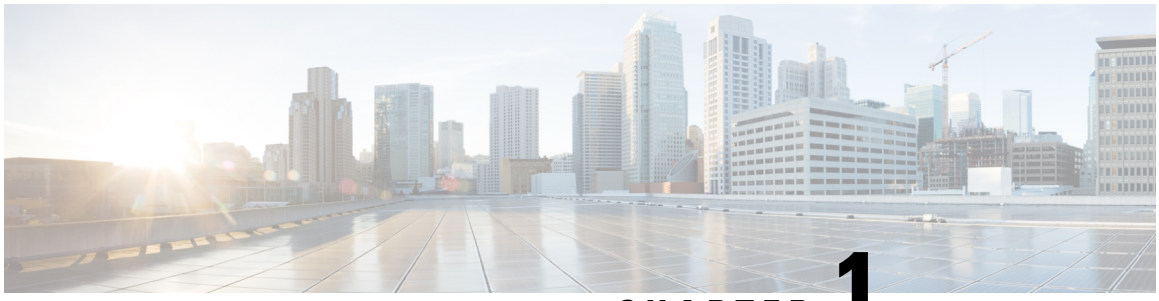
LAN Switching Configuration Guide, Cisco IOS XE Everest 3.18SP (Cisco NCS 4200 Series)

First Published: 2016-07-29

Last Modified: 2021-04-07

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

CONTENTS



CHAPTER 1

UniDirectional Link Detection (UDLD) Protocol

The UniDirectional Link Detection protocol is a Layer 2 protocol that detects and disables one-way connections before they create undesired situation such as Spanning Tree loops.

Information About the UDLD Protocol

UDLD Overview

The Cisco-proprietary UDLD protocol allows the devices connected through fiber optic or copper (for example, Category 5 cabling) Ethernet cables that are connected to the LAN ports to monitor the physical configuration of the cables and detect whether a unidirectional link exists. When a unidirectional link is detected, the UDLD shuts down the affected LAN port and alerts the corresponding user, because unidirectional links 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. In Layer 1, auto negotiation takes care of physical signaling and fault detection. UDLD performs tasks that auto negotiation cannot perform, such as detecting the identities of neighbors and shutting down misconnected LAN ports. When you enable both auto negotiation 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 the traffic transmitted by a local device over a link is received by a 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, the link does not stay up as long as the auto negotiation is active. In such a scenario, the logical link is undetermined, and the UDLD does not take any action. If both the fibers are working normally in Layer 1, the UDLD in Layer 2 determines whether those fibers are connected correctly and whether the traffic is flowing bidirectionally between the correct neighbors. This check cannot be performed by auto negotiation because auto negotiation operates in Layer 1.

The router periodically transmits the UDLD packets to the neighbor devices on LAN ports where UDLD is enabled. If the packets are echoed back within a specific timeframe and they are lacking 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 for the protocol to successfully identify and disable the unidirectional links.

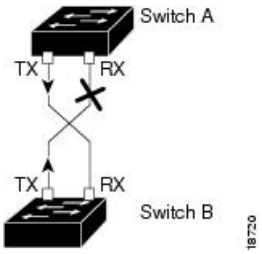
UDLD detects and disables unidirectional links on Ethernet fiber and copper interfaces due to miswiring or malfunctioning of the interfaces.

Note UDLD is disabled by default on all ports to avoid sending unnecessary traffic.

To configure fibre-optic interfaces, enable the **udld** command at the global level. For copper interfaces, enable the **udld port** command at the interface level.

The figure displays the UDLD mechanism.

Figure 1: Unidirectional Link



UDLD supports two modes of operation: normal (the default) and aggressive. In normal mode, UDLD can detect unidirectional links due to misconnected interfaces 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 interfaces on fiber-optic links.

UDLD Normal Mode

In normal mode, UDLD detects the unidirectional link when fiber strands in a fiber-optic interface are misconnected and the Layer 1 mechanisms do not detect this misconnection. If the interfaces are connected correctly, but the traffic is one way, UDLD does not detect the unidirectional link because the Layer 1 mechanism, which is supposed to detect this condition, does not do so. In case, the logical link is considered undetermined, and UDLD does not disable the interface. If one of the fiber strands in a pair is disconnected and autonegotiation is active, the link does not stay up because the Layer 1 mechanisms did not detect a physical problem with the link. In this case, UDLD does not take any action, and the logical link is considered undetermined.

UDLD Aggressive Mode

The UDLD aggressive mode is configured only on the point-to-point link between the network devices that support the UDLD aggressive mode. With UDLD aggressive mode enabled, a port on a bidirectional link that has a UDLD neighbor relationship established stops receiving the UDLD packets. The UDLD tries to re-establish the connection with the neighbor; the port is disabled after eight failed retries.

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 the UDLD aggressive mode is enabled, the UDLD can error disable the ports on the link to prevent the traffic from being discarded under the following scenarios:

- One side of a link has a port (either Tx and Rx) stuck.

- One side of a link remains up while the other side of the link has gone down.

UDLD Functions

UDLD performs the following functions

- Sends a probe packet on every active interface on which UDLD is configured to keep each device informed about its neighbors.
- Learns about the neighbors and keeps the updated neighbor information in a cache table
- Sends several echo messages whenever it detects a new neighbor sending UDLD packets or whenever a neighbor requests a resynchronization of the caches
- Shuts down the affected port and notifies the user when one-way connection is detected. Devices on both ends of the link must support UDLD in order for the protocol to successfully identify and disable unidirectional links
- Reestablishes the connection with the neighbor when a port on a bidirectional link stops receiving UDLD packets if aggressive mode is enabled. After eight failed retries, the port goes into disabled state

Detecting Unidirectional Links

UDLD operates by using two mechanisms:

Neighbor database maintenance

UDLD learns about other UDLD-capable neighbors by periodically sending a hello packet (also called an advertisement or probe) on every active interface to keep each device informed about its neighbors. When the switch receives a hello message, it caches the information until the age time (hold time or time-to-live) expires. If the switch receives a new hello message before an older cache entry ages, the switch replaces the older entry with the new one. Whenever an interface is disabled and UDLD is running, whenever UDLD is disabled on an interface, or whenever the switch is reset, UDLD clears all existing cache entries for the interfaces affected by the configuration change. UDLD sends at least one message to inform the neighbors to flush the part of their caches affected by the status change. The message is intended to keep the caches synchronized.

Event-driven detection and echoing

UDLD relies on echoing as its detection mechanism. 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 is 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 interface is shut down. If UDLD in normal mode is in the advertisement or in the detection phase and all the neighbor cache entries are aged out, UDLD restarts the link-up sequence to resynchronize with any potentially out-of-sync neighbors. If you enable aggressive mode when all the neighbors of a port have aged out either in the advertisement or in the detection phase, UDLD restarts the link-up sequence to resynchronize with any potentially out-of-sync neighbor. UDLD shuts down the port if, after the fast train of messages, the link state is still undetermined.

How to Configure UDLD Protocol

Enabling UDLD Protocol

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **udld** {**enable** | **aggressive**}
4. **end**

DETAILED STEPS

	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	**udld** {**enable**	**aggressive**} **Example:** `Router(config)# udld enable`	Enables UDLD protocol on the router.
Step 4	**end** **Example:** `Device(config-erp-profile)# end`	Returns to user EXEC mode.

Enabling UDLD Protocol at Interface Level

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **udld port** [**aggressive**]
5. **end**

DETAILED STEPS

	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:** `Router(config)# interface gigabitethernet0/0/1`	Enter interface configuration mode. Valid interfaces are physical ports.
Step 4	**udld port** [**aggressive**] **Example:** `Router(config)# udld port aggressive`	Enables UDLD on a specific port. Enter the aggressive keyword to enable the aggressive mode. On a fiber-optic LAN port, this command overrides the udld enable global configuration command setting. Use the **no** form of this command to disable the UDLD on a non fiber-optic LAN port.
Step 5	**end** **Example:** `Device(config-erp-profile)# end`	Returns to user EXEC mode.

Enabling UDLD Probe Message Interval

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **udld message time** *interval*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**udld message time** *interval* **Example:** `Router(config)# udld message time 90`	Set the time in seconds between UDLD probe messages. The valid range is from 7 to 90 seconds. The default is 15 seconds
Step 4	**end** **Example:** `Device(config-erp-profile)# end`	Returns to user EXEC mode.

Recovering the UDLD Protocol

UDLD recovery when enabled, attempts to bring an UDLD error-disabled port out of reset. Tthe default recovery timer is 300 seconds.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **udld recovery** *inteval*
4. **end**

DETAILED STEPS

	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	**udld recovery** *inteval* **Example:** `Router(config)# udld recovery`	Enables UDLD recovery on the router. • *inteval*—Sets the recovery time interval. The valid range is from 30 to 86400 seconds. The default value is 300 seconds.

