CDP, LLDP, MAC, and UDLD Configuration Guide

First Published: 2025-09-15

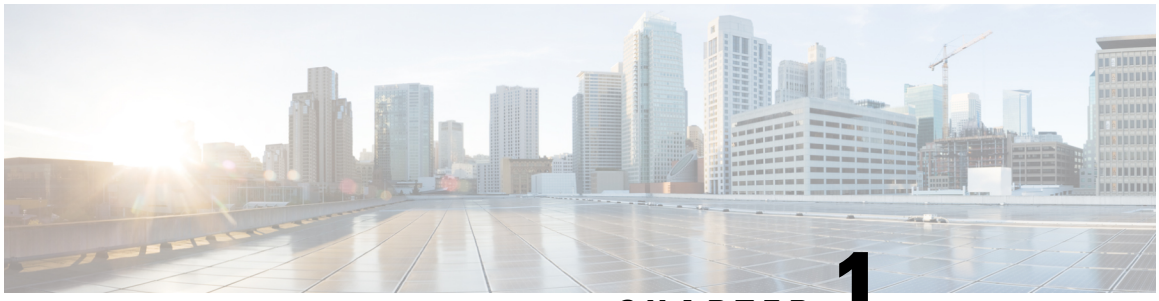
Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Read Me First

Only supported features are documented. To confirm or clarify all the supported features for a platform, go to Cisco Feature Navigator.

Read Me First

CONTENTS



CHAPTER 1

Configure CDP

Feature History for CDP

This table provides release and platform support information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Table 1:

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	Cisco Discovery Protocol (CDP) is a proprietary Layer 2 network protocol that operates on Cisco devices. It enables devices to discover directly connected Cisco equipment and gather vital information about neighboring devices, facilitating network management and troubleshooting.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

Cisco Discovery Protocol

Cisco Discovery Protocol (CDP) is a proprietary Layer 2 network protocol that operates on Cisco devices. It enables devices to discover directly connected Cisco equipment and gather vital information about neighboring devices, facilitating network management and troubleshooting.

CDP operates in two primary versions:

- **CDPv1:** This initial version provides basic information about directly connected Cisco devices. This includes details such as their device ID, the local interface connected, the platform type, the device's capabilities (e.g., router, switch, host), its Cisco IOS version, and its IP address.
- **CDPv2:** This enhanced version builds upon CDPv1 by adding more detailed information. This includes the VLAN Trunking Protocol (VTP) domain name, the native VLAN used on the connected port, the duplex mode of the interface, and power consumption details for Power over Ethernet (PoE) devices.

CDP offers several advantages for network administrators, enhancing network visibility and simplifying operations:

- **Network Topology Discovery:** CDP automatically maps the network topology by identifying how Cisco devices connect to each other. This eliminates the need for manual mapping, which is particularly beneficial in large or complex network environments.
- **Troubleshooting Connectivity Issues:** By displaying comprehensive information about directly connected devices and their configurations, CDP helps you quickly identify and resolve common connectivity problems, such as duplex mismatches or incorrect VLAN assignments.
- **Inventory Management:** CDP provides a real-time inventory of connected Cisco devices, including their specific hardware platforms and software versions. This assists in asset tracking and compliance.
- **Simplified Configuration:** CDP can facilitate the auto-configuration of certain devices, such as Cisco IP phones, by providing them with necessary network parameters like VLAN information.

After understanding Cisco Discovery Protocols, you will be able to:

- Identify the version of CDP running on your devices.
- Verify information about directly connected Cisco devices.
- Leverage CDP for network discovery and basic troubleshooting.

How Cisco Discovery Protocol Works

CDP-enabled devices periodically send multicast advertisements to directly connected neighboring devices. These advertisements contain various pieces of information about the sending device, including its identity, capabilities, and configuration details. Receiving devices store this information in a local CDP table. You can then view this table using Command Line Interface (CLI) commands to gain insights into your network's immediate connections.

Configure Cisco Discovery Protocol

Procedure

	Command or Action	Purpose

Step 1	cdp run
	Example:
	`Device# configure terminal`
	`Device(config)# cdp run`
	`Device(config)# end`
		To enable CDP globally on a device, enter global configuration mode. Use the cdp run command.

	Command or Action	Purpose

Step 2	no cdp run
	Example:
	`Device# configure terminal`
	`Device(config)# no cdp run`
	`Device(config)# end`
		To disable CDP globally, enter global configuration mode. Use the no cdp run command.

Step 3	cdp enable
	Example:
	`Device# configure terminal`
	`Device(config)# interface GigabitEthernet0/1`
	`Device(config-if)# cdp enable`
	`Device(config-if)# end`
		To enable CDP on a specific interface, enter interface configuration mode for the desired interface. Use the cdp enable command.

Step 4	no cdp enable
	Example:
	`Device# configure terminal`
	`Device(config)# interface GigabitEthernet0/1`
	`Device(config-if)# no cdp enable`
	`Device(config-if)# end`
		To disable CDP on a specific interface, enter interface configuration mode for the desired interface. Use the no cdp enable command.

Verify Cisco Discovery Protocol

You can verify CDP status and view the information about neighboring devices using various show cdp commands.

Procedure

	Command or Action	Purpose

Step 1	show cdp neighbors
	Example:
	`Router# show cdp neighbors`
	`Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge, S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone, D - Remote, C - CVTA, M - Two-port Mac Relay`
	`Device ID Local Intrfce Holdtme Capability Platform Port ID`
	`Switch1 GigabitEthernet0/1 120 S I WS-C9350-24P GigabitEthernet1/0/1`
	`IP_Phone GigabitEthernet0/2 175 H P IP Phone Port 1`
		To view information about directly connected Cisco devices, enter the show cdp neighbors command in privileged EXEC mode.

Cisco Discovery Protocol Bypass

Cisco Discovery Protocol (CDP) Bypass describes scenarios or configurations where the standard CDP discovery mechanism is intentionally or unintentionally circumvented, preventing devices from exchanging information about their neighbors. This concept is crucial for understanding network visibility and security implications.

Understanding CDP Bypass enables you to:

- **Manage Network Visibility:** Control which devices are discoverable via CDP, particularly in multi-vendor environments or security-sensitive zones.
- **Enhance Security Posture:** Prevent the leakage of network topology and device information to unauthorized entities or across specific network boundaries.
- **Troubleshoot Connectivity:** Identify why certain devices are not appearing in CDP neighbor tables, which might be due to intentional bypass configurations.

After understanding CDP Bypass, you will be able to:

- Evaluate the implications of CDP's presence or absence on network segments.
- Implement configurations that align with your network's security and operational requirements regarding device discovery.

Scenarios of CDP Bypass

CDP Bypass typically occurs in the following situations:

- **CDP is Disabled:**
  - **Globally on a Device:** When CDP is turned off for the entire switch or router, it will neither send nor receive CDP advertisements, effectively bypassing any CDP discovery.
  - **On a Specific Interface:** CDP can be disabled on individual interfaces. This is common for interfaces connecting to untrusted networks, non-Cisco devices, or in security zones where information disclosure is undesirable.
- **Non-Cisco Devices:** CDP is a proprietary Cisco protocol. Devices from other vendors do not support CDP and therefore inherently "bypass" its discovery capabilities. They will not send or process CDP advertisements.
- **Security Zone Boundaries:** In network designs that enforce strict security boundaries, CDP is often intentionally disabled on interfaces connecting different security zones (e.g., connecting a DMZ to an internal network) to prevent internal network topology details from being exposed.

