

# Ethernet Protection Switching Ring (EPSR)

## Feature Overview and Configuration Guide

### Introduction

This guide describes EPSR and how to configure it.

Putting a ring of Ethernet switches at the core of a network is a simple way to increase the network's resilience—such a network is no longer susceptible to a single point of failure. However, the ring must be protected from Layer 2 loops. Traditionally, STP-based technologies are used to protect rings, but they are relatively slow to recover from link failure. This can create problems for applications that have strict loss requirements, such as voice and video traffic, where the speed of recovery is highly significant.

This guide describes a fast alternative to STP: Ethernet Protection Switched Ring (EPSR). EPSR enables rings to recover rapidly from link or node failures—within as little as 50ms, depending on port type and configuration. This is much faster than STP at 30 seconds or even RSTP at 1 to 3 seconds.

In a separate section, this guide also describes the EPSR SuperLoop Prevention (EPSR-SLP) feature, which is an enhancement to the existing EPSR feature in AlliedWare Plus. EPSR-SLP prevents “SuperLoops” forming in certain EPSR multi-ring topologies. This functionality makes it possible for EPSR-SLP protected rings to have data VLANs in common on their respective ring domains.

#### List of Terms:

**EPSR**

Ethernet Protection Switched Ring.

**ER**

Enhanced Recovery.

**EPSR Domain**

An EPSR Domain is created from individual switch nodes connected as a ring, where all nodes are configured with an EPSR instance with the same set of EPSR Data VLANs.

**Failover timer expiry**

The Failover timer expires when several healthcheck messages fail to circumnavigate the ring, due to a break in the ring. This causes the master node to undertake subsequent fault recovery actions.

**Health messages**

Messages sent to check the condition of the ring. Also known as Healthcheck messages.

# Contents

Introduction .....	1
Products and software version that apply to this guide .....	4
What information will you find in this document? .....	4
Allied Telesis products that support EPSR and their ring limits.....	4
How EPSR Works.....	6
Establishing a ring.....	7
Detecting a fault.....	9
Recovering from a fault.....	9
Fault in a link or a transit node .....	9
Fault in the master node.....	10
Enhanced Recovery .....	10
Restoring normal operation .....	11
Master node .....	11
Transit nodes with one port down .....	11
Transit nodes with both ports down.....	12
Connecting G.8032 and EPSR .....	12
How to Configure EPSR .....	16
Configuring EPSR .....	16
Example 1: A Basic Ring .....	18
Configure the master node (A) .....	18
Configure the transit nodes (B and C) .....	19
Example 2: A Double Ring .....	20
Example 3: EPSR and RSTP .....	24
Example 4: EPSR with Nested VLANs .....	27
Example 5: EPSR with an iMAP .....	32
Configure the AT-TN7100 iMAP as master node .....	32
Checking the master node configuration .....	32
Configure the AT-TN7100 iMAP as a transit node .....	34
Checking the transit node configuration .....	34
Ports and Recovery Times .....	35
IGMP Snooping and Recovery Times .....	36
Query flooding protection .....	36
Health Message Priority .....	37
Topology Changes and Optimizing EPSR rings with Large Numbers of Data VLANs .....	37
EPSR State and Settings .....	38

Master node in a complete ring .....	38
Master node in a failed ring .....	38
Transit node in a fully forwarding state .....	39
SNMP Traps .....	40
Counters .....	41
Master node in a complete ring .....	41
Transit node in a ring that had failures .....	41
Debugging .....	42
Link down between master node and transit node .....	42
Master node (Node A) debug output .....	42
Transit node (Node B) debug output .....	49
Link Down between two transit nodes .....	53
Master node (Node A) debug output .....	53
Transit node (Node B) debug output .....	59
EPSR SuperLoop Prevention .....	63
Overview .....	63
What is a SuperLoop? .....	63
Why do SuperLoops occur? .....	63
Example diagram .....	63
How does EPSR-SLP work? .....	65
Fault detection and recovery without EPSR-SLP .....	66
Fault detection and recovery with EPSR-SLP .....	66
Common segments .....	67
Master node behavior .....	67
Transit node behavior .....	69
Transit node behaviour if the other port is still down .....	70
EPSR Enhanced Recovery when SLP is enabled .....	71
Transit nodes and Enhanced Recovery .....	71
Physical and logical port control .....	71
EPSR show commands .....	74
Best practice guidelines for EPSR-SLP deployment .....	77
EPSR-SLP priorities .....	77
EPSR-SLP Data VLANs .....	77
Placement of EPSR master node .....	77
Master node on common segment: avoiding inappropriate blocking .....	78
Co-existence with non-SLP EPSR instances .....	78
Caution about Enhanced Recovery with EPSR-SLP topologies with 3 or more rings .....	78
Precaution rule for 3-or-more rings EPSR-SLP topologies .....	79

Cases where manual intervention may be required.....	80
High priority master reboot when ring is down.....	80
Split ring recovery techniques and warning.....	80
Manual fix.....	81
To return to normal configuration .....	81

## Products and software version that apply to this guide

This guide applies to AlliedWare Plus products that support EPSR, running version **5.4.4** or later.

However, support and implementation of EPSR varies between products. To see whether a product supports a particular feature or command, see the following documents:

- The [product's Datasheet](#)
- The product's [Command Reference](#)

These documents are available from the above links on our website at [alliedtelesis.com](http://alliedtelesis.com).

Most features described in this document are supported from AlliedWare Plus 5.4.4 or later. These features are available in later releases:

- Version 5.4.7-1.1 and later support combining EPSR with G.8032 sub-rings.

Feature support may change in later software versions. For the latest information, see the above documents.

## What information will you find in this document?

This guide begins by describing EPSR. Then it gives step-by-step configuration details and examples, followed by discussing important implementation details, troubleshooting information in the following sections, and finally, it explains EPSR-SLP.

## Allied Telesis products that support EPSR and their ring limits

The following table shows whether each product can be configured as a master node, and the maximum number of rings (domains) supported by each product. Note that if your ring has a mixture of products that support 16 rings and products that support 64 rings, the maximum number of rings is 16.

PRODUCT NAME	MASTER	RING LIMIT
SwitchBlade x8100 CFC400 and CFC960	Y	64
SwitchBlade x908 GEN2	Y	64
x950 Series	Y	64
x930 Series	Y	64
x550 Series	Y	64

PRODUCT NAME	MASTER	RING LIMIT
x530 and x530L Series	Y	64
x320 Series	Y	64
x230 and x230L Series	-	16
x220 Series	-	64
IE510 Series	Y	16
IE340 and IE340L Series	Y	16
IE300 Series	Y	16
IE210L Series	Y	16
XS900MX Series	Y	16
GS980MX Series	-	64
GS980EM Series	-	64
GS980M Series	-	64
GS970M Series	-	16
FS980M Series	-	16
SwitchBlade x908	Y	64
DC2552XS/L3	Y	16
x900 Series	Y	64
x610 Series	Y	64
x510 Series	Y	16
IX5	Y	16
x310 Series	Y	16
x210 Series	-	16
IE200 Series	Y	16
MiniMAP 9100	Y	64
iMAP 9700	Y	64
iMAP 9810 Chassis	Y	64
MicroMap 9001	Y	64

## How EPSR Works

EPSR operates on physical rings of switches (note, not on meshed networks). When all nodes and links in the ring are up, EPSR prevents a loop by blocking data transmission across one port. When a node or link fails, EPSR detects the failure rapidly and responds by unblocking the blocked port so that data can flow around the ring.

In EPSR, each ring of switches forms an EPSR domain. One of the domain's switches is the master node and the others are transit nodes. Each node connects to the ring via two ports.

One or more data VLANs send data around the ring, and a control VLAN sends EPSR messages. A physical ring can have more than one EPSR domain, but each domain operates as a separate logical group of VLANs and has its own control VLAN and master node.

On the master node, one port is the primary port and the other is the secondary port. When all the nodes in the ring are up, EPSR prevents loops by blocking the data VLAN on the secondary port.

The master node does not need to block any port on the control VLAN because loops never form on the control VLAN. This is because the master node never forwards any EPSR messages that it receives.

### EPSR Components:

#### **EPSR domain**

A protection scheme for an Ethernet ring that consists of one or more data VLANs and a control VLAN.

#### **Master node**

The controlling node for a domain, responsible for polling the ring state, collecting error messages, and controlling the flow of traffic in the domain.

#### **Transit node**

Other nodes in the domain.

#### **Ring port**

A port that connects the node to the ring. On the master node, each ring port is either the primary port or the secondary port. On transit nodes, ring ports do not have roles.

#### **Primary port**

A ring port on the master node. This port determines the direction of the traffic flow, and is always operational.

#### **Secondary port**

A second ring port on the master node. This port remains active, but blocks all protected VLANs from operating unless the ring fails. Similar to the blocking port in an STP/RSTP instance.

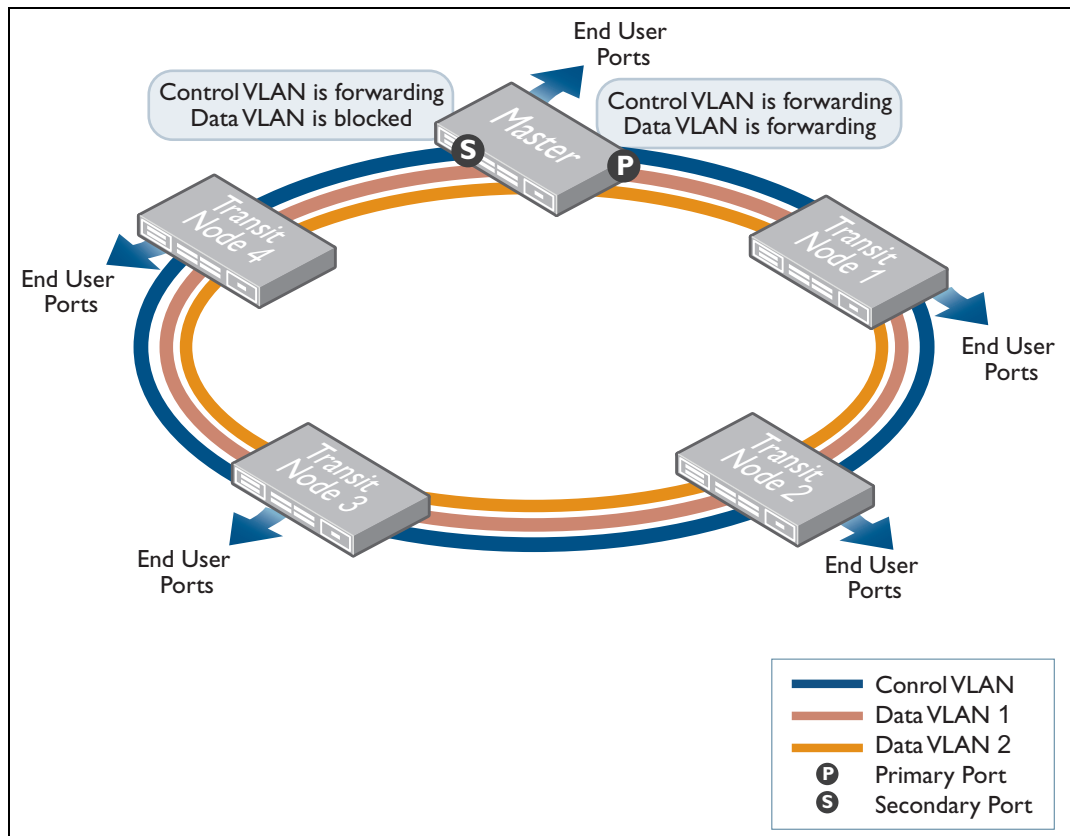
#### **Control VLAN**

The VLAN over which all control messages are sent and received. EPSR never blocks this VLAN.

#### **Data VLAN**

A VLAN that needs to be protected from loops. Each EPSR domain has one or more data VLANs.

The following diagram shows a basic ring with all the switches in the ring up:



## Establishing a ring

Once you have configured EPSR on the switches, the following steps complete the EPSR ring:

1. The master node creates an EPSR Health message and sends it out the primary port. This increments the master node's Transmit: Health counter in the `show epsr count` command.
2. The first transit node receives the Health message on one of its two ring ports and, using a hardware filter, sends the message out its other ring port.

**Note:** Transit nodes never generate Health messages, only receive them and forward them with their switching hardware. This does not increment the transit node's Transmit: Health counter. However, it does increment the Transmit counter in the `show switch port` command.

The hardware filter also copies the Health message to the CPU. This increments the transit node's Receive: Health counter. The CPU processes this message as required by the state machines, but does not send the message anywhere because the switching hardware has already done this.

3. The Health message continues around the rest of the transit nodes, being copied to the CPU and forwarded in the switching hardware.
4. The master node eventually receives the Health message on its secondary port. The master node's hardware filter copies the packet to the CPU (which increments the master node's

Receive: Health counter). Because the Master received the Health message on its secondary port, it knows that all links and nodes in the ring are up.

When the master node receives the Health message back on its secondary port, it resets the Failover timer. If the Failover timer expires before the master node receives the Health message back, it concludes that the ring must be broken.

The master node does not send that particular Health message out again. If it did, the packet would be continuously flooded around the ring. Instead, the master node generates a new Health message when the Hello timer expires.



## Detecting a fault

EPSR uses a fault detection scheme that alerts the ring when a break occurs, instead of using a spanning tree-like calculation to determine the best path. The ring then automatically heals itself by sending traffic over a protected reverse path.

EPSR uses the following two methods to detect when a transit node or a link goes down:

- **Master node polling fault detection**

To check the condition of the ring, the master node regularly sends Health messages out its primary port, as described in ["Establishing a ring" on page 7](#). If all links and nodes in the ring are up, the messages arrive back at the master node on its secondary port.

This can be a relatively slow detection method, because it depends on how often the node sends Health messages. The master node only ever sends Health messages out its primary port. If its primary port goes down, it does not send Health messages.

- **Transit node unsolicited fault detection**

To speed up fault detection, EPSR transit nodes directly communicate when one of their interfaces goes down. When a transit node detects a fault at one of its interfaces, it immediately sends a Link-Down message over the link that remains up. This notifies the master node that the ring is broken and causes it to respond immediately.

## Recovering from a fault

### Fault in a link or a transit node

When the master node detects an outage somewhere in the ring, using either detection method, it restores traffic flow by:

1. Declaring the ring to be in a Failed state
2. Unblocking its secondary port, which enables data VLAN traffic to pass between its primary and secondary ports
3. Flushing its own forwarding database (FDB) for the two ring ports
4. Sending an EPSR Ring-Down-Flush-FDB control message to all the transit nodes, via both its primary and secondary ports.

The transit nodes respond to the Ring-Down-Flush-FDB message by flushing their forwarding databases for each of their ring ports. As the data starts to flow in the ring's new configuration, the nodes (Master and Transit) re-learn their Layer 2 addresses. During this period, the master node continues to send Health messages over the control VLAN. This situation continues until the faulty link or node is repaired.

### Master Node States:

#### Complete

The state when there are no link or node failures on the ring.

#### Failed

The state when there is a link or node failure on the ring. This state indicates that the master node received a Link-Down message or that the failover timer expired before the master node's secondary port received a Health message.

### Transit node states:

#### Idle

The state when EPSR is first configured, before the master node determines that all links in the ring are up. In this state, both ports on the node are blocked for the data VLAN. From this state, the node can move to LinksUp or LinksDown.

#### LinksUp

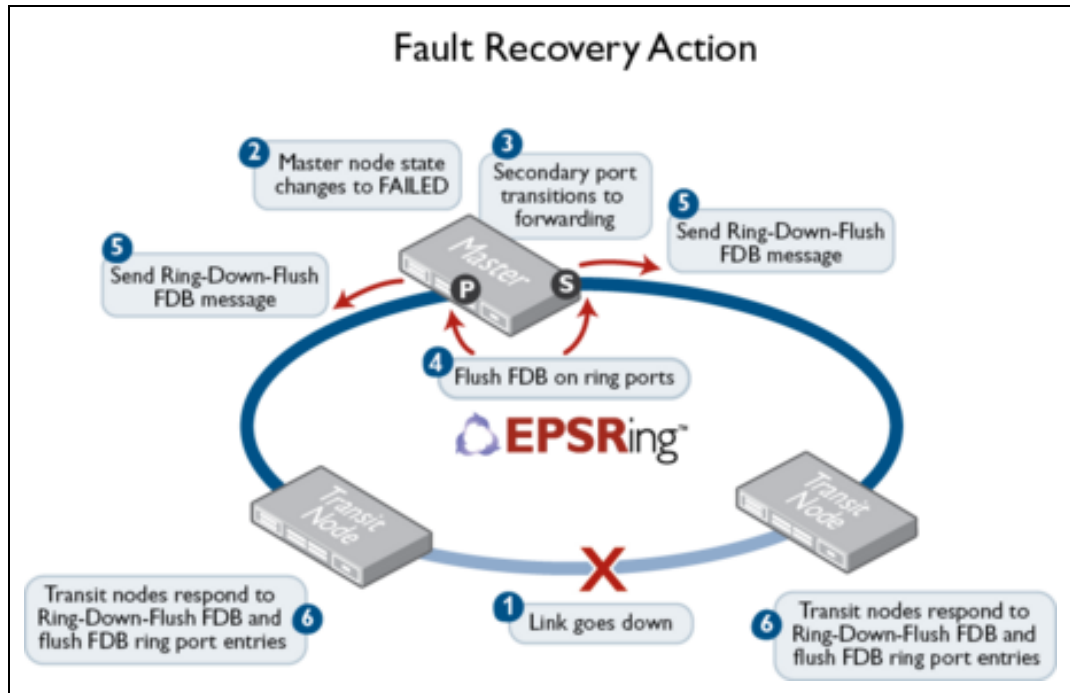
The state when both the node's ring ports are up and forwarding. From this state, the node can move to LinksDown.

#### LinksDown

The state when one or both of the node's ring ports are down. From this state, the node can move to Pre-forwarding.

#### Pre-forwarding

The state when both ring ports are up, but one has only just



For a multi-domain ring, this process occurs separately for each domain within the ring.

### Fault in the master node

If the master node goes down, the transit nodes simply continue forwarding traffic around the ring—their operation does not change.

The only observable effects on the transit nodes are that:

- They stop receiving Health messages and other messages from the master node.
- The transit nodes connected to the master node experience a broken link, so they send Link-Down messages. If the master node is down these messages are simply dropped.

Neither of these symptoms affect how the transit nodes forward traffic.

Once the master node recovers, it continues its function as the master node.

### Enhanced Recovery

A transit node port enters the Pre-forwarding state when the ring port becomes electrically available. Enhanced Recovery can speed a node's recovery from the Pre-forwarding state.

With Enhanced Recovery, the transit node port can exit the Pre-forwarding state without the entire ring becoming complete. It does this in one of two ways:

- When entering the Pre-forwarding state, the transit node sends a Link-Forward-Request message and waits for a response from the master node. When the Master receives this message, it sends a special healthcheck message. If the Master does not receive the healthcheck

back within x seconds, the Master sends a Permission-Link-Forward message to the transit node. The transit node can then start forwarding on both ports.

- If the transit node doesn't receive a Permission-Link-Forward message within x seconds, it makes the decision that the Master is not reachable, and starts forwarding anyway.

Without Enhanced Recovery, the transit node port waits in the Pre-forwarding state until it receives the Ring Up Flush message from the Master. This occurs when the Master receives back its healthcheck messages, and the ring is declared complete.

**Note:** Version 5.4.6-1.x extends EPSR SuperLoop Protection (SLP) to **allow** multiple ring EPSR scenarios where there are multiple ring masters on a common segment, as long as none of the master secondary ports are on the common segment. However, in such scenarios, it is not advisable to use EPSR Enhanced Recovery on transit nodes.

## Restoring normal operation

### Master node

Once the fault has been fixed, the master node's Health messages traverse the whole ring and arrive at the master node's secondary port. The master node then restores normal conditions by:

1. Declaring the ring to be in a state of Complete
2. Blocking its secondary port for data VLAN traffic (but not for the control VLAN)
3. Flushing its forwarding database for its two ring ports
4. Sending a Ring-Up-Flush-FDB message from its primary port, to all transit nodes.

### Transit nodes with one port down

As soon as the fault has been fixed, the transit nodes on each side of the (previously) faulty link section detect that link connectivity has returned. They change their ring port state from LinksDown to Pre-Forwarding, and wait for the master node to send a Ring-Up-Flush-FDB control message.

Once these transit nodes receive the Ring-Up-Flush-FDB message, they:

- Flush the forwarding databases for both their ring ports
- Change the state of their ports from blocking to forwarding for the data VLAN, which allows data to flow through their previously-blocked ring ports

The transit nodes do not start forwarding traffic on the previously-down ports until after they receive the Ring-Up-Flush-FDB message. This makes sure the previously-down transit node ports stay blocked until after the master node blocks its secondary port. Otherwise, the ring could form a loop because it had no blocked ports.

### Transit nodes with both ports down

The Allied Telesis implementation includes an extra feature to improve handling of double link failures. If both ports on a transit node are down and one port comes up, the node:

1. Puts the port immediately into the forwarding state and starts forwarding data out that port. It does not need to wait, because the node knows there is no loop in the ring—because the other ring port on the node is down.
2. Remains in the LinksDown state
3. Starts a DoubleFailRecovery timer with a timeout of four seconds
4. Waits for the timer to expire. At that time, if one port is still up and one is still down, the transit node sends a Ring-Up-Flush-FDB message out the port that is up. This message is usually called a “Fake Ring Up message”. Sending this message allows any ports on other transit nodes that are blocking or in the Pre-forwarding state to move to forwarding traffic in the LinksUp state. The timer delay lets the device at the other end of the link that came up configure its port appropriately, so that it is ready to receive the transmitted message.

**Note:** The master node would not send a Ring-Up-Flush-FDB message in these circumstances, because the ring is not in a state of Complete. The master node’s secondary port remains unblocked.