	Command or Action	Purpose
Step 4	**end** **Example:** `Device(config-erp-profile)# end`	Returns to user EXEC mode.

Resetting Ports

SUMMARY STEPS

1. **enable**
2. **udld reset**
3. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**udld reset** **Example:** `Router# udld reset`	Resets ports that are shut down by UDLD.
Step 3	**end** **Example:** `Device(config-erp-profile)# end`	Returns to user EXEC mode.

Configuration Examples

Example: Configuring UDLD Protocol

This example shows UDLD on the router.

```
show running-config | i udld
udld enable
udld message time 7
udld recovery
udld recovery interval 30
```

Verifying UDLD Protocol

Example: Verifying UDLD Protocol

Use the **show udld** command to view the status of the UDLD protocol on the ports.

- This example shows UDLD protocol on all ports the router.

```
Router# show udld
 Interface Te0/0/0
---
Port enable administrative configuration setting: Follows device default
Port enable operational state: Enabled
Current bidirectional state: Bidirectional
Current operational state: Advertisement - Single neighbor detected
Message interval: 15
Time out interval: 5

    Entry 1
    ---
    Expiration time: 40
    Cache Device index: 1
    Current neighbor state: Bidirectional
    Device ID: FOX1736P0JP
    Port ID: Te0/1/0
    Neighbor echo 1 device: FOX1709P3D0
    Neighbor echo 1 port: Te0/0/0

    Message interval: 15
    Time out interval: 5
    CDP Device name: RSP1B

Interface Gi0/2/0
---
Port enable administrative configuration setting: Follows device default
Port enable operational state: Enabled
Current bidirectional state: Bidirectional
Current operational state: Advertisement - Single neighbor detected
Message interval: 15
Time out interval: 5

    Entry 1
    ---
    Expiration time: 33
    Cache Device index: 1
    Current neighbor state: Bidirectional
    Device ID: FOC1528V27K
    Port ID: Gi0/2
    Neighbor echo 1 device: FOX1709P3D0
    Neighbor echo 1 port: Gi0/2/0

    Message interval: 15
    Time out interval: 5
    CDP Device name: RSP1A

Interface Gi0/2/1
---
Port enable administrative configuration setting: Follows device default
Port enable operational state: Enabled
Current bidirectional state: Bidirectional
```

```
Current operational state: Advertisement - Single neighbor detected
Message interval: 15
Time out interval: 5

    Entry 1
    ---
    Expiration time: 33
    Cache Device index: 1
    Current neighbor state: Bidirectional
    Device ID: FOC1639V1Z4
    Port ID: Gi0/4
    Neighbor echo 1 device: FOX1709P3D0
    Neighbor echo 1 port: Gi0/2/1

    Message interval: 15
    Time out interval: 5
    CDP Device name: RSP1A

Interface Gi0/2/2
---
Port enable administrative configuration setting: Follows device default
Port enable operational state: Enabled
Current bidirectional state: Unknown
Current operational state: Advertisement
Message interval: 15
Time out interval: 5
No neighbor cache information stored

Interface Gi0/2/3
---
Port enable administrative configuration setting: Follows device default
Port enable operational state: Enabled
Current bidirectional state: Unknown
Current operational state: Link down
Message interval: 15
Time out interval: 5
No neighbor cache information stored

Interface Gi0/2/4
---
Port enable administrative configuration setting: Follows device default
Port enable operational state: Disabled
Current bidirectional state: Unknown

Interface Gi0/2/5
---
Port enable administrative configuration setting: Disabled
Port enable operational state: Disabled
Current bidirectional state: Unknown

Interface Gi0/2/6
---
Port enable administrative configuration setting: Disabled
Port enable operational state: Disabled
Current bidirectional state: Unknown
.
.
.
```

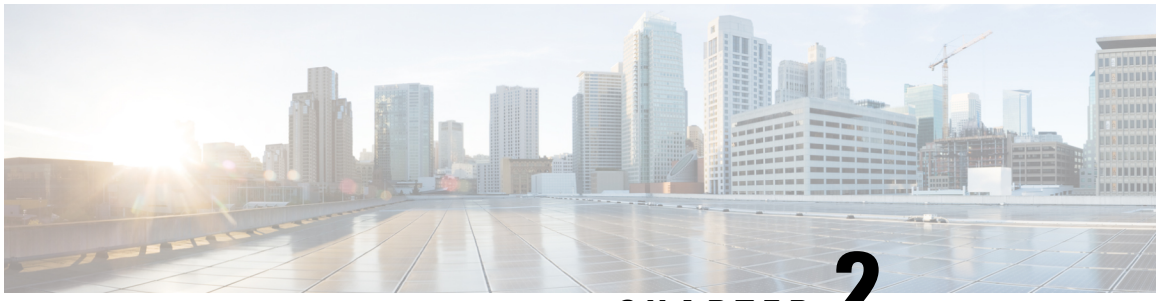
- This example shows UDLD protocol on the Ten Gigabit Ethernet interface.

```
Router# show udld tengigabitethernet 0/0/0

Interface Te0/0/0
---
```

```
Port enable administrative configuration setting: Follows device default
Port enable operational state: Enabled
Current bidirectional state: Bidirectional
Current operational state: Advertisement - Single neighbor detected
Message interval: 15
Time out interval: 5

    Entry 1
    ---
    Expiration time: 43
    Cache Device index: 1
    Current neighbor state: Bidirectional
    Device ID: FOX1736P0JP
    Port ID: Te0/1/0
    Neighbor echo 1 device: FOX1709P3D0
    Neighbor echo 1 port: Te0/0/0

    Message interval: 15
    Time out interval: 5
    CDP Device name: RSP1B

Router# show running-config | i udld
udld enable
udld message time 15
udld recovery
udld recovery interval 30
```

- This example shows the UDLD protocol neighbors.

```
Router# show udld neighbors

Port          Device Name     Device ID     Port ID        Neighbor State
------------ ------------- ------------ ------------ ----------------
Te0/0/0         FOX1736P0JP     1            Te0/1/0         Bidirectional
Gi0/2/0         FOC1528V27K     1            Gi0/2           Bidirectional
Gi0/2/1         FOC1639V1Z4     1            Gi0/4           Bidirectional
```



CHAPTER 2

ITU-T G.8032 Ethernet Ring Protection Switching

The ITU-T G.8032 Ethernet Ring Protection Switching feature implements protection switching mechanisms for Ethernet layer ring topologies. This feature uses the G.8032 Ethernet Ring Protection (ERP) protocol, defined in ITU-T G.8032, to provide protection for Ethernet traffic in a ring topology, while ensuring that no loops are within the ring at the Ethernet layer. The loops are prevented by blocking traffic on either a predetermined link or a failed link.

Prerequisites for Configuring ITU-T G.8032 Ethernet Ring Protection Switching

- The Ethernet Flow Points (EFPs) and Trunk Ethernet Flow Points (TEFPs) must be configured.

About ITU-T G.8032 Ethernet Ring Protection Switching

Ring Protection Links

An Ethernet ring consists of multiple Ethernet ring nodes. Each Ethernet ring node is connected to adjacent Ethernet ring nodes using two independent ring links. A ring link prohibits formation of loops that affect the network. The Ethernet ring uses a specific link to protect the entire Ethernet ring. This specific link is called the Ring Protection Link (RPL). A ring link is bound by two adjacent Ethernet ring nodes and a port for a ring link (also known as a ring port). There must be at least two Ethernet ring nodes in an Ethernet ring.

ITU-T G.8032 Ethernet Ring Protection Switching Functionality

The Ethernet ring protection functionality includes the following:

- Loop avoidance

- The use of learning, forwarding, and Filtering Database (FDB) mechanisms

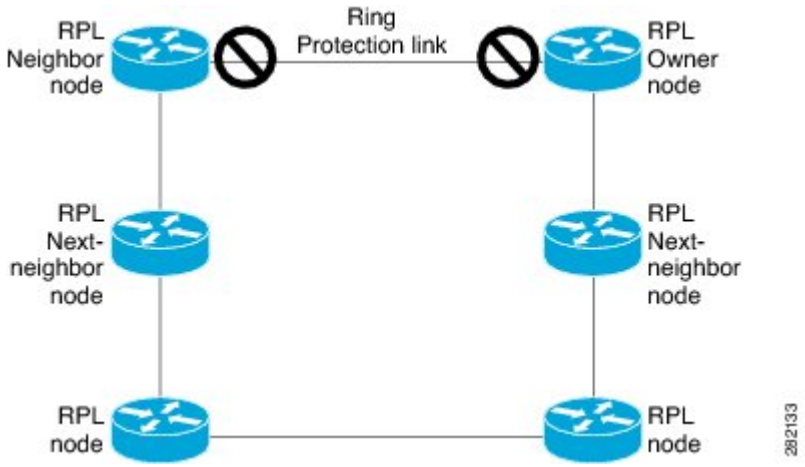
Loop avoidance in an Ethernet ring is achieved by ensuring that, at any time, traffic flows on all but the Ring Protection Link (RPL).