Implications of CDP Bypass

When CDP is bypassed, either intentionally or unintentionally, consider these implications:

- **Reduced Automated Discovery:** You lose the benefit of automatic network topology mapping and rapid neighbor identification that CDP provides.

- **Increased Security:** By limiting the information exchanged, you reduce the attack surface and prevent potential reconnaissance by malicious actors.
- **Manual Documentation Required:** Without CDP, maintaining accurate network documentation becomes a manual process, requiring up-to-date records of device connections and configurations.
- **Impact on Features:** Some Cisco features, such as Power over Ethernet (PoE) negotiation for IP phones or certain auto-configuration capabilities, may rely on CDP. Disabling CDP might impact the functionality of these features.

Managing Cisco Discovery Protocol

You can manage CDP's operation to achieve a "bypass" state when required.

Cisco Discovery Protocol IPv6 Support

Cisco Discovery Protocol (CDP) IPv6 support refers to the capability of CDP to discover and exchange IPv6 address information between directly connected Cisco devices. This feature extends CDP's traditional IPv4 discovery functions to modern IPv6 networks, providing comprehensive visibility and simplifying management in dual-stack or IPv6-only environments.

CDP IPv6 support enables you to:

- **Gain Visibility in IPv6 Networks:** Discover neighboring Cisco devices and their IPv6 addresses, including link-local and global unicast addresses, in networks where IPv6 is deployed.
- **Simplify Troubleshooting:** Quickly identify IPv6 connectivity issues, verify neighbor reachability, and confirm interface configurations by leveraging CDP's neighbor discovery capabilities.
- **Enhance Network Management:** Maintain an accurate inventory of devices and their IPv6 addressing schemes, crucial for network planning and operational efficiency in IPv6 deployments.

After understanding CDP IPv6 support, you will be able to:

- Verify IPv6 neighbor information using CDP.
- Troubleshoot basic IPv6 connectivity issues between Cisco devices.
- Incorporate CDP IPv6 information into your network documentation and management practices.

How Cisco Discovery Protocol IPv6 Support Works

When CDP IPv6 support is enabled on a device and its interfaces, CDP messages (advertisements) are updated to include IPv6 address information in addition to IPv4 details. These advertisements are sent periodically as multicast frames (both IPv4 and IPv6 multicast addresses are used) to directly connected CDP-enabled neighbors. Receiving devices parse these messages and update their local CDP neighbor tables with the discovered IPv6 addresses, along with other device details.

This allows network administrators to use familiar CDP commands to inspect IPv6 neighbor information, providing a consistent operational experience across IPv4 and IPv6 addressing schemes.

Configure Cisco Discovery Protocol IPv6 Support

Procedure

	Command or Action	Purpose

Step 1	cdp run
	Example:
	`Device# configure terminal`
	`Device(config)# cdp run`
	`Device(config)# end`
		To enable CDP globally, enter global configuration mode. Use the cdp run command.

Step 2	cdp enable
	Example:
	`Device(config)# interface GigabitEthernet1/0/1`
	`Device(config-if)# cdp enable`
	`Device(config-if)# end`
		To enable CDP on a specific interface, enter interface configuration mode. Use the cdp enable command.

Verify Cisco Discovery Protocol IPv6 Support

You can verify CDP IPv6 support and view the IPv6 addresses of neighboring devices using the show cdp neighbors detail command.

Procedure

	Command or Action	Purpose

Step 1	show cdp neighbors detail
	Example:
	`Router# show cdp neighbors detail`
	`-------------------------`
	`Device ID: Switch2`
	`Entry address(es):`
	`  IP address: 192.168.1.2`
	`  IPv6 address: 2001:DB8:0:1::2`
	`Platform: cisco WS-C9350-24P,`
	`Capabilities: Switch IGMP`
	`Interface: GigabitEthernet1/0/1, Port ID (outgoing port): GigabitEthernet1/0/2`
	`Holdtime : 162 sec`
		To view detailed information about directly connected Cisco devices, including their IPv6 addresses, enter the show cdp neighbors detail command in privileged EXEC mode.

Considerations for CDP IPv6 Support

- **Link-Local Addresses:** CDP frequently exchanges link-local IPv6 addresses, which are automatically configured on interfaces and are only valid for communication on the local link. These are essential for neighbor discovery even without global unicast addresses.
- **Security:** As with IPv4, CDP can reveal network topology information. In security-sensitive environments, consider disabling CDP on interfaces connected to untrusted networks to prevent information leakage.

- **Resource Usage:** CDP uses minimal network and CPU resources. However, in very large-scale deployments, managing CDP on all interfaces might require careful planning.

Cisco Discovery Protocol Location Support

Cisco Discovery Protocol (CDP) Location Support refers to CDP's capability to carry and exchange physical location information about connected devices. This feature extends CDP's traditional device discovery functions by providing geographical or civic address details for endpoints like IP phones, wireless access points, or other network devices. It is particularly valuable for location-based services, emergency services, and asset management.

CDP Location Support enables you to:

- **Enhance Emergency Services (E911):** Provide accurate physical location data for devices, such as IP phones, to emergency responders, ensuring faster and more precise assistance.
- **Improve Asset Tracking:** Maintain a real-time inventory of device locations within a physical environment, simplifying asset management and auditing.
- **Aid Network Planning and Operations:** Understand the physical distribution of network devices and endpoints, which is crucial for capacity planning, troubleshooting, and physical security.

After understanding CDP Location Support, you will be able to:

- Configure physical location information on Cisco switches and devices.
- Verify that CDP is correctly advertising location data to connected neighbors.
- Leverage location information obtained via CDP for various network services and management tasks.

How Cisco Discovery Protocol Location Support Works

CDP Location Support functions by embedding location-specific Type-Length-Value (TLV) fields within standard CDP advertisements. When a Cisco device (e.g., a switch) is configured with its physical location, or when a connected endpoint (e.g., an IP phone) reports its location, this information can be included in the CDPv2 messages exchanged between devices.

The location information can be:

- **Civic Location:** Details such as street address, building, floor, or room number.
- **Geographical Coordinates:** Latitude, longitude, and altitude.

The sending device includes these TLVs in its periodic CDP advertisements. Receiving devices parse these TLVs and store the location information in their local CDP neighbor tables. This allows network administrators to query the switch for the location of connected devices, providing a dynamic and up-to-date physical topology.

Configure Cisco Discovery Protocol Location Support

Procedure

	Command or Action	Purpose
Step 1	cdp advertise-v2 **Example:** `Device(config)# cdp advertise-v2`	Enable CDPv2 TLV advertising globally.
Step 2	location civic-location street-address <address> **Example:** `Device(config)# location civic-location street-address "123 Main St" city "Anytown" state "CA" postal-code "90210" floor "3" room "305"`	Configure the device's civic location.
Step 3	location coordinate latitude <co-ordinates> **Example:** `Device(config)# location coordinate latitude 34.0522 longitude -118.2437 altitude 100`	Configure the device's geographical coordinates (optional, alternative to civic address).

Verify Cisco Discovery Protocol Location Support

You can verify that CDP is advertising location information by examining the detailed CDP neighbor output.