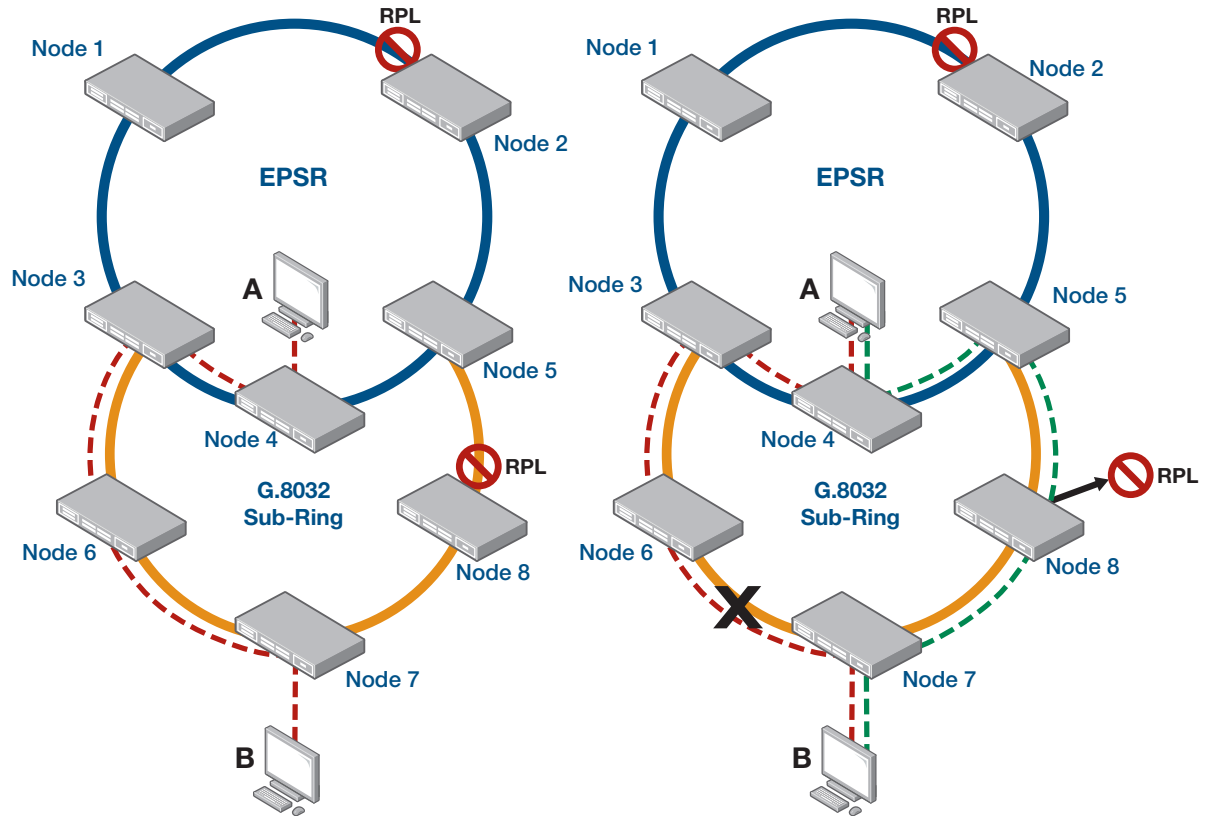
### Connecting G.8032 and EPSR

From software version 5.4.7-1.1 onwards, G.8032 can also interact with EPSR. A G.8032 sub-ring may be connected to and interact with an EPSR ring.

**Note:** NOTE: Only a G.8032 sub-ring connected to an EPSR ring is supported. A G.8032 major ring connected to an EPSR ring is not supported.

In the following diagram, a G.8032 sub-ring that is made up of nodes 3, 6, 7, 8, and 5 is connected to an EPSR ring made up of nodes 1, 2, 3, 4, and 5. The G.8032 sub-ring is protecting the same Data VLANs as the EPSR ring. In this scenario, any topology changes seen in the G.8032 sub-ring may need to be propagated to the EPSR ring. This requires EPSR on the Interconnecting node (nodes 3 and 5) to inform all the other nodes in the EPSR instance to flush their FDB using the FLUSH-FDB message.

Figure 1: Topology Change in G.8032 and EPSR



The path from A to B for the data VLANs that are being protected in the EPSR ring and the G.8032 sub-ring is shown in the red dotted line, going through nodes 4-3-6-7. When a failure occurs in the G.8032 sub-ring as shown, a block is performed around the failure, while the G.8032 sub-ring's RPL is opened up for traffic to flow. The new active topology of the same data VLANs has changed. The path from A to B is different as shown by the green dotted line going through nodes 4-5-8-7.

However, node 4 (as well as nodes 1 and 2) in the EPSR ring do not know about the topology change in the G.8032 sub-ring and node 4 continues to forward traffic from A to B along the red dotted line. To overcome this, the nodes in the EPSR ring need to flush their FDB. Within the interconnected node (nodes 3 and 5), this requires the sub-ring ERP instance to notify the EPSR ring domain instance so that EPSR in nodes 3 and 5 can perform a flush and in turn notify all the other nodes in the EPSR ring to also perform an FDB flush.

Although the G.8032 ERP instance and EPSR domain instance are independent of one another, the interconnected node has knowledge of which instances are protecting the same data VLANs. As such, the sub-ring ERP instance can determine which EPSR ring domain instance to notify of topology changes seen by the G.8032 sub-ring.

Once the EPSR ring domain instance is notified of the TCN by the G.8032 sub-ring ERP instance, the EPSR ring domain instance can send out a FLUSH-FDB message. This message will go around the EPSR ring causing all the other EPSR nodes in the EPSR ring to perform a FDB flush. After which, MAC re-learning will occur and traffic along the protected data VLANs can flow properly along the green path. Because of added delays of the G.8032 ring informing the EPSR ring of the topology change, switchover times of the data VLANs may not meet 50ms objectives.

**Note:** Not all scenarios require the non-interconnected EPSR ring nodes to perform an FDB flush, and as such the sending of the FLUSH-FDB message is optional.

A G.8032 ERP instance detecting a topology change sends TCNs to the EPSR instance(s) that is protecting the same data VLAN(s) as the ERP instance if the EPSR instance has two ring ports and is enabled. This is called the “target EPSR instance”. Once a TCN is received by the target EPSR instance, the target EPSR instance performs an FDB Flush of its two ring ports. The target EPSR instance also performs an ARP cache flush, and sends Query Solicit, and gratARP if the VLAN is enabled for such.

To enable an EPSR instance to send out a FLUSH-FDB message after being notified by an ERP instance, use the following command:

```
awplus(config)#epsr <epsr-instance-name> topology-change g8032
```

To disable this functionality, use the following command:

```
awplus(config)#no epsr <epsr-instance-name> topology-change g8032
```

where:

**<epsr-instance-name>** identifies the EPSR ring domain instance.

**g8032** is the protocol that EPSR will allow as the trigger for the sending of an FLUSH-FDB message.

By default, the **topology-change** is disabled for the EPSR instance, and thus the target EPSR instance will not send a FLUSH-FDB message.

To see which EPSR target instances were found, use the following command:

```
awplus#show g8032 erp-instance {<erp-instance-name>|all}
```

An example of the output can be seen in the ERPS instance section. The relevant section:

```
awplus#show g8032 erp-instance all
-----
...
TTCN To Inst      : EPSR-red
TCN Flush Event    : G8032
Wait-To-Restore     : -
...
```

Once a target EPSR instance is informed of a TCN, if enabled by configuration, it will in turn send FLUSH-FDB messages out both of its ring ports. Upon receipt of a FLUSH-FDB message (not from itself), the EPSR instance performs an FDB flush on both of its ring ports. The EPSR instance also performs an ARP cache flush, and send Query Solicit, and gratARP if the VLAN is enabled for such.

To see the EPSR configuration for topology change, use the following command:

```
awplus#show epsr {<instance-name>|null}
```

This generates the following example output:

```
awplus#show epsr testpsr2
EPSR Information
-----
Name ..... testpsr2
Mode ..... Transit
Status ..... Enabled
State ..... Links-Down
Control Vlan ..... 400
Data VLAN(s) ..... 2000,2199
First Port ..... port1.0.7
  Status ..... Down
  Direction ..... Unknown
  Is On Common Segment ..... No
  Blocking Control ..... Physical
Second Port ..... port1.0.8
  Status ..... Down
  Direction ..... Unknown
  Is On Common Segment ..... No
  Blocking Control ..... Physical
Trap ..... Enabled
Master Node ..... Unknown
Enhanced Recovery ..... Disabled
G.8032 TCN Flush Event ..... Enabled
Priority ..... 0 [superloop prevention disabled]
```

To see the EPSR counts for sending and receiving FLUSH-FDB messages, use the following command:

```
awplus#show epsr {<instance-name>|null} counters
```

This generates the following example output:

```
awplus#show epsr testpsr2 counters
EPSR Counters
-----
Name: testpsr2
Receive:                                Transmit:
Total EPSR Packets                      0      Total EPSR Packets      0
Health                                  0      Health                  0
Ring Up                                0      Ring Up                 0
Ring Down                              0      Ring Down               0
Link Down                              0      Link Down               0
Link Forward Request                    0      Link Forward Request    0
Permit Link Forward                     0      Permit Link Forward     0
Flush FDB                             0      Flush FDB             0
Invalid EPSR Packets                    0
-----
```

# How to Configure EPSR

This section first outlines, step-by-step, how to configure EPSR. Then it discusses changing the settings for the control VLAN, if you need to do this after initial configuration.

## Configuring EPSR

### Step 1. Connect your switches into a ring.

EPSR does not in itself limit the number of nodes that can exist on any given ring. For information on ring limits, see the section titled: ["Allied Telesis products that support EPSR and their ring limits" on page 4.](#)

If you already have a ring in a live network, disconnect the cable between any two of the nodes before you start configuring EPSR, to prevent a loop.

### Step 2. On each switch, configure EPSR.

On each switch, perform the following configuration steps. Configuration of the master node and each transit node is very similar.

#### 1. Configure the control and data VLAN.

This step creates the control and data VLANs for EPSR. Enter global configuration mode and enter the following commands:

```
awplus(config)#vlan database
awplus(config-vlan)#vlan <control-vid> name <control-vlan-name>
awplus(config-vlan)#vlan <data-vid> name <data-vlan-name>
```

#### 2. Configure the switch ports.

This step sets the rings ports to VLAN trunk mode and adds the control and data VLANs.

Enter global configuration mode and enter the following commands:

```
awplus(config)#interface <port-numbers>
awplus(config-if)#switchport mode trunk
awplus(config-if)#switchport trunk allowed vlan add <control-vid,data-vid>
awplus(config-if)#switchport trunk native vlan none
```



The final command removes the native VLAN (vlan1) from the ring ports. If you leave all the ring ports in the native VLAN, they will create a loop, unless vlan1 is part of the EPSR domain. To avoid loops, you need to do one of the following:

- make vlan1 a data VLAN, or
- remove the ring ports from vlan1, or
- remove at least one of the ring ports from vlan1 on at least one of the switches. We do not recommend this option, because the action you have taken is less obvious when maintaining the network later.

In this document, we remove the ring ports from the native VLAN (vlan1).

### 3. Configure the EPSR domain.

This step creates the domain, specifying whether the switch is the master node or a transit node. It also specifies which VLAN is the control VLAN, and on the master node which port is the primary port.

Enter global configuration mode and enter the following commands.

On the master node:

```
awplus(config)#epsr configuration
awplus(config-epsr)#epsr <name> mode master
controlvlan <control-vid> primaryport <port-numbers>
awplus(config-epsr)#epsr <name> datavlan <data-vid>
```

On each transit node:

```
awplus(config)#epsr configuration
awplus(config-epsr)#epsr <name> mode transit controlvlan <control-vid>
awplus(config-epsr)#epsr <name> datavlan <data-vid>
```

### 4. Change the timers if necessary.

You can change how often the master sends health messages (hellotime), how long the master waits for a returning health check message before entering the failed state (failovertime), and the minimum time a master must stay in the failed state (ringflaptime). The failovertime should be more than double the hellotime.

For most networks, the default timer settings are suitable. But if you are using VCStack with EPSR, you must increase the EPSR failovertime to at least 5 seconds, to avoid broadcast storms during stack failover. Broadcast storms may occur if the stack failover isn't finished before the EPSR failover timer expires.

To change the timers (in seconds), use the commands:

```
awplus(config)#epsr configuration
awplus(config-epsr)#epsr <name> failovertime <2-65535>
awplus(config-epsr)#epsr <name> hellotime <1-32767>
awplus(config-epsr)#epsr <name> ringflaptime <0-65535>
```

## 5. Enable EPSR.

This step enables the domain on each switch.

Enter global configuration mode and enter the following commands:

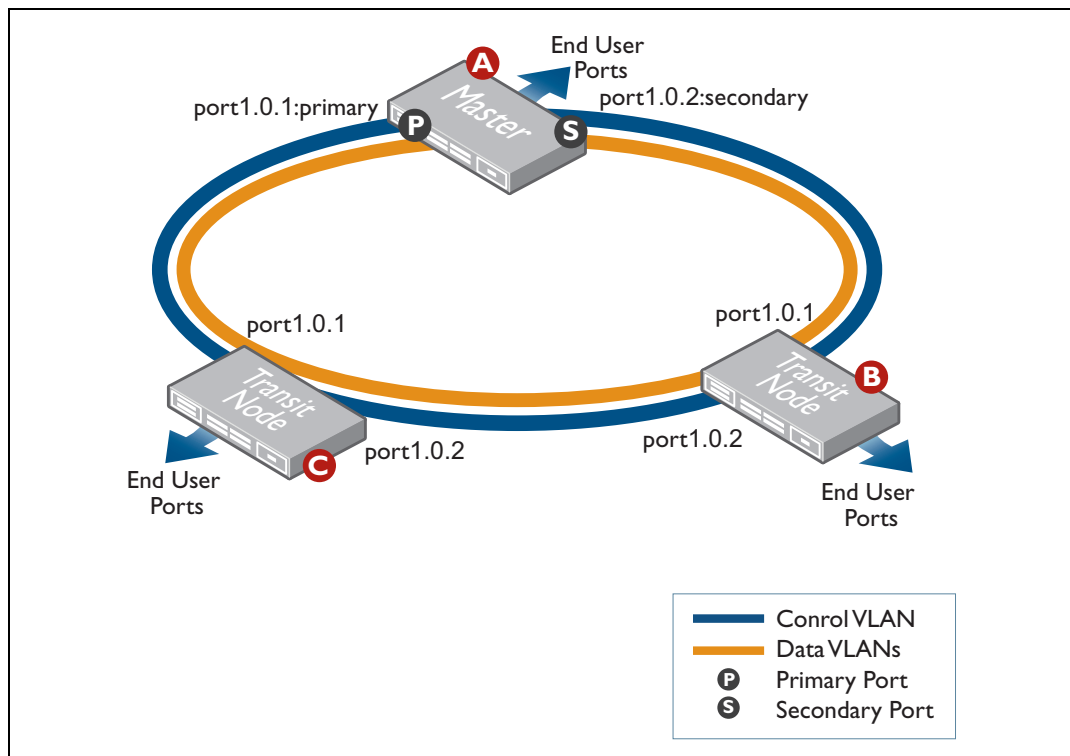
```
awplus(config-epsr)#epsr <name> state enabled
```

### Step 3. Configure other ports and protocols as required.

On each switch, configure the other ports and protocols that are required for your network.

## Example 1: A Basic Ring

This example builds a simple 3-switch ring with one data VLAN, as shown in the following diagram. Control packets are transmitted around the ring on vlan 1000 and data packets on VLAN 2.



## Configure the master node (A)

### Step 1. Configure the control and data VLANs.

Enter global configuration mode and enter the following commands:

```
awplus(config)#vlan database
awplus(config-vlan)#vlan 1000 name epsr-control
awplus(config-vlan)#vlan 2 name data
```

**Step 2. Configure the switch ports.**

```
awplus(config-vlan)#interface port1.0.1-port1.0.2
awplus(config-if)#switchport mode trunk
awplus(config-if)#switchport trunk allowed vlan add 1000,2
awplus(config-if)#switchport trunk native vlan none
```

**Step 3. Configure the EPSR domain.**

Create the domain, specifying that this switch is the master node. Also specify which VLAN is the control VLAN and which port is the primary port. In this example the EPSR domain is called **awplus**.

```
awplus(config-if)#epsr configuration
awplus(config-epsr)#epsr awplus mode master controlvlan 1000 primaryport
port1.0.1
awplus(config-epsr)#epsr awplus datavlan 2
```

**Step 4. Enable EPSR.**

```
awplus(config-epsr)#epsr awplus state enabled
```

**Configure the transit nodes (B and C)**

Each of the transit nodes has the same EPSR configuration in this example.

**Step 1. Configure the control and data VLAN.**

Enter global configuration mode and enter the following commands:

```
awplus(config)#vlan database
awplus(config-vlan)#vlan 1000 name epsr-control
awplus(config-vlan)#vlan 2 name data
```

**Step 2. Configure the switch ports.**

```
awplus(config-vlan)#interface port1.0.1-port1.0.2
awplus(config-if)#switchport mode trunk
awplus(config-if)#switchport trunk allowed vlan add 1000,2
awplus(config-if)#switchport trunk native vlan none
```

**Step 3. Configure the EPSR domain.**

```
awplus(config-if)#epsr configuration
awplus(config-epsr)#epsr awplus mode transit controlvlan 1000
awplus(config-epsr)#epsr awplus datavlan 2
```

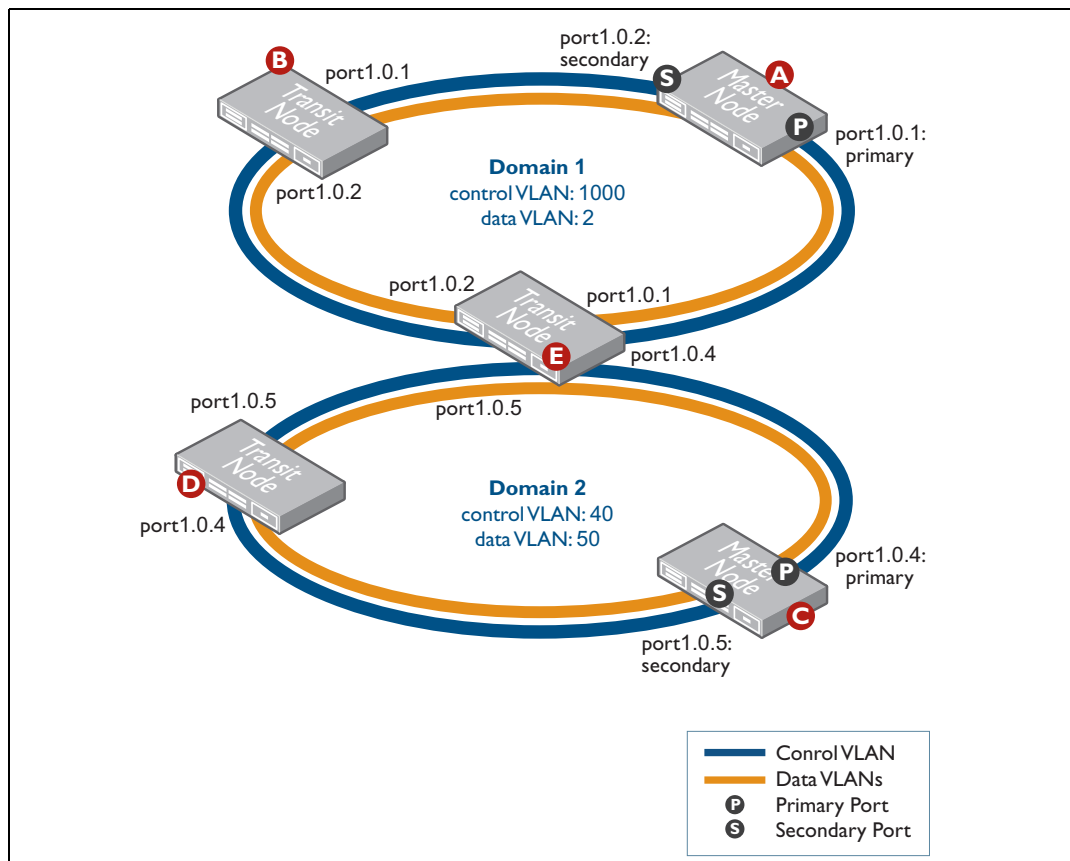
**Step 4. Enable EPSR.**

The two ring ports must belong to both the control VLAN and all data VLANs.

```
awplus(config-epsr)#epsr awplus state enabled
```

## Example 2: A Double Ring

This example adds to the previous ring by making two domains, as shown in the following diagram.



**Step 1. Configure the master node (switch A) for domain 1.**

The master node for domain 1 is the same as in the previous example (except that the domain has been renamed).

Enter global configuration mode and enter the following commands:

Configure the control and data VLANs:

```
awplus(config)#vlan database
awplus(config-vlan)#vlan 1000 name epsr-control
awplus(config-vlan)#vlan 2 name data
```

Configure the switch ports:

```
awplus(config-vlan)#interface port1.0.1-port1.0.2
awplus(config-if)#switchport mode trunk
awplus(config-if)#switchport trunk allowed vlan add 1000,2
awplus(config-if)#switchport trunk native vlan none
```

Configure the EPSR domain. This device is an EPSR master node.

```
awplus(config-if)#epsr configuration
awplus(config-epsr)#epsr domain1 mode master controlvlan 1000 primaryport
port1.0.1
awplus(config-epsr)#epsr domain1 datavlan 2
```

Enable EPSR:

```
awplus(config-epsr)#epsr domain1 state enabled
awplus(config-epsr)#end
```

**Step 2. Configure the transit node (switch B) that belongs just to domain 1.**

This transit node is the same as in the previous example (except that the domain has been renamed).

