Configuring Layer 2 Interfaces

This chapter contains the following sections:

Information About Ethernet Interfaces

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

On a Cisco Nexus 3000 Series switch, the Ethernet interfaces are enabled by default.

About the 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:

- Interface type—All physical Ethernet interfaces use the **ethernet** keyword.
- Slot number
  - Slot 1 includes all the fixed ports.
  - Slot 2 includes the ports on the upper expansion module (if populated).
  - Slot 3 includes the ports on the lower expansion module (if populated).
  - Slot 4 includes the ports on the lower expansion module (if populated).

Note Slot 4 is only available on the Cisco Nexus 5596T switch.

- Port number
  - Port number within the group.

The interface numbering convention is extended to support use with a Cisco Nexus 2000 Series Fabric Extender as follows:

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

- Chassis ID is an optional entry 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 via the interface. The chassis ID ranges from 100 to 199.

About the 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.

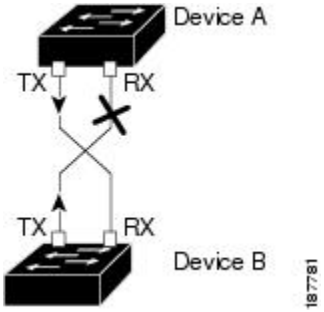
A unidirectional link occurs whenever traffic transmitted by the local device over a link is received by the neighbor but traffic transmitted from the neighbor is not received by the local device. If one of the fiber strands in a pair is disconnected, as long as autonegotiation is active, the link does 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 3000 Series switch 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.

Note By default, UDLD is locally disabled on copper LAN ports to avoid sending unnecessary control traffic on this type of media.

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.

Interface Speed

Cisco Nexus 3000 Series switches have a number of fixed 10-Gigabit ports, each equipped with SFP+ interface adapters.

About the 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 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 2: Default CDP Configuration

Feature	Default Setting
CDP interface state	Enabled
CDP timer (packet update frequency)	60 seconds
CDP holdtime (before discarding)	180 seconds
CDP Version-2 advertisements	Enabled

About the 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.

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.

About MTU Configuration

The Cisco Nexus 3000 Series 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.

Configuring Ethernet Interfaces

The section includes the following topics:

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.

Procedure

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

This example shows how to enable the 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
```

Changing an Interface Port Mode

You can configure a Quad small form-factor pluggable (QSFP+) port by using the **hardware profile portmode** command. To restore the defaults, use the **no** form of this command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **copy running-config bootflash**: *my-config.cfg*	Copies the running configuration to the bootflash. You can use this file to configure your device later.
Step 3	switch(config)# **write erase**	Removes all the interface configurations.
Step 4	switch(config)# **reload**	Reloads the Cisco Nexus 3000 Series switch software.
Step 5	switch(config)# [**no**] **hardware profile portmode** *portmode*	Changes the interface port mode.
Step 6	switch(config)# **hardware profile portmode** *portmode* **2-tuple**	(Optional) Displays the port names in 2-tuple mode instead of the default 3-tuple convention mode.
Step 7	switch(config)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 8	switch(config)# **reload**	Reloads the Cisco Nexus 3000 Series switch software. Manually apply all the interface configuration. You can refer to the configuration file that you saved earlier. **Note** The interface numbering changes if the ports are changed from 40G mode to 4x10G mode or vice versa.

This example shows how to change the port mode to 48x10g+4x40g for QSFP+ ports:

```
switch# configure terminal
switch(config) copy running-config bootflash:my-config.cfg
switch(config)# write erase
switch(config)# reload
WARNING: This command will reboot the system
Do you want to continue? (y/n) [n] y
switch(config)# hardware profile portmode 48x10g+4x40g
Warning: This command will take effect only after saving the configuration and reload!
Port configurations could get lost when port mode is changed!
switch(config)# copy running-config startup-config
switch(config)# reload
WARNING: This command will reboot the system
Do you want to continue? (y/n) [n] y
```

This example shows how to change the port mode to 48x10g+4x40g for QSFP+ ports and verify the changes:

