first.
Step 4	switch(config)# **no shutdown**	Brings the interface up administratively and enables the interface to recover manually from the err-disabled state.
Step 5	switch(config)# **show interface status err-disabled**	Displays information about err-disabled interfaces.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable the err-disabled detection in all cases:

```
switch# configure terminal
switch(config)# errdisable detect cause all
switch(config)# shutdown
switch(config)# no shutdown
switch(config)# show interface status err-disabled
switch(config)# copy running-config startup-config
```

Enabling the Error-Disabled Recovery

You can specify the application to bring the interface out of the error-disabled (err-disabled) state and retry coming up. It retries after 300 seconds, unless you configure the recovery timer (see the **errdisable recovery interval** command).

SUMMARY STEPS

1. switch# **configure terminal**

2. switch(config)# **errdisable recovery cause** {*all* | *udld* | *bpduguard* | *link-flap* | *failed-port-state* | *pause-rate-limit* | *loopback*}
3. switch(config)# **show interface status err-disabled**
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **errdisable recovery cause** {*all* | *udld* | *bpduguard* | *link-flap* | *failed-port-state* | *pause-rate-limit* | *loopback*}	Specifies a condition under which the interface automatically recovers from the err-disabled state, and the device retries bringing the interface up. The device waits 300 seconds to retry. The default is disabled.
Step 3	switch(config)# **show interface status err-disabled**	Displays information about err-disabled interfaces.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable err-disabled recovery under all conditions:

```
switch# configure terminal
switch(config)# errdisable recovery cause loopback
switch(config)# show interface status err-disabled
switch(config)# copy running-config startup-config
```

Configuring the Error-Disabled Recovery Interval

You can use this procedure to configure the err-disabled recovery timer value. The range is from 30 to 65535 seconds. The default is 300 seconds.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **errdisable recovery interval** *interval*
3. switch(config)# **show interface status err-disabled**
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **errdisable recovery interval** *interval*	Specifies the interval for the interface to recover from the err-disabled state. The range is from 30 to 65535 seconds. The default is 300 seconds.
Step 3	switch(config)# **show interface status err-disabled**	Displays information about err-disabled interfaces.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable err-disabled recovery under all conditions:

```
switch# configure terminal
switch(config)# errdisable recovery interval 32
switch(config)# show interface status err-disabled
switch(config)# copy running-config startup-config
```

Disabling the Error-Disabled Recovery

You can disable recovery of an interface from the err-disabled state.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **no errdisable recovery cause** {*all* | *udld* | *bpduguard* | *link-flap* | *failed-port-state* | *pause-rate-limit* | *loopback*}
3. (Optional) switch(config)# **show interface status err-disabled**
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no errdisable recovery cause** {*all* | *udld* | *bpduguard* | *link-flap* | *failed-port-state* | *pause-rate-limit* | *loopback*}	Specifies a condition under which the interface reverts back to the default err-disabled state.
Step 3	(Optional) switch(config)# **show interface status err-disabled**	Displays information about err-disabled interfaces.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to disable err-disabled recovery:

```
switch# configure terminal
switch(config)# no errdisable recovery cause loopback
switch(config)# show interface status err-disabled
switch(config)# copy running-config startup-config
```

Configuring the Debounce Timer

You can enable the debounce timer for Ethernet ports by specifying a debounce time, in milliseconds (ms), or disable the timer by specifying a debounce time of 0. By default, the debounce timer is set to 100 ms, which results in the debounce timer not running.

You can show the debounce times for all of the Ethernet ports by using the **show interface debounce** command.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot*/*port*
3. switch(config-if)# **link debounce time** *milliseconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Enters interface configuration mode for the specified interface.
Step 3	switch(config-if)# **link debounce time** *milliseconds*	Enables the debounce timer for the amount of time (1 to 5000 ms) specified. Disables the debounce timer if you specify 0 milliseconds.

Example

This example shows how to enable the debounce timer and set the debounce time to 1000 ms for an Ethernet interface:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# link debounce time 1000
```

This example shows how to disable the debounce timer for an Ethernet interface:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# link debounce time 0
```

Configuring the Description Parameter

You can provide textual interface descriptions for the Ethernet ports.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot*/*port*
3. switch(config-if)# **description** *test*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Enters interface configuration mode for the specified interface.
Step 3	switch(config-if)# **description** *test*	Specifies the description for the interface.