The following is a list of RPL types (or RPL nodes) and their functions:

- RPL owner—Responsible for blocking traffic over the RPL so that no loops are formed in the Ethernet traffic. There can be only one RPL owner in a ring.
- RPL neighbor node—An Ethernet ring node adjacent to the RPL. It is responsible for blocking its end of the RPL under normal conditions. This node type is optional and prevents RPL usage when protected.
- RPL next-neighbor node—Next-neighbor node is an Ethernet ring node adjacent to an RPL owner node or RPL neighbor node. It is mainly used for FDB flush optimization on the ring. This node is also optional.

The following figure illustrates the G.8032 Ethernet ring topology.

Figure 2: G.8032 Ethernet Ring Topology



R-APS Control Messages

Nodes on the ring use control messages called Ring Automatic Protection Switching (R-APS) messages to coordinate the activities of switching the ring protection link (RPL) on and off. Any failure along the ring triggers a R-APS Signal Failure (R-APS SF) message in both directions of the nodes adjacent to the failed link, after the nodes have blocked the port facing the failed link. On obtaining this message, the RPL owner unblocks the RPL port.

Note A single link failure in the ring ensures a loop-free topology.

CFM Protocols and Link Failures

Connectivity Fault Management (CFM) and line status messages are used to detect ring link and node failure. During the recovery phase, when the failed link is restored, the nodes adjacent to the restored link send Ring Automatic Protection Switching (R-APS) No Request (R-APS NR) messages. On obtaining this message, the

ring protection link (RPL) owner blocks the RPL port and sends R-APS NR and R-APS RPL (R-APS NR, RB) messages. These messages cause all other nodes, other than the RPL owner in the ring, to unblock all blocked ports. The Ethernet Ring Protection (ERP) protocol works for both unidirectional failure and multiple link failure scenarios in a ring topology.

Note The G.8032 Ethernet Ring Protection (ERP) protocol uses CFM Continuity Check Messages (CCMs) at an interval of 3.3 milliseconds (ms). At this interval (which is supported only on selected platforms), SONET-like switching time performance and loop-free traffic can be achieved.

G.8032 Ring-Supported Commands and Functionality

A G.8032 ring supports these basic operator administrative commands:

- Force switch (FS)—Allows the operator to forcefully block a particular ring port. Note the following points about Force Switch commands:
  - Effective even if there is an existing SF condition
  - Multiple FS commands for ring are supported
  - May be used to allow immediate maintenance operations
- Manual switch (MS)—Allows the operator to manually block a particular ring port. Note the following points about MS commands:
  - Ineffective in an existing FS or signal failure (SF) condition
  - Overridden by new FS or SF conditions
  - When multiple MS commands are executed more than once on the same device, all MS commands are cancelled.

    When multiple MS commands are executed on different devices in the ring, for the same instance, then the command executed on the second device is rejected.
- Clear—Cancels an existing FS or MS command on the ring port. The Clear command is used at the ring protection link (RPL) owner to clear a nonrevertive mode condition.

A G.8032 ring can support multiple instances. An instance is a logical ring running over a physical ring. Such instances are used for various reasons, such as load-balancing VLANs over a ring. For example, odd-numbered VLANs may go in one direction of the ring, and even-numbered VLANs may go in the other direction. Specific VLANs can be configured under only one instance. They cannot overlap multiple instances. Otherwise, data traffic or Ring Automatic Protection Switching (R-APS) messages may cross logical rings, which is not desirable.

Note G.8032 Ethernet Ring Protection Switching Version 1 and Version 2 are supported.

G.8032 ERP Timers

The G.8032 Ethernet Ring Protection (ERP) protocol specifies the use of different timers to avoid race conditions and unnecessary switching operations:

- Delay timers—Used by the Ring Protection Link (RPL) owner to verify that the network has stabilized before blocking the RPL. Note the following points about delay timers.
  - After a signal failure (SF) condition, a Wait-to-Restore (WTR) timer is used to verify that the SF is not intermittent.
  - The WTR timer can be configured by the operator. The default time interval is 5 minutes; the time interval ranges from 1 to 12 minutes.
  - After a force switch (FS) or a manual switch (MS) command is issued, a Wait-to-Block (WTB) timer is used to verify that no background condition exists.

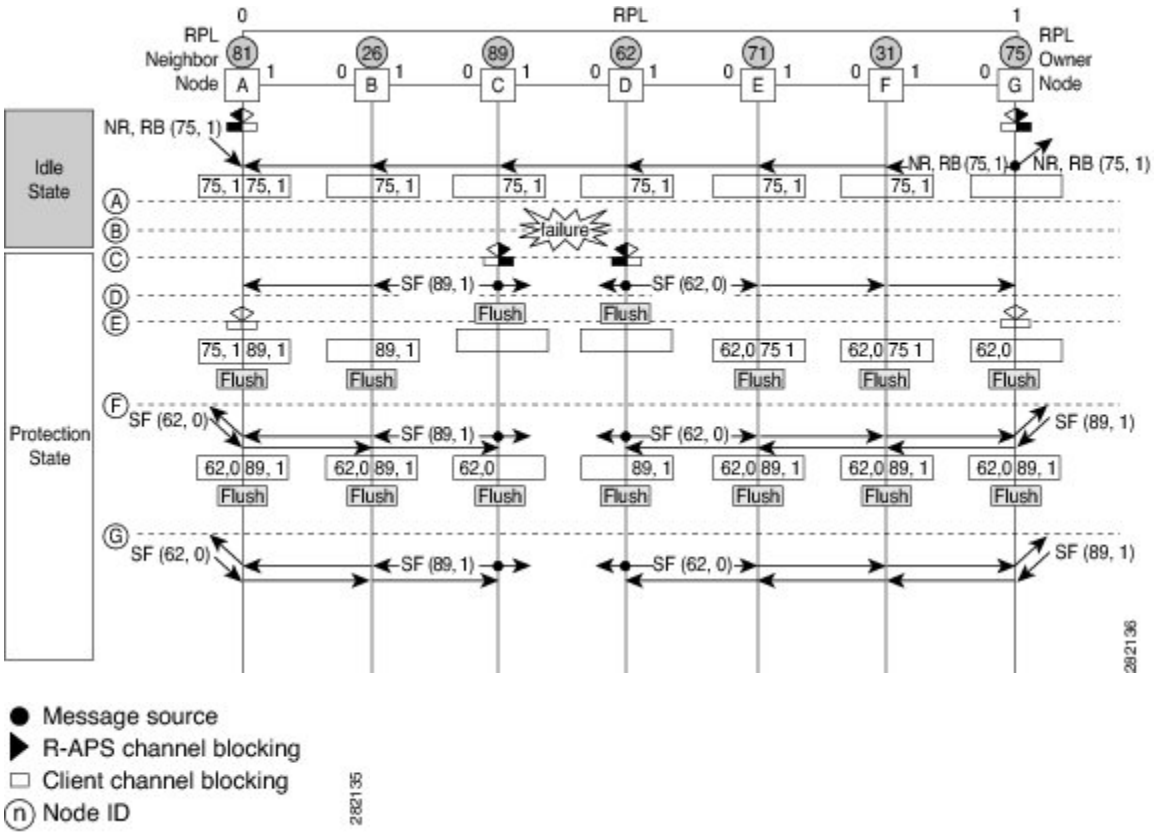
Note The WTB timer interval may be shorter than the WTR timer interval.

- Guard timer—Used by all nodes when changing state; the guard timer blocks latent outdated messages from causing unnecessary state changes. The guard timer can be configured. The default time interval is 500 ms; the time interval ranges from 10 to 2000 ms.
- The recommended Guard Timer for Cisco RSP2 and RSP3 routers is 500 ms.
- Hold-off timers—Used by the underlying Ethernet layer to filter out intermittent link faults. The hold-off timer can be configured. The default time interval is 0 seconds; the time interval ranges from 0 to 10 seconds. Faults are reported to the ring protection mechanism only if this timer expires.

Protection Switching Functionality in a Single Link Failure and Recovery

The following figure illustrates protection switching functionality in a single-link failure.