Enter global configuration mode and enter the following commands:

Configure the data and control VLANs:

```
awplus(config)#vlan database
awplus(config-vlan)#vlan 1000 name epsr-control
awplus(config-vlan)#vlan 2 name data
```

Configure the switch ports:

```
awplus(config-vlan)#interface port1.0.1-port1.0.2
```

```
awplus(config-if)#switchport mode trunk
awplus(config-if)#switchport trunk allowed vlan add 1000,2
awplus(config-if)#switchport trunk native vlan none
```

Configure the EPSR domain. This device is a transit node:

```
awplus(config-if)#epsr configuration
awplus(config-epsr)#epsr domain1 mode transit controlvlan 1000
awplus(config-epsr)#epsr domain1 datavlan 2
```

Enable EPSR:

```
awplus(config-epsr)#epsr domain1 state enabled
awplus(config-epsr)#end
```

### Step 3. Configure the master node (switch C) for domain 2.

Enter global configuration mode and enter the following commands:

```
awplus(config-epsr)#vlan database
awplus(config-vlan)#vlan 40 name epsr-control
awplus(config-vlan)#vlan 50 name data
awplus(config-vlan)#interface port1.0.4-port1.0.5
awplus(config-if)#switchport mode trunk
awplus(config-if)#switchport trunk allowed vlan add 50,40
awplus(config-if)#switchport trunk native vlan none
awplus(config-if)#epsr configuration
awplus(config-epsr)#epsr domain2 mode master controlvlan 40 primaryport
port1.0.4
awplus(config-epsr)#epsr domain2 datavlan 50
awplus(config-epsr)#epsr domain2 state enabled
awplus(config-epsr)#end
```

### Step 4. Configure the transit node (switch D) that belongs just to domain 2.

Enter global configuration mode and enter the following commands:

```
awplus(config)#vlan database
awplus(config-vlan)#vlan 40 name epsr-control
awplus(config-vlan)#vlan 50 name data
awplus(config-vlan)#interface port1.0.4-port1.0.5
awplus(config-if)#switchport mode trunk
awplus(config-if)#switchport trunk allowed vlan add 40,50
```

```
awplus(config-if)#switchport trunk native vlan none
awplus(config-if)#epsr configuration
awplus(config-epsr)#epsr domain2 mode transit controlvlan 40
awplus(config-epsr)#epsr domain2 datavlan 50
awplus(config-epsr)#epsr domain2 state enabled
awplus(config-epsr)#end
```

### Step 5. Configure the transit node (switch E) that belongs to both domains.

Two separate EPSR domains are configured on this device. Enter global configuration mode and enter the following commands:

Configure the data and control VLANs for domain 1:

```
awplus(config)#vlan database
awplus(config-vlan)#vlan 1000 name epsr-control
awplus(config-vlan)#vlan 2 name data
```

Configure the data and control VLANs for domain 2:

```
awplus(config-vlan)#vlan 40 name epsr-control
awplus(config-vlan)#vlan 50 name data
```

Configure the switch ports for domain 1:

```
awplus(config-vlan)#interface port1.0.1-port1.0.2
awplus(config-if)#switchport mode trunk
awplus(config-if)#switchport trunk allowed vlan add 1000,2
awplus(config-if)#switchport trunk native vlan none
```

Configure the switch ports for domain 2:

```
awplus(config-if)#interface port1.0.4-port1.0.5
awplus(config-if)#switchport mode trunk
awplus(config-if)#switchport trunk allowed vlan add 40,50
awplus(config-if)#switchport trunk native vlan none
```

Configure EPSR for domain 1. This device is a transit node:

```
awplus(config-if)#epsr configuration
awplus(config-epsr)#epsr domain1 mode transit controlvlan 1000
awplus(config-epsr)#epsr domain1 datavlan 2
```

Configure EPSR for domain 2. This device is a transit node:

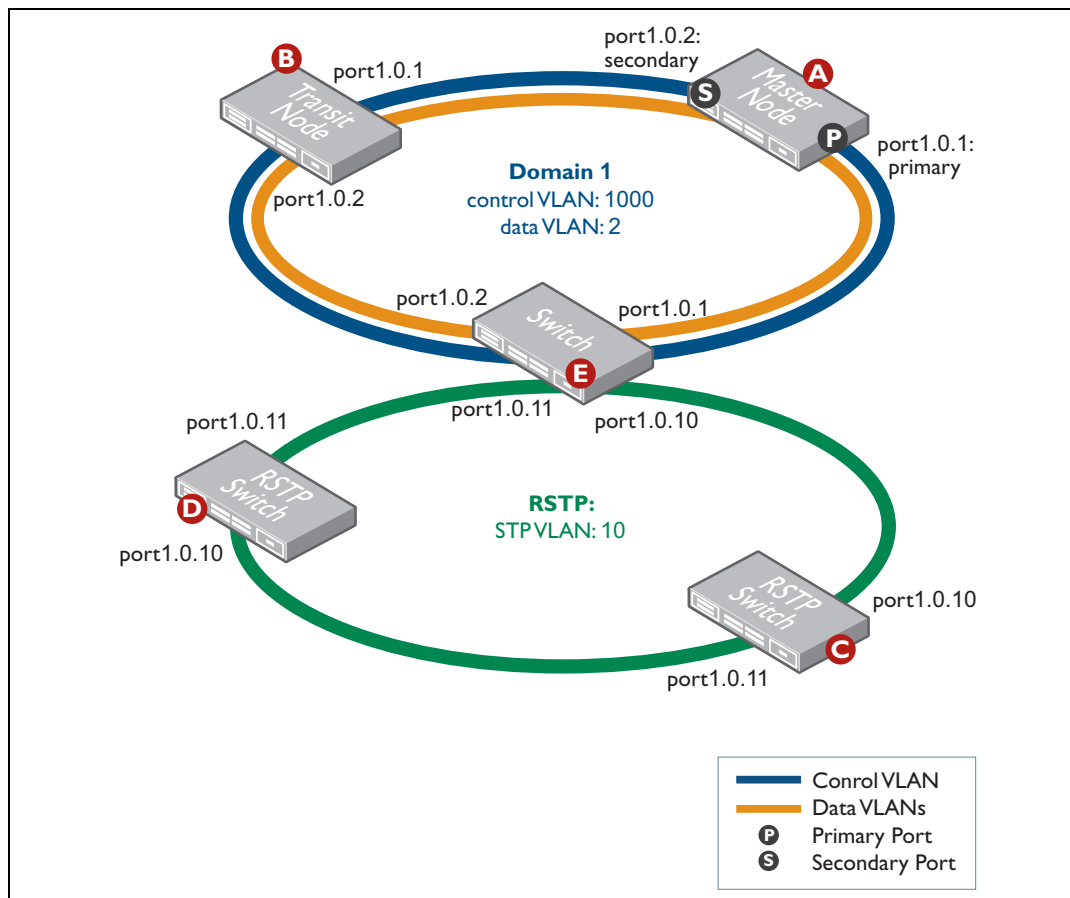
```
awplus(config-epsr)#epsr domain2 mode transit controlvlan 40
awplus(config-epsr)#epsr domain2 datavlan 50
```

Enable EPSR for both domains:

```
awplus(config-epsr)#epsr domain1 state enabled
awplus(config-epsr)#epsr domain2 state enabled
awplus(config-epsr)#end
```

## Example 3: EPSR and RSTP

This example shows how to configure both EPSR and RSTP in the same network. It is possible to configure both protocols on a single device. However, it is not possible to run both EPSR and RSTP on the same ports. RSTP is automatically disabled on a port when it is added to an EPSR VLAN.



### Step 1. Configure the master node (switch A) for the EPSR domain.

The master node is the same as in the previous example.

Enter global configuration mode and enter the following commands:

```
awplus(config)#vlan database
awplus(config-vlan)#vlan 1000 name epsr-control
awplus(config-vlan)#vlan 2 name data
awplus(config-vlan)#interface port1.0.1-port1.0.2
```



```
awplus(config-if)#switchport mode trunk
awplus(config-if)#switchport trunk allowed vlan add 1000,2
awplus(config-if)#switchport trunk native vlan none
awplus(config-if)#epsr configuration
awplus(config-epsr)#epsr domain1 mode master controlvlan 1000 primaryport
port1.0.1
awplus(config-epsr)#epsr domain1 datavlan 2
awplus(config-epsr)#epsr domain1 state enabled
awplus(config-epsr)#end
```

**Step 2. Configure the transit node switch (B) that belongs just to the EPSR domain.**

This transit node (B) is the same as in the previous example.

Enter global configuration mode and enter the following commands:

```
awplus(config)#vlan database
awplus(config-vlan)#vlan 1000 name epsr-control
awplus(config-vlan)#vlan 2 name data
awplus(config-vlan)#interface port1.0.1-port1.0.2
awplus(config-if)#switchport mode trunk
awplus(config-if)#switchport trunk allowed vlan add 1000,2
awplus(config-if)#switchport trunk native vlan none
awplus(config-if)#epsr configuration
awplus(config-epsr)#epsr domain1 mode transit controlvlan 1000
awplus(config-epsr)#epsr domain1 datavlan 2
awplus(config-epsr)#epsr domain1 state enabled
awplus(config-epsr)#end
```

**Step 3. Configure switches belonging to the RSTP instance (switches C and D).**

Switches C and D have the same configuration in this example.

Enter global configuration mode and enter the following commands:

```
awplus(config)#vlan database
awplus(config-vlan)#vlan 10 name rstp-domain
awplus(config-vlan)#interface port1.0.10-1.0.11
awplus(config-if)#switchport mode trunk
awplus(config-if)#switchport trunk allowed vlan add 10
awplus(config-if)#end
```

Note that RSTP is enabled by default on AlliedWare Plus switches.

#### Step 4. Configure switch E for EPSR and RSTP.

Enter global configuration mode and enter the following commands.

Configure the data and control VLANs for the EPSR domain:

```
awplus(config)#vlan database
awplus(config-vlan)#vlan 1000 name epsr-control
awplus(config-vlan)#vlan 2 name data
```

Configure the VLAN for the RSTP domain:

```
awplus(config-vlan)#vlan 10 name rstp-vlan
```

Configure the switch ports for the EPSR domain:

```
awplus(config-vlan)#interface port1.0.1-port1.0.2
awplus(config-if)#switchport mode trunk
awplus(config-if)#switchport trunk allowed vlan add 1000,2
awplus(config-if)#switchport trunk native vlan none
```

Configure the switch ports for the RSTP domain:

```
awplus(config-if)#interface port1.0.10-port1.0.11
awplus(config-if)#switchport mode trunk
awplus(config-if)#switchport trunk allowed vlan add 10
```

Configure the EPSR domain. This device is a transit node:

```
awplus(config-if)#epsr configuration
awplus(config-epsr)#epsr domain1 mode transit controlvlan 1000
awplus(config-epsr)#epsr domain1 datavlan 2
```

Enable EPSR:

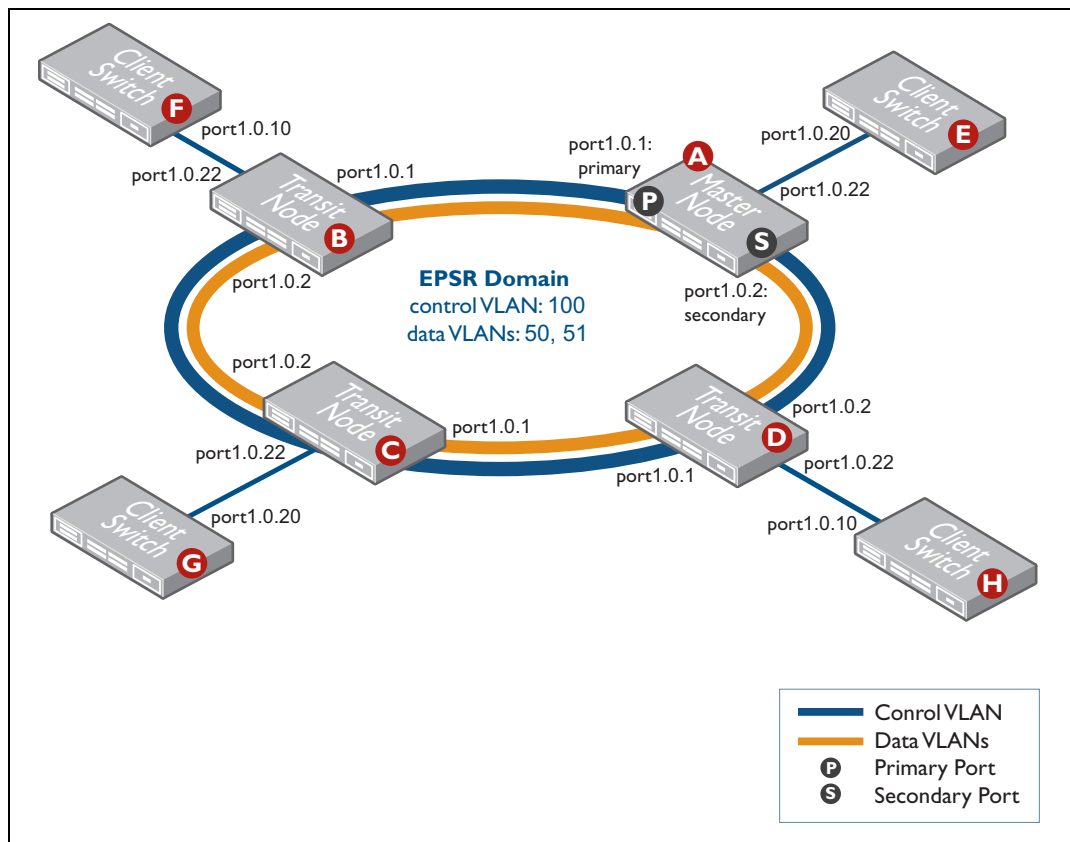
```
awplus(config-epsr)#epsr domain1 state enabled
awplus(config-epsr)#end
```

**Note:** Although RSTP is enabled by default on AlliedWare Plus switches, it is automatically disabled on EPSR ports.

## Example 4: EPSR with Nested VLANs

In this example:

- Client switches A and C are in the same end-user VLAN (vlan20)
- Client switches B and D are in the same end-user VLAN (vlan200)
- Traffic for vlan20 is nested inside vlan 50 for transmission around the core
- Traffic for vlan200 is nested inside vlan51 for transmission around the core
- The data VLANs for the EPSR domain are vlan50 and vlan51
- The control VLAN for the EPSR domain is vlan100



### Step 1. Configure the master node (switch A) for the EPSR domain.

Enter global configuration mode and enter the following commands:

Configure the EPSR control and data VLANs:

```
awplus(config)#vlan database
awplus(config-vlan)#vlan 100 name epsr-control
awplus(config-vlan)#vlan 50 name data-c1
awplus(config-vlan)#vlan 51 name data-c2
```

In this example the data VLAN is also the nested VLAN.

Configure the switch ports:

```
awplus(config-vlan)#interface port1.0.22
awplus(config-if)#switchport access vlan 50
awplus(config-if)#switchport vlan-stacking customer-edge-port
awplus(config-if)#interface port1.0.1-port1.0.2
awplus(config-if)#switchport mode trunk
awplus(config-if)#switchport trunk allowed vlan add 100,50,51
awplus(config-if)#switchport trunk native vlan none
awplus(config-if)#switchport vlan-stacking provider-port
```

Configure the EPSR domain. This switch is the master node:

```
awplus(config-if)#epsr configuration
awplus(config-epsr)#epsr awplus mode master controlvlan 100 primaryport
port1.0.1
awplus(config-epsr)#epsr awplus datavlan 50-51
```

Enable EPSR:

```
awplus(config-epsr)#epsr awplus state enabled
awplus(config-epsr)#exit
```

## Step 2. Configure transit node C for the EPSR domain.

Each of the transit nodes has the same EPSR configuration in this example.

Enter global configuration mode and enter the following commands.

Configure the EPSR control and data VLANs:

```
awplus(config)#vlan database
awplus(config-vlan)#vlan 100 name epsr-control
awplus(config-vlan)#vlan 50 name data-c1
awplus(config-vlan)#vlan 51 name data-c2
```

**Note:** In this example the control VLAN is also the nested VLAN.

Configure the switch ports:

```
awplus(config-vlan)#interface port1.0.22
awplus(config-if)#switchport access vlan 50
awplus(config-if)#switchport vlan-stacking customer-edge-port
awplus(config-if)#interface port1.0.1-port1.0.2
awplus(config-if)#switchport mode trunk
```

```
awplus(config-if)#switchport trunk allowed vlan add 100,50,51
awplus(config-if)#switchport trunk native vlan none
awplus(config-if)#switchport vlan-stacking provider-port
```

Configure the EPSR domain:

```
awplus(config-if)#epsr configuration
awplus(config-epsr)#epsr awplus mode transit controlvlan 100
awplus(config-epsr)#epsr awplus datavlan 50-51
```

Enable EPSR:

```
awplus(config-epsr)#epsr awplus state enabled
awplus(config-epsr)#exit
```

### Step 3. Configure transit nodes B and D for the EPSR domain.

Each of the transit nodes has the same EPSR configuration in this example.

Enter global configuration mode and enter the following commands.

Configure the EPSR control and data VLANs:

```
awplus(config)#vlan database
awplus(config-vlan)#vlan 100 name epsr-control
awplus(config-vlan)#vlan 50 name data-c1
awplus(config-vlan)#vlan 51 name data-c2
```

**Note:** In this example the control VLAN is also the nested VLAN.

Configure the switch ports:

```
awplus(config-vlan)#interface port1.0.22
awplus(config-if)#switchport access vlan 51
awplus(config-if)#switchport vlan-stacking customer-edge-port
awplus(config-if)#interface port1.0.1-port1.0.2
awplus(config-if)#switchport mode trunk
awplus(config-if)#switchport trunk allowed vlan add 100,50,51
awplus(config-if)#switchport trunk native vlan none
awplus(config-if)#switchport vlan-stacking provider-port
```

Configure the EPSR domain:

```
awplus(config-if)#epsr configuration
awplus(config-epsr)#epsr awplus mode transit controlvlan 100
awplus(config-epsr)#epsr awplus datavlan 50-51
```

**Enable EPSR:**

```
awplus(config-epsr)#epsr awplus state enabled
awplus(config-epsr)#exit
```

**Step 4. Configure client switch E (connected to the master node).**

Enter global configuration mode and enter the following commands:

**Configure the end-user VLAN:**

```
awplus(config)#vlan database
awplus(config-vlan)#vlan 20 name customer20
```

**Configure an IP address on the end-user VLAN:**

```
awplus(config-vlan)#interface vlan20
awplus(config-if)#ip address 192.168.20.10/24
```

**Configure the port connected to the service provider network:**

```
awplus(config-if)#interface port1.0.20
awplus(config-if)#switchport mode trunk
awplus(config-if)#switchport trunk allowed vlan add 20
awplus(config-if)#end
```

**Step 5. Configure client switch F (connected to transit node B).**

Enter global configuration mode and enter the following commands:

```
awplus(config)#vlan database
awplus(config-vlan)#vlan 200 name customer200
awplus(config-vlan)#interface vlan200
awplus(config-if)#ip address 192.168.200.1/24
awplus(config-if)#interface port1.0.10
awplus(config-if)#switchport mode trunk
awplus(config-if)#switchport trunk allowed vlan add 20
awplus(config-if)#end
```

**Step 6. Configure client switch G (connected to transit node C).**

Enter global configuration mode and enter the following commands:

```
awplus(config)#vlan database
awplus(config-vlan)#vlan 20 name customer20
awplus(config-vlan)#interface vlan20
awplus(config-if)#ip address 192.168.20.1/24
awplus(config-if)#interface port1.0.20
awplus(config-if)#switchport mode trunk
awplus(config-if)#switchport trunk allowed vlan add 20
awplus(config-if)#end
```

**Step 7. Configure client switch H (connected to transit node D).**

Enter global configuration mode and enter the following commands:

```
awplus(config)#vlan database
awplus(config-vlan)#vlan 200 name customer200
awplus(config-vlan)#interface vlan200
awplus(config-if)#ip address 192.168.200.10/24
awplus(config-if)#interface port1.0.10
awplus(config-if)#switchport mode trunk
awplus(config-if)#switchport trunk allowed vlan add 200
awplus(config-if)#end
```

## Example 5: EPSR with an iMAP

This example is the same as "Example 1: A Basic Ring" on page 18 except that one of the three switches is an iMAP. We used an AT-TN7100 iMAP running 6.1.10. The ring ports on the iMAP are 5.0 and 5.1. The example first shows the configuration script for the iMAP as the master node, then as the transit node. For the configuration of the other two switches, see Example 1.

### Configure the AT-TN7100 iMAP as master node

The following screenshot shows a partial script for the iMAP, with the commands for configuring it as a EPSR master node and other relevant commands.

```
ADD IP INTERFACE=MGMT IPADDRESS=172.28.9.3 SUBNETMASK=255.255.255.0
CARD=ACTCFC GATEWAY=172.28.9.1
#
SET SWITCH AGEINGTIMER=300
#
SET SYSTEM PROVMODE=AUTO
SET SYSTEM GATEWAY=172.28.9.1
#
CREATE EPSR=awplus MASTER HELLOTIME=1 FAILOVERTIME=2 RINGFLAPTIME=0
#
CREATE VLAN=vlan2 VID=2 FORWARDINGMODE=STD
CREATE VLAN=vlan1000 VID=1000 FORWARDINGMODE=STD
#
ADD VLAN=2 INTERFACE=ETH:[5.0-1] FRAME=TAGGED
ADD VLAN=1000 INTERFACE=ETH:[5.0-1] FRAME=TAGGED
#
DELETE VLAN=1 INTERFACE=ETH:[5.0-1]
#
SET INTERFACE=ETH:[5.0-1] ACCEPTABLE=VLAN
#
ADD EPSR=awplus INTERFACE=ETH:[5.0] TYPE=PRIMARY
ADD EPSR=awplus INTERFACE=ETH:[5.1] TYPE=SECONDARY
ADD EPSR=awplus VLAN=1000 TYPE=CONTROL
ADD EPSR=awplus VLAN=2 TYPE=DATA
#
ENABLE EPSR=awplus
```

### Checking the master node configuration

To see a summary, use the command:

```
show epsr
```

The following screenshot shows the expected output.

```
--- EPSR Domain Information -----
```

EPSR Domain	Node Type	Domain Status/ State	Control Vlan	Interface(s) (PhysicalState, Type, State)
awplus	MASTER	EN/COMPLETE	1000	5.0 (UP,DNSTRM,FWDING ), 5.1 (UP,DNSTRM,BLOCKED)

```
-----
```



To see details, use the command:

```
show epsr=awplus
```

The following screenshot shows the expected output.

```

--- EPSR Domain Information -----
EPSR Domain Name..... awplus
EPSR Domain Node Type..... Master
EPSR Domain State..... COMPLETE
MAC Address of Master Node..... 00:00:CD:28:06:19
EPSR Domain Status..... Enabled
Control Vlan..... 1000
Primary Interface..... 5.0
Physical State of Primary Interface... UP
Primary Interface Type..... DOWNSTREAM
Primary Interface State..... FORWARDING
Secondary Interface..... 5.1
Physical State of Secondary Interface. UP
Secondary Interface Type..... DOWNSTREAM
Secondary Interface State..... BLOCKED
Hello Timer (seconds)..... 1
Failover Timer (seconds)..... 2
RingFlap Timer (seconds)..... 0
Hello Time Remaining (seconds)..... 1
Failover Time Remaining (seconds)..... 0
RingFlap Time Remaining (seconds)..... 0
Hello Sequence..... 526
Data Vlans..... 2
-----

```

## Configure the AT-TN7100 iMAP as a transit node

The following screenshot shows a partial script for the iMAP, with the commands for configuring it as a transit node.

```
CREATE EPSR=awplus TRANSIT
#
CREATE VLAN=vlan2 VID=2 FORWARDINGMODE=STD
CREATE VLAN=vlan1000 VID=1000 FORWARDINGMODE=STD
#
DISABLE INTERFACE=0.0-0.15,1.0-1.15,2.0-2.15,4.0-4.1,5.0-5.1
#
ADD VLAN=2 INTERFACE=ETH:[5.0-1] FRAME=TAGGED
ADD VLAN=1000 INTERFACE=ETH:[5.0-1] FRAME=TAGGED
#
DELETE VLAN=1 INTERFACE=ETH:[5.0-1]
#
SET INTERFACE=0.0-0.15,1.0-1.15,2.0-2.15,4.0-4.1,5.0-5.1
PROFILE=AutoProv
SET INTERFACE=ETH:[5.0-1] ACCEPTABLE=VLAN
#
ADD EPSR=awplus INTERFACE=ETH:[5.0-1]
ADD EPSR=awplus VLAN=1000 TYPE=CONTROL
ADD EPSR=awplus VLAN=2 TYPE=DATA
#
ENABLE EPSR=awplus
#
ENABLE INTERFACE=0.0-0.15,1.0-1.15,2.0-2.15,4.0-4.1,5.0-5.1
```

### Checking the transit node configuration

To see a summary, use the command:

```
show epsr
```

The following screenshot shows the expected output.

```
--- EPSR Domain Information -----
      EPSR Domain   Node Type Domain Status/   Control Interface(s) (PhysicalState,
                        State          Vlan          Type, State)
-----
awplus                TRANSIT   EN/LINKS-UP    1000   5.0 (UP,UPSTRM,FWDING ),
                                           5.1 (UP,DNSTRM,FWDING )
-----
```

To see details, use the command:

```
show epsr=awplus
```

The following screenshot shows the expected output.

```

--- EPSR Domain Information -----

EPSR Domain Name..... awplus
EPSR Domain Node Type..... Transit
EPSR Domain State..... LINKS-UP
MAC Address of Master Node..... 00:00:CD:24:02:4F
EPSR Domain Status..... Enabled
Control Vlan..... 1000
Ring Interface # 1..... 5.0
Physical State of Ring Interface # 1.. UP
Ring Interface # 1 Type..... UPSTREAM
Ring Interface # 1 State..... FORWARDING
Ring Interface # 2..... 5.1
Physical State of Ring Interface # 2.. UP
Ring Interface # 2 Type..... DOWNSTREAM
Ring Interface # 2 State..... FORWARDING
Hello Timer (seconds)..... N/A
Failover Timer (seconds)..... N/A
Ringflap Timer (seconds)..... N/A
Hello Time Remaining (seconds)..... N/A
Failover Time Remaining (seconds)..... N/A
Ringflap Time Remaining (seconds)..... N/A
Hello Sequence..... N/A
Data Vlans..... 2

-----

```

## Ports and Recovery Times

In practice, recovery time in an EPSR ring is generally between 50 and 100ms. However, it depends on the port type, because this determines how long it takes for the port to report that it is down and send a Link-Down message.