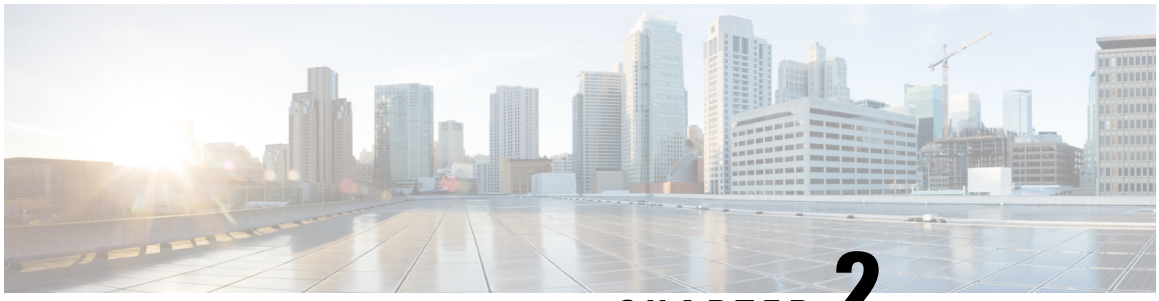
To view detailed information about directly connected Cisco devices, including their advertised location, enter the show cdp neighbors detail command in privileged EXEC mode.

Procedure

	Command or Action	Purpose

Step 1	show cdp neighbors detail
	Example:
	`Router# show cdp neighbors detail`
	`-------------------------`
	`Device ID: IP_Phone_1`
	`Entry address(es):`
	`  IP address: 10.1.1.100`
	`Platform: Cisco IP Phone 8841,`
	`Capabilities: Host Phone`
	`Interface: GigabitEthernet1/0/1, Port ID (outgoing port): Port 1`
	`Holdtime : 165 sec`
	`Version :`
	`Cisco IP Phone OS`
	`Location:`
	`  Civic Address: street-address 123 Main St, city Anytown, state CA, postal-code 90210, floor 3, room 305`	

	Command or Action	Purpose
	Coordinates: latitude 34.0522, longitude -118.2437, altitude 100	



CHAPTER 2

Configure LLDP

Feature History for LLDP

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	Link Layer Discovery Protocol (LLDP) is an Layer 2 protocol that enables network devices to advertise their identity, capabilities, and connectivity information to directly connected devices.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

Link Layer Discovery Protocol

Link Layer Discovery Protocol (LLDP) is an Layer 2 protocol that enables network devices to advertise their identity, capabilities, and connectivity information to directly connected devices.

On Cisco switches, LLDP allows the switch to discover neighboring devices and learn about their configuration, regardless of the device type or vendor. This information is exchanged between devices on a port-by-port basis. LLDP operates by sending and receiving LLDP Data Units (LLDPDUs) that contain Type-Length-Value (TLV) elements. These TLVs carry specific pieces of information about the sending device and its port.

Key information exchanged

LLDP advertisements typically include details such as:

- **Chassis ID:** A unique identifier for the device, often its MAC address.
- **Port ID:** A unique identifier for the specific port (e.g., port number, interface name).
- **Port Description:** A textual description of the port.

- **System Name:** The hostname of the device.
- **System Description:** A description of the operating system software and hardware.
- **System Capabilities:** The major functions of the device (e.g., router, switch, WLAN AP).
- **Management Address:** The IP address for managing the device.
- **Power over Ethernet (PoE) Information:** Details about power requirements and capabilities.

Benefits of LLDP

LLDP is invaluable for network administrators for:

- **Network topology discovery:** Quickly mapping out network connections and understanding physical connectivity.
- **Troubleshooting:** Identifying connectivity issues, verifying physical connections, and detecting misconfigurations.
- **Asset management:** Gaining visibility into connected devices and their attributes, aiding in inventory and documentation.
- **Deployment and configuration:** Informing other protocols or automated systems, particularly for Power over Ethernet (PoE) devices, to ensure proper power delivery and network policy application.

Restrictions and limitations of LLDP

LLDP, while highly beneficial, has specific operational characteristics and potential considerations:

- **Layer 2 only operation:** LLDP operates strictly at Layer 2 (the data link layer) of the OSI model. This means it can only discover and exchange information with directly connected neighbors on the same physical segment. It does not provide information about devices across Layer 3 boundaries.
- **Information disclosure:** LLDP broadcasts device information, which could potentially be leveraged by unauthorized parties if enabled on untrusted or publicly accessible ports. It is not an authentication mechanism and does not verify the identity of the neighbor.
- **Resource consumption:** Although generally minimal, enabling LLDP on a large number of ports, especially in very dense environments, can consume a small amount of CPU and memory resources for processing and storing neighbor information.
- **No configuration enforcement:** LLDP is a discovery protocol; it gathers and advertises information. It does not automatically apply or propagate configurations to neighboring devices. Other protocols or manual configurations are required to act upon the discovered information.
- **Vendor-specific TLVs:** While LLDP is an open standard, vendors may implement proprietary TLVs for specific features or information. These vendor-specific TLVs might not be understood or processed by devices from other manufacturers.
- **Unidirectional operation:** If LLDP is configured in send-only or receive-only mode on a port, the full discovery capability between two devices might be limited. For complete neighbor discovery, both devices should be configured to send and receive LLDP packets.

Types of LLDP

LLDP is primarily a single protocol, but its functionality can be extended, and it operates in different modes.

- **Base LLDP:** This refers to the core LLDP protocol (IEEE 802.1AB) that provides the fundamental device discovery and information exchange capabilities. It includes the standard TLVs for chassis ID, port ID, system name, system description, and capabilities.
- **LLDP-MED (Media Endpoint Devices):** This is an extension to the base LLDP standard (ANSI/TIA-1057) designed specifically for Voice over IP (VoIP) phones, video conferencing units, and other media endpoint devices. LLDP-MED provides additional TLVs that are crucial for these applications, such as:
  - **Network policy:** Advertises VLAN IDs, Layer 2 priority (802.1p), and Layer 3 Differentiated Services Code Point (DSCP) values for voice and video traffic.
  - **Power over MDI (Media Dependent Interface):** Provides detailed power management information for PoE devices, including power type, source, priority, and allocated power.
  - **Inventory:** Allows discovery of hardware and software versions of the endpoint.
  - **Location identification:** Enables the endpoint to send its physical location (e.g., civic address, coordinate-based location) to the switch, which is critical for emergency services (E911).
- **LLDP modes of operation:** On Cisco switches, LLDP can be configured to operate in different modes on a per-interface basis:
  - **Transmit (Tx):** The interface sends LLDPDUs but does not process incoming LLDPDUs.
  - **Receive (Rx):** The interface processes incoming LLDPDUs but does not send its own LLDPDUs.
  - **Transmit and Receive (TxRx):** The interface sends and receives LLDPDUs. This is typically the default and recommended mode for full discovery.
  - **Disabled:** LLDP is not active on the interface.

Configure LLDP

Follow these steps to configure LLDP on your switch:

Before you begin

You must have privileged EXEC mode access to the switch.