Figure 3: G.8032 Ethernet Ring Protection Switching in a Single-Link Failure



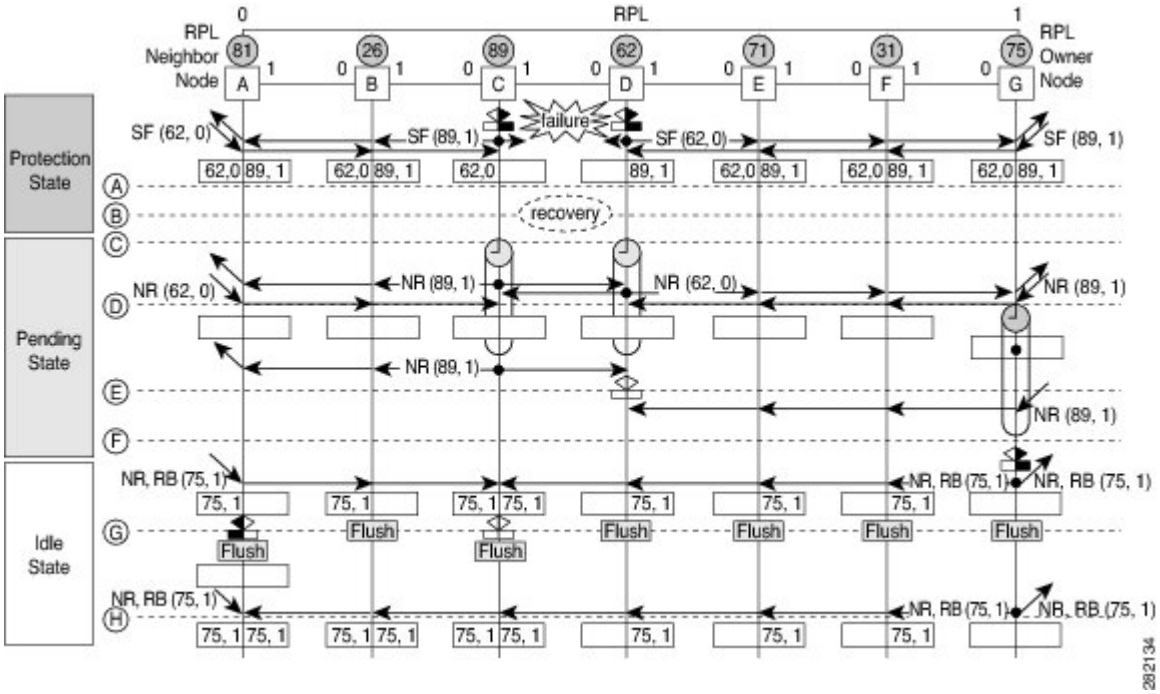
The figure represents an Ethernet ring topology consisting of seven Ethernet ring nodes. The ring protection link (RPL) is the ring link between Ethernet ring nodes A and G. In this topology, both ends of the RPL are blocked. Ethernet ring node G is the RPL owner node, and Ethernet ring node A is the RPL neighbor node.

The following sequence describes the steps followed in the single-link failure:

1. A link operates in the normal condition.
2. A failure occurs.
3. Ethernet ring nodes C and D detect a local signal failure (SF) condition and after the hold-off time interval, block the failed ring port and perform the FDB flush.
4. Ethernet ring nodes C and D start sending Ring Automatic Protection Switching (R-APS) SF messages periodically along with the (node ID and bidirectional path-protected ring (BPR) identifier pair) on both ring ports while the SF condition persists.
5. All Ethernet ring nodes receiving an R-APS SF message perform the FDB flush. When the RPL owner node G and RPL neighbor node A receive an R-APS SF message, the Ethernet ring node unblocks its end of the RPL and performs the FDB flush.
6. All Ethernet ring nodes receiving a second R-APS SF message perform the FDB flush again; the additional FDB flush is because of the node ID and BPR-based configuration.
7. R-APS SF messages are detected on the Ethernet Ring indicating a stable SF condition. Further R-APS SF messages trigger no further action.

The following figure illustrates the steps taken in a revertive operation in a single-link failure.

Figure 4: Single-Link Failure Recovery (Revertive Operation)



The following sequence describes the steps followed in the single-link failure revertive (recovery) operation:

1. A link operates in the stable SF condition.
2. Recovery of link failure occurs.
3. Ethernet ring nodes C and D detect clearing of the SF condition, start the guard timer, and initiate periodic transmission of the R-APS No Request (NR) messages on both ring ports. (The guard timer prevents the reception of R-APS messages.)
4. When the Ethernet ring nodes receive an R-APS NR message, the node ID and BPR identifier pair of a receiving ring port is deleted and the RPL owner node starts the Wait-to-Restore (WTR) timer.
5. When the guard timer expires on Ethernet ring nodes C and D, the nodes may accept the new R-APS messages, if any. Ethernet ring node D receives an R-APS NR message with a higher node ID from Ethernet ring node C, and unblocks its nonfailed ring port.
6. When the WTR timer expires, the RPL owner node blocks its end of the RPL, sends R-APS (NR or route blocked [RB]) message with the (node ID and BPR identifier pair), and performs the FDB flush.
7. When Ethernet ring node C receives an R-APS (NR or RB) message, the node removes the block on its blocked ring ports, and stops sending R-APS NR messages. On the other hand, when the RPL neighbor node A receives an R-APS NR or RB message, the node blocks its end of the RPL. In addition, Ethernet ring nodes A to F perform the FDB flush when receiving an RAPS NR or RB message because of the node ID and BPR-based configuration.

Ethernet Flow Points

An Ethernet flow point (EFP) is a forwarding decision point in the provider edge (PE) router, which gives network designers flexibility to make many Layer 2 flow decisions within the interface. Many EFPs can be configured on a single physical port. (The number varies from one device to another.) EFPs are the logical demarcation points of an Ethernet virtual connection (EVC) on an interface. An EVC that uses two or more user network interfaces (UNIs) requires an EFP on the associated ingress and egress interfaces of every device that the EVC passes through.

EFPs can be configured on any Layer 2 traffic port; however, they are usually configured on UNI ports. The following parameters (matching criteria) can be configured on the EFP:

- Frames of a specific VLAN, a VLAN range, or a list of VLANs (100-150 or 100,103,110)
- Frames with no tags (untagged)
- Frames with identical double-tags (VLAN tags) as specified
- Frames with identical Class of Service (CoS) values

A frame passes each configured match criterion until the correct matching point is found. If a frame does not fit any of the matching criteria, it is dropped. Default criteria can be configured to avoid dropping frames.

You can configure a new type of TEFP called TEFP with encapsulation from bridge domain (BD). All the BDs configured on the switch are part of the VLAN list of the encapsulated TEFP. The TEFP is encapsulated using the **encapsulation dot1q from-bd** command. The feature brings about the following interaction between the Ethernet-EFP and Layer2-bridge domain components:

- If BDs exist in the system and a TEFP with encapsulation from bridge domain is created, then all the BDs get added to the VLAN list of TEFP with encapsulation from bridge domain.
- If TEFP with encapsulation from bridge domain exists in the system and a new BD is created, then the BD is added to the VLAN list of all the TEFP with encapsulation from bridge domain in the system.
- If TEFP with encapsulation from bridge domain exists in the system and a BD gets deleted, and if the deleted BD is not part of an existing TEFP or EFP then it gets deleted from all the TEFP with encapsulation from bridge domain in the system.

The following types of commands can be used in an EFP:

- Rewrite commands—In each EFP, VLAN tag management can be specified with the following actions:
  - Pop—1) pops out a tag; 2) pops out two tags
  - Push— pushes in a tag
  - Translate—1 to 1) changes a tag value; 1 to 2) pops one tag and pushes two tags; 2 to 1) pops two tags and pushes one tag; 2 to 2) changes the value for two tags
- Forwarding commands—Each EFP specifies the forwarding command for the frames that enter the EFP. Only one forwarding command can be configured per EFP. The forwarding options are as follows:
  - Layer 2 point-to-point forwarding to a pseudowire tunnel
  - Multipoint bridge forwarding to a bridge domain entity
  - Local switch-to-switch forwarding between two different interfaces

- Feature commands—In each EFP, the QoS features or parameters can be changed and the ACL can be updated.

Service Instances and Associated EFPs

Configuring a service instance on a Layer 2 port creates a pseudoport or EFP on which you configure EVC features. Each service instance has a unique number per interface, but you can use the same number on different interfaces because service instances on different ports are not related.

An EFP classifies frames from the same physical port to one of the multiple service instances associated with that port, based on user-defined criteria. Each EFP can be associated with different forwarding actions and behavior.

When an EFP is created, the initial state is UP. The state changes to DOWN under the following circumstances:

- The EFP is explicitly shut down by a user.
- The main interface to which the EFP is associated is down or removed.
- If the EFP belongs to a bridge domain, the bridge domain is down.
- The EFP is forced down as an error-prevention measure of certain features.