Example

This example shows how to set the interface description to Server 3 interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/3
switch(config-if)# description Server 3 Interface
```

Disabling and Restarting Ethernet Interfaces

You can shut down and restart an Ethernet interface. This action disables all of the interface functions and marks the interface as being down on all monitoring displays.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot*/*port*
3. switch(config-if)# **shutdown**
4. switch(config-if)# **no shutdown**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Enters interface configuration mode for the specified interface.
Step 3	switch(config-if)# **shutdown**	Disables the interface.

	Command or Action	Purpose
Step 4	switch(config-if)# **no shutdown**	Restarts the interface.

Example

This example shows how to disable an Ethernet port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# shutdown
```

This example shows how to restart an Ethernet interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# no shutdown
```

Configuring MAC addresses Limitation on a VLAN

Cisco Nexus 3600 Series switches provides the ability to set an upper limit for the number of MAC addresses that can reside inside MAC address table of a Line-card Expansion-module (LEM). You can configure the limitations at System, VLAN, port, trunk and tunnel levels. For instance if the specified VLAN limitation is 2000 MACs, the Layer 2 Forwarding Manager (L2FM) accepts the first 2000 MACs it receives and reject the remaining MACs. To configure MAC address limitation on VLAN, complete the following steps:

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **mac address-table limit system** *value*
3. switch(config)# **mac address-table limit vlan** *value*
4. switch(config)# **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **mac address-table limit system** *value*	Specifies an upper limit for MAC learning at system level.
Step 3	switch(config)# **mac address-table limit vlan** *value*	Specifies an upper limit for MAC learning at VLAN level.
Step 4	switch(config)# **exit**	Exits configuration mode.

Example

This example shows how to configure the upper limit for MAC learning at system and VLAN levels:

```
switch# configure terminal
```

```
switch(config)# mac address-table limit system 10000
Configuring Mac address limit will result in flushing existing Macs in the specified
VLAN/System.Proceed(yes/no)? [no] yes
switch(config)# mac address-table limit vlan 30 3000
Configuring Mac address limit will result in flushing existing Macs in the specified
VLAN/System.Proceed(yes/no)? [no] yes
switch(config)# exit
```

This example shows how to display the MAC address limitations:

```
switch# configure terminal
switch(config)# sh mac address-table limit

System Limit: 10000


Vlan    Learning Limit
----    ----------
1       196000
20      196000
30      3000
100     196000
switch(config)# exit
```

Configuring Custom EtherType or Tag Protocol Identifier (TPID)

The switch uses a default ethertype of 0x8100 for 802.1Q and Q-in-Q encapsulations. You can configure EtherTypes 0x9100, 0x9200 and 0x88a8 on a per port basis by enabling the **dot1q ethertype** command on the switchport interface. You can configure a custom EtherType field value on a port to support network devices that do not use the standard 0x8100 EtherType field value on 802.1Q-tagged or 802.1p-tagged frames.

Note You must set the EtherType or TPID only on the egress trunk interface that carries double tagged frames. EtherType value impacts all the tagged packets that go out on the interface (on both Q-in-Q and 802.1Q packets).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface ethernet** *slot/port*
3. switch(config-if)# **switchport**
4. switch(config-if)# **switchport mode**
5. switch(config-if)# **switchport dot1q ethertype** *value*
6. (Optional) switch(config-if)# **switchport access vlan** *value*
7. switch(config-if)# **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **interface ethernet** *slot/port*	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **switchport**	Sets the interface as a Layer 2 switching port.
Step 4	switch(config-if)# **switchport mode**	Sets the interface as a Layer 2 switching port mode.
Step 5	switch(config-if)# **switchport dot1q ethertype** *value*	Sets the EtherType for the Q-in-Q tunnel on the port.
Step 6	(Optional) switch(config-if)# **switchport access vlan** *value*	Sets the interface access VLAN.
Step 7	switch(config-if)# **exit**	Exits configuration mode.

Example

This example shows how to configure custom ethertype on an 802.1Q tunnel port:

```
switch# configure terminal
switch(config)# interface ethernet 1/1
switch(config-if)# switchport
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# switchport dot1q ethertype 0x9100
switch(config-if)# switchport access vlan 30
switch(config-if)# exit
switch(config)# exit
```

Configuring Downlink Delay

You can operationally enable uplink SFP+ ports before downlink RJ-45 ports after a reload on a Cisco Nexus 3048 switch by delaying enabling the RJ-45 ports in the hardware until the SFP+ ports are enabled.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **downlink delay enable** | **disable** [**timeout** *time-out*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **downlink delay enable** | **disable** [**timeout** *time-out*]	Enables or disables downlink delay and configures the timeout.

Example

This example shows how to enable downlink delay and configure the delay timeout on the switch:

```
switch# configure terminal
switch(config)# downlink delay enable timeout 45
```

Displaying Interface Information

To view configuration information about the defined interfaces, perform one of these tasks:

Command	Purpose
switch# **show interface** *type slot/port*	Displays the detailed configuration of the specified interface.
switch# **show interface** *type slot/port* **capabilities**	Displays detailed information about the capabilities of the specified interface. This option is available only for physical interfaces.
switch# **show interface** *type slot/port* **transceiver**	Displays detailed information about the transceiver connected to the specified interface. This option is available only for physical interfaces.
switch# **show interface brief**	Displays the status of all interfaces.
switch# **show interface flowcontrol**	Displays the detailed listing of the flow control settings on all interfaces.

The **show interface** command is invoked from EXEC mode and displays the interface configurations. Without any arguments, this command displays the information for all the configured interfaces in the switch.

This example shows how to display the physical Ethernet interface:

```
switch# show interface ethernet 1/1
  Ethernet1/1 is up
  Hardware is 1000/10000 Ethernet, address is 000d.eca3.5f08 (bia 000d.eca3.5f08)
  MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 190/255, rxload 192/255
  Encapsulation ARPA
  Port mode is trunk
  full-duplex, 10 Gb/s, media type is 1/10g
  Input flow-control is off, output flow-control is off
  Auto-mdix is turned on
  Rate mode is dedicated
  Switchport monitor is off
  Last clearing of "show interface" counters never
  5 minute input rate 942201806 bytes/sec, 14721892 packets/sec
  5 minute output rate 935840313 bytes/sec, 14622492 packets/sec
  Rx
    129141483840 input packets 0 unicast packets 129141483847 multicast packets
    0 broadcast packets 0 jumbo packets 0 storm suppression packets
    8265054965824 bytes
    0 No buffer 0 runt 0 Overrun
    0 crc 0 Ignored 0 Bad etype drop
    0 Bad proto drop
  Tx
    119038487241 output packets 119038487245 multicast packets
   0 broadcast packets 0 jumbo packets
    7618463256471 bytes
    0 output CRC 0 ecc
    0 underrun 0 if down drop     0 output error 0 collision 0 deferred
```

```
    0 late collision 0 lost carrier 0 no carrier
    0 babble
    0 Rx pause 8031547972 Tx pause 0 reset
```

This example shows how to display the physical Ethernet capabilities:

```
switch# show interface ethernet 1/1 capabilities
Ethernet1/1
  Model:                734510033
  Type:                 10Gbase-(unknown)
  Speed:                1000,10000
  Duplex:               full
  Trunk encap. type:    802.1Q
  Channel:              yes
  Broadcast suppression: percentage(0-100)
  Flowcontrol:          rx-(off/on),tx-(off/on)
  Rate mode:            none
  QOS scheduling:       rx-(6q1t),tx-(1p6q0t)
  CoS rewrite:          no
  ToS rewrite:          no
  SPAN:                 yes
  UDLD:                 yes
  MDIX:                 no
  FEX Fabric:           yes
```

This example shows how to display the physical Ethernet transceiver:

```
switch# show interface ethernet 1/1 transceiver
Ethernet1/1
    sfp is present
    name is CISCO-EXCELIGHT
    part number is SPP5101SR-C1
    revision is A
    serial number is ECL120901AV
    nominal bitrate is 10300 MBits/sec
    Link length supported for 50/125mm fiber is 82 m(s)
    Link length supported for 62.5/125mm fiber is 26 m(s)
    cisco id is --
    cisco extended id number is 4
```

This example shows how to display a brief interface status (some of the output has been removed for brevity):

```
switch# show interface brief

--------------------------------------------------------------------------------
Ethernet      VLAN   Type Mode   Status  Reason                   Speed     Port
Interface                                                                  Ch #
--------------------------------------------------------------------------------
Eth1/1        200    eth  trunk  up      none                       10G(D) --
Eth1/2        1      eth  trunk  up      none                       10G(D) --
Eth1/3        300    eth  access down    SFP not inserted           10G(D) --
Eth1/4        300    eth  access down    SFP not inserted           10G(D) --
Eth1/5        300    eth  access down    Link not connected        1000(D) --
Eth1/6        20     eth  access down    Link not connected         10G(D) --
Eth1/7        300    eth  access down    SFP not inserted           10G(D) --
...
```

This example shows how to display the CDP neighbors:

```
switch# show cdp neighbors
Capability Codes: R - Router, T - Trans-Bridge, B - Source-Route-Bridge
```

```
                  S - Switch, H - Host, I - IGMP, r - Repeater,
                  V - VoIP-Phone, D - Remotely-Managed-Device,
                  s - Supports-STP-Dispute
Device ID             Local Intrfce   Hldtme  Capability  Platform      Port ID
d13-dist-1              mgmt0           148     S I      WS-C2960-24TC  Fas0/9
n5k(FLC12080012)        Eth1/5          8       S I s    N5K-C5020P-BA  Eth1/5
```