Procedure

	Command or Action	Purpose
Step 1	Switch# enable	Enables privileged EXEC mode. Enter your password if prompted
Step 2	Switch# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	Switch(config)# lldp run	Enables LLDP globally on the switch. This command allows the LLDP process to start running on the device.
Step 4	Switch(config)# interface GigabitEthernet0/1	Specifies the interface on which to configure LLDP, and enters interface configuration mode. Replace type and number with your specific interface (e.g., GigabitEthernet0/1, FastEthernet0/5).
Step 5	Switch(config-if)# lldp transmit	Configures the interface to send LLDPDUs.
Step 6	Switch(config-if)# lldp receive	Configures the interface to receive LLDPDUs. Note: For full bidirectional discovery, you typically configure both lldp transmit and lldp receive on the interface. The lldp transmit receive command is a shortcut for both.
Step 7	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 8	Switch# copy running-config startup-config	Saves the running configuration to the startup configuration. This ensures that your LLDP configuration is retained after a switch reload. Press Enter when prompted to confirm the destination filename.

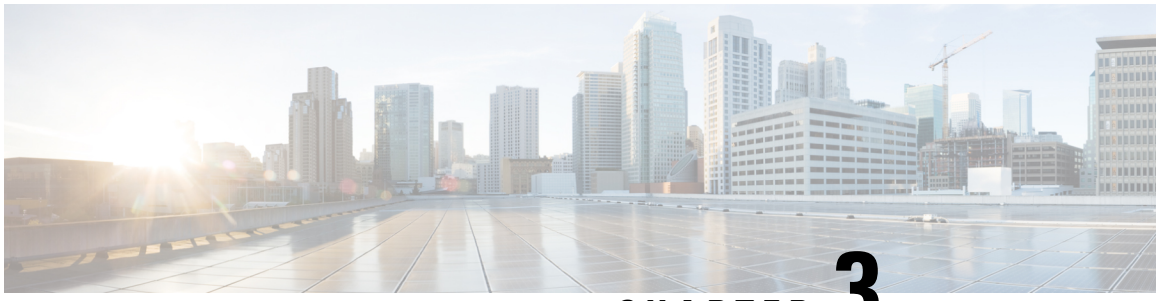
Verify LLDP

After configuring LLDP, use these commands in privileged EXEC mode to verify its operation:

Table 1: Commands for displaying LLDP status

Command	Purpose	Example Output
show lldp	Displays the global LLDP status, including whether it's enabled and the default timer values.	LLDP is enabled LLDP advertisements are sent every 30 seconds LLDP hold time is 120 seconds
show lldp interface [type number]	Shows LLDP status for all interfaces or a specific interface, indicating whether transmit and receive are enabled.	GigabitEthernet0/1: Tx: enabled Rx: enabled FastEthernet0/5 is not being advertised)

show lldp neighbors	Displays a summary of discovered LLDP neighbors, including their local port, device ID, and remote port ID.	Device ID Local Intf Hold-time Capabilities Port ID --- Router1 Gig 0/1 120 R, B GigabitEthernet0/0 IPPhone Gig 0/2 120 T MAC:aabb.ccdd.eeff
show lldp neighbors detail	Provides detailed information about all discovered LLDP neighbors, including all advertised TLVs (system name, description, capabilities, management IP, etc.).	--- Chassis id: 0011.2233.4455 Port id: GigabitEthernet0/0 Port Description: Uplink to Switch System Name: Router1 System Description: Cisco IOS Software …
show lldp entry [device-id]	Displays detailed LLDP information for a specific neighbor.	(Similar detailed output as show lldp neighbors detail but filtered for a single device.)



CHAPTER 3

Configure MAC

Feature History for MAC

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	A MAC address is a Layer 2 address used to identify individual devices on a local area network (LAN). It is a globally unique identifier hardcoded into the network adapter by the manufacturer.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

MAC Address and MAC Address Table Management

MAC Address Table Management (MATM) refers to the processes by which a Layer 2 switch builds, maintains, and uses its MAC address table to efficiently forward Ethernet frames.

A Layer 2 switch's primary function is to forward frames based on MAC addresses. To do this efficiently, the switch maintains a MAC address table that maps MAC addresses to specific switch ports and VLANs. MATM encompasses the mechanisms of how entries are added, updated, and removed from this table.

Key MATM processes:

- **MAC address learning:**
  - When a switch receives a frame on a port, it inspects the **source MAC address** of the frame.
  - If the source MAC address is not already in the MAC address table, or if it is in the table but associated with a different port, the switch adds or updates the entry, associating the MAC address with the incoming port and VLAN.

- This allows the switch to know which port to use when sending frames *to* that MAC address in the future.

- **Frame forwarding:**
  - When a switch receives a frame, it inspects the **destination MAC address**.
  - The switch then looks up this destination MAC address in its MAC address table.
  - **Known destination:** If the destination MAC address is found and associated with a specific port (different from the incoming port), the switch forwards the frame only out of that specific port. This is called **unicast forwarding**.
  - **Unknown destination (flooding):** If the destination MAC address is not found in the table, the switch floods the frame out of all ports within the same VLAN, except the port on which it was received. This ensures the frame reaches its intended destination, and the destination's response will allow the switch to learn its MAC address.
  - **Same port (filtering):** If the destination MAC address is found and associated with the same port on which the frame was received, the switch drops (filters) the frame. This prevents unnecessary traffic from being sent back out the same segment.
  - **Broadcast/multicast:** Broadcast frames (destination MAC FF:FF:FF:FF:FF:FF) and unknown multicast frames are always flooded out of all ports within the same VLAN, except the incoming port.

- **MAC address aging:**
  - MAC address entries in the table are not permanent. Each entry has an associated aging timer.
  - If the switch does not receive any frames from a particular source MAC address on a specific port within the aging time, that entry is removed from the MAC address table.
  - Aging ensures that the table remains current and does not become cluttered with stale entries from devices that are no longer connected or have moved to a different port. The default aging time is typically 300 seconds (5 minutes).

Benefits of MATM:

- **Efficient frame delivery:** Frames are sent only to the necessary destination port, reducing unnecessary traffic on other segments.
- **Reduced collisions (in half-duplex environments):** By segmenting collision domains, switches improve network performance.
- **Enhanced security:** Limits the exposure of traffic to only the relevant devices, although not a security feature in itself.
- **Dynamic adaptation:** The table automatically updates as devices connect, disconnect, or move within the network.

Restrictions and limitations of MAC and MATM

Both MAC addresses and MATM processes have inherent restrictions and limitations:

MAC address limitations:

- **Flat address space:** MAC addresses provide no hierarchical information for routing beyond the local segment. They do not indicate network location or subnet, making them unsuitable for large-scale routing.
- **Burned-in nature:** MAC addresses are generally hardcoded into the network interface controller (NIC) and are not easily changed by end-users. While they can be spoofed (changed programmatically), this is typically a violation of network policy.
- **Security vulnerabilities:** MAC addresses are not inherently secure. They are susceptible to spoofing, where a malicious actor impersonates another device by using its MAC address, and to MAC flooding attacks, which can overwhelm a switch's MAC address table.
- **Scalability in broadcast domains:** While unique, managing an extremely large number of MAC addresses within a single broadcast domain (VLAN) can lead to very large CAM tables, consuming significant switch memory and potentially impacting lookup performance.

MATM limitations:

- **CAM table size limits:** Switches have finite Content Addressable Memory (CAM) table sizes. If the number of learned MAC addresses exceeds this limit, the switch may resort to flooding unknown unicast frames out of all ports in a VLAN, effectively behaving like a hub for those frames. This impacts performance and can be exploited for security breaches.
- **MAC flooding attacks:** Malicious actors can deliberately send frames with many unique source MAC addresses to rapidly fill the CAM table, forcing the switch into a flooding state.
- **MAC address flapping:** A MAC address learned on multiple ports (or rapidly moving between ports) indicates a network loop or misconfiguration. While switches detect and log these events, constant flapping can lead to CAM table instability, CPU utilization spikes, and service disruptions.
- **Aging time impact:** An aging time that is too short can cause frequent re-learning of MAC addresses, increasing switch overhead. An aging time that is too long can keep stale entries in the table, leading to frames being sent to incorrect ports until the entry ages out or is re-learned.
- **No Layer 3 awareness:** MATM operates purely at Layer 2. It does not understand IP addresses, higher-layer protocols, or network topology beyond direct connectivity.