Use the **service instance ethernet** interface configuration command to create an EFP on a Layer 2 interface and to enter service instance configuration mode. Service instance configuration mode is used to configure all management and control data plane attributes and parameters that apply to the service instance on a per-interface basis. The service instance number is the EFP identifier.

After the device enters service instance configuration mode, you can configure these options:

- default--Sets a command to its defaults
- description--Adds a service instance-specific description
- encapsulation--Configures Ethernet frame match criteria
- exit--Exits from service instance configuration mode
- no--Negates a command or sets its defaults
- shutdown--Takes the service instance out of service

Restrictions for Configuring ITU-T G.8032 Ethernet Ring Protection Switching

- G.8032 is supported only on EFP bridgedomains on the physical interface and port-channel interface.

Note G.8032 is supported only on TEFP. Port-channel is not supported.

- G.8032 is supported only on EFP with dot1q, dot1ad, QinQ, or dot1ad-dot1Q encapsulation type.

Note G.8032 is supported only on TEFP with dot1q.

- G.8032 is not supported on xconnect interface.
- G.8032 does not support more than two ERP instances per ring.
- Link flap occurs while configuring the inclusion or exclusion VLAN list.
- Admin shut down is highly recommended before making any changes in Connectivity Fault Management (CFM) configuration.
- The **efd notify** command must be used under CFM configuration to notify G.8032 of failures, if any.
- BFD IPv4 and IPv6 Single Hop is supported. BFD Echo Mode is not supported.
- Modification of APS VLAN will not be effective until you delete and reconfigure the G8032 ring configuration.

How to Configure ITU-T G.8032 Ethernet Ring Protection Switching

Configuring the Ethernet Ring Profile

To configure the Ethernet ring profile, complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet ring g8032 profile** *profile-name*
4. **timer** {**guard** *seconds* | **hold-off** *seconds* | **wtr** *minutes*}
5. **non-revertive**
6. **end**

DETAILED STEPS

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

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**ethernet ring g8032 profile** *profile-name* **Example:** `Device(config)# ethernet ring g8032 profile profile1`	Creates the Ethernet ring profile and enters Ethernet ring profile configuration mode.
Step 4	**timer** {**guard** *seconds* | **hold-off** *seconds* | **wtr** *minutes*} **Example:** `Device(config-erp-profile)# timer hold-off 5`	Specifies the time interval for the guard, hold-off, and Wait-to-Restore (WTR) timers.
Step 5	**non-revertive** **Example:** `Device(config-erp-profile)# non-revertive`	Specifies a nonrevertive Ethernet ring instance. • By default, Ethernet ring instances are revertive.
Step 6	**end** **Example:** `Device(config-erp-profile)# end`	Returns to user EXEC mode.

Configuring Ethernet CFM MEPs

Configuring Ethernet Connectivity Fault Management (CFM) maintenance endpoints (MEPs) is optional although recommended for fast failure detection and CFM monitoring. When CFM monitoring is configured, note the following points:

- Static remote MEP (RMEP) checking should be enabled.
- The MEPs should be configured to enable Ethernet fault detection.

For information about configuring Ethernet Connectivity Fault Management (CFM) maintenance endpoints (MEPs), see the "Configuring Ethernet Connectivity Fault Management in a Service Provider Network" module of the *Carrier Ethernet Configuration Guide*.

Enabling Ethernet Fault Detection for a Service

To enable Ethernet Fault Detection (EFD) for a service to achieve fast convergence, complete the following steps

Note Link protection is not supported on the RSP3 Module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm global**
4. **link-protection enable**
5. **link-protection group management vlan** *vlan-id*
6. **link-protection group** *group-number* **pccm vlan** *vlan-id*
7. **ethernet cfm domain***domain-name* **level** *level-id* [**direction outward**]
8. **service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]]
9. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
10. **efd notify g8032**
11. **end**

DETAILED STEPS

	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	**ethernet cfm global** **Example:** `Device(config)# ethernet cfm global`	Enables Ethernet CFM globally.
Step 4	**link-protection enable** **Example:** `Device(config)# link-protection enable`	Enables link protection globally on the router.
Step 5	**link-protection group management vlan** *vlan-id* **Example:** `Device(config)# link-protection group management vlan 51`	Defines the management VLAN used for link protection.
Step 6	**link-protection group** *group-number* **pccm vlan** *vlan-id* **Example:** `Device(config)# link-protection group 2 pccm vlan 16`	Specifies an ODU-to-ODU continuity check message (P-CCM) VLAN.

	Command or Action	Purpose
Step 7	**ethernet cfm domain***domain-name* **level** *level-id* [**direction outward**] **Example:** `Device(config)# ethernet cfm domain G8032 level 4`	Configures the CFM domain for ODU 1 and enters Ethernet CFM configuration mode.
Step 8	**service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]] **Example:** `Device(config-ecfm)# service 8032_service evc 8032-evc vlan 1001 direction down`	Defines a maintenance association for ODU 1 and enters Ethernet CFM service instance configuration mode.
Step 9	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Device(config-ecfm-srv)# continuity-check interval 3.3ms`	Enables the transmission of continuity check messages (CCMs).
Step 10	**efd notify g8032** **Example:** `Device(config-ecfm-srv)# efd notify g8032`	Enables CFM to notify registered protocols when a defect is detected or cleared, which matches the current fault alarm priority.
Step 11	**end** **Example:** `Device(config-ecfm-srv)# end`	Returns to user EXEC mode.

Configuring the Ethernet Protection Ring

To configure the Ethernet Protection Ring (EPR), complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet ring g8032** *ring-name*
4. **port0 interface** *type number*
5. **monitor service instance** *instance-id*
6. **exit**
7. **port1** {**interface***type number* | **none**}
8. **monitor service instance** *instance-id*
9. **exit**

10. **exclusion-list vlan-ids** *vlan-id*
11. **open-ring**
12. **instance** *instance-id*
13. **description** *descriptive-name*
14. **profile** *profile-name*
15. **rpl** {**port0** | **port1**} {**owner** | **neighbor** | **next-neighbor** }
16. **inclusion-list vlan-ids** *vlan-id*
17. **aps-channel**
18. **level** *level-value*
19. **port0 service instance** *instance-id*
20. **port1 service instance** {*instance-id* | **none** }
21. **end**

DETAILED STEPS

	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	**ethernet ring g8032** *ring-name* **Example:** `Device(config)# ethernet ring g8032 ring1`	Specifies the Ethernet ring and enters Ethernet ring port configuration mode.
Step 4	**port0 interface** *type number* **Example:** `Device(config-erp-ring)# port0 interface gigabitethernet 0/1/0`	Connects port0 of the local node of the interface to the Ethernet ring and enters Ethernet ring protection mode.
Step 5	**monitor service instance** *instance-id* **Example:** `Device(config-erp-ring-port)# monitor service instance 1`	Assigns the Ethernet service instance to monitor the ring port (port0) and detect ring failures.
Step 6	**exit** **Example:** `Device(config-erp-ring-port)# exit`	Exits Ethernet ring port configuration mode.

	Command or Action	Purpose
Step 7	**port1** {**interface***type* *number* | **none**} **Example:** `Device(config-erp-ring)# port1 interface gigabitethernet 0/1/1`	Connects port1 of the local node of the interface to the Ethernet ring and enters Ethernet ring protection mode.
Step 8	**monitor service instance** *instance-id* **Example:** `Device(config-erp-ring-port)# monitor service instance 2`	Assigns the Ethernet service instance to monitor the ring port (port1) and detect ring failures. • The interface (to which port1 is attached) must be a subinterface of the main interface.
Step 9	**exit** **Example:** `Device(config-erp-ring-port)# exit`	Exits Ethernet ring port configuration mode.
Step 10	**exclusion-list vlan-ids** *vlan-id* **Example:** `Device(config-erp-ring)# exclusion-list vlan-ids 2`	Specifies VLANs that are unprotected by the Ethernet ring protection mechanism.
Step 11	**open-ring** **Example:** `Device(config-erp-ring)# open-ring`	Specifies the Ethernet ring as an open ring.
Step 12	**instance** *instance-id* **Example:** `Device(config-erp-ring)# instance 1`	Configures the Ethernet ring instance and enters Ethernet ring instance configuration mode.
Step 13	**description** *descriptive-name* **Example:** `Device(config-erp-inst)# description cisco_customer_instance`	Specifies a descriptive name for the Ethernet ring instance.
Step 14	**profile** *profile-name* **Example:** `Device(config-erp-inst)# profile profile1`	Specifies the profile associated with the Ethernet ring instance.
Step 15	**rpl** {**port0** | **port1**} {**owner** | **neighbor** | **next-neighbor** } **Example:**	Specifies the Ethernet ring port on the local node as the RPL owner, neighbor, or next neighbor.