```
switch# configure terminal
switch(config)# hardware profile portmode 48x10g+4x40g
Warning: This command will take effect only after saving the configuration and r
eload! Port configurations could get lost when port mode is changed!
switch(config)# show running-config
!Command: show running-config
!Time: Thu Aug 25 07:39:37 2011
version 5.0(3)U2(1)
feature telnet
```

```
no feature ssh
feature lldp
username admin password 5 $1$OOV4MdOM$BAB5RkD22YanT4empqqSM0 role network-admin
ip domain-lookup
switchname BLR-QG-5
ip access-list my-acl
10 deny ip any 10.0.0.1/32
20 deny ip 10.1.1.1/32 any
class-map type control-plane match-any copp-arp
class-map type control-plane match-any copp-bpdu
:
:
control-plane
service-policy input copp-system-policy
hardware profile tcam region arpacl 128
hardware profile tcam region ifacl 256
hardware profile tcam region racl 256
hardware profile tcam region vacl 512
hardware profile portmode 48x10G+4x40G
snmp-server user admin network-admin auth md5 0xdd1d21ee42e93106836cdefd1a60e062
<--Output truncated-->
switch#
```

This example shows how to restore the default port mode for QSFP+ ports:

```
switch# configure terminal
switch(config)# no hardware profile portmode
Warning: This command will take effect only after saving the configuration and r
eload! Port configurations could get lost when port mode is changed!
switch(config)#
```

Configuring Interface Speed

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 Gigabits.

Procedure

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

	Command or Action	Purpose
		• 10Gbps • automatic

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
```

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.

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

Note We do not recommend that you enable auto negotiation on 10-Gigabit ports. Enabling auto-negotiation on 10-Gigabit ports brings the link down. By default, link negotiation is disabled on 10-Gigabit ports.

Procedure

	Command or Action	Purpose
Step 1	switch# configure terminal	Enters 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	switch(config-if)# negotiate auto	(Optional) Enables link negotiation on the selected Ethernet interface. The default for 1-Gigabit ports is enabled. **Note** This command is not applicable for 10GBase-T ports. It should not be used on 10GBase-T ports.

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)#
```

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.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# [**no**] **cdp advertise** {**v1** | **v2** }	(Optional) 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	switch(config)# [**no**] **cdp format device-id** {**mac-address** | **serial-number** | **system-name**}	(Optional) 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	switch(config)# [**no**] **cdp holdtime** *seconds*	(Optional) 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	switch(config)# [**no**] **cdp timer** *seconds*	(Optional) 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.

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.

Procedure

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

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.

Procedure

	Command or Action	Purpose

Step 1	**config t**
	
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**errdisable detect cause** {*all* | *link-flap* | *loopback*}
	
	Example:
	`switch(config)# errdisable detect cause all`
	`switch(config)#`
		Specifies a condition under which to place the interface in an err-disabled state. The default is enabled.

	Command or Action	Purpose

Step 3	**shutdown**
	
	Example:
	`switch(config)# shutdown`
	`switch(config)#`
		Brings the interface down administratively. To manually recover the interface from the err-disabled state, enter this command first.

Step 4	**no shutdown**
	
	Example:
	`switch(config)# no shutdown`
	`switch(config)#`
		Brings the interface up administratively and enables the interface to recover manually from the err-disabled state.

Step 5	**show interface status err-disabled** **Example:** `switch(config)# show interface status err-disabled`	Displays information about err-disabled interfaces.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

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

```
switch(config)#errdisable detect cause all
switch(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).

Procedure

	Command or Action	Purpose

Step 1	**config t**
	
	Example:
	`switch#config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**errdisable recovery cause** {*all* | *udld* | *bpduguard* | *link-flap* | *failed-port-state* | *pause-rate-limit*}	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.