Types of MAC addresses

This section describes different classifications of MAC addresses and various methods for managing MAC address entries within the switch's table.

Types of MAC addresses:

- **Unicast MAC addresses:** These are unique identifiers assigned to a single network interface. They are the most common type and are used for one-to-one communication. The least significant bit of the first octet is 0.
- **Multicast MAC addresses:** These addresses are used for one-to-many communication within a local network segment. Frames sent to a multicast MAC address are received by all devices configured to listen for that specific multicast group. The least significant bit of the first octet is 1. IPv4 multicast addresses typically map to MAC addresses in the 01:00:5E:XX:XX:XX range.
- **Broadcast MAC addresses:** The broadcast MAC address is FF:FF:FF:FF:FF:FF. Frames sent to this address are delivered to all devices within the same broadcast domain (VLAN).

- **Universally administered addresses (UAAs):** These are the most common type of MAC address. They are burned into the NIC by the manufacturer and are globally unique. The OUI identifies the manufacturer.
- **Locally administered addresses (LAAs):** These are MAC addresses that can be manually configured or assigned by a network administrator, overriding the burned-in address. LAAs are not globally unique and must be managed carefully to avoid conflicts within a network.

Types of MAC address table entries (MATM methods):

- **Dynamic MAC addresses:** These are the most common type of entry. The switch learns them automatically when it receives a frame from a source MAC address on a specific port. Dynamic entries have an aging timer and are removed if no traffic is seen from that MAC address within the aging period.
- **Static MAC addresses:** These entries are manually configured by an administrator to permanently associate a MAC address with a specific port and VLAN. Static entries do not age out and remain in the CAM table until they are manually removed. They are useful for critical devices that should always be accessible on a specific port.
- **Sticky MAC addresses:** These combine aspects of dynamic and static learning. When enabled on a port, the switch dynamically learns MAC addresses and converts them into sticky entries. These sticky entries are then stored in the running configuration. If the running configuration is saved to the startup configuration, these entries persist across reboots. If a device associated with a sticky MAC address moves to another port, it triggers a security violation.
- **Secure MAC addresses (Port Security):** These are MAC addresses that are learned or configured under the port security feature. Port security allows you to limit the number of MAC addresses that can be learned on a port and define actions to take if a violation occurs (e.g., shutdown the port, restrict traffic). Secure MAC addresses can be dynamic, static, or sticky.

Configuration steps for MAC

-

Configuring global MAC address table aging time

Sticky MAC addresses are a port security feature that prevents unauthorized devices from connecting to a switch port. This feature controls which specific devices are allowed on certain ports.

Port security with the sticky feature enabled allows a switch port to learn MAC addresses dynamically. The switch then stores these learned MAC addresses in its running configuration. This makes the MAC addresses persistent.

If a device with a new MAC address attempts to connect to a port with sticky MAC addresses, a security violation occurs. A security violation triggers predefined actions. These actions include:

- **Shutdown:** The port enters an error-disabled state. Manual intervention is required to reactivate the port.
- **Restrict:** The switch drops packets from the unauthorized source MAC address. A violation counter increments, and a Simple Network Management Protocol (SNMP) trap message is sent to the administrator. Authorized traffic continues to flow.
- **Protect:** The switch drops packets from the unauthorized source MAC address. No log messages, SNMP traps, or violation counter increments occur.

The feature also supports configuring a maximum number of sticky MAC addresses per port. Reaching this limit with a new MAC address triggers a violation.

Benefits

- **Enhanced security:** Prevents unauthorized device access to the network.
- **Network stability:** Reduces disruptions from unauthorized MAC address changes.
- **Simplified management:** Automates MAC address learning and persistence, reducing manual configuration.
- **Persistence:** Learned MAC addresses remain associated with the port across reboots (if saved to startup configuration).

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** `Switch> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	interface type number **Example:** `Switch(config)# interface GigabitEthernet0/2`	Specifies the interface for port security, and enters interface configuration mode.
Step 4	switchport mode access **Example:** `Switch(config-if)# switchport mode access`	Configures the interface as an access port. Port security is typically configured on access ports.
Step 5	switchport port-security **Example:** `Switch(config-if)# switchport port-security`	Enables port security on the interface.
Step 6	switchport port-security maximum value **Example:** `Switch(config-if)# switchport port-security maximum 5`	Configures the maximum number of secure MAC addresses allowed on the port. For value, the range is typically 1 to a maximum supported by the switch model.
Step 7	switchport port-security mac-address sticky **Example:** `Switch(config-if)# switchport port-security mac-address sticky`	Enables sticky learning on the port. Dynamically learned MAC addresses are converted to sticky secure MAC addresses

	Command or Action	Purpose
Step 8	switchport port-security violation {protect | restrict | shutdown} **Example:** `Switch(config-if)# switchport port-security violation shutdown`	Configures the action to take when a security violation occurs. * protect: Drops packets from unknown sources until a secure MAC address is removed. * restrict: Drops packets from unknown sources, sends an SNMP trap, and increments a violation counter. * shutdown: Shuts down the interface, sends an SNMP trap, and increments a violation counter. The interface remains shut down until manually re-enabled or error-disabled recovery is configured.
Step 9	end **Example:** `Switch(config-if)# end`	Returns to priviledged EXEC mode
Step 10	copy running-config startup-config **Example:** `Switch# copy running-config startup-config`	Saves the running configuration to the startup configuration

Configuring a static MAC address entry

Sticky MAC addresses are a port security feature that prevents unauthorized devices from connecting to a switch port. This feature controls which specific devices are allowed on certain ports.

Port security with the sticky feature enabled allows a switch port to learn MAC addresses dynamically. The switch then stores these learned MAC addresses in its running configuration. This makes the MAC addresses persistent.

If a device with a new MAC address attempts to connect to a port with sticky MAC addresses, a security violation occurs. A security violation triggers predefined actions. These actions include:

- **Shutdown:** The port enters an error-disabled state. Manual intervention is required to reactivate the port.
- **Restrict:** The switch drops packets from the unauthorized source MAC address. A violation counter increments, and a Simple Network Management Protocol (SNMP) trap message is sent to the administrator. Authorized traffic continues to flow.
- **Protect:** The switch drops packets from the unauthorized source MAC address. No log messages, SNMP traps, or violation counter increments occur.

The feature also supports configuring a maximum number of sticky MAC addresses per port. Reaching this limit with a new MAC address triggers a violation.

Benefits

- **Enhanced security:** Prevents unauthorized device access to the network.
- **Network stability:** Reduces disruptions from unauthorized MAC address changes.
- **Simplified management:** Automates MAC address learning and persistence, reducing manual configuration.
- **Persistence:** Learned MAC addresses remain associated with the port across reboots (if saved to startup configuration).