	Command or Action	Purpose
	`Device(config-erp-inst)# rpl port0 neighbor`	
Step 16	**inclusion-list vlan-ids** *vlan-id* **Example:** `Device(config-erp-inst)# inclusion-list vlan-ids 11`	Specifies VLANs that are protected by the Ethernet ring protection mechanism. **Note** VLANs should be within or equal to VLAN configured in the interface.
Step 17	**aps-channel** **Example:** `Device(config-erp-inst)# aps-channel`	Enters Ethernet ring instance aps-channel configuration mode.
Step 18	**level** *level-value* **Example:** `Device(config-erp-inst-aps)# level 5`	Specifies the Automatic Protection Switching (APS) message level for the node on the Ethernet ring. • All nodes in the Ethernet ring must be configured with the same level.
Step 19	**port0 service instance** *instance-id* **Example:** `Device(config-erp-inst-aps)# port0 service instance 100`	Associates APS channel information with port0.
Step 20	**port1 service instance** {*instance-id* **| none** } **Example:** `Device(config-erp-inst-aps)# port1 service instance 100`	Associates APS channel information with port1.
Step 21	**end** **Example:** `Device(config-erp-inst-aps)# end`	Returns to user EXEC mode.

Configuring Topology Change Notification Propagation

To configure topology change notification (TCN) propagation, complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet tcn-propagation G8032 to {REP | G8032}**
4. **end**

DETAILED STEPS

	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	**ethernet tcn-propagation G8032 to {REP | G8032}** **Example:** `Device(config)# ethernet tcn-propagation G8032 to G8032`	Allows topology change notification (TCN) propagation from a source protocol to a destination protocol. • Source and destination protocols vary by platform and release.
Step 4	**end** **Example:** `Device(config)# end`	Returns to user EXEC mode.

Configuring a Service Instance

To configure a service instance, complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **service instance** *instance-id* **ethernet** [*evc-id*]
5. **encapsulation dot1q** *vlan-id* [**native**]
6. **bridge-domain** *bridge-id* [**split-horizon** [**group** *group-id*]]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 0/1/0`	Specifies the interface type and number.
Step 4	**service instance** *instance-id* **ethernet** [*evc-id*] **Example:** `Device(config-if)# service instance 101 ethernet`	Creates a service instance (an instance of an EVC) on an interface and enters service instance configuration mode.
Step 5	**encapsulation dot1q** *vlan-id* [**native**] **Example:** `Device(config-if-srv)# encapsulation dot1q 13`	Defines the matching criteria to be used in order to map ingress dot1q frames on an interface to the appropriate service instance.
Step 6	**bridge-domain** *bridge-id* [**split-horizon** [**group** *group-id*]] **Example:** `Device(config-if-srv)# bridge-domain 12`	Binds the service instance to a bridge domain instance.
Step 7	**end** **Example:** `Device(config-if-srv)# end`	Exits service instance configuration mode.

Verifying the Ethernet Ring Protection (ERP) Switching Configuration

To verify the ERP switching configuration, use one or more of the following commands in any order.

Note Follow these rules while adding or deleting VLANs from the inclusion list:

- While adding VLAN into the inclusion list, it has to be first added on the interface and then in the G.8032 inclusion list.
- While removing VLAN from the inclusion list, it has to be removed from the G.8032 inclusion list and then from the interface.

Addition or Deletion of VLANs in exclusion list is not supported.

SUMMARY STEPS

1. **enable**
2. **show ethernet ring g8032 status** [*ring-name*] [**instance** [*instance-id*]]
3. **show ethernet ring g8032 brief** [*ring-name*] [**instance** [*instance-id*]]
4. **show ethernet ring g8032 summary**
5. **show ethernet ring g8032 statistics** [*ring-name*] [**instance** [*instance-id*]]
6. **show ethernet ring g8032 profile** [*profile-name*]
7. **show ethernet ring g8032 port status interface** [*type number*]
8. **show ethernet ring g8032 configuration** [*ring-name*] **instance** [*instance-id*]
9. **show ethernet ring g8032 trace** {**ctrl** [*ring-name* **instance** *instance-id*] | **sm**}
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ethernet ring g8032 status** [*ring-name*] [**instance** [*instance-id*]] **Example:** `Device# show ethernet ring g8032 status RingA instance 1`	Displays a status summary for the ERP instance.
Step 3	**show ethernet ring g8032 brief** [*ring-name*] [**instance** [*instance-id*]] **Example:** `Device# show ethernet ring g8032 brief`	Displays a brief description of the functional state of the ERP instance.
Step 4	**show ethernet ring g8032 summary** **Example:** `Device# show ethernet ring g8032 summary`	Displays a summary of the number of ERP instances in each state of the ERP switching process.
Step 5	**show ethernet ring g8032 statistics** [*ring-name*] [**instance** [*instance-id*]] **Example:** `Device# show ethernet ring g8032 statistics RingA instance 1`	Displays the number of events and Ring Automatic Protection Switching (R-APS) messages received for an ERP instance.
Step 6	**show ethernet ring g8032 profile** [*profile-name*] **Example:**	Displays the settings for one or more ERP profiles.

	Command or Action	Purpose
	`Device# show ethernet ring g8032 profile gold`	
Step 7	**show ethernet ring g8032 port status interface** [*type number*] **Example:** `Device# show ethernet ring g8032 port status interface gigabitethernet 0/0/1`	Displays Ethernet ring port status information for the interface.
Step 8	**show ethernet ring g8032 configuration** [*ring-name*] **instance** [*instance-id*] **Example:** `Device# show ethernet ring g8032 configuration RingA instance 1`	Displays the details of the ERP instance configuration manager.
Step 9	**show ethernet ring g8032 trace** {**ctrl** [*ring-name* **instance** *instance-id*] | **sm**} **Example:** `Device# show ethernet ring g8032 trace sm`	Displays information about ERP traces.
Step 10	**end** **Example:** `Device# end`	Returns to privileged EXEC mode.

Configuration Examples for ITU-T G.8032 Ethernet Ring Protection Switching

Example: Configuring Ethernet Ring Protection Switching

The following is an example of an Ethernet Ring Protection (ERP) switching configuration:

```
ethernet ring g8032 profile profile_ABC
 timer wtr 1
 timer guard 100
 timer hold-off  1

ethernet ring g8032 major_ring_ABC
 exclusion-list vlan-ids 1000
 port0 interface  GigabitEthernet 0/0/1
  monitor service instance 103
 port1 interface  GigabitEthernet 0/1/0
  monitor service instance 102
 instance 1
  profile profile_ABC
```

```
  rpl port0 owner
  inclusion-list vlan-ids 100
  aps-channel
   port0 service instance 100
   port1 service instance 100
  !
interface GigabitEthernet0/1/0
mtu 9216
 no ip address
 negotiation auto
 service instance trunk 1 ethernet
  encapsulation dot1q 60-61
  rewrite ingress tag pop 1 symmetric
  bridge-domain from-encapsulation

 !
!
```

Example: Enabling Ethernet Fault Detection for a Service

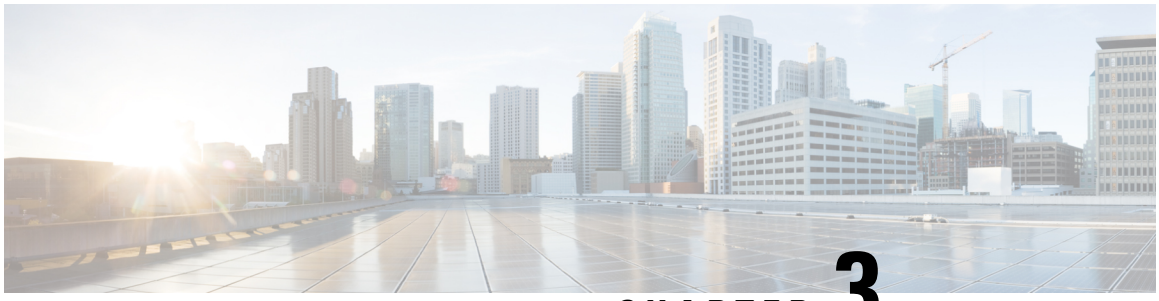
```
ethernet cfm domain G8032 level 4
service 8032_service evc 8032-evc vlan 1001 direction down
  continuity-check
  continuity-check interval 3.3ms
  offload sampling 1000
  efd notify g8032
ethernet ring g8032 profile TEST
timer wtr 1
timer guard 100
ethernet ring g8032 open
open-ring
port0 interface GigabitEthernet0/1/3
  monitor service instance 1001
port1 none
instance 1
  profile TEST
  inclusion-list vlan-ids 2-500,1001
  aps-channel
   port0 service instance 1001
   port1 none
  !
!
instance 2
  profile TEST
  rpl port0 owner
  inclusion-list vlan-ids 1002,1005-2005
  aps-channel
   port0 service instance 1002
   port1 none
  !

interface GigabitEthernet0/1/3
no ip address
load-interval 30
shutdown
negotiation auto
storm-control broadcast level 10.00
storm-control multicast level 10.00
storm-control unicast level 90.00
service instance 1 ethernet
  encapsulation untagged
```

```
  l2protocol peer lldp
  bridge-domain 1
!
service instance trunk 10 ethernet
  encapsulation dot1q 2-500,1005-2005
  rewrite ingress tag pop 1 symmetric
  bridge-domain from-encapsulation
!
service instance 1001 ethernet 8032-evc
  encapsulation dot1q 1001
  rewrite ingress tag pop 1 symmetric
  bridge-domain 1001
  cfm mep domain G8032 mpid 20
!
service instance 1002 ethernet 8032-evc-1
  encapsulation dot1q 1002
  rewrite ingress tag pop 1 symmetric
  bridge-domain 1002
!
End
```