	Command or Action	Purpose

	Example:
	`switch(config)#errdisable recovery cause all`
	`switch(config-if)#`	

Step 3	**show interface status err-disabled** **Example:** `switch(config)#show interface status err-disabled`	Displays information about err-disabled interfaces.
Step 4	**copy running-config startup-config** **Example:** `switch(config)#copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

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

```
switch(config)#errdisable recovery cause all
switch(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.

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch#config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**errdisable recovery interval** *interval*
	Example:
	`switch(config)#errdisable recovery interval 32`
	`switch(config-if)#`
		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	**show interface status err-disabled** **Example:** `switch(config)#show interface status err-disabled`	Displays information about err-disabled interfaces.

	Command or Action	Purpose
Step 4	**copy running-config startup-config** **Example:** `switch(config)#copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

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

```
switch(config)#errdisable recovery cause all
switch(config)#
```

Configuring the Description Parameter

You can provide textual interface descriptions for the Ethernet ports.

Procedure

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

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. This information is communicated to other network servers through all dynamic routing protocols. When shut down, the interface is not included in any routing updates.

Procedure

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

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

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
```

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 only available for physical interfaces
switch# **show interface** *type slot/port* **transceiver**	Displays detailed information about the transceiver connected to the specified interface. This option is only available 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
```

Displaying Input Packet Discard Information

Beginning with Cisco NX-OS Release 5.0(3)U2(1), you can get detailed information on what specific condition led to an input discard on a given interface. Use the **show hardware internal interface indiscard-stats front-port** *x* command to determine the condition that could be potentially responsible for the input discards that are seen on port eth1/x. The switch output shows the discards for IPv4, STP, input policy, ACL specific discard, generic receive drop, and VLAN related discards.

This example shows how to determine the condition that could be potentially responsible for the input discards:

```
switch# show hardware internal interface indiscard-stats front-port 1

+----------------------------------------+-----------------+----------------+
|       Counter Description              |     Count       |                |
+----------------------------------------+-----------------+----------------+
        IPv4 Discards                           0
        STP Discards                            0
        Policy Discards                         100
        ACL Drops                               0
        Receive Drops                           0
        Vlan Discards                           33
+----------------------------------------+-----------------+----------------+
```

Counter Information:

- IPv4 Discards--- IPv4 Discards represent errors at the IP layer, for example the IP checksum error.
- STP Discards--- STP Discards are incremented when the receive interface STP state is not forwarding the packets received.
- Policy Discards--- Policy Discards are incremented when there are discards because of input policy on the interface.
- ACL Drops---ACL drops indicate that incoming packets match an ACL entry with a drop action.
- Receive Drops--- This drop increment represents a condition when no output port is determined for an ingress packet. Receive drops happen because of variety of reasons including IPv4, STP and policy discards. The drop counter increments in conjunction with one of the above counters or separately.
- Vlan Discard--- Vlan Discard indicates vlan-based discards. For example, a vlan tagged packet ingressing on a port which is not a member of the vlan.

This example shows how to clear all the input discard counters which is useful for debugging purposes:

```
Switch# show hardware internal interface indiscard-stats front-port 1 clear

+----------------------------------+---------------+--------------+-------------------------------+
|       Counter Description              |     Count       | Last Increment |    Last
Increment Time              |
```

```
+-------------------------------------+----------------+--------------+---------------------------------+

 Discard Stats have been reset
+-------------------------------------+----------------+--------------+---------------------------------+
```

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.

MIBs for Layer 2 Interfaces

MIB	MIB Link
IF-MIB	To locate and download MIBs, go to the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml
MAU-MIB Limited support includes only the following MIB Objects: • ifMauType (Read-only) GET • ifMauAutoNegSupported (Read-only) GET • ifMauTypeListBits (Read-only) GET • ifMauDefaultType (Read-write) GET-SET • ifMauAutoNegAdminStatus (Read-write) GET-SET • ifMauAutoNegCapabilityBits (Read-only) GET • ifMauAutoNegAdvertisedBits (Read-write) GET-SET	