The following ports report that they are down immediately or within a few milliseconds, which leads to an EPSR recovery time of 50 to 100ms:

- 10/100M copper RJ-45 ports
- Tri-speed copper RJ-45 ports operating at 10 or 100M
- Fiber 1000M ports
- 10G ports

However, for tri-speed copper RJ-45 ports operating at 1000M, there is a short delay before the port reports that it is down. For almost all networks, this slight delay in recovery has no practical effect. However, for networks with extremely stringent failover requirements, we recommend using fiber 1000M ports instead of copper.

## IGMP Snooping and Recovery Times

From Software Version 5.3.2 onwards, IGMP snooping includes query solicitation, a feature that minimizes loss of multicast data after a topology change.

When IGMP snooping is enabled on a VLAN, and EPSR changes the underlying link layer topology of that VLAN, this can interrupt multicast data flow for a significant length of time. Query solicitation prevents this by monitoring the VLAN for any topology changes. When it detects a change, it generates a special IGMP Leave message known as a Query Solicit, and floods the Query Solicit message to all ports. When the IGMP Querier receives the message, it responds by sending a General Query. This refreshes snooped group membership information in the network.

Query solicitation functions by default (without you enabling it) on the EPSR master node. By default, the master node always sends a Query Solicit message when the topology changes.

On other switches in the network, the query solicitation is disabled by default, but you can enable it by using the command:

```
awplus(config)#ip igmp snooping tcn query solicit
```

If you enable query solicitation on an EPSR transit node, both that node and the master node send a Query Solicit message.

Once the Querier receives the Query Solicit message, it sends out a General Query and waits for responses, which update the snooping information throughout the network. If necessary, you can reduce the time this takes by tuning the IGMP timers, especially the general query messages using the ***ip igmp query-interval*** command.

Query solicitation also works with networks that use Spanning Tree (STP, RSTP, or MSTP).

### Query flooding protection

It is possible for an IGMP Querier to be flooded with Query Solicit packets and, in response, generate large numbers of IGMP Queries. This could potentially congest the network.

To prevent this kind of flooding, the AlliedWare Plus OS has an IGMP Query-Hold Interval. This is the time, starting from the last Query sent, that an IGMP Querier refrains from sending any more IGMP Queries. You can configure this time period on each VLAN interface, using the command:

```
awplus(config-if)#ip igmp query-holdtime <100-5000>
```

where <100-5000> is the time in milliseconds for the hold interval. The default is 500 milliseconds. This hold time is always enabled, and does not require Query Solicitation to be enabled.

## Health Message Priority

EPSR uses Health messages to check that the ring is intact. If switches in the ring were to drop Health messages, this could make the ring unstable. Therefore, Health messages are sent to the highest priority queue (queue 7), which uses strict priority scheduling by default. This makes sure that the switches forward Health messages even if the network is congested.

We recommend that you leave queue 7 as the highest priority queue, leave it using strict priority scheduling, and only send essential control traffic to it.

In the unlikely event that this is impossible, you can increase the failover time so that the master node only changes the ring topology if several Health messages in a row fail to arrive. By default, the failover time is set to two seconds, which means that the master node decides that the ring is down if two Health messages in a row fail to arrive.

## Topology Changes and Optimizing EPSR rings with Large Numbers of Data VLANs

When a Master node detects an outage (topology change) somewhere in the ring, it sends a Ring-Down-Flush message to its nodes telling them to delete the entries in their forwarding databases.

Once a fault in the ring or node has been rectified, the Master sends a Ring-Up-Flush message to the Transit nodes telling them to change their port states from blocking to forwarding (if necessary) and to delete entries in their forwarding databases. This restores normal conditions and allows data to flow again.

This flushing of Layer 2 entries could take some time for EPSR rings with large numbers of data VLANs. To help reduce latency caused during EPSR topology changes you can use the command:

```
awplus (config-epsr)# epsr [<epsr-name>] flush-type {interface|vlan}
```

This command is available from software version **5.4.9-1.1** onwards.

It allows you to set how EPSR flushes Layer 2 entries when a topology change occurs. It can be configured to flush all Layer 2 entries on its EPSR interfaces or only flush the Layer 2 entries on its EPSR data VLANs.

Select the **interface** parameter as the flush-type to help reduce latency, as this type of flushing is quicker and less granular than flushing per data VLAN.

Note that it will require relearning on any VLANs that are on an EPSR interface but not part of the EPSR configuration.

## EPSR State and Settings

To display the EPSR state, the attached VLANs, the ring ports, and the timer values, use the command:

```
show epsr
```

### Master node in a complete ring

The following screenshot shows the output for a master node in a ring that is in a state of Complete. As well as giving the state as Complete, it also shows that port1.0.1 is the primary port and port1.0.2 is the secondary port. The secondary port is blocked, so does not forward packets over the data VLAN (vlan2).

```

EPSR Information
-----
Name ..... test
Mode ..... Master
Status ..... Enabled
State ..... Complete
Control Vlan ..... 1000
Data Vlan(s) ..... 2
Primary Port ..... port1.0.1
Primary Port Status ..... Forwarding
Secondary Port ..... port1.0.2
Secondary Port Status ..... Blocked
Hello Time ..... 1 s
Failover Time ..... 2 s
Ring Flap Time ..... 0 s
Trap ..... Enabled
-----

```

### Master node in a failed ring

The following screenshot shows the output for a master node in a ring that is in a Failed state. Both ring ports are now forwarding.

```

EPSR Information
-----
Name ..... domain1
Mode ..... Master
Status ..... Enabled
State ..... Failed
Control Vlan ..... 1000
Data VLAN(s) ..... 2
Primary Port ..... port1.0.1
Primary Port Status ..... Forwarding
Secondary Port ..... port1.0.2
Secondary Port Status ..... Forwarding
Hello Time ..... 1 s
Failover Time ..... 2 s
Ring Flap Time ..... 0 s
Trap ..... Enabled
-----

```

## Transit node in a fully forwarding state

In contrast, the following screenshot shows the output for a transit node when both its ring ports are forwarding.

```

EPSR Information
-----
Name ..... test
Mode ..... Transit
Status ..... Enabled
State ..... Links-Up
Control Vlan ..... 1000
Data VLAN(s) ..... 2
First Port ..... port1.0.1
First Port Status ..... Forwarding
First Port Direction ..... Upstream
Second Port ..... port1.0.2
Second Port Status ..... Forwarding
Second Port Direction ..... Downstream
Trap ..... Enabled
Master Node ..... 00-00-cd-28-06-19
-----

```

## SNMP Traps

From Software Version 5.3.1 onwards, you can use SNMP traps to notify you when events occur in the EPSR ring.

The EPSR Group has the object identifier prefix epsrv2 (module 536), and contains a collection of objects and traps for monitoring EPSR states.

The following trap is defined under the epsrv2Events subtree:

- atEpsrv2NodeTrap is the trap type of the EPSR node (master/transit).

The following objects are defined under the epsrv2EventVariables subtree:

- atEpsrv2NodeType is the type of the EPSR node (master/transit).
- atEpsrv2DomainName is the name assigned to the EPSR domain.
- atEpsrv2DomainID is a domain index variable used by the AlliedWare Plus GUI.
- atEpsrv2FromState is the defined state that an EPSR domain is transitioning from.
- atEpsrv2CurrentState is the state that an EPSR domain is transitioning to.
- atEpsrv2ControlVlanId is the VLAN identifier for the control VLAN.
- atEpsrv2PrimaryIfIndex is the ifIndex of the primary interface.
- atEpsrv2PrimaryIfState is the current state of the primary interface.
- atEpsrv2SecondaryIfIndex is the ifIndex of the secondary interface.
- atEpsrv2SecondaryIfState is the current state of the secondary interface.



## Counters

The EPSR counters record the number of EPSR messages that the CPU received and transmitted. To display the counters, use the command:

```
show epsr <epsr-name> count
```

### Master node in a complete ring

The following screenshot shows the counters for a master node in a ring that has never had a link or node fail.

EPSR Counters			
-----			
Name: domain1			
Receive:		Transmit:	
Total EPSR Packets	1093	Total EPSR Packets	1093
Health	1092	Health	1092
Ring Up	1	Ring Up	1
Ring Down	0	Ring Down	0
Link Down	0	Link Down	0
Invalid EPSR Packets	0		
-----			

The node has generated 1093 EPSR packets (and sent them out its primary port) and has received the same number of EPSR packets (on its secondary port). However, it is very common to see a few Link Down, Ring Down, and Ring Up entries in the output of a ring that has never been in a Failed state. These messages are produced when you first enable EPSR, if some ring nodes establish before others.

### Transit node in a ring that had failures

In contrast, the following screenshot shows the counters for a transit node in a ring that has been in a Failed state twice.

EPSR Counters			
-----			
Name: domain1			
Receive:		Transmit:	
Total EPSR Packets	1425	Total EPSR Packets	2
Health	1421	Health	0
Ring Up	2	Ring Up	0
Ring Down	0	Ring Down	0
Link Down	0	Link Down	2
Invalid EPSR Packets	0		
-----			

Here, the transit node has received 1421 Health messages, which it will have forwarded on if its ports were up. These messages do not show in the transmit counters because they are transmitted by the switching hardware, not the CPU. The node has also generated two Link-Down messages, indicating that on two separate occasions one of its links has gone down.

## Debugging

This section walks you through the EPSR debugging output as links go down and come back up again. The debugging output comes from the ring in "Example 1: A Basic Ring" on page 18. The output shows what happened when we took down two separate links in turn:

- First, the link between the master node's primary port and transit node B
- Second, the link between the two transit nodes B and C

To enable debugging, enter the commands:

```
awplus#terminal monitor
awplus#debug epsr all
```

The ***terminal monitor*** command causes the switch to display **terminal logging** messages on the console. By default, debug messages are terminal logging messages. You can change this by using the ***log terminal*** command in global configuration mode. You can see which messages are saved into each type of log by using the ***show log config*** command.

**Note:** The master node transmits Health messages every second by default. The debugging displays every message, including all Health messages. Therefore, we recommend that you capture the debugging output for separate analysis, to make analysis simpler.

### Link down between master node and transit node

This section shows the debugging output when the link between the master node's primary port and transit node B goes down and comes back up again. It shows the debugging output for the complete failure and recovery cycle:

- First on the master node
- Then on the transit node

#### Master node (Node A) debug output

The following debugging output starts with the ring established and in a state of Complete.

**Step 1. The master node sends Health messages.**

Each time the Hello timer expires, the master node sends a Health message out its primary port (port1.0.1). As long as the ring is in a state of Complete, it receives each Health message again on its secondary port (port1.0.2). In the System field, this output shows the MAC address of the source of the message—the master node in this case.

```

13:49:18 EPSR[1296]: EPSR: epsrHelloTimeout: EPSR awplus Hello Timer expired
13:49:18 EPSR[1296]: EPSR: port1.0.1 Tx:
13:49:18 EPSR[1296]: EPSR: 00e02b00 00040000 cd240331 8100e3e8 005caaaa 0300e02b
13:49:18 EPSR[1296]: EPSR: 00bb0100 005405f1 00000000 0000cd24 0331990b 00400105
13:49:18 EPSR[1296]: EPSR: 03e80000 00000000 cd240331 00010002 01001fcf 00000000
13:49:18 EPSR[1296]: EPSR: port1.0.1 Tx:
13:49:18 EPSR[1296]: EPSR: -----
13:49:18 EPSR[1296]: EPSR:          TYPE = HEALTH                      STATE = COMPLETE
13:49:18 EPSR[1296]: EPSR:          CTRL VLAN = 1000                SYSTEM = 00-00-cd-24-03-31
13:49:18 EPSR[1296]: EPSR:          HELLO TIME = 1                  FAIL TIME = 2
13:49:18 EPSR[1296]: EPSR:          HELLO SEQ = 8143
13:49:18 EPSR[1296]: EPSR: -----
13:49:18 EPSR[1296]: EPSR: port1.0.2 Rx:
13:49:18 EPSR[1296]: EPSR: 00e02b00 00040000 cd240331 810003e8 005caaaa 0300e02b
13:49:18 EPSR[1296]: EPSR: 00bb0100 005405f1 00000000 0000cd24 0331990b 00400105
13:49:18 EPSR[1296]: EPSR: 03e80000 00000000 cd240331 00010002 01001fcf 00000000
13:49:18 EPSR[1296]: EPSR: port1.0.2 Rx:
13:49:18 EPSR[1296]: EPSR: -----
13:49:18 EPSR[1296]: EPSR:          TYPE = HEALTH                      STATE = COMPLETE
13:49:18 EPSR[1296]: EPSR:          CTRL VLAN = 1000                SYSTEM = 00-00-cd-24-03-31
13:49:18 EPSR[1296]: EPSR:          HELLO TIME = 1                  FAIL TIME = 2
13:49:18 EPSR[1296]: EPSR:          HELLO SEQ = 8143
13:49:18 EPSR[1296]: EPSR: -----

```

**Step 2. The master node continues sending Health messages.**

The master node continues sending Health messages, and increments the Hello Sequence number with each message. If all nodes and links in the ring are intact, these Health messages are the only debugging output you see.

```
.
.
.
13:52:10 EPSR[1296]: EPSR: epsrHelloTimeout: EPSR awplus Hello Timer expired
13:52:10 EPSR[1296]: EPSR: port1.0.1 Tx:
13:52:10 EPSR[1296]: EPSR: 00e02b00 00040000 cd240331 8100e3e8 005caaaa 0300e02b
13:52:10 EPSR[1296]: EPSR: 00bb0100 00540545 00000000 0000cd24 0331990b 00400105
13:52:10 EPSR[1296]: EPSR: 03e80000 00000000 cd240331 00010002 0100207b 00000000
13:52:10 EPSR[1296]: EPSR: port1.0.1 Tx:
13:52:10 EPSR[1296]: EPSR: -----
13:52:10 EPSR[1296]: EPSR:          TYPE = HEALTH                      STATE = COMPLETE
13:52:10 EPSR[1296]: EPSR:          CTRL VLAN = 1000                SYSTEM = 00-00-cd-24-03-31
13:52:10 EPSR[1296]: EPSR:          HELLO TIME = 1                  FAIL TIME = 2
13:52:10 EPSR[1296]: EPSR:          HELLO SEQ = 8420
13:52:10 EPSR[1296]: EPSR: -----
13:52:10 EPSR[1296]: EPSR: port1.0.2 Rx:
13:52:10 EPSR[1296]: EPSR: 00e02b00 00040000 cd240331 810003e8 005caaaa 0300e02b
13:52:10 EPSR[1296]: EPSR: 00bb0100 00540545 00000000 0000cd24 0331990b 00400105
13:52:10 EPSR[1296]: EPSR: 03e80000 00000000 cd240331 00010002 0100207b 00000000
13:52:10 EPSR[1296]: EPSR: port1.0.2 Rx:
13:52:10 EPSR[1296]: EPSR: -----
13:52:10 EPSR[1296]: EPSR:          TYPE = HEALTH                      STATE = COMPLETE
13:52:10 EPSR[1296]: EPSR:          CTRL VLAN = 1000                SYSTEM = 00-00-cd-24-03-31
13:52:10 EPSR[1296]: EPSR:          HELLO TIME = 1                  FAIL TIME = 2
13:52:10 EPSR[1296]: EPSR:          HELLO SEQ = 8420
13:52:10 EPSR[1296]: EPSR: -----
```

**Step 3. The primary port goes down.**

The link between the master node's primary port and the neighboring transit node goes down. Therefore, the master node detects that its primary port (port1.0.1) has gone down.

```
13:53:45 EPSR[1296]: EPSR: EPSR awplus, ifIndex 5001 down
13:53:45 EPSR[1296]: EPSR: Flush FDB EPSR: awplus vid: 2
13:53:45 EPSR[1296]: EPSR: EPSR awplus: port 5001 is down
```

**Step 4. The master node receives a Link-Down message on its secondary port.**

The master node receives a Link-Down message on its secondary port (port1.0.2) from transit node B, which is at the other end of the broken link.

```

13:53:45 EPSR[1296]: EPSR: port1.0.2 Rx:
13:53:45 EPSR[1296]: EPSR: 00e02b00 00040000 cd127808 810003e8 005caaaa 0300e02b
13:53:45 EPSR[1296]: EPSR: 00bb0100 00543935 00000000 0000cd12 7808990b 00400108
13:53:45 EPSR[1296]: EPSR: 03e80000 00000000 cd127808 00000000 04000000 00000000
13:53:45 EPSR[1296]: EPSR: port1.0.2 Rx:
13:53:45 EPSR[1296]: EPSR: -----
13:53:45 EPSR[1296]: EPSR:      TYPE = LINK-DOWN                STATE = LINK-DOWN
13:53:45 EPSR[1296]: EPSR:      CTRL VLAN = 1000                SYSTEM = 00-00-cd-12-78-08
13:53:45 EPSR[1296]: EPSR:      HELLO TIME = 0                FAIL TIME = 0
13:53:45 EPSR[1296]: EPSR:      HELLO SEQ = 0
13:53:45 EPSR[1296]: EPSR: -----
13:53:45 EPSR[1296]: EPSR: EPSR awplus: link down msg from 00-00-cd-12-78-08

```

In the System field, this output shows the MAC address of the source of the message—the transit node in this case.

**Step 5. The master node transmits a Ring-Down-Flush-FDB message.**

The master switch responds to the break in the ring by sending a Ring-Down-Flush-FDB message, which tells each transit node to learn the new topology. The master node also unblocks its secondary port for the data VLAN (vlan2), flushes its FDB, sends an SNMP trap, and changes the EPSR state to Failed. The master node sends the Ring-Down-Flush-FDB message only out its secondary port, because the link between the primary port and the neighboring transit node is down.

```

13:53:45 EPSR[1296]: EPSR: port1.0.2 Tx:
13:53:45 EPSR[1296]: EPSR: 00e02b00 00040000 cd240331 8100e3e8 005caaaa 0300e02b
13:53:45 EPSR[1296]: EPSR: 00bb0100 005424c1 00000000 0000cd24 0331990b 00400107
13:53:45 EPSR[1296]: EPSR: 03e80000 00000000 cd240331 00000000 02000000 00000000
13:53:45 EPSR[1296]: EPSR: port1.0.2 Tx:
13:53:45 EPSR[1296]: EPSR: -----
13:53:45 EPSR[1296]: EPSR:      TYPE = RING-DOWN-FLUSH-FDB          STATE = FAILED
13:53:45 EPSR[1296]: EPSR:      CTRL VLAN = 1000                SYSTEM = 00-00-cd-24-03-31
13:53:45 EPSR[1296]: EPSR:      HELLO TIME = 0                FAIL TIME = 0
13:53:45 EPSR[1296]: EPSR:      HELLO SEQ = 0
13:53:45 EPSR[1296]: EPSR: -----
13:53:45 EPSR[1296]: EPSR: Unblock EPSR:awplus port:5002 VLAN:2
13:53:45 EPSR[1296]: EPSR: EPSR awplus: port 5002 is forwarding
13:53:45 EPSR[1296]: EPSR: Flush FDB EPSR: awplus vid: 2
13:53:45 EPSR[1296]: EPSR: awplus oldState:COMPLETE newState:FAILED
13:53:45 EPSR[1296]: EPSR: EPSR awplus: ring failed

```

**Step 6. The Hello timer expires.**

The Hello timer expires, which would normally trigger the master node to send a Health message out the primary port. However, the link between the primary port and the neighboring transit node is down, so the master node does not send the Health message.

```
13:53:46 EPSR[1296]: EPSR: epsrHelloTimeout: EPSR awplus Hello Timer expired
13:53:47 EPSR[1296]: EPSR: epsrHelloTimeout: EPSR awplus Hello Timer expired
13:53:48 EPSR[1296]: EPSR: epsrHelloTimeout: EPSR awplus Hello Timer expired
13:53:49 EPSR[1296]: EPSR: epsrHelloTimeout: EPSR awplus Hello Timer expired
```

**Step 7. The primary port comes back up.**

The primary port comes back up. The master node immediately blocks that port for vlan2 to prevent a loop.

```
14:05:03 EPSR[1296]: EPSR: EPSR awplus, ifIndex 5001 up
14:05:03 EPSR[1296]: EPSR: Block EPSR:awplus port:5001 VLAN:2
14:05:03 EPSR[1296]: EPSR: EPSR awplus: port 5001 is blocking
```

**Step 8. The Hello timer expires again.**

The Hello timer expires again. Port1.0.1 is now up, so this time the master node sends a Health message. The Health message shows that the EPSR state is Failed.