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** `Switch> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	interface type number **Example:** `Switch(config)# interface GigabitEthernet0/2`	Specifies the interface for port security, and enters interface configuration mode.
Step 4	switchport mode access **Example:** `Switch(config-if)# switchport mode access`	Configures the interface as an access port. Port security is typically configured on access ports.
Step 5	switchport port-security **Example:** `Switch(config-if)# switchport port-security`	Enables port security on the interface.
Step 6	switchport port-security maximum value **Example:** `Switch(config-if)# switchport port-security maximum 5`	Configures the maximum number of secure MAC addresses allowed on the port. For value, the range is typically 1 to a maximum supported by the switch model.
Step 7	switchport port-security mac-address sticky **Example:** `Switch(config-if)# switchport port-security mac-address sticky`	Enables sticky learning on the port. Dynamically learned MAC addresses are converted to sticky secure MAC addresses
Step 8	switchport port-security violation {protect | restrict | shutdown}	Configures the action to take when a security violation occurs.

	Command or Action	Purpose
	Example: `Switch(config-if)# switchport port-security violation shutdown`	* protect: Drops packets from unknown sources until a secure MAC address is removed. * restrict: Drops packets from unknown sources, sends an SNMP trap, and increments a violation counter. * shutdown: Shuts down the interface, sends an SNMP trap, and increments a violation counter. The interface remains shut down until manually re-enabled or error-disabled recovery is configured.
Step 9	end **Example:** `Switch(config-if)# end`	Returns to priviledged EXEC mode
Step 10	copy running-config startup-config **Example:** `Switch# copy running-config startup-config`	Saves the running configuration to the startup configuration

Configuring port security with sticky MAC addresses

Sticky MAC addresses are a port security feature that prevents unauthorized devices from connecting to a switch port. This feature controls which specific devices are allowed on certain ports.

Port security with the sticky feature enabled allows a switch port to learn MAC addresses dynamically. The switch then stores these learned MAC addresses in its running configuration. This makes the MAC addresses persistent.

If a device with a new MAC address attempts to connect to a port with sticky MAC addresses, a security violation occurs. A security violation triggers predefined actions. These actions include:

- **Shutdown:** The port enters an error-disabled state. Manual intervention is required to reactivate the port.
- **Restrict:** The switch drops packets from the unauthorized source MAC address. A violation counter increments, and a Simple Network Management Protocol (SNMP) trap message is sent to the administrator. Authorized traffic continues to flow.
- **Protect:** The switch drops packets from the unauthorized source MAC address. No log messages, SNMP traps, or violation counter increments occur.

The feature also supports configuring a maximum number of sticky MAC addresses per port. Reaching this limit with a new MAC address triggers a violation.

Benefits

- **Enhanced security:** Prevents unauthorized device access to the network.
- **Network stability:** Reduces disruptions from unauthorized MAC address changes.

- **Simplified management:** Automates MAC address learning and persistence, reducing manual configuration.
- **Persistence:** Learned MAC addresses remain associated with the port across reboots (if saved to startup configuration).

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** `Switch> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	interface type number **Example:** `Switch(config)# interface GigabitEthernet0/2`	Specifies the interface for port security, and enters interface configuration mode.
Step 4	switchport mode access **Example:** `Switch(config-if)# switchport mode access`	Configures the interface as an access port. Port security is typically configured on access ports.
Step 5	switchport port-security **Example:** `Switch(config-if)# switchport port-security`	Enables port security on the interface.
Step 6	switchport port-security maximum value **Example:** `Switch(config-if)# switchport port-security maximum 5`	Configures the maximum number of secure MAC addresses allowed on the port. For value, the range is typically 1 to a maximum supported by the switch model.
Step 7	switchport port-security mac-address sticky **Example:** `Switch(config-if)# switchport port-security mac-address sticky`	Enables sticky learning on the port. Dynamically learned MAC addresses are converted to sticky secure MAC addresses
Step 8	switchport port-security violation {protect | restrict | shutdown} **Example:** `Switch(config-if)# switchport port-security violation shutdown`	Configures the action to take when a security violation occurs. * protect: Drops packets from unknown sources until a secure MAC address is removed.

	Command or Action	Purpose
		* restrict: Drops packets from unknown sources, sends an SNMP trap, and increments a violation counter. * shutdown: Shuts down the interface, sends an SNMP trap, and increments a violation counter. The interface remains shut down until manually re-enabled or error-disabled recovery is configured.
Step 9	end **Example:** `Switch(config-if)# end`	Returns to priviledged EXEC mode
Step 10	copy running-config startup-config **Example:** `Switch# copy running-config startup-config`	Saves the running configuration to the startup configuration

Configuring MAC address move notification

Sticky MAC addresses are a port security feature that prevents unauthorized devices from connecting to a switch port. This feature controls which specific devices are allowed on certain ports.

Port security with the sticky feature enabled allows a switch port to learn MAC addresses dynamically. The switch then stores these learned MAC addresses in its running configuration. This makes the MAC addresses persistent.

If a device with a new MAC address attempts to connect to a port with sticky MAC addresses, a security violation occurs. A security violation triggers predefined actions. These actions include:

- **Shutdown:** The port enters an error-disabled state. Manual intervention is required to reactivate the port.
- **Restrict:** The switch drops packets from the unauthorized source MAC address. A violation counter increments, and a Simple Network Management Protocol (SNMP) trap message is sent to the administrator. Authorized traffic continues to flow.
- **Protect:** The switch drops packets from the unauthorized source MAC address. No log messages, SNMP traps, or violation counter increments occur.

The feature also supports configuring a maximum number of sticky MAC addresses per port. Reaching this limit with a new MAC address triggers a violation.

Benefits

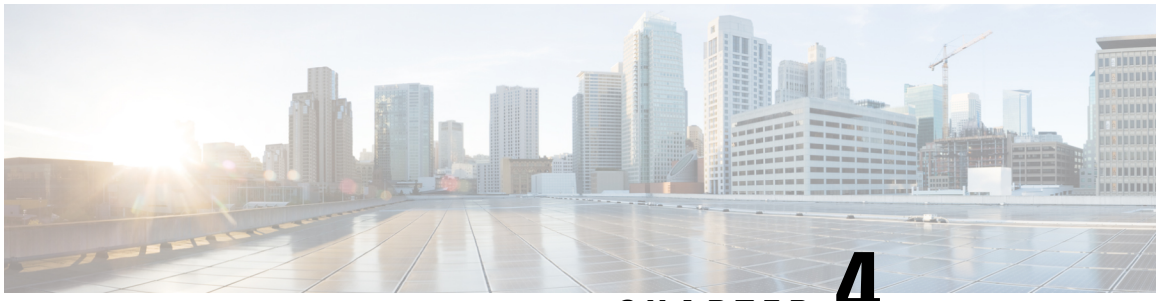
- **Enhanced security:** Prevents unauthorized device access to the network.
- **Network stability:** Reduces disruptions from unauthorized MAC address changes.
- **Simplified management:** Automates MAC address learning and persistence, reducing manual configuration.

- **Persistence:** Learned MAC addresses remain associated with the port across reboots (if saved to startup configuration).