Example: Verifying the Ethernet Ring Protection Configuration

The following is sample output from the **show ethernet ring g8032 configuration** command. Use this command to verify if the configuration entered is valid and to check for any missing configuration parameters.

```
Device# show ethernet ring g8032 configuration

ethernet ring ring0
 Port0: GigabitEthernet0/0/0 (Monitor: GigabitEthernet0/0/0)
 Port1: GigabitEthernet0/0/4 (Monitor: GigabitEthernet0/0/4)
 Exclusion-list VLAN IDs: 4001-4050
 Open-ring: no
 Instance 1
  Description:
  Profile:     opp
  RPL:
  Inclusion-list VLAN IDs: 2,10-500
  APS channel
   Level: 7
   Port0: Service Instance 1
   Port1: Service Instance 1
  State: configuration resolved
```



CHAPTER 3

Multiple Spanning Tree Protocol

The Multiple Spanning Tree Protocol (MSTP) is an STP variant that allows multiple and independent spanning trees to be created over the same physical network. The parameters for each spanning tree can be configured separately, so as to cause a different network devices to be selected as the root bridge or different paths to be selected to form the loop-free topology. Consequently, a given physical interface can be blocked for some of the spanning trees and unblocked for others.

Having set up multiple spanning trees, the set of VLANs in use can be partitioned among them; for example, VLANs 1 - 100 can be assigned to spanning tree 1, VLANs 101 - 200 can be assigned to spanning tree 2, VLANs 201 - 300 can be assigned to spanning tree 3, and so on. Since each spanning tree has a different active topology with different active links, this has the effect of dividing the data traffic among the available redundant links based on the VLAN - a form of load balancing.

Restrictions for configuring MSTP

- RSTP is not supported. To support RSTP, all vlans are mapped to MSTI 0 when no instance is created for MSTP.
- PVSTP is *not* supported.
- Supports only 16 instances.
- Untagged EVCs do not participate in MST loop detection.

How to Configure MST Protocol

This section describes the procedure for configuring MSTP:

Enabling Multiple Spanning Tree Protocol

By default, MSTP is disabled on all interfaces. MSTP need not be enabled explicitly on each interfaces. By turning the global configuration on, it is enabled on all interfaces.

Configuring Multiple Spanning Tree Protocol

Describes steps to configure MST

SUMMARY STEPS

1. **configure**
2. **spanning-tree mode mst**
3. **spanning-tree mst configuration**
4. **instance** *vlan-id* **vlan** *vlan-range*
5. **name** *region*
6. **revision** *revision -number*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `Device> configure`	Enters global configuration mode.
Step 2	**spanning-tree mode mst** **Example:** `Device> spanning-tree mode mst`	Enables MSTP configuration mode.
Step 3	**spanning-tree mst configuration** **Example:** `Device(config)#spanning-tree mst configuration`	Enters the MSTP configuration submode.
Step 4	**instance** *vlan-id* **vlan** *vlan-range* **Example:** `Device(config-mstp-inst)# instance 1 vlan 450-480`	Maps the VLANs to an MST instance
Step 5	**name** *region* **Example:** `Device(config-mstp)# name m1`	Sets the name of the MSTP region.
Step 6	**revision** *revision -number* **Example:** `Device(config-mstp)#)revision 1`	Sets the revision level of the MSTP region.
Step 7	**end** **Example:** `Device(config-mstp-if)# end`	Returns to privileged EXEC mode.

Configuring Untagged EFP over MST Interface

Describes steps to configure untagged EFP over MST:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface number*
4. **no ip address**
5. **service instance** *number* **ethernet** *[name]*
6. **bridge-domain** *bridge-id*
7. **encapsulation untagged dot1q** {any|vlan-id [,vlan-id [-vlan-d]]}
8. **l2protocol peer stp**
9. **end**

DETAILED STEPS

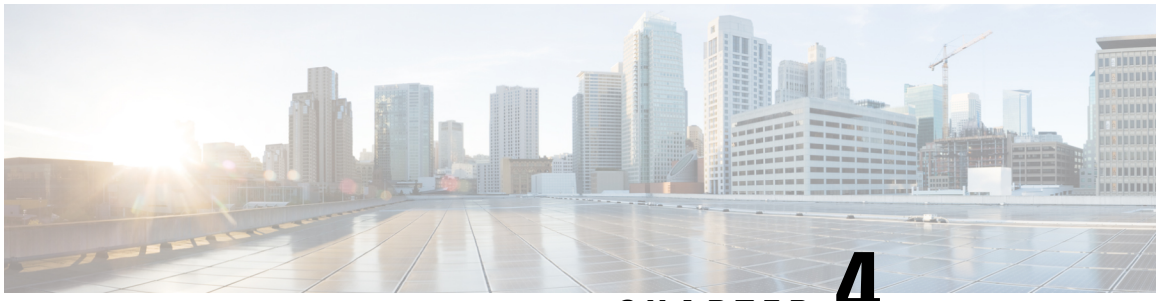
	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface number* **Example:** `Router(config)# interface gigabitEthernet 0/0/5`	Specifies the Gigabit Ethernet interface to configure, where: slot/subslot/port-Specifies the location of the interface.
Step 4	**no ip address** **Example:** `Router (config-if)# no ip address`	Disables the IP address on the interface.
Step 5	**service instance** *number* **ethernet** *[name]* **Example:** `Router (config-if)#service instance 200 ethernet`	Configure an EFP (service instance) and enter service instance configuration mode.
Step 6	**bridge-domain** *bridge-id* **Example:** `Router (config-if-srv)# bridge-domain from-encapsulation`	Creates a list of bridge domains for an EFP trunk port using the bridge-domain IDs derived from the encapsulation VLAN numbers.
Step 7	**encapsulation untagged dot1q** {any|vlan-id [,vlan-id [-vlan-d]]} **Example:**	Configures the encapsulation. Defines the matching criteria that maps the ingress dot1q or untagged frames on an interface for the appropriate service instance.

	Command or Action	Purpose
	`Router (config-if-srv)# encapsulation dot1q 20`	
Step 8	**l2protocol peer stp** **Example:** `Router (config-if-srv)# l2protocol peer stp`	Configures STP to peer with a neighbor on a port that has an EFP service instance.
Step 9	**end** **Example:** `Device(config-mstp-if)# end`	Returns to privileged EXEC mode.

Configuration Example

This example shows how to configure STP to peer with a neighbor on a service instance.

```
interface GigabitEthernet0/0/0
no ip address
negotiation auto
service instance trunk 10 ethernet
  encapsulation dot1q 10-20
  bridge-domain from-encapsulation
!
service instance 1024 ethernet
  encapsulation untagged
  l2protocol peer stp
  bridge-domain 1024
!
end
```



CHAPTER 4

Configuring Flex Links

This chapter describes how to configure Flex Links, a pair of Layer 2 interfaces, where one interface is configured to act as a backup to the other.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for Configuring Flex Links

- Flex Links is supported on Cisco RSP2 module only.
- Flex Links is supported on NCS 4201 and NCS 4202 routers only.
- You can configure only one Flex Link backup link for any active link, and it must be a different interface from the active interface.
- An interface can belong to only one Flex Link pair. An interface can be a backup link for only one active link. An active link cannot belong to another Flex Link pair.
- Neither of the links can be a port that belongs to an EtherChannel nor port channel
- A backup link does not have to be the same type (Fast Ethernet, Gigabit Ethernet) as the active link.
- STP is disabled on Flex Link ports. If STP is configured on the switch, Flex Links do not participate in STP in all VLANs in which STP is configured. With STP not running, be sure that there are no loops in the configured topology.