The Hello Sequence number increments from the number it was before the primary port went down, because the master node could not transmit Health messages while the port was down.

```
14:05:03 EPSR[1296]: EPSR: epsrHelloTimeout: EPSR awplus Hello Timer expired
14:05:03 EPSR[1296]: EPSR: port1.0.1 Tx:
14:05:03 EPSR[1296]: EPSR: 00e02b00 00040000 cd240331 8100e3e8 005caaaa 0300e02b
14:05:03 EPSR[1296]: EPSR: 00bb0100 005403db 00000000 0000cd24 0331990b 00400105
14:05:03 EPSR[1296]: EPSR: 03e80000 00000000 cd240331 00010002 020020e5 00000000
14:05:03 EPSR[1296]: EPSR: port1.0.1 Tx:
14:05:03 EPSR[1296]: EPSR: -----
14:05:03 EPSR[1296]: EPSR:      TYPE = HEALTH                      STATE = FAILED
14:05:03 EPSR[1296]: EPSR:      CTRL VLAN = 1000                SYSTEM = 00-00-cd-24-03-31
14:05:03 EPSR[1296]: EPSR:      HELLO TIME = 1                  FAIL TIME = 2
14:05:03 EPSR[1296]: EPSR:      HELLO SEQ = 8421
14:05:03 EPSR[1296]: EPSR: -----
```

**Step 9. The master node receives the Health message on its secondary port.**

The master node receives the Health message on its secondary port (port1.0.2). This tells it that all links on the ring are up again.

```

14:05:03 EPSR[1296]: EPSR: port1.0.2 Rx:
14:05:03 EPSR[1296]: EPSR: 00e02b00 00040000 cd240331 810003e8 005caaaa 0300e02b
14:05:03 EPSR[1296]: EPSR: 00bb0100 005403db 00000000 0000cd24 0331990b 00400105
14:05:03 EPSR[1296]: EPSR: 03e80000 00000000 cd240331 00010002 020020e5 00000000
14:05:03 EPSR[1296]: EPSR: port1.0.2 Rx:
14:05:03 EPSR[1296]: EPSR: -----
14:05:03 EPSR[1296]: EPSR:         TYPE = HEALTH                      STATE = FAILED
14:05:03 EPSR[1296]: EPSR:         CTRL VLAN = 1000                SYSTEM = 00-00-cd-24-03-31
14:05:03 EPSR[1296]: EPSR:         HELLO TIME = 1                  FAIL TIME = 2
14:05:03 EPSR[1296]: EPSR:         HELLO SEQ = 8421
14:05:03 EPSR[1296]: EPSR: -----

```

**Step 10. The master node returns the ring to a state of Complete.**

The master node blocks its secondary port for the data VLAN, unblocks its primary port, transmits a Ring-Up-Flush-FDB message, flushes its FDB, sends a trap, and changes the EPSR state to Complete.

```

14:05:03 EPSR[1296]: EPSR: Block EPSR:awplus port:5002 VLAN:2
14:05:03 EPSR[1296]: EPSR: EPSR awplus: port 5002 is blocking
14:05:03 EPSR[1296]: EPSR: Unblock EPSR:awplus port:5001 VLAN:2
14:05:03 EPSR[1296]: EPSR: EPSR awplus: port 5001 is forwarding
14:05:03 EPSR[1296]: EPSR: port1.0.1 Tx:
14:05:03 EPSR[1296]: EPSR: 00e02b00 00040000 cd240331 8100e3e8 005caaaa 0300e02b
14:05:03 EPSR[1296]: EPSR: 00bb0100 005425c2 00000000 0000cd24 0331990b 00400106
14:05:03 EPSR[1296]: EPSR: 03e80000 00000000 cd240331 00000000 01000000 00000000
14:05:03 EPSR[1296]: EPSR: port1.0.1 Tx:
14:05:03 EPSR[1296]: EPSR: -----
14:05:03 EPSR[1296]: EPSR:         TYPE = RING-UP-FLUSH-FDB          STATE = COMPLETE
14:05:03 EPSR[1296]: EPSR:         CTRL VLAN = 1000                SYSTEM = 00-00-cd-
24-03-31
14:05:03 EPSR[1296]: EPSR:         HELLO TIME = 0                  FAIL TIME = 0
14:05:03 EPSR[1296]: EPSR:         HELLO SEQ = 0
14:05:03 EPSR[1296]: EPSR: -----
14:05:03 EPSR[1296]: EPSR: Flush FDB EPSR: awplus vid: 2
14:05:03 EPSR[1296]: EPSR: awplus oldState:FAILED newState:COMPLETE
14:05:03 EPSR[1296]: EPSR: EPSR awplus: ring complete

```

**Step 11. The master node receives the Ring-Up-Flush-FDB message on port1.0.2.**

The master node receives the Ring-Up-Flush-FDB message back on its secondary port, because the packet traversed the whole ring. The master node ignores the message.

```

14:05:03 EPSR[1296]: EPSR: port1.0.2 Rx:
14:05:03 EPSR[1296]: EPSR: 00e02b00 00040000 cd240331 810003e8 005caaaa 0300e02b
14:05:03 EPSR[1296]: EPSR: 00bb0100 005425c2 00000000 0000cd24 0331990b 00400106
14:05:03 EPSR[1296]: EPSR: 03e80000 00000000 cd240331 00000000 01000000 00000000
14:05:03 EPSR[1296]: EPSR: port1.0.2 Rx:
14:05:03 EPSR[1296]: EPSR: -----
14:05:03 EPSR[1296]: EPSR:         TYPE = RING-UP-FLUSH-FDB             STATE = COMPLETE
14:05:03 EPSR[1296]: EPSR:         CTRL VLAN = 1000                SYSTEM = 00-00-cd-24-03-31
14:05:03 EPSR[1296]: EPSR:         HELLO TIME = 0                    FAIL TIME = 0
14:05:03 EPSR[1296]: EPSR:         HELLO SEQ = 0
14:05:03 EPSR[1296]: EPSR: -----

```

**Step 12. The master node transmits and receives Health messages.**

The master node continues transmitting and receiving Health messages for as long as the ring stays in a state of Complete.

```

14:05:04 EPSR[1296]: EPSR: epsrHelloTimeout: EPSR awplus Hello Timer expired
14:05:04 EPSR[1296]: EPSR: port1.0.1 Tx:
14:05:04 EPSR[1296]: EPSR: 00e02b00 00040000 cd240331 8100e3e8 005caaaa 0300e02b
14:05:04 EPSR[1296]: EPSR: 00bb0100 005404da 00000000 0000cd24 0331990b 00400105
14:05:04 EPSR[1296]: EPSR: 03e80000 00000000 cd240331 00010002 010020e6 00000000
14:05:04 EPSR[1296]: EPSR: port1.0.1 Tx:
14:05:04 EPSR[1296]: EPSR: -----
14:05:04 EPSR[1296]: EPSR:         TYPE = HEALTH                STATE = COMPLETE
14:05:04 EPSR[1296]: EPSR:         CTRL VLAN = 1000                SYSTEM = 00-00-cd-24-03-31
14:05:04 EPSR[1296]: EPSR:         HELLO TIME = 1                    FAIL TIME = 2
14:05:04 EPSR[1296]: EPSR:         HELLO SEQ = 8422
14:05:04 EPSR[1296]: EPSR: -----
14:05:04 EPSR[1296]: EPSR: port1.0.2 Rx:
14:05:04 EPSR[1296]: EPSR: 00e02b00 00040000 cd240331 810003e8 005caaaa 0300e02b
14:05:04 EPSR[1296]: EPSR: 00bb0100 005404da 00000000 0000cd24 0331990b 00400105
14:05:04 EPSR[1296]: EPSR: 03e80000 00000000 cd240331 00010002 010020e6 00000000
14:05:04 EPSR[1296]: EPSR: port1.0.2 Rx:
14:05:04 EPSR[1296]: EPSR: -----
14:05:04 EPSR[1296]: EPSR:         TYPE = HEALTH                STATE = COMPLETE
14:05:04 EPSR[1296]: EPSR:         CTRL VLAN = 1000                SYSTEM = 00-00-cd-24-03-31
14:05:04 EPSR[1296]: EPSR:         HELLO TIME = 1                    FAIL TIME = 2
14:05:04 EPSR[1296]: EPSR:         HELLO SEQ = 8422
14:05:04 EPSR[1296]: EPSR: -----

```



**Transit node (Node B) debug output**

The following debugging shows the same sequence of events as the previous section, but on the transit node instead of the master node. It starts with the ring established and in a state of Complete.

**Note:** The following debug was captured at a different time (during a different ring-down event) from the master node debug in the previous section. This is why the times and hello sequence numbers do not match.

**Step 1. The transit node receives Health messages.**

The transit node receives Health messages on port1.0.1, because that port is connected to the master node's primary port. In the System field, this output shows the MAC address of the source of the message—the master node in this case.

```

10:45:38 EPSR[1284]: EPSR: port1.0.1 Rx:
10:45:38 EPSR[1284]: EPSR: 00e02b00 00040000 cd240331 810003e8 0058aaaa 0300e02b
10:45:38 EPSR[1284]: EPSR: 00bb0100 005422a9 00000000 0000cd24 0331990b 00400105
10:45:38 EPSR[1284]: EPSR: 03e80000 00000000 cd240331 00010002 0100052d 00000000
10:45:38 EPSR[1284]: EPSR: port1.0.1 Rx:
10:45:38 EPSR[1284]: EPSR: -----
10:45:38 EPSR[1284]: EPSR:          TYPE = HEALTH                      STATE = COMPLETE
10:45:38 EPSR[1284]: EPSR:          CTRL VLAN = 1000                SYSTEM = 00-00-cd-24-03-31
10:45:38 EPSR[1284]: EPSR:          HELLO TIME = 1                      FAIL TIME = 2
10:45:38 EPSR[1284]: EPSR:          HELLO SEQ = 1325
10:45:38 EPSR[1284]: EPSR: -----
10:45:39 EPSR[1284]: EPSR: port1.0.1 Rx:
10:45:39 EPSR[1284]: EPSR: 00e02b00 00040000 cd240331 810003e8 0058aaaa 0300e02b
10:45:39 EPSR[1284]: EPSR: 00bb0100 005422a8 00000000 0000cd24 0331990b 00400105
10:45:39 EPSR[1284]: EPSR: 03e80000 00000000 cd240331 00010002 0100052e 00000000
10:45:39 EPSR[1284]: EPSR: port1.0.1 Rx:
10:45:39 EPSR[1284]: EPSR: -----
10:45:39 EPSR[1284]: EPSR:          TYPE = HEALTH                      STATE = COMPLETE
10:45:39 EPSR[1284]: EPSR:          CTRL VLAN = 1000                SYSTEM = 00-00-cd-24-03-31
10:45:39 EPSR[1284]: EPSR:          HELLO TIME = 1                      FAIL TIME = 2
10:45:39 EPSR[1284]: EPSR:          HELLO SEQ = 1326
10:45:39 EPSR[1284]: EPSR: -----

```

**Step 2. Port1.0.1 on the transit node goes down.**

The transit node detects that port1.0.1 (between the transit node and the master node) has gone down. The transit node flushes its forwarding database, blocks port1.0.1 for the data VLAN (to prevent a loop from forming when the master node comes back up), sends a Link-Down message towards the master node, sends a trap, and changes the EPSR state to Link-Down.

```

10:46:10 EPSR[1284]: EPSR: EPSR awplus, ifIndex 5001 down
10:46:10 EPSR[1284]: EPSR: Flush FDB EPSR: awplus vid: 2
10:46:10 EPSR[1284]: EPSR: EPSR awplus: port 5001 is down
10:46:10 EPSR[1284]: EPSR: Block EPSR:awplus port:5001 VLAN:2
10:46:10 EPSR[1284]: EPSR: port1.0.2 Tx:
10:46:10 EPSR[1284]: EPSR: 00e02b00 00040000 cd247808 8100e3e8 005caaaa 0300e02b
10:46:10 EPSR[1284]: EPSR: 00bb0100 005422c0 00000000 0000cd24 7808990b 00400108
10:46:10 EPSR[1284]: EPSR: 03e80000 00000000 cd247808 00000000 04000000 00000000
10:46:10 EPSR[1284]: EPSR: port1.0.2 Tx:
10:46:10 EPSR[1284]: EPSR: -----
10:46:10 EPSR[1284]: EPSR:          TYPE = LINK-DOWN                STATE = LINK-DOWN
10:46:10 EPSR[1284]: EPSR:          CTRL VLAN = 1000            SYSTEM = 00-00-cd-24-78-08
10:46:10 EPSR[1284]: EPSR:          HELLO TIME = 0              FAIL TIME = 0
10:46:10 EPSR[1284]: EPSR:          HELLO SEQ = 0
10:46:10 EPSR[1284]: EPSR: -----
10:46:10 EPSR[1284]: EPSR: awplus oldState:LINK-UP newState:LINK-DOWN

```

**Step 3. The transit node receives a Ring-Down-Flush-FDB message.**

In response to the Link-Down message, the master node sends a Ring-Down-Flush-FDB message. However, this transit node does not need to flush its database—it already did.

```

10:46:10 EPSR[1284]: EPSR: port1.0.2 Rx:
10:46:10 EPSR[1284]: EPSR: 00e02b00 00040000 cd240331 810003e8 0058aaaa 0300e02b
10:46:10 EPSR[1284]: EPSR: 00bb0100 005426d7 00000000 0000cd24 0331990b 00400107
10:46:10 EPSR[1284]: EPSR: 03e80000 00000000 cd240331 00000000 02000000 00000000
10:46:10 EPSR[1284]: EPSR: port1.0.2 Rx:
10:46:10 EPSR[1284]: EPSR: -----
10:46:10 EPSR[1284]: EPSR:          TYPE = RING-DOWN-FLUSH-FDB        STATE = FAILED
10:46:10 EPSR[1284]: EPSR:          CTRL VLAN = 1000            SYSTEM = 00-00-cd-24-03-31
10:46:10 EPSR[1284]: EPSR:          HELLO TIME = 0              FAIL TIME = 0
10:46:10 EPSR[1284]: EPSR:          HELLO SEQ = 0
10:46:10 EPSR[1284]: EPSR: -----

```

**Step 4. Port1.0.1 comes back up.**

The transit node detects that port1.0.1 has come back up. It sends a trap and changes the EPSR state to Pre-forwarding. It leaves port1.0.1 blocked for vlan2, to make sure there are no loops.

```
10:47:27 EPSR[1284]: EPSR: Block EPSR:awplus port:5001 VLAN:2
10:47:27 EPSR[1284]: EPSR: EPSR awplus: port 5001 is blocking
10:47:27 EPSR[1284]: EPSR: EPSR awplus, ifIndex 5001 up
10:47:27 EPSR[1284]: EPSR: awplus oldState:LINK-DOWN newState:PRE-FORWARDING
```

**Step 5. The transit node receives a Health message.**

Now that the master node's primary port is up again, it sends a Health message. Now that the transit node's port1.0.1 is up again for the control VLAN, the transit node receives the message. This demonstrates that the transit node has only blocked port1.0.1 for the data VLAN, not the control VLAN. EPSR control messages never loop because the master node never forwards them between its ring ports.

The Hello Sequence number increments from the number it was before the primary port went down, because the master node could not transmit Health messages while the port was down.

```
10:47:28 EPSR[1284]: EPSR: port1.0.1 Rx:
10:47:28 EPSR[1284]: EPSR: 00e02b00 00040000 cd240331 810003e8 0058aaaa 0300e02b
10:47:28 EPSR[1284]: EPSR: 00bb0100 00542188 00000000 0000cd24 0331990b 00400105
10:47:28 EPSR[1284]: EPSR: 03e80000 00000000 cd240331 00010002 0200054e 00000000
10:47:28 EPSR[1284]: EPSR: port1.0.1 Rx:
10:47:28 EPSR[1284]: EPSR: -----
10:47:28 EPSR[1284]: EPSR:      TYPE = HEALTH                      STATE = FAILED
10:47:28 EPSR[1284]: EPSR:      CTRL VLAN = 1000                SYSTEM = 00-00-cd-24-03-31
10:47:28 EPSR[1284]: EPSR:      HELLO TIME = 1                  FAIL TIME = 2
10:47:28 EPSR[1284]: EPSR:      HELLO SEQ = 1358
10:47:28 EPSR[1284]: EPSR: -----
```

**Step 6. The transit node receives a Ring-Up-Flush-FDB message.**

The Health message from the previous step reaches the master node and shows it that all links in the ring are now up. The master node sends a Ring-Up-Flush-FDB message. When it receives the message, the transit node unblocks port1.0.1 for vlan2, flushes its FDB, sends a trap, and changes the state to Link-Up.

This is equivalent to the packet shown in step 10 of the master node debug output for ["Link Down between two transit nodes"](#) on page 53.

```

10:47:28 EPSR[1284]: EPSR: port1.0.1 Rx:
10:47:28 EPSR[1284]: EPSR: 00e02b00 00040000 cd240331 810003e8 0058aaaa 0300e02b
10:47:28 EPSR[1284]: EPSR: 00bb0100 005427d8 00000000 0000cd24 0331990b 00400106
10:47:28 EPSR[1284]: EPSR: 03e80000 00000000 cd240331 00000000 01000000 00000000
10:47:28 EPSR[1284]: EPSR: port1.0.1 Rx:
10:47:28 EPSR[1284]: EPSR: -----
10:47:28 EPSR[1284]: EPSR:      TYPE = RING-UP-FLUSH-FDB      STATE = COMPLETE
10:47:28 EPSR[1284]: EPSR:      CTRL VLAN = 1000      SYSTEM = 00-00-cd-24-03-31
10:47:28 EPSR[1284]: EPSR:      HELLO TIME = 0      FAIL TIME = 0
10:47:28 EPSR[1284]: EPSR:      HELLO SEQ = 0
10:47:28 EPSR[1284]: EPSR: -----
10:47:28 EPSR[1284]: EPSR: Unblock EPSR:awplus port:5001 VLAN:2
10:47:28 EPSR[1284]: EPSR: EPSR awplus: port 5001 is forwarding
10:47:28 EPSR[1284]: EPSR: Flush FDB EPSR: awplus vid: 2
10:47:28 EPSR[1284]: EPSR: awplus oldState:PRE-FORWARDING newState:LINK-UP

```

**Step 7. The transit node receives Health messages.**

The transit node continues receiving Health messages for as long as the ring stays in a state of Complete.

```

10:47:29 EPSR[1284]: EPSR: port1.0.1 Rx:
10:47:29 EPSR[1284]: EPSR: 00e02b00 00040000 cd240331 810003e8 0058aaaa 0300e02b
10:47:29 EPSR[1284]: EPSR: 00bb0100 00542287 00000000 0000cd24 0331990b 00400105
10:47:29 EPSR[1284]: EPSR: 03e80000 00000000 cd240331 00010002 0100054f 00000000
10:47:29 EPSR[1284]: EPSR: port1.0.1 Rx:
10:47:29 EPSR[1284]: EPSR: -----
10:47:29 EPSR[1284]: EPSR:      TYPE = HEALTH      STATE = COMPLETE
10:47:29 EPSR[1284]: EPSR:      CTRL VLAN = 1000      SYSTEM = 00-00-cd-24-03-31
10:47:29 EPSR[1284]: EPSR:      HELLO TIME = 1      FAIL TIME = 2
10:47:29 EPSR[1284]: EPSR:      HELLO SEQ = 1359
10:47:29 EPSR[1284]: EPSR: -----
10:47:30 EPSR[1284]: EPSR: port1.0.1 Rx:
10:47:30 EPSR[1284]: EPSR: 00e02b00 00040000 cd240331 810003e8 0058aaaa 0300e02b
10:47:30 EPSR[1284]: EPSR: 00bb0100 00542286 00000000 0000cd24 0331990b 00400105
10:47:30 EPSR[1284]: EPSR: 03e80000 00000000 cd240331 00010002 01000550 00000000
10:47:30 EPSR[1284]: EPSR: port1.0.1 Rx:
10:47:30 EPSR[1284]: EPSR: -----
10:47:30 EPSR[1284]: EPSR:      TYPE = HEALTH      STATE = COMPLETE
10:47:30 EPSR[1284]: EPSR:      CTRL VLAN = 1000      SYSTEM = 00-00-cd-24-03-31
10:47:30 EPSR[1284]: EPSR:      HELLO TIME = 1      FAIL TIME = 2
10:47:30 EPSR[1284]: EPSR:      HELLO SEQ = 1360
10:47:30 EPSR[1284]: EPSR: -----

```

## Link Down between two transit nodes

This section shows the debugging output when the link between transit node B and transit node C goes down and comes back up again. It shows the debugging output for a complete failure and recovery cycle:

- On the master node, and then
- On transit node B.

### Master node (Node A) debug output

The following debugging output starts with the ring established and in a state of Complete.