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** `Switch> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal **Example:** `Switch# configure terminal`	Enters global configuration mode.
Step 3	interface type number **Example:** `Switch(config)# interface GigabitEthernet0/2`	Specifies the interface for port security, and enters interface configuration mode.
Step 4	switchport mode access **Example:** `Switch(config-if)# switchport mode access`	Configures the interface as an access port. Port security is typically configured on access ports.
Step 5	switchport port-security **Example:** `Switch(config-if)# switchport port-security`	Enables port security on the interface.
Step 6	switchport port-security maximum value **Example:** `Switch(config-if)# switchport port-security maximum 5`	Configures the maximum number of secure MAC addresses allowed on the port. For value, the range is typically 1 to a maximum supported by the switch model.
Step 7	switchport port-security mac-address sticky **Example:** `Switch(config-if)# switchport port-security mac-address sticky`	Enables sticky learning on the port. Dynamically learned MAC addresses are converted to sticky secure MAC addresses
Step 8	switchport port-security violation {protect | restrict | shutdown} **Example:** `Switch(config-if)# switchport port-security violation shutdown`	Configures the action to take when a security violation occurs. * protect: Drops packets from unknown sources until a secure MAC address is removed. * restrict: Drops packets from unknown sources, sends an SNMP trap, and increments a violation counter.

	Command or Action	Purpose
		* shutdown: Shuts down the interface, sends an SNMP trap, and increments a violation counter. The interface remains shut down until manually re-enabled or error-disabled recovery is configured.
Step 9	end **Example:** `Switch(config-if)# end`	Returns to priviledged EXEC mode
Step 10	copy running-config startup-config **Example:** `Switch# copy running-config startup-config`	Saves the running configuration to the startup configuration



CHAPTER 4

Configure UDLD

Feature History for UDLD

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature Name and Description	Supported Platform
Cisco IOS XE 17.18.1	UniDirectional Link Detection (UDLD) is a Layer 2 protocol that enables devices connected through fiber-optic or twisted-pair Ethernet cables to monitor the physical configuration of the cables.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

Overview of UniDirectional Link Detection

UniDirectional Link Detection (UDLD) is a Layer 2 protocol that:

- Enables devices connected through fiber-optic or twisted-pair Ethernet cables to monitor the physical configuration of the cables,
- Detects when a unidirectional link exists, and
- Alerts and disables the affected port upon detection.

UDLD prevents issues such as spanning-tree topology loops by identifying unidirectional links. Devices in the network must support UDLD to function effectively.

Restrictions for Configuring UniDirectional Link Detection

The following restrictions apply to configuring UniDirectional Link Detection (UDLD):

- A UDLD-capable port cannot detect a unidirectional link if it is connected to a UDLD-incapable port on another device.
- When configuring the UDLD mode (normal or aggressive), ensure that the same mode is configured on both sides of the link.
- Fast UDLD can only be enabled on up to 32 ports initially. If this limit is reached, Fast UDLD will not be enabled on additional ports, and an error message will be displayed on the console.

UDLD: hundredGigE<> not enabled for fast hello, maximum number of fast hello ports (32) reached.

- If UDLD is disabled when Fast UDLD is configured, the entire UDLD configuration is removed.
- The "Alert" option for UDLD ports is not supported.
- During manual reloads, switches with UDLD aggressive mode enabled (and lower-value timers) may enter an error-disabled state. This is because the system requires time to gracefully restart while CPU resources prioritize higher-priority processes.

Mitigation: To prevent this, either increase the hello interval value or shut down all active interfaces before reloading.

Fast UniDirectional Link Detection Capabilities

Fast UDLD capability is a feature of network devices that:

- Supports timers in the few-hundred milliseconds range,
- Enables subsecond unidirectional link detection, and
- Facilitates transition between different modes based on configuration.

Fast UniDirectional Link Detection (Fast UDLD) allows detection of unidirectional links in less than a second, with links status messages exchanged every 200 milliseconds. A link's transition from slow mode to fast mode occurs when both sides of a link have Fast UDLD configured and negotiated successfully to move into fast mode. Conversely, a transition from fast mode to slow mode happens when one of the configured ports has its Fast UDLD configuration removed. Fast UDLD supports a wide range of devices.

Modes of operation for UDLD

UDLD and Fast UDLD supports two modes of operation: normal (the default) and aggressive. In normal mode, UDLD can detect unidirectional links due to misconnected ports on fiber-optic connections. In aggressive mode, UDLD can also detect unidirectional links due to one-way traffic on fiber-optic and twisted-pair links and to misconnected ports on fiber-optic links.

In normal and aggressive modes, UDLD works with the Layer 1 mechanisms to learn 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 ports. When you enable both autonegotiation and UDLD, the 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 sent by a local device is received by its neighbor but traffic from the neighbor is not received by the local device.

Normal mode of operation for UDLD

A UDLD normal mode is an operational mode of the Unidirectional Link Detection (UDLD) protocol that:

- Detects unidirectional links caused by fiber misconnection when Layer 1 mechanisms fail,
- Leaves the logical link undetermined if traffic is unidirectional but Layer 1 mechanisms do not detect this condition, and
- Does not take action when autonegotiation detects a physical link issue caused by disconnection of one or more fiber strands.

Aggressive mode of operation for UDLD

UDLD aggressive mode is a protocol mode for Unidirectional Link Detection (UDLD) that:

- Detects unidirectional links using previously defined methods,
- Shuts down affected ports when a unidirectional link is detected, and
- Ensures bidirectional traffic flow on point-to-point links.

In aggressive mode, UDLD not only monitors the health of point-to-point links but also actively disables ports when unidirectional traffic patterns are detected. It is particularly effective in scenarios where device failures occur, such as:

- A port is unable to send or receive traffic,
- One port is down while the other remains up, or
- A fiber strand is disconnected within the cable.

UDLD differs from Layer 1 autonegotiation by operating at a higher layer, ensuring that traffic flows bidirectionally between the correct neighbors.

Example:

- On fiber-optic or twisted-pair links, one of the ports cannot send or receive traffic.
- One of the fiber strands in the cable is disconnected.

In a point-to-point link, UDLD hello packets can be considered as a "heart beat," ensuring the health of the link. A loss of the "heart beat" indicates a failure that requires the link to be disabled unless a bidirectional connection can be reestablished.

Unidirectional Link Detection Methods

UDLD operates by using two methods:

- Neighbor database maintenance
- Event-driven detection and echoing

How UDLD maintains neighbor databases

The process of UDLD neighbor database maintenance ensures that each device stays updated about its neighbors, maintains synchronized caches, and handles exceptional cases like errors or configuration changes.

The key components involved in the process are:

- UDLD-capable neighbors: Devices that exchange UDLD hello packets for neighbor discovery and synchronization.
- Hello packets: Periodic packets sent to exchange device information and keep caches updated.
- Cache entries: Stored information about neighbors with an expiration mechanism (age time).
- Configuration changes: Events such as port status changes, UDLD enable/disable actions, or device resets that trigger cache clearing.

The process involves the following stages:

1. Hello packet exchange and neighbor discovery
   - UDLD periodically sends hello packets (advertisements or probes) on active ports.
   - Devices receiving these packets cache the neighbor's details until the age time (hold time or time-to-live) expires.
2. Cache updates
   - If a new hello packet is received before an older cache entry ages, the device replaces the old entry with the new one.
3. Cache clearing during configuration changes
   - When a port is disabled, UDLD is disabled on a port, or the device is reset, all cache entries for affected ports are cleared.
   - UDLD sends a message to neighbors to inform them of the change, prompting the synchronization of their caches.
4. Handling multiple UDLD neighbors per interface
   - Interfaces do not support multiple UDLD neighbors.
   - If a UDLD PDU (protocol data unit) with multiple device IDs in the echo TLV (type, length, value) is received, the interface enters an error-disabled state to prevent ambiguity.