- Flex link is only supported on trunk EFP.
- In bi-directional traffic, FlexLink Convergence will be high in one-direction due to mac address black holing.

Information About Flex Links

The feature provides an alternative solution to the Spanning Tree Protocol (STP), allowing you to turn off STP and still provide basic link redundancy. Flex Links are typically configured in service provider or enterprise networks, where, you do not want to run STP on the router. If the router is running STP, it is not necessary to configure Flex Links, because STP already provides link-level redundancy or backup. Flex Links are supported only on Trunk EFP and are not supported on other EVCs.

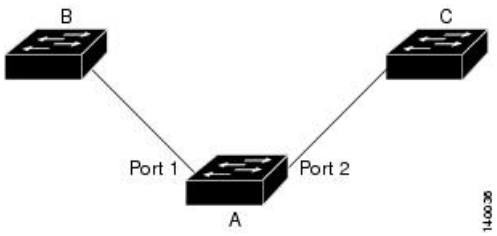
Following are the two flex link modes supported:

- Active-Alone Forwarding Method
- Active-Backup-Both Forwarding Method

Active-Alone forwarding Method

From the schematic representation, ports 1 and 2 on switch A are connected to uplink switches B and C. Because they are configured as Flex Link in active-backup both forwarding mode, both the interfaces will be forwarding traffic. If port 1 is the active link, all mutually inclusive VLANs (common VLANs configured in both active / backup interface) would be forwarded on active interface and mutually exclusive VLANs would be forwarded from the respective active / backup interfaces. If port 1 goes down, then port 2 will start forwarding only the traffic for the common VLANs along with its specific exclusive vlans. All traffic belonging to the exclusive VLANS as part of active interface configuration would be dropped until port 1 comes back to operational state.

Figure 5: Active-Alone Forwarding Method



Configuring Active Alone Forwarding Method

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **no shutdown**
5. **ethernet backup interface** *interface-id*

6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Router(config)# interface gigabitEthernet 0/0/5`	Specify the interface, and enter interface configuration mode. The interface can be a physical Layer 2 interface or a port channel (logical interface). The port-channel range is 1 to 48.
Step 4	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enable the port, if necessary. By default, UNIs are disabled, and NNIs are enabled.
Step 5	**ethernet backup interface** *interface-id* **Example:** `Router(config)# ethernet backup interface gigabitEthernet 0/0/5`	Configure a physical Layer 2 interface (or port channel) as part of a Flex Link pair with the interface. When one link is forwarding traffic, the other interface is in standby mode.
Step 6	**end** **Example:** `Router(config-if)# end`	Return to privileged EXEC mode.

Configuration Example

```
On Active interface(Port 5)

Router> enable
Router# configure terminal
Router# service instance trunk 1000 ethernet
Router# encapsulation dot1q 1-1000
Router# rewrite ingress tag pop 1 symmetric
Router# bridge-domain from-encapsulation


Backup interface (Port 6)

Router> enable
Router# configure terminal
Router# service instance trunk 1000 ethernet
Router# encapsulation dot1q 1-1000
Router# rewrite ingress tag pop 1 symmetric
```

```
Router# bridge-domain from-encapsulation
```

Flexlink Configuration

```
Router> enable
Router# configure terminal
Router(config)# interface gigabitEthernet 0/0/5
Router(config-if)# no shutdown
Router(config-if)# ethernet backup interface gigabitEthernet 0/0/6
Router(config-if)# end
```

Verifying Active Alone Forwarding Method Configuration

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **show ethernet backup detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**show ethernet backup detail** **Example:** `Router# show ethernet backup detail`	This displays the flex link configuration.

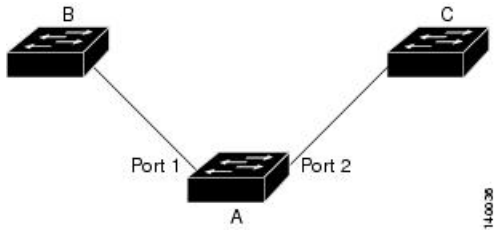
Configuration Output

```
Switch Backup Interface Pairs:
Active Interface        Backup Interface        State
------------------------------------------------------------------------
GigabitEthernet0/0/5    Te0/0/12                Active Up/Backup Standby
        Preemption Mode  : off
        Multicast Fast Convergence  : Off
        Bandwidth : 1000000 Kbit (Gi0/0/3), 1000000 Kbit (Te0/0/12)
        Mac Address Move Update Vlan : auto
        Forwarding  : Active-Only
```

Active-Backup-Both forwarding Method

From the schematic representation, ports 1 and 2 on switch A are connected to uplink switches B and C. Because they are configured as Flex Link in active-backup both forwarding mode, both the interfaces will be forwarding traffic. If port 1 is the active link, all mutually inclusive vlans (common vlans configured in both active / backup interface) would be forwarded on active interface and mutually exclusive vlans would be forwarded from the respective active / backup interfaces. If port 1 goes down, then port 2 will start forwarding only the traffic for the common vlans along with its specific exclusive vlans. All traffic belonging to the exclusive vlans as part of active interface configuration would be dropped until port 1 comes back to operational state.

Figure 6: Active-Backup-Both Forwarding Method



Configuring Active Backup Both Forwarding Method

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **no shutdown**
5. **ethernet backup interface** *interface-id* **prefer forwarding**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Router(config)# interface gigabitEthernet 0/0/8`	Specify the interface, and enter interface configuration mode. The interface can be a physical Layer 2 interface or a port channel (logical interface). The port-channel range is 1 to 48.

	Command or Action	Purpose
Step 4	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enable the port, if necessary. By default, UNIs are disabled, and NNIs are enabled.
Step 5	**ethernet backup interface** *interface-id* **prefer forwarding** **Example:** `Router(config)# ethernet backup interface gigabitEthernet 0/0/8 prefer forwarding`	Configure a physical Layer 2 interface (or port channel) as part of a Flex Link pair with the interface. When one link is forwarding traffic, the other interface is in standby mode.
Step 6	**end** **Example:** `Router(config-if)# end`	Return to privileged EXEC mode.

Configuration Example

```
On Active interface(Port 7)

Router> enable
Router# configure terminal
Router# service instance trunk 1000 ethernet
Router# encapsulation dot1q 1-512
Router# rewrite ingress tag pop 1 symmetric
Router# bridge-domain from-encapsulation



Backup interface (Port 8)

Router> enable
Router# configure terminal
Router# service instance trunk 1000 ethernet
Router# encapsulation dot1q 512-1000
Router# rewrite ingress tag pop 1 symmetric
Router# bridge-domain from-encapsulation



Flexlink Configuration

Router> enable
Router# configure terminal
Router(config)# interface gigabitEthernet 0/0/8
Router(config-if)# no shutdown
Router(config-if)# ethernet backup interface gigabitEthernet 0/0/8 prefer forwarding

Router(config-if)# end
```

Verifying Active-Backup-Both Forwarding Method Configuration

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **show ethernet backup detail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**show ethernet backup detail** **Example:** `Router# show ethernet backup detail`	This displays the flex link configuration.

Configuration Output

```
Switch Backup Interface Pairs:
Active Interface        Backup Interface        State
------------------------------------------------------------------------
GigabitEthernet0/0/3    Te0/0/12                Active Up/Backup Standby
        Preemption Mode  : off
        Multicast Fast Convergence  : Off
        Bandwidth : 1000000 Kbit (Gi0/0/3), 1000000 Kbit (Te0/0/12)
        Mac Address Move Update Vlan : auto
        Forwarding  : Active-Backup-Both
```

Unsupported Functions

Following functions are not supported:

- MMU Notification
- IGMP Fast convergence
- Preemption Support
- Flex links support on a Port channel interface.
- Flex links support on EVC
- Flex links with VLB
- Flex links on IP configured Physical interface.
- Flexlink cannot be configured on a REP / G8032 configured interface and vice-versa.

- STP can be enabled globally but will not be applied on flex link configured interfaces alone.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/mcl/allreleasemcl/all-book.html

Standards and RFCs

Standard/RFC	Title
No specific Standards and RFCs are supported by the features in this document.	—

MIBs

MIB	MIBs Link
—	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Flex Links

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for Flex Links

Feature Name	Releases	Feature Information
Flex Links	Cisco IOS XE Release 3.13.0S	This feature was introduced on the Cisco ASR 920 Series Aggregation Services Router (ASR-920-12CZ-A, ASR-920-12CZ-D, ASR-920-4SZ-A, ASR-920-4SZ-D).