#### Step 1. The master node sends Health messages.

Each time the Hello timer expires, the master node sends a Health message out its primary port (port1.0.1). As long as the ring is in a state of Complete, it receives each Health message again on its secondary port (port1.0.2).

```

10:52:14 EPSR[1283]: EPSR: epsrHelloTimeout: EPSR awplus Hello Timer expired
10:52:14 EPSR[1283]: EPSR: port1.0.1 Tx:
10:52:14 EPSR[1283]: EPSR: 00e02b00 00040000 cd240331 8100e3e8 005caaaa 0300e02b
10:52:14 EPSR[1283]: EPSR: 00bb0100 005425a3 00000000 0000cd24 0331990b 00400105
10:52:14 EPSR[1283]: EPSR: 03e80000 00000000 cd240331 00010002 0100001d 00000000
10:52:14 EPSR[1283]: EPSR: port1.0.1 Tx:
10:52:14 EPSR[1283]: EPSR: -----
10:52:14 EPSR[1283]: EPSR:         TYPE = HEALTH                     STATE = COMPLETE
10:52:14 EPSR[1283]: EPSR:         CTRL VLAN = 1000                SYSTEM = 00-00-cd-24-03-31
10:52:14 EPSR[1283]: EPSR:         HELLO TIME = 1                     FAIL TIME = 2
10:52:14 EPSR[1283]: EPSR:         HELLO SEQ = 29
10:52:14 EPSR[1283]: EPSR: -----
10:52:14 EPSR[1283]: EPSR: port1.0.2 Rx:
10:52:14 EPSR[1283]: EPSR: 00e02b00 00040000 cd240331 810003e8 0058aaaa 0300e02b
10:52:14 EPSR[1283]: EPSR: 00bb0100 005425a3 00000000 0000cd24 0331990b 00400105
10:52:14 EPSR[1283]: EPSR: 03e80000 00000000 cd240331 00010002 0100001d 00000000
10:52:14 EPSR[1283]: EPSR: port1.0.2 Rx:
10:52:14 EPSR[1283]: EPSR: -----
10:52:14 EPSR[1283]: EPSR:         TYPE = HEALTH                     STATE = COMPLETE
10:52:14 EPSR[1283]: EPSR:         CTRL VLAN = 1000                SYSTEM = 00-00-cd-24-03-31
10:52:14 EPSR[1283]: EPSR:         HELLO TIME = 1                     FAIL TIME = 2
10:52:14 EPSR[1283]: EPSR:         HELLO SEQ = 29
10:52:14 EPSR[1283]: EPSR: -----

```

**Step 2. The link between the two transit nodes goes down.**

When the link goes down, the master node transmits a Health message but does not receive it on its secondary port.

```

10:53:21 EPSR[1283]: EPSR: port1.0.1 Tx:
10:53:21 EPSR[1283]: EPSR: 00e02b00 00040000 cd240331 8100e3e8 005caaaa 0300e02b
10:53:21 EPSR[1283]: EPSR: 00bb0100 00542560 00000000 0000cd24 0331990b 00400105
10:53:21 EPSR[1283]: EPSR: 03e80000 00000000 cd240331 00010002 01000060 00000000
10:53:21 EPSR[1283]: EPSR: port1.0.1 Tx:
10:53:21 EPSR[1283]: EPSR: -----
10:53:21 EPSR[1283]: EPSR:         TYPE = HEALTH                      STATE = COMPLETE
10:53:21 EPSR[1283]: EPSR:         CTRL VLAN = 1000                SYSTEM = 00-00-cd-24-03-31
10:53:21 EPSR[1283]: EPSR:         HELLO TIME = 1                      FAIL TIME = 2
10:53:21 EPSR[1283]: EPSR:         HELLO SEQ = 96
10:53:21 EPSR[1283]: EPSR: -----

```

**Step 3. The master node receives a Link-Down message on its secondary port.**

The master node receives a Link-Down message, which tells it that a link in the ring is broken. This message came from the transit node on one side of the broken link.

```

10:53:21 EPSR[1283]: EPSR: port1.0.2 Rx:
10:53:21 EPSR[1283]: EPSR: 00e02b00 00040000 cd127808 810003e8 0058aaaa 0300e02b
10:53:21 EPSR[1283]: EPSR: 00bb0100 00543935 00000000 0000cd12 7808990b 00400108
10:53:21 EPSR[1283]: EPSR: 03e80000 00000000 cd127808 00000000 04000000 00000000
10:53:21 EPSR[1283]: EPSR: port1.0.2 Rx:
10:53:21 EPSR[1283]: EPSR: -----
10:53:21 EPSR[1283]: EPSR:         TYPE = LINK-DOWN                      STATE = LINK-DOWN
10:53:21 EPSR[1283]: EPSR:         CTRL VLAN = 1000                SYSTEM = 00-00-cd-12-78-08
10:53:21 EPSR[1283]: EPSR:         HELLO TIME = 0                      FAIL TIME = 0
10:53:21 EPSR[1283]: EPSR:         HELLO SEQ = 0
10:53:21 EPSR[1283]: EPSR: -----
10:53:21 EPSR[1283]: EPSR: EPSR awplus: link down msg from 00-00-cd-12-78-08

```

**Step 4. The master node transmits a Ring-Down-Flush-FDB message.**

In response to the Link-Down message, the master node transmits a Ring-Down-Flush-FDB message out both its primary and secondary ports. The message has to go out both ports to make sure it reaches the nodes on both sides of the broken link. The master node also unblocks its secondary port for vlan2, flushes its forwarding database, sends a trap, and changes the EPSR state to Failed.

```

10:53:21 EPSR[1283]: EPSR: port1.0.1 Tx:
10:53:21 EPSR[1283]: EPSR: 00e02b00 00040000 cd240331 8100e3e8 005caaaa 0300e02b
10:53:21 EPSR[1283]: EPSR: 00bb0100 005424c1 00000000 0000cd24 0331990b 00400107
10:53:21 EPSR[1283]: EPSR: 03e80000 00000000 cd240331 00000000 02000000 00000000
10:53:21 EPSR[1283]: EPSR: port1.0.1 Tx:
10:53:21 EPSR[1283]: EPSR: -----
10:53:21 EPSR[1283]: EPSR:      TYPE = RING-DOWN-FLUSH-FDB      STATE = FAILED
10:53:21 EPSR[1283]: EPSR:      CTRL VLAN = 1000      SYSTEM = 00-00-cd-24-03-31
10:53:21 EPSR[1283]: EPSR:      HELLO TIME = 0      FAIL TIME = 0
10:53:21 EPSR[1283]: EPSR:      HELLO SEQ = 0
10:53:21 EPSR[1283]: EPSR: -----
10:53:21 EPSR[1283]: EPSR: port1.0.2 Tx:
10:53:21 EPSR[1283]: EPSR: 00e02b00 00040000 cd240331 8100e3e8 005caaaa 0300e02b
10:53:21 EPSR[1283]: EPSR: 00bb0100 005424c1 00000000 0000cd24 0331990b 00400107
10:53:21 EPSR[1283]: EPSR: 03e80000 00000000 cd240331 00000000 02000000 00000000
10:53:21 EPSR[1283]: EPSR: port1.0.2 Tx:
10:53:21 EPSR[1283]: EPSR: -----
10:53:21 EPSR[1283]: EPSR:      TYPE = RING-DOWN-FLUSH-FDB      STATE = FAILED
10:53:21 EPSR[1283]: EPSR:      CTRL VLAN = 1000      SYSTEM = 00-00-cd-24-03-31
10:53:21 EPSR[1283]: EPSR:      HELLO TIME = 0      FAIL TIME = 0
10:53:21 EPSR[1283]: EPSR:      HELLO SEQ = 0
10:53:21 EPSR[1283]: EPSR: -----
10:53:21 EPSR[1283]: EPSR: Unblock EPSR:awplus port:5002 VLAN:2
10:53:21 EPSR[1283]: EPSR: EPSR awplus: port 5002 is forwarding
10:53:21 EPSR[1283]: EPSR: Flush FDB EPSR: awplus vid: 2
10:53:21 EPSR[1283]: EPSR: awplus oldState:COMPLETE newState:FAILED

```

**Step 5. The master node receives a second Link-Down message.**

The master node receives a Link-Down message from the transit node on the other side of the broken link. This message arrived after a delay because the ring ports are 1000M ports (see ["Ports and Recovery Times" on page 35](#)). The master node does not take any action in response to this message, because it already responded to the broken link.

```

10:53:21 EPSR[1283]: EPSR: port1.0.1 Rx:
10:53:21 EPSR[1283]: EPSR: 00e02b00 00040000 cd240226 810003e8 0058aaaa 0300e02b
10:53:21 EPSR[1283]: EPSR: 00bb0100 005424d6 00000000 0000cd24 0226990b 00400108
10:53:21 EPSR[1283]: EPSR: 03e80000 00000000 cd240226 00000000 04000000 00000000
10:53:21 EPSR[1283]: EPSR: port1.0.1 Rx:
10:53:21 EPSR[1283]: EPSR: -----
10:53:21 EPSR[1283]: EPSR:      TYPE = LINK-DOWN      STATE = LINK-DOWN
10:53:21 EPSR[1283]: EPSR:      CTRL VLAN = 1000      SYSTEM = 00-00-cd-24-02-26
10:53:21 EPSR[1283]: EPSR:      HELLO TIME = 0      FAIL TIME = 0
10:53:21 EPSR[1283]: EPSR:      HELLO SEQ = 0
10:53:21 EPSR[1283]: EPSR: -----

```

**Step 6. The master node continues sending Health messages.**

The master node continues sending Health messages out its primary port. It does not receive any of these at the secondary port, which tells it that the link is still down.

```

10:53:22 EPSR[1283]: EPSR: port1.0.1 Tx:
10:53:22 EPSR[1283]: EPSR: 00e02b00 00040000 cd240331 8100e3e8 005caaaa 0300e02b
10:53:22 EPSR[1283]: EPSR: 00bb0100 0054245f 00000000 0000cd24 0331990b 00400105
10:53:22 EPSR[1283]: EPSR: 03e80000 00000000 cd240331 00010002 02000061 00000000
10:53:22 EPSR[1283]: EPSR: port1.0.1 Tx:
10:53:22 EPSR[1283]: EPSR: -----
10:53:22 EPSR[1283]: EPSR:         TYPE = HEALTH                     STATE = FAILED
10:53:22 EPSR[1283]: EPSR:         CTRL VLAN = 1000                SYSTEM = 00-00-cd-24-03-31
10:53:22 EPSR[1283]: EPSR:         HELLO TIME = 1                     FAIL TIME = 2
10:53:22 EPSR[1283]: EPSR:         HELLO SEQ = 97
10:53:22 EPSR[1283]: EPSR: -----
.
.
.

```

**Step 7. The master node receives a Health message.**

The master node transmits a Health message and receives it at the secondary port. This indicates that the link is back up.

```

10:59:12 EPSR[1283]: EPSR: port1.0.1 Tx:
10:59:12 EPSR[1283]: EPSR: -----
10:59:12 EPSR[1283]: EPSR:         TYPE = HEALTH                     STATE = FAILED
10:59:12 EPSR[1283]: EPSR:         CTRL VLAN = 1000                SYSTEM = 00-00-cd-24-03-31
10:59:12 EPSR[1283]: EPSR:         HELLO TIME = 1                     FAIL TIME = 2
10:59:12 EPSR[1283]: EPSR:         HELLO SEQ = 447
10:59:12 EPSR[1283]: EPSR: -----
10:59:12 EPSR[1283]: EPSR: port1.0.2 Rx:
10:59:12 EPSR[1283]: EPSR: 00e02b00 00040000 cd240331 810003e8 0058aaaa 0300e02b
10:59:12 EPSR[1283]: EPSR: 00bb0100 00542301 00000000 0000cd24 0331990b 00400105
10:59:12 EPSR[1283]: EPSR: 03e80000 00000000 cd240331 00010002 020001bf 00000000
10:59:12 EPSR[1283]: EPSR: port1.0.2 Rx:
10:59:12 EPSR[1283]: EPSR: -----
10:59:12 EPSR[1283]: EPSR:         TYPE = HEALTH                     STATE = FAILED
10:59:12 EPSR[1283]: EPSR:         CTRL VLAN = 1000                SYSTEM = 00-00-cd-24-03-31
10:59:12 EPSR[1283]: EPSR:         HELLO TIME = 1                     FAIL TIME = 2
10:59:12 EPSR[1283]: EPSR:         HELLO SEQ = 447
10:59:12 EPSR[1283]: EPSR: -----

```



**Step 8. The master node returns the ring to a state of Complete.**

Now that the ring is back up, the master node blocks its secondary port for the data VLAN, transmits a Ring-Up-Flush-FDB message, flushes its FDB, sends a trap, and changes the EPSR state to Complete.

```

10:59:12 EPSR[1283]: EPSR: Block EPSR:awplus port:5002 VLAN:2
10:59:12 EPSR[1283]: EPSR: EPSR awplus: port 5002 is blocking
10:59:12 EPSR[1283]: EPSR: port1.0.1 Tx:
10:59:12 EPSR[1283]: EPSR: 00e02b00 00040000 cd240331 8100e3e8 005caaaa 0300e02b
10:59:12 EPSR[1283]: EPSR: 00bb0100 005425c2 00000000 0000cd24 0331990b 00400106
10:59:12 EPSR[1283]: EPSR: 03e80000 00000000 cd240331 00000000 01000000 00000000
10:59:12 EPSR[1283]: EPSR: port1.0.1 Tx:
10:59:12 EPSR[1283]: EPSR: -----
10:59:12 EPSR[1283]: EPSR:          TYPE = RING-UP-FLUSH-FDB          STATE = COMPLETE
10:59:12 EPSR[1283]: EPSR:          CTRL VLAN = 1000              SYSTEM = 00-00-cd-24-03-31
10:59:12 EPSR[1283]: EPSR:          HELLO TIME = 0                FAIL TIME = 0
10:59:12 EPSR[1283]: EPSR:          HELLO SEQ = 0
10:59:12 EPSR[1283]: EPSR: -----
10:59:12 EPSR[1283]: EPSR: Flush FDB EPSR: awplus vid: 2
10:59:12 EPSR[1283]: EPSR: awplus oldState:FAILED newState:COMPLETE
10:59:12 EPSR[1283]: EPSR: EPSR awplus: ring complete

```

**Step 9. The master node receives the Ring-Up-Flush-FDB message on port1.0.2.**

The master node receives the Ring-Up-Flush-FDB message back on its secondary port, because the packet traversed the whole ring. The master node ignores the message.

```

10:59:12 EPSR[1283]: EPSR: port1.0.2 Rx:
10:59:12 EPSR[1283]: EPSR: 00e02b00 00040000 cd240331 810003e8 0058aaaa 0300e02b
10:59:12 EPSR[1283]: EPSR: 00bb0100 005425c2 00000000 0000cd24 0331990b 00400106
10:59:12 EPSR[1283]: EPSR: 03e80000 00000000 cd240331 00000000 01000000 00000000
10:59:12 EPSR[1283]: EPSR: port1.0.2 Rx:
10:59:12 EPSR[1283]: EPSR: -----
10:59:12 EPSR[1283]: EPSR:          TYPE = RING-UP-FLUSH-FDB          STATE = COMPLETE
10:59:12 EPSR[1283]: EPSR:          CTRL VLAN = 1000              SYSTEM = 00-00-cd-24-03-31
10:59:12 EPSR[1283]: EPSR:          HELLO TIME = 0                FAIL TIME = 0
10:59:12 EPSR[1283]: EPSR:          HELLO SEQ = 0
10:59:12 EPSR[1283]: EPSR: -----

```

**Step 10. The master node transmits and receives Health messages.**

The master node continues transmitting and receiving Health messages for as long as the ring stays in a state of Complete.

```

10:59:13 EPSR[1283]: EPSR: port1.0.1 Tx:
10:59:13 EPSR[1283]: EPSR: 00e02b00 00040000 cd240331 8100e3e8 005caaaa 0300e02b
10:59:13 EPSR[1283]: EPSR: 00bb0100 00542400 00000000 0000cd24 0331990b 00400105
10:59:13 EPSR[1283]: EPSR: 03e80000 00000000 cd240331 00010002 010001c0 00000000
10:59:13 EPSR[1283]: EPSR: port1.0.1 Tx:
10:59:13 EPSR[1283]: EPSR: -----
10:59:13 EPSR[1283]: EPSR:          TYPE = HEALTH                      STATE = COMPLETE
10:59:13 EPSR[1283]: EPSR:          CTRL VLAN = 1000                SYSTEM = 00-00-cd-24-03-31
10:59:13 EPSR[1283]: EPSR:          HELLO TIME = 1                      FAIL TIME = 2
10:59:13 EPSR[1283]: EPSR:          HELLO SEQ = 448
10:59:13 EPSR[1283]: EPSR: -----
10:59:13 EPSR[1283]: EPSR: port1.0.2 Rx:
10:59:13 EPSR[1283]: EPSR: 00e02b00 00040000 cd240331 810003e8 0058aaaa 0300e02b
10:59:13 EPSR[1283]: EPSR: 00bb0100 00542400 00000000 0000cd24 0331990b 00400105
10:59:13 EPSR[1283]: EPSR: 03e80000 00000000 cd240331 00010002 010001c0 00000000
10:59:13 EPSR[1283]: EPSR: port1.0.2 Rx:
10:59:13 EPSR[1283]: EPSR: -----
10:59:13 EPSR[1283]: EPSR:          TYPE = HEALTH                      STATE = COMPLETE
10:59:13 EPSR[1283]: EPSR:          CTRL VLAN = 1000                SYSTEM = 00-00-cd-24-03-31
10:59:13 EPSR[1283]: EPSR:          HELLO TIME = 1                      FAIL TIME = 2
10:59:13 EPSR[1283]: EPSR:          HELLO SEQ = 448
10:59:13 EPSR[1283]: EPSR: -----
.
.
.

```

**Transit node (Node B) debug output**

The following debugging shows the same sequence of events as the previous section, but on the transit node instead of the master node. It starts with the ring established and in a state of Complete.

**Note:** The following debug was captured at a different time (during a different ring-down event) from the master node debug in the previous section. This is why the times and hello sequence numbers do not match.

**Step 1. The transit node receives Health messages.**

The transit node receives Health messages on port1.0.1, because that port is connected to the master node's primary port. The message shows that the ring state is Complete.

Also note the Hello sequence number, which is very close to the maximum of 65535. Once the number reaches 65535, it restarts at zero.

```

10:23:12 EPSR[1284]: EPSR: port1.0.1 Rx:
10:23:12 EPSR[1284]: EPSR: 00e02b00 00040000 cd240331 810003e8 0058aaaa 0300e02b
10:23:12 EPSR[1284]: EPSR: 00bb0100 005427e9 00000000 0000cd24 0331990b 00400105
10:23:12 EPSR[1284]: EPSR: 03e80000 00000000 cd240331 00010002 0100ffec 00000000
10:23:12 EPSR[1284]: EPSR: port1.0.1 Rx:
10:23:12 EPSR[1284]: EPSR: -----
10:23:12 EPSR[1284]: EPSR:          TYPE = HEALTH                      STATE = COMPLETE
10:23:12 EPSR[1284]: EPSR:          CTRL VLAN = 1000                SYSTEM = 00-00-cd-24-03-31
10:23:12 EPSR[1284]: EPSR:          HELLO TIME = 1                  FAIL TIME = 2
10:23:12 EPSR[1284]: EPSR:          HELLO SEQ = 65516
10:23:12 EPSR[1284]: EPSR: -----
.
.
.

```

**Step 2. The link between the two transit nodes goes down.**

The transit node receives Health message 29. At this stage, the message does not indicate that anything is wrong. However, between messages 28 and 29, the link went down. This means that message 29 will not make it back to the master node.

The hello sequence counter has wrapped and is now counting up from zero.

```

10:24:01 EPSR[1284]: EPSR: port1.0.1 Rx:
10:24:01 EPSR[1284]: EPSR: 00e02b00 00040000 cd240331 810003e8 0058aaaa 0300e02b
10:24:01 EPSR[1284]: EPSR: 00bb0100 005427b9 00000000 0000cd24 0331990b 00400105
10:24:01 EPSR[1284]: EPSR: 03e80000 00000000 cd240331 00010002 0100001d 00000000
10:24:01 EPSR[1284]: EPSR: port1.0.1 Rx:
10:24:01 EPSR[1284]: EPSR: -----
10:24:01 EPSR[1284]: EPSR:          TYPE = HEALTH                      STATE = COMPLETE
10:24:01 EPSR[1284]: EPSR:          CTRL VLAN = 1000                SYSTEM = 00-00-cd-24-03-31
10:24:01 EPSR[1284]: EPSR:          HELLO TIME = 1                  FAIL TIME = 2
10:24:01 EPSR[1284]: EPSR:          HELLO SEQ = 29
10:24:01 EPSR[1284]: EPSR: -----

```

**Step 3. The transit node receives a Ring-Down-Flush-FDB message.**

In the meanwhile, the master node has received a Link-Down message from the switch at the other end of the broken link. Therefore, the master node realizes that the ring is broken and acts accordingly. As part of the recovery process, the master node sends a Ring-Down-Flush-FDB message. The transit node receives this message and flushes its forwarding database.

```

10:24:02 EPSR[1284]: EPSR: port1.0.1 Rx:
10:24:02 EPSR[1284]: EPSR: 00e02b00 00040000 cd240331 810003e8 0058aaaa 0300e02b
10:24:02 EPSR[1284]: EPSR: 00bb0100 005426d7 00000000 0000cd24 0331990b 00400107
10:24:02 EPSR[1284]: EPSR: 03e80000 00000000 cd240331 00000000 02000000 00000000
10:24:02 EPSR[1284]: EPSR: port1.0.1 Rx:
10:24:02 EPSR[1284]: EPSR: -----
10:24:02 EPSR[1284]: EPSR:      TYPE = RING-DOWN-FLUSH-FDB      STATE = FAILED
10:24:02 EPSR[1284]: EPSR:      CTRL VLAN = 1000      SYSTEM = 00-00-cd-24-03-31
10:24:02 EPSR[1284]: EPSR:      HELLO TIME = 0      FAIL TIME = 0
10:24:02 EPSR[1284]: EPSR:      HELLO SEQ = 0
10:24:02 EPSR[1284]: EPSR: -----

```

**Step 4. The transit node sends a Link-Down message.**

The transit node realizes that its port is down, sends a Link-Down message, sends a trap, and changes its state to Link-Down. The transit node sends this message some time after the link actually went down, because the ring ports are 1000M ports (see ["Ports and Recovery Times" on page 35](#)). By this stage the ring has already changed topology to restore traffic flow. The master node detected the link failure by receiving a Link-Down message from the other side of the link.

```

10:24:02 EPSR[1284]: EPSR: port1.0.1 Tx:
10:24:02 EPSR[1284]: EPSR: 00e02b00 00040000 cd240226 8100e3e8 005caaaa 0300e02b
10:24:02 EPSR[1284]: EPSR: 00bb0100 005422c0 00000000 0000cd24 0226990b 00400108
10:24:02 EPSR[1284]: EPSR: 03e80000 00000000 cd240226 00000000 04000000 00000000
10:24:02 EPSR[1284]: EPSR: port1.0.1 Tx:
10:24:02 EPSR[1284]: EPSR: -----
10:24:02 EPSR[1284]: EPSR:      TYPE = LINK-DOWN      STATE = LINK-DOWN
10:24:02 EPSR[1284]: EPSR:      CTRL VLAN = 1000      SYSTEM = 00-00-cd-24-02-26
10:24:02 EPSR[1284]: EPSR:      HELLO TIME = 0      FAIL TIME = 0
10:24:02 EPSR[1284]: EPSR:      HELLO SEQ = 0
10:24:02 EPSR[1284]: EPSR: -----
10:24:02 EPSR[1284]: EPSR: awplus oldState:LINK-UP newState:LINK-DOWN

```

**Step 5. The transit node receives Health messages.**

The transit node receives Health messages from the master node. These have a state of Failed, which shows that the ring is still broken.

```

10:24:02 EPSR[1284]: EPSR: port1.0.1 Rx:
10:24:02 EPSR[1284]: EPSR: 00e02b00 00040000 cd240331 810003e8 0058aaaa 0300e02b
10:24:02 EPSR[1284]: EPSR: 00bb0100 005426b8 00000000 0000cd24 0331990b 00400105
10:24:02 EPSR[1284]: EPSR: 03e80000 00000000 cd240331 00010002 0200001e 00000000
10:24:02 EPSR[1284]: EPSR: port1.0.1 Rx:
10:24:02 EPSR[1284]: EPSR: -----
10:24:02 EPSR[1284]: EPSR:          TYPE = HEALTH                      STATE = FAILED
10:24:02 EPSR[1284]: EPSR:          CTRL VLAN = 1000                SYSTEM = 00-00-cd-24-03-31
10:24:02 EPSR[1284]: EPSR:          HELLO TIME = 1                  FAIL TIME = 2
10:24:02 EPSR[1284]: EPSR:          HELLO SEQ = 30
10:24:02 EPSR[1284]: EPSR: -----
.
.
.