The process ensures accurate and synchronized neighbor database maintenance. However, interfaces receiving multiple device IDs will be placed in an error-disabled state to avoid misconfiguration.

Event-driven detection and echoing

UDLD relies on echoing as its detection operation. Whenever a UDLD device learns about a new neighbor or receives a resynchronization request from an out-of-sync neighbor, it restarts the detection window on its side of the connection and sends echo messages in reply. Because this behavior is the same on all UDLD neighbors, the sender of the echoes expects to receive an echo in reply.

If the detection window ends and no valid reply message are received, the link might shut down, depending on the UDLD mode. When UDLD is in normal mode, the link might be considered undetermined and might

not be shut down. When UDLD is in aggressive mode, the link is considered unidirectional, and the port is disabled.

UDLD reset options

If an interface becomes disabled by UDLD, you can use one of the following options to reset UDLD:

- The **udld reset** interface command.
- The **shutdown** interface configuration command followed by the **no shutdown** interface configuration command restarts the disabled port.
- The **no udld** { **aggressive** | **enable**} global configuration command followed by the **udld** { **aggressive** | **enable**} global configuration command reenables the disabled ports.
- The **no udld port** interface configuration command followed by the **udld port** [**aggressive**] interface configuration command reenables the disabled fiber-optic port.
- The **errdisable recovery cause udld** global configuration command enables the timer to automatically recover from the UDLD error-disabled state, and the **errdisable recovery interval** *interval* global configuration command specifies the time to recover from the UDLD error-disabled state.

The **udld port disable** command disables UDLD on fiber-optic LAN ports.

Note:

This command is only supported on fiber-optic LAN ports.

Default UDLD configuration

Default UniDirectional Link Detection (UDLD) configuration governs network communication integrity by detecting unidirectional links.

The default settings for UDLD configuration are as follows:

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

This set of configurations ensures that UDLD operates under default settings until explicitly enabled or modified to suit specific network environments.

Configure UniDirectional Link Detection

The following sections provide information about configuring UDLD:

Default UDLD configuration

Default UniDirectional Link Detection (UDLD) configuration governs network communication integrity by detecting unidirectional links.

The default settings for UDLD configuration are as follows:

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

This set of configurations ensures that UDLD operates under default settings until explicitly enabled or modified to suit specific network environments.

Enable UniDirectional Link Detection globally

Follow these steps either to enable UDLD in the aggressive or normal mode or to disable UDLD on a port.

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** `Device# enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal **Example:** `Device# configure terminal`	Enters global configuration mode
Step 3	interface interface-id **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Specifies the port to be enabled for UDLD, and enters interface configuration mode.
Step 4	**udld port** [**aggressive**] **Example:** `Device(config-if)# udld port aggressive`	UDLD is disabled by default. • **udld port**—Enables UDLD in normal mode on the specified port. • **udld port aggressive**(Optional) Enables UDLD in aggressive mode on the specified port.

	Command or Action	Purpose
Step 5	end **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode. UDLD is configured on the specified interface, ensuring link detection and error prevention according to the chosen mode.

Enabling UDLD on an interface

Follow these steps either to enable UDLD in the aggressive or normal mode or to disable UDLD on a port.

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** `Device# enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal **Example:** `Device# configure terminal`	Enters global configuration mode
Step 3	interface interface-id **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Specifies the port to be enabled for UDLD, and enters interface configuration mode.
Step 4	**udld port** [**aggressive**] **Example:** `Device(config-if)# udld port aggressive`	UDLD is disabled by default. • **udld port**—Enables UDLD in normal mode on the specified port. • **udld port aggressive**(Optional) Enables UDLD in aggressive mode on the specified port.
Step 5	end **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode. UDLD is configured on the specified interface, ensuring link detection and error prevention according to the chosen mode.

Enable Fast UniDirectional Link Detection on an interface

Follow these steps either to enable UDLD in the aggressive or normal mode or to disable UDLD on a port.

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** `Device# enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal **Example:** `Device# configure terminal`	Enters global configuration mode
Step 3	interface interface-id **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Specifies the port to be enabled for UDLD, and enters interface configuration mode.
Step 4	**udld port** [**aggressive**] **Example:** `Device(config-if)# udld port aggressive`	UDLD is disabled by default. • **udld port**—Enables UDLD in normal mode on the specified port. • **udld port aggressive**(Optional) Enables UDLD in aggressive mode on the specified port.
Step 5	end **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode. UDLD is configured on the specified interface, ensuring link detection and error prevention according to the chosen mode.

Enable Fast UDLD error reporting

Follow these steps either to enable UDLD in the aggressive or normal mode or to disable UDLD on a port.

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** `Device# enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal **Example:** `Device# configure terminal`	Enters global configuration mode

	Command or Action	Purpose
Step 3	interface interface-id **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Specifies the port to be enabled for UDLD, and enters interface configuration mode.
Step 4	**udld port** [**aggressive**] **Example:** `Device(config-if)# udld port aggressive`	UDLD is disabled by default. • **udld port**—Enables UDLD in normal mode on the specified port. • **udld port aggressive**(Optional) Enables UDLD in aggressive mode on the specified port.
Step 5	end **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode. UDLD is configured on the specified interface, ensuring link detection and error prevention according to the chosen mode.

Disable UDLD on fiber-optic LAN interfaces

Follow these steps either to enable UDLD in the aggressive or normal mode or to disable UDLD on a port.

Procedure

	Command or Action	Purpose
Step 1	enable **Example:** `Device# enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal **Example:** `Device# configure terminal`	Enters global configuration mode
Step 3	interface interface-id **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Specifies the port to be enabled for UDLD, and enters interface configuration mode.
Step 4	**udld port** [**aggressive**] **Example:** `Device(config-if)# udld port aggressive`	UDLD is disabled by default. • **udld port**—Enables UDLD in normal mode on the specified port. • **udld port aggressive**(Optional) Enables UDLD in aggressive mode on the specified port.

	Command or Action	Purpose
Step 5	end **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode. UDLD is configured on the specified interface, ensuring link detection and error prevention according to the chosen mode.

UDLD commands and their purposes

UDLD commands are used to monitor and verify the integrity of connections over network ports. Below is a list of UDLD commands with their specific purposes:

Command	Purpose
show udld [*interface-id* | **neighbors**]	Displays the UDLD status for the specified port or for all ports.
show udld fast-hello [*interface-id*]	Displays fast-hello information for the specified port or for all ports.

Console error messages for fast UDLD

Fast UDLD (Unidirectional Link Detection) generates error messages in the console when it detects a link failure. The type of error and the resulting action depend on whether **udld fast-hello error-reporting** is configured. Below are the possible error messages:

- If a unidirectional link is detected and the link is err-disabled by UDLD, the following message is displayed

  %UDLD-4-UDLD_PORT_DISABLED: UDLD disabled interface Hu1/0/10, unidirectional link detected

- If **udld fast-hello error-reporting** is configured, UDLD reports the link failure without err-disabling the affected port. The following message is displayed instead

  %UDLD-SP-4-UDLD_PORT_FAILURE: UDLD failure reported per user request, interface HU1/0/10, fast udld unidirectional link detected

- To clear the UDLD port state in either case, use the **udld reset** command. This command resets the port and resolves the error condition if the issue is fixed.