```

**Step 6. The link comes back up.**

The transit node detects that the broken link has come back up. It blocks the port to prevent a loop from occurring, sends a trap, and changes the EPSR state to Pre-forwarding.

```

10:42:49 EPSR[1284]: EPSR: Block EPSR:awplus port:5002 VLAN:2
10:42:49 EPSR[1284]: EPSR: EPSR awplus: port 5002 is blocking
10:42:49 EPSR[1284]: EPSR: EPSR awplus, ifIndex 5002 up
10:42:49 EPSR[1284]: EPSR: awplus oldState:LINK-DOWN newState:PRE-FORWARDING

```

**Step 7. The transit node receives another Health message.**

The transit node receives another Health message. This message will make it back to the master node's secondary port, because the link between the two transit nodes is now up.

```

10:42:50 EPSR[1284]: EPSR: port1.0.1 Rx:
10:42:50 EPSR[1284]: EPSR: 00e02b00 00040000 cd240331 810003e8 0058aaaa 0300e02b
10:42:50 EPSR[1284]: EPSR: 00bb0100 00542251 00000000 0000cd24 0331990b 00400105
10:42:50 EPSR[1284]: EPSR: 03e80000 00000000 cd240331 00010002 02000485 00000000
10:42:50 EPSR[1284]: EPSR: port1.0.1 Rx:
10:42:50 EPSR[1284]: EPSR: -----
10:42:50 EPSR[1284]: EPSR:          TYPE = HEALTH                      STATE = FAILED
10:42:50 EPSR[1284]: EPSR:          CTRL VLAN = 1000                SYSTEM = 00-00-cd-24-03-31
10:42:50 EPSR[1284]: EPSR:          HELLO TIME = 1                  FAIL TIME = 2
10:42:50 EPSR[1284]: EPSR:          HELLO SEQ = 1157
10:42:50 EPSR[1284]: EPSR: -----

```

**Step 8. The transit node receives a Ring-Up-Flush-FDB message.**

The transit node receives a Ring-Up-Flush-FDB message, which indicates that the master node knows that all links in the ring are up again. The transit node unblocks port1.0.2 for vlan2, flushes its FDB, sends a trap, and changes state to Link-Up.

```

10:42:50 EPSR[1284]: EPSR: port1.0.1 Rx:
10:42:50 EPSR[1284]: EPSR: 00e02b00 00040000 cd240331 810003e8 0058aaaa 0300e02b
10:42:50 EPSR[1284]: EPSR: 00bb0100 005427d8 00000000 0000cd24 0331990b 00400106
10:42:50 EPSR[1284]: EPSR: 03e80000 00000000 cd240331 00000000 01000000 00000000
10:42:50 EPSR[1284]: EPSR: port1.0.1 Rx:
10:42:50 EPSR[1284]: EPSR: -----
10:42:50 EPSR[1284]: EPSR:          TYPE = RING-UP-FLUSH-FDB          STATE = COMPLETE
10:42:50 EPSR[1284]: EPSR:          CTRL VLAN = 1000          SYSTEM = 00-00-cd-24-03-31
10:42:50 EPSR[1284]: EPSR:          HELLO TIME = 0          FAIL TIME = 0
10:42:50 EPSR[1284]: EPSR:          HELLO SEQ = 0
10:42:50 EPSR[1284]: EPSR: -----
10:42:50 EPSR[1284]: EPSR: Unblock EPSR:awplus port:5002 VLAN:2
10:42:50 EPSR[1284]: EPSR: EPSR awplus: port 5002 is forwarding
10:42:50 EPSR[1284]: EPSR: Flush FDB EPSR: awplus vid: 2
10:42:50 EPSR[1284]: EPSR: awplus oldState:PRE-FORWARDING newState:LINK-UP

```

**Step 9. The transit node receives Health messages.**

The transit node continues receiving Health messages for as long as the ring stays in a state of Complete.

This is equivalent to the packet shown on **step 10** of ["Link Down between two transit nodes"](#) on [page 53](#) of the master node debug output.

```

10:42:51 EPSR[1284]: EPSR: port1.0.1 Rx:
10:42:51 EPSR[1284]: EPSR: 00e02b00 00040000 cd240331 810003e8 0058aaaa 0300e02b
10:42:51 EPSR[1284]: EPSR: 00bb0100 00542350 00000000 0000cd24 0331990b 00400105
10:42:51 EPSR[1284]: EPSR: 03e80000 00000000 cd240331 00010002 01000486 00000000
10:42:51 EPSR[1284]: EPSR: port1.0.1 Rx:
10:42:51 EPSR[1284]: EPSR: -----
10:42:51 EPSR[1284]: EPSR:          TYPE = HEALTH          STATE = COMPLETE
10:42:51 EPSR[1284]: EPSR:          CTRL VLAN = 1000          SYSTEM = 00-00-cd-24-03-31
10:42:51 EPSR[1284]: EPSR:          HELLO TIME = 1          FAIL TIME = 2
10:42:51 EPSR[1284]: EPSR:          HELLO SEQ = 1158
10:42:51 EPSR[1284]: EPSR: -----
.
.
.

```

# EPSR SuperLoop Prevention

## Overview

EPSR SuperLoop Prevention (EPSR-SLP) is an enhancement to the existing EPSR feature in AlliedWare Plus. EPSR-SLP prevents “SuperLoops” forming in certain EPSR multi-ring topologies. EPSR-SLP was introduced in AlliedWare Plus Version 5.4.2.

### What is a SuperLoop?

To achieve redundancy, you may wish to deploy multiple EPSR rings that have the same set of protected VLANs. If these rings share a common segment, and that common segment fails, a loop forms. This loop is known as a SuperLoop.

### Why do SuperLoops occur?

In normal EPSR operation (that is, without EPSR-SLP), the Masters on both rings separately put their secondary ports into the Forwarding state when they detect a link going down. As illustrated in the diagram on the following page, this creates a Forwarding loop.

### Example diagram

The following diagram shows how EPSR without the EPSR-SLP enhancement can lead to a SuperLoop. It also shows the topology of the resultant SuperLoop.

#### List of terms:

##### **SuperLoop**

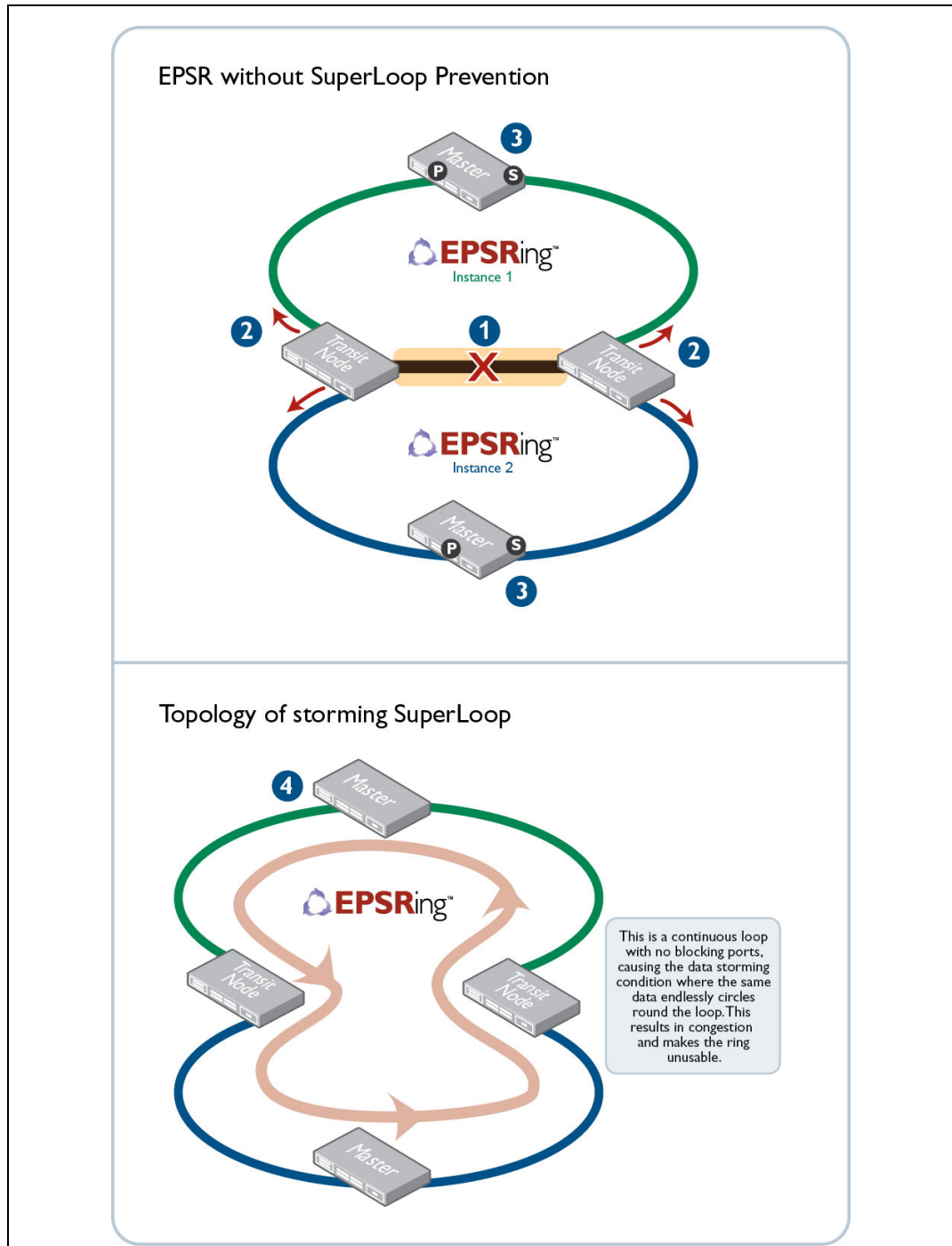
Multiple rings, each with their own EPSR Domains, may be connected together in a topology. If these domains share the same set of Data VLANs, and also share a common segment, then the failure of that common segment leads to a SuperLoop.

##### **Common Segment**

A common segment is a link (or links) in the network that are shared by two or more rings, and which has a common set of Data VLANs.

##### **SLP**

SuperLoop Prevention.



The sequence of events without EPSR-SLP, as shown above, is:

1. The common link goes down.
2. The transit nodes at each end of the common link send Link Down messages to both master nodes.
3. The master nodes both unblock their secondary ports.
4. As shown in the lower half of the diagram, this results in a loop. Data circulates continuously around this loop, congesting the network.

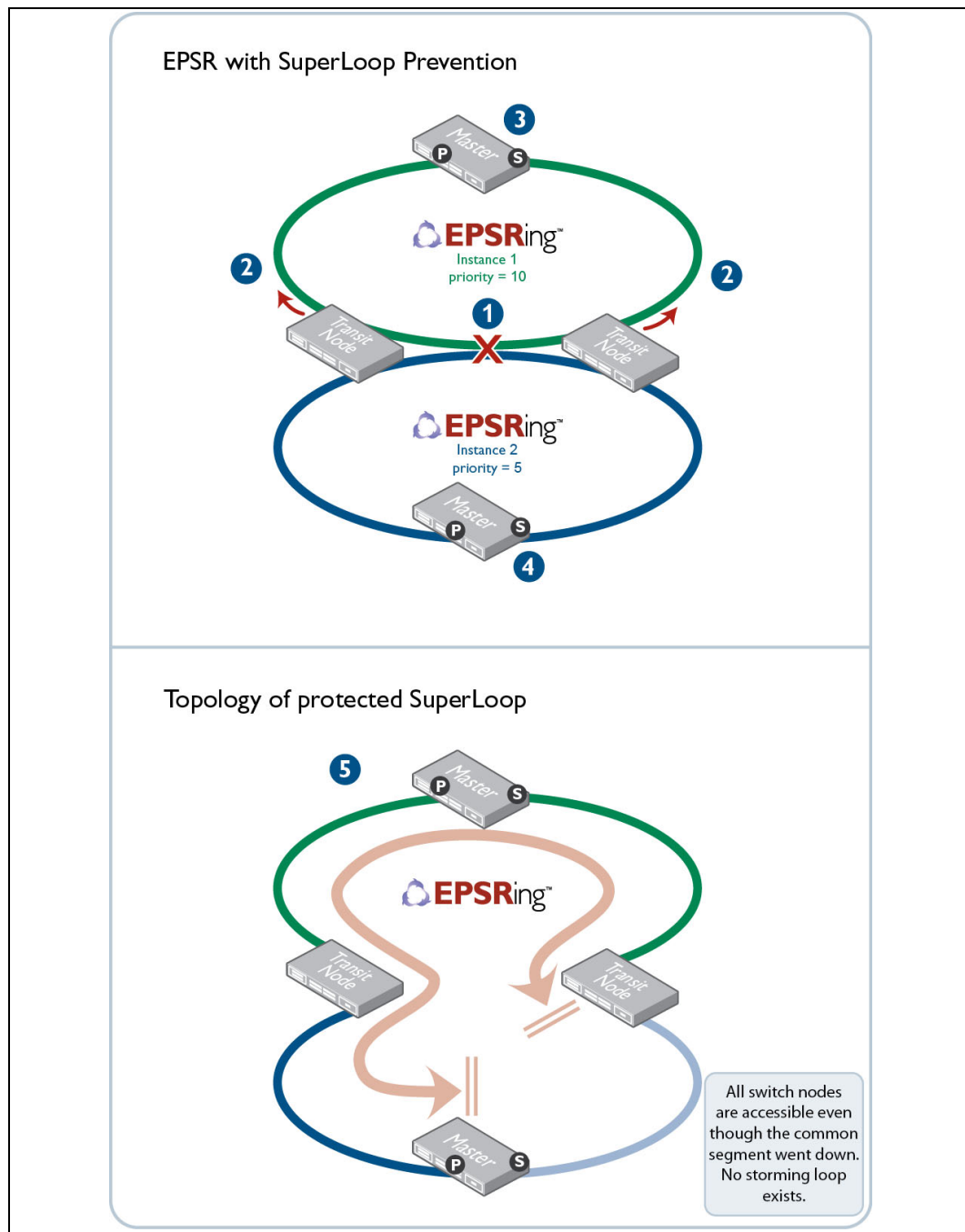


## How does EPSR-SLP work?

EPSR-SLP prevents SuperLoops forming in the following way:

1. It assigns a priority to each EPSR ring.
2. It ensures that common segment transit nodes send Link Down messages only to the Master of the highest priority ring.
3. When a link in a common segment goes down, only the Master of the highest priority ring opens its secondary port.

The following diagram illustrates how EPSR-SLP prevents SuperLoops forming.



The sequence of events with EPSR-SLP, as shown above, is:

1. The common link goes down.
2. The transit nodes at each end of the common link send Link Down messages only to their higher-priority master nodes.
3. The higher-priority master node unblocks its secondary ports.
4. The other master node performs no action.
5. The end result is a new topology in which all nodes retain connectivity, but one link is blocked to prevent packet storming.

## Fault detection and recovery without EPSR-SLP

It is important at this stage to review original EPSR functionality (prior to the introduction of EPSR-SLP).

For information about how EPSR detects outages in a node, or a link in the ring, see ["Detecting a fault" on page 9](#).

For information about the fault recovery actions EPSR takes, see ["Recovering from a fault" on page 9](#).

For information about Enhanced Recovery, see ["Enhanced Recovery" on page 10](#).

**Note:** Enhanced Recovery behaviour is the same with EPSR-SLP enabled, however some differences exist for a master node. For more information, see ["EPSR Enhanced Recovery when SLP is enabled" on page 71](#).

## Fault detection and recovery with EPSR-SLP

The key concept that underlies EPSR-SLP is that of domain priority. For a network to utilize EPSR-SLP, you need to assign all EPSR Domains a priority level value between 1 and 127.

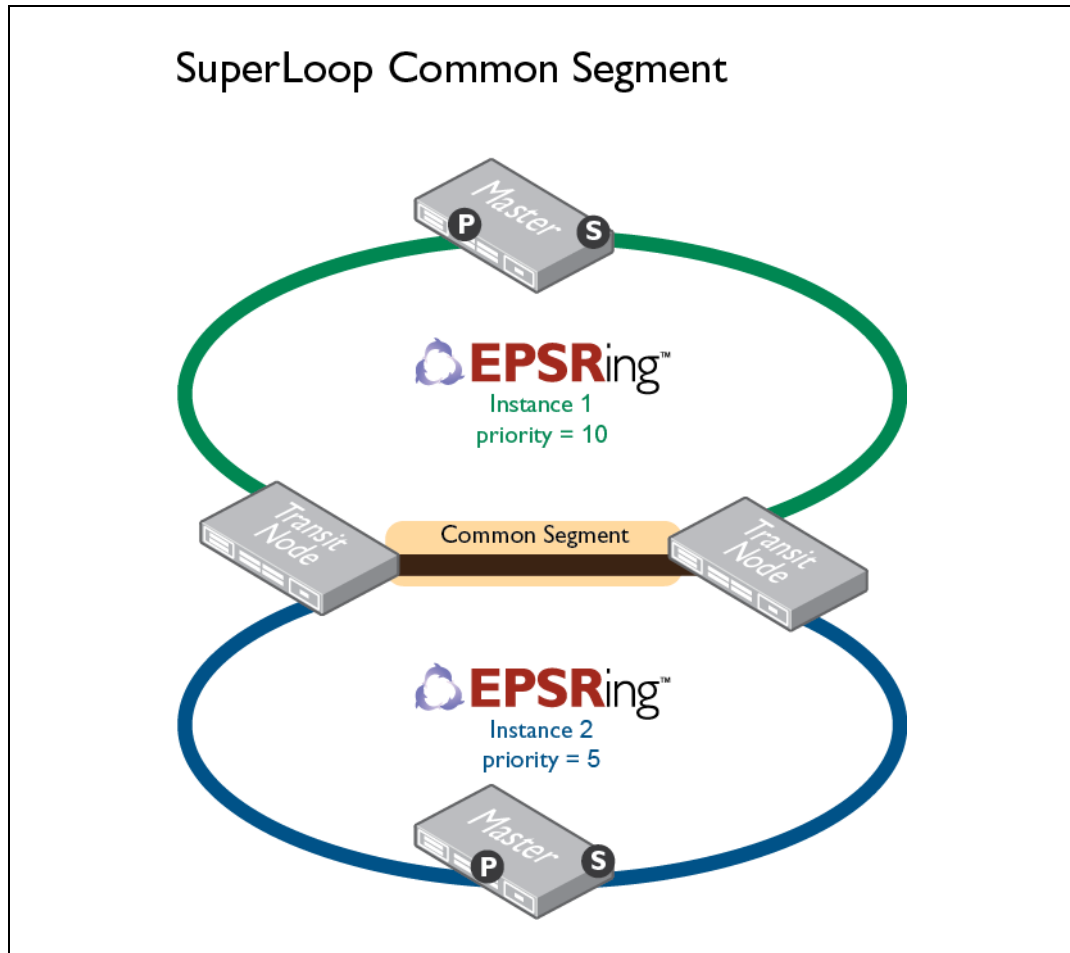
A value of 1 represents the lowest priority level, and 127 the highest priority. Assigning a priority of 1 or greater enables EPSR-SLP.

**Note:** A value of 0 effectively disables EPSR-SLP, returning the switch to standard EPSR behaviour.

### Common segments

A common segment is a link in the network that is shared by two or more rings, and which has a common set of data VLANs. In other words, the data VLANs passing through the common segment also extend into both the rings that share the segment.

The following diagram illustrates a common segment.



How the switch applies SuperLoop Protection depends on the role of the node within the ring:

- Whether it is a master node or a transit node
- Whether or not the node is connected to a common segment

### Master node behavior

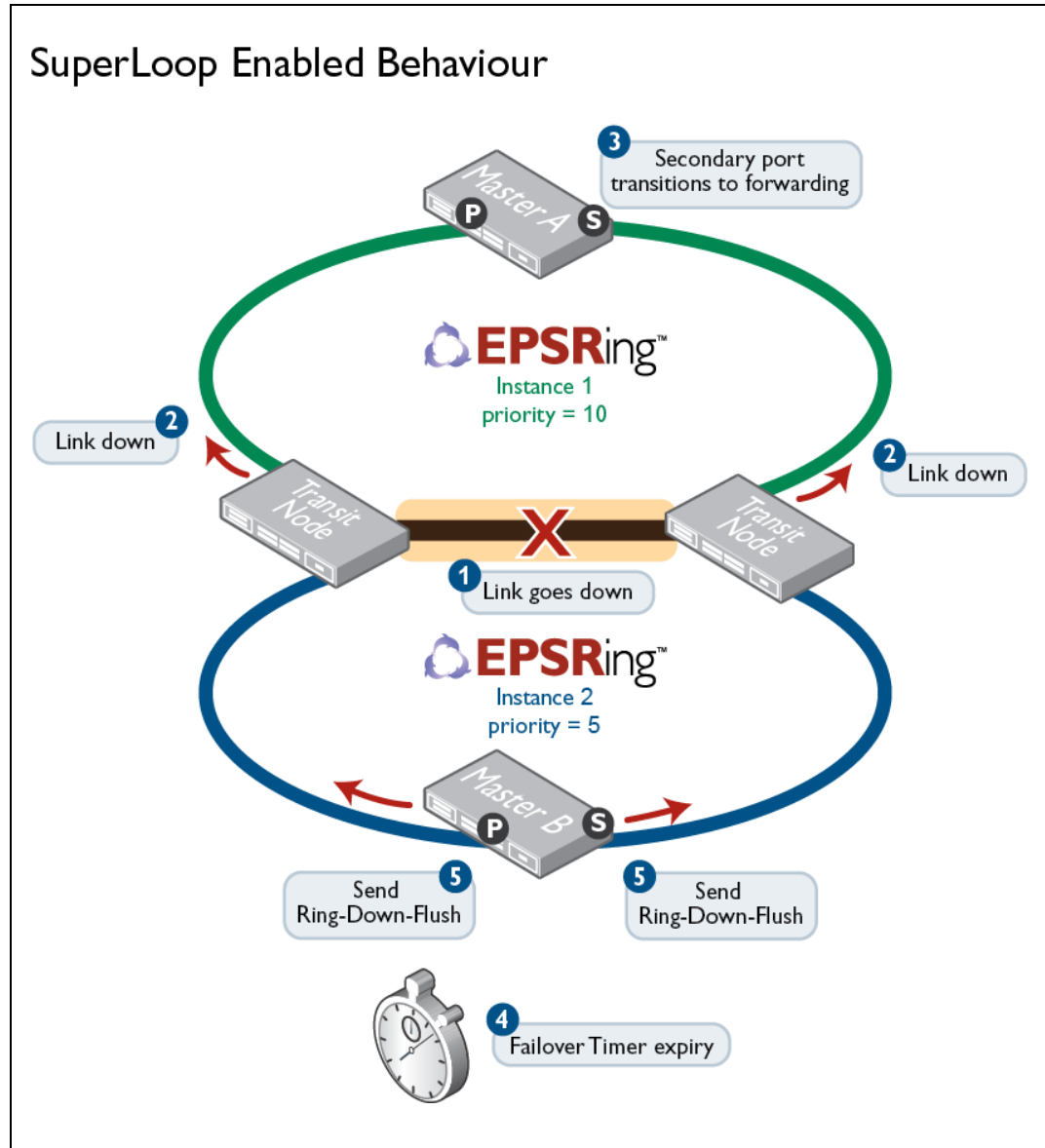
When a domain's Failover timer expires, the master node does not unblock its secondary port, but it does:

- Transition to the Failed state
- Send a Ring-Down-Flush message

The only situation in which the master node does unblock its secondary port is if:

- It receives a Link Down message from a transit node, and
- The Link Down message arrives before the failover timer expires.

### Example



#### In this example:

Master A is the higher-priority master node with a priority level of 10. Therefore, transit nodes send Link Down messages to Master A.

Master A receives the Link Down messages before its failover timer expires. This means it will:

1. transition its secondary port to the Forwarding state
2. transition to the Failed state
3. send a Ring-Down-Flush message to enable new MAC address learning

Master B does not receive Link Down messages. Therefore its Failover timer will expire without having received any Link Down messages. So, it will:

1. not unblock its secondary port
2. transition to the Failed state
3. send a Ring-Down-Flush message from both ports.

**Timing is important:**

- Link Down messages are normally received from transit nodes before Failover timer expiry. In this case, the secondary port transitions to the Forwarding state.
- If a Link Down message is received after the Failover timer expires, the secondary port remains in the Blocking state. see ["Transit node behavior" on page 69](#)

This behavior can sometimes result in cases where the secondary port seems to be unexpectedly blocked. See ["High priority master reboot when ring is down" on page 80](#).

**Transit node behavior**

A transit node that is not connected to a common segment is not affected by its EPSR priority. It behaves as it would without SuperLoop Protection enabled, simply sending a Link Down message if it detects a failed link.

A transit node that is connected to a common segment is affected by its EPSR priority. It changes its behavior in the following ways:

- It compares the EPSR priority of each of the instances that share the common segment.
- If the common segment fails, the transit node only issues a Link Down message on the instance with the highest priority.

This is illustrated in the example diagram under ["Master node behavior" on page 67](#). At step 2 in this diagram, the transit nodes on the common link send Link Down messages only to the higher-priority Master.

## Transit node behaviour if the other port is still down

Without SuperLoop Protection, if both ring ports that connect a transit node to a given EPSR instance go down, and then one of those ports comes back up again, the switch will end up putting that newly recovered port into forwarding.

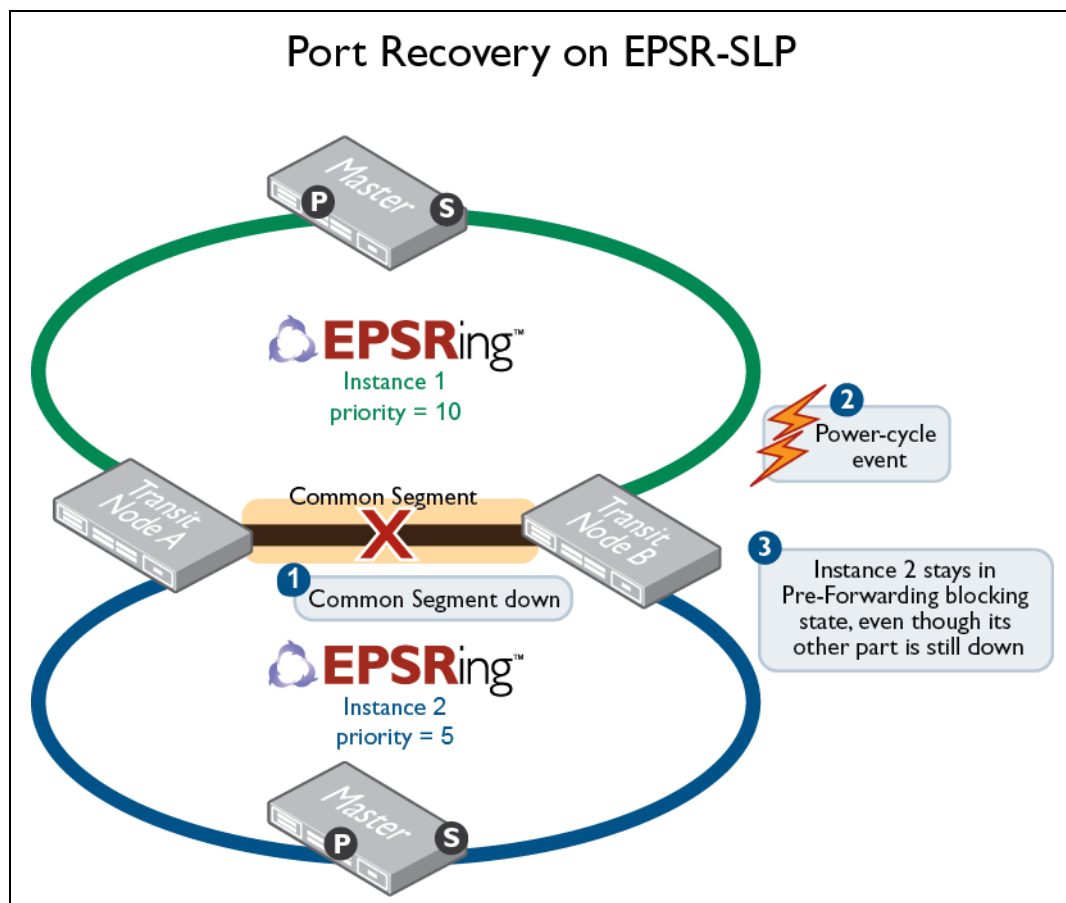
With SuperLoop Protection, there are cases where this does not happen. Specifically if:

- One of the ring ports that went down is connected to a common segment, and
- The ring port that recovers is not connected to the common segment, and is not connected to the highest priority EPSR instance that shares the common segment, then the newly recovered ring port is not transitioned to forwarding.

This avoids the risk of SuperLoops that could form in some topologies.

**Example** Consider the case illustrated in the diagram below. If the switch at the right-hand end of the common segment is power cycled while the common segment is down, then when the switch comes up, the port that connects to EPSR instance 2 will remain Blocking.

The reason for this is that this transit node cannot know for certain whether the secondary port on the Master switch in the lower-priority ring is still Blocking. If that Master's secondary port is not Blocking, then if the transit node puts its port into Forwarding, a SuperLoop would form. Hence, to be safe, that port remains Blocking.



## EPSR Enhanced Recovery when SLP is enabled

For information about EPSR Enhanced Recovery without SuperLoop enabled, see ["Enhanced Recovery" on page 10](#).

The following sections address the behavior of EPSR Enhanced Recovery on SuperLoop-enabled nodes.

**Note:** Enhanced Recovery should not be used in EPSR-SLP topologies that include 3 or more rings. For more information, see ["Caution about Enhanced Recovery with EPSR-SLP topologies with 3 or more rings" on page 78](#).

### Transit nodes and Enhanced Recovery

In most situations, transit nodes using Enhanced Recovery behave in the same way regardless of whether or not SuperLoop Protection is enabled. In general, when a transit node receives the Permission-Link-Forward response from the Master, it moves the newly-recovered port from the Pre-forwarding state to the Forwarding state.

However, there are some over-riding behaviors that can cause a port to remain in the Blocking state:

- If the instance that receives permission to forward is not the highest priority on a common segment, the port may still be subject to the physical blocking of a higher priority instance. For more information, see ["Physical and logical port control" on page 71](#).
- The transit node behaviour explained in ["Transit node behaviour if the other port is still down" on page 70](#).

## Physical and logical port control

In the context of EPSR-SLP, it is important to understand the difference between physical and logical control of ports.

On nodes that have ports connected to common segments, only the highest priority EPSR instance has physical control of those ports. The other EPSR instances are deemed to have logical control of the common segment ports.

The EPSR instance that has physical control of the ring ports is the one that sets the port states, for example blocking, pre-forwarding or forwarding.

The state that the other, lower-priority instances that share the ring ports would put the ports into, if they had control of them, is referred to as the logical state of the ports for those instances. This logical state has no effect on the operation of the ports. The logical state is tracked mostly so that you can check that those other instances are maintaining internal consistency, and are making the correct state transitions.

If the EPSR instance that has physical control of a port is physically blocking the port, it is also blocking access to that port for all other instances as well.

You can see whether a port is physically or logically blocking by using the `show epsr` command:

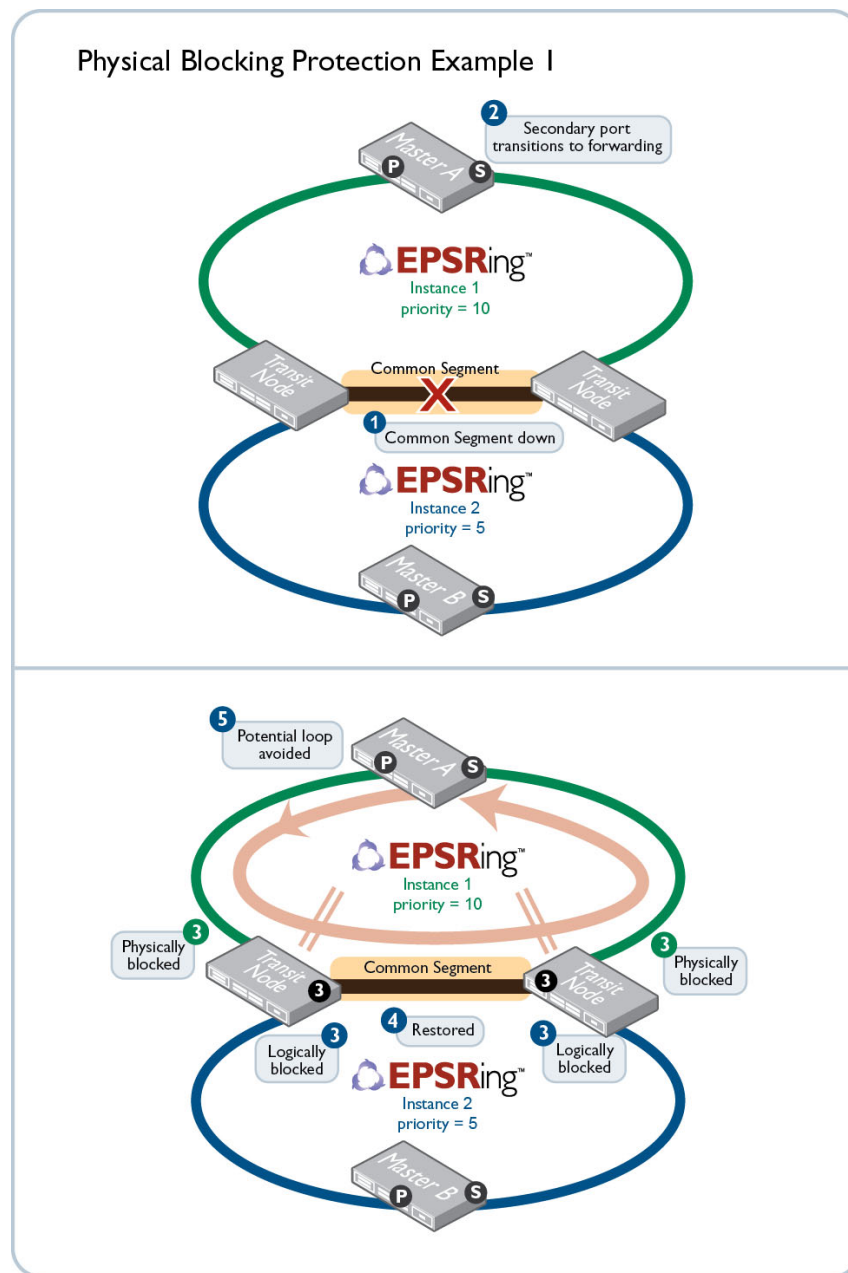
```
epsr4# show epsr
```

```
-----
EPSR Information
-----
```

```
Name ..... B
Mode ..... Master
Status ..... Enabled
State ..... Idle
Control Vlan ..... 6
Data VLAN(s) ..... 40
Interface Mode ..... Channel Groups Only
Primary Port ..... sa2
    Status ..... Down
    Is On Common Segment ..... No
    Blocking Control ..... Physical << Here it is physical
Secondary Port ..... sa1
    Status ..... Down
    Is On Common Segment ..... No
    Blocking Control ..... Physical << Here it is physical
Hello Time ..... 1 s
Failover Time ..... 2 s
Ring Flap Time ..... 0 s
Trap ..... Enabled
Enhanced Recovery ..... Enabled
Priority ..... 5
-----
```



**Example** The following example illustrates the distinction between physical and logical control.



On the common segment, only the highest priority EPSR instance has physical control of the ports. So, when a common segment port fails, only the highest priority instance on that common segment physically blocks the port. Other instances on the common segment put their ring ports into a logical blocking state.

When the link goes up again, the port is initially held in the Pre-forwarding state. While in the Pre-forwarding state, the highest priority instance is physically blocking. This also blocks all other instances on the port.

Once the highest priority Master has put its secondary port into the Blocking state, it can inform the transit nodes attached to the common segment to transition their Pre-forwarding ports to Forwarding.

At that point, these ports will also go to Forwarding for the other EPSR instance. So, when the physical blocking is removed, the logical blocking is also removed.

The port remains in the Pre-forwarding state until:

- Without Enhanced Recovery enabled:  
the node receives a Ring-Up-Flush message from the highest-priority Master.
- With Enhanced Recovery enabled:  
the node receives a Permission-Link-Forward message from the highest-priority Master.

The key point here is that it is packets from the highest priority Master that determine when the ports can return to the Forwarding state. Therefore, it must be the highest priority EPSR instance that has control of this.

## EPSR show commands

Some show commands enable monitoring of EPSR-SLP.

**Command** `show epsr common-segments`

This show command gives you information about common segments.

EPSR Common Segments							
Common Seg Port	EPSR Instance	Mode	Prio	Port Type	Phys Ctrl of Port	Ring Port Status	
port1.0.2	blue	Transit	120	Second	Yes	Fwding	
	green	Transit	60	First	No	Fwding (logical)	

**Parameters explained**

PARAMETER	MEANING
Common Seg Port	The ring port that identifies the common segment
EPSR Instance	Corresponds to IMASK/EMASK fields on the IMASK table. Shows which port numbers packets will be matched on.
Mode	The mode in which the EPSR instance is configured to operate - either Master or Transit
Prio	The EPSR instance's priority
Port Type	The type of ring port in the instance - Primary or Secondary for a master node; First or Second for a transit node
Phys Ctrl of Port	Whether the instance has physical control of the common ring port's blocking in the instances' data VLANs
Ring Port Status	Whether the EPSR instance's ring port is currently in the Forwarding, Blocking, or Link Down state

**Command** `show epsr summary`

This command lets you view summary information for all EPSR instances. Information specific to common segments is present in this output.

```

EPSR Summary Information

Abbreviations:
  M = Master node
  T = Transit node
  C = is on a Common Segment with other instances
  P = instance on a Common Segment has physical control of the shared
port's
      data VLAN blocking
  Blocked (SLP) = master secondary ring port is blocked for EPSR-SLP

EPSR
Instance      Mode Enabled  State      Ctrl      Primary/1st  Secondary/2nd
              VLAN  Prio  Port Status  Port Status
-----
blue          T    Yes    Links-Up    5    120  Fwding      Fwding (C,P)
green         T    Yes    Links-Up    6    60   Fwding (C)  Fwding
-----

```

**Parameters explained**

PARAMETER	MEANING
EPSR Instance	The name of the EPSR instance
Mode	The mode in which the EPSR instance is configured to operate - either Master (M) or Transit (T)
Enabled	Whether the EPSR instance is enabled or disabled
State	The state of the EPSR instance's state machine
Ctrl VLAN	The VLAN ID of the EPSR instance's control VLAN
Prio	The EPSR instance's priority

PARAMETER	MEANING
Primary/1st Port Status	For a master node, this is the EPSR instance's primary ring port. For a transit node, this is the EPSR instance's first port. C indicates the ring port is on a common segment with other instances. P indicates the instance has physical control of the shared port's data VLAN blocking.

**Command** `show epsr [<instance>] config-check`

PARAMETER	MEANING
<instance>	Name of the EPSR instance to check on.

This command checks the configuration of a specified EPSR instance, or all EPSR instances. If an instance is enabled, this command will check for the following errors or warnings:

- The control VLAN has the wrong number of ports.
- There are no data VLANs.
- Some of the data VLANs are not assigned to the ring ports.
- The failover time is less than 5 seconds, for a stacked device.
- The instance is a master that shares a common segment with a higher priority instance.
- The instance is a master that shares a common segment with another master.
- The instance is a master with its secondary port on a common segment.

To check the configuration of all EPSR instances and display the results, use the command:

`awplus# show epsr config-check`

```

EPSR Instance   Status   Description
-----
red             Warning  Failover time is 2s but should be 5s because
              device is stacked.
white           OK.
blue            Warning  Primary port is not data VLANs 29-99
orange          OK.

Don't forget to check that this node's configuration is consistent with
all other nodes in the ring.
-----

```

## Best practice guidelines for EPSR-SLP deployment

### EPSR-SLP priorities

- To enable EPSR SuperLoop Protection, EPSR master nodes and common segment transit nodes must have an EPSR instance priority greater than zero.
- All member nodes of an EPSR-SLP domain should have a consistent EPSR priority value.
- On common segment nodes, ensure that all the different instances have unique priorities.

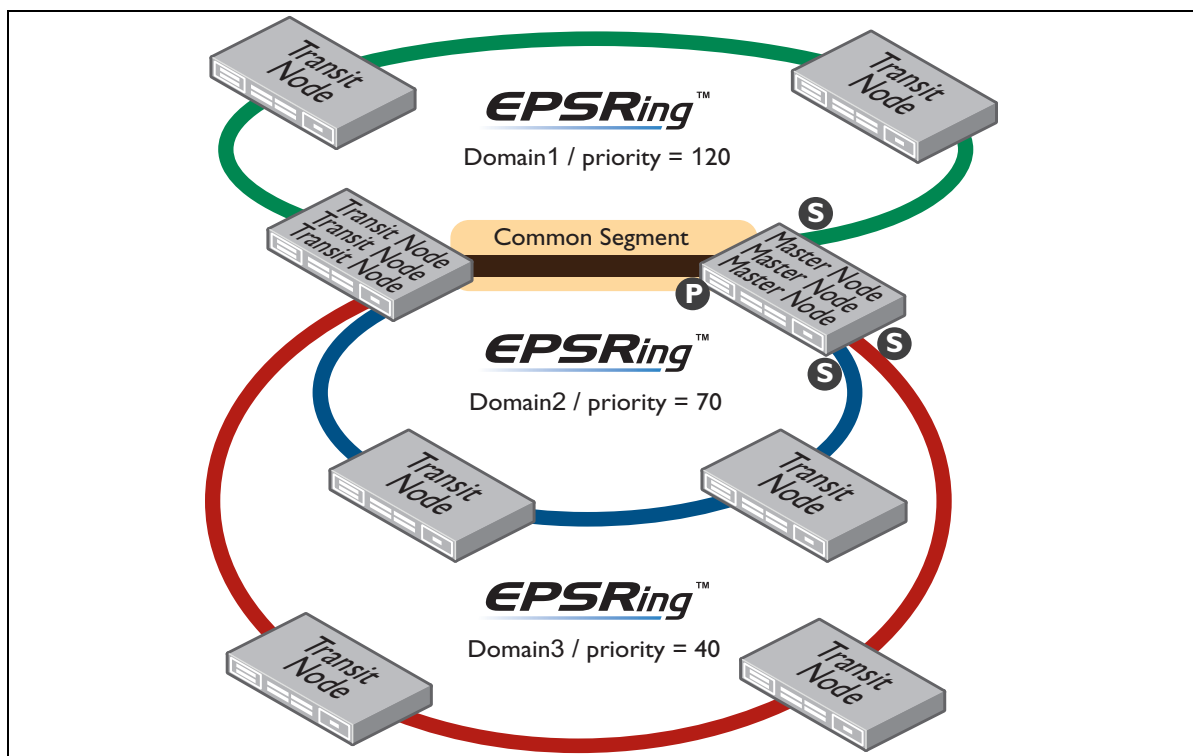
### EPSR-SLP Data VLANs

- During deployment, you need to define the same set of Data VLANs for all member nodes of EPSR-SLP domains.
- When configuring multiple EPSR-SLP instances on a common segment node, the switch performs checks to ensure that all instances on any identified common segment ports share the same set of data VLANs. If any of these checks fail, the switch does not accept the command, and returns an error message.
- Either remove the native VLAN from ring ports, or ensure that the native VLAN is specified as an EPSR Data VLAN.

### Placement of EPSR master node

- Master nodes can be placed on a common segment, but it is generally better not to.
- Each Master's port that connects to the common segment must be configured as the **primary** port.

**Note:** Remember you can check EPSR configuration by using the **show** commands, see "[EPSR show commands](#)" on page 74.



### Master node on common segment: avoiding inappropriate blocking

Master nodes can be placed on a common segment, but it is generally better not to.

If a master node is located on a common segment, the port connecting to the common segment on the master node's Master instance must be a primary port.

This is to avoid inappropriate physical blocking. A master node's secondary ports must not connect to the common segment, because in normal operation secondary ports are blocking. In the case of the highest priority Master, this would result in physical blocking, which would unnecessarily prevent lower priority domains from having access to the common segment.

Also note that if you have:

- a device on the common segment acting as the master node for multiple EPSR rings,
- and you disable the rings and later re-enable them (e.g., when doing EPSR configuration changes),
- then you must re-enable the ring with the highest EPSR SLP priority **last**.

If you do not re-enable the rings in that order, the primary port may end up permanently in a blocking state. If the primary port does end up in a permanent blocking state, recover by disabling and re-enabling the EPSR ring that has the highest priority.

### Co-existence with non-SLP EPSR instances

If a node with an EPSR-SLP instance also has other non-SLP EPSR instances present, these instances are not protected by EPSR-SLP. The non-SLP instances cannot have any Data VLANs in common with the EPSR-SLP instances.

Therefore, if an EPSR instance has to share any VLANs with other EPSR instances, then EPSR-SLP must be enabled on all those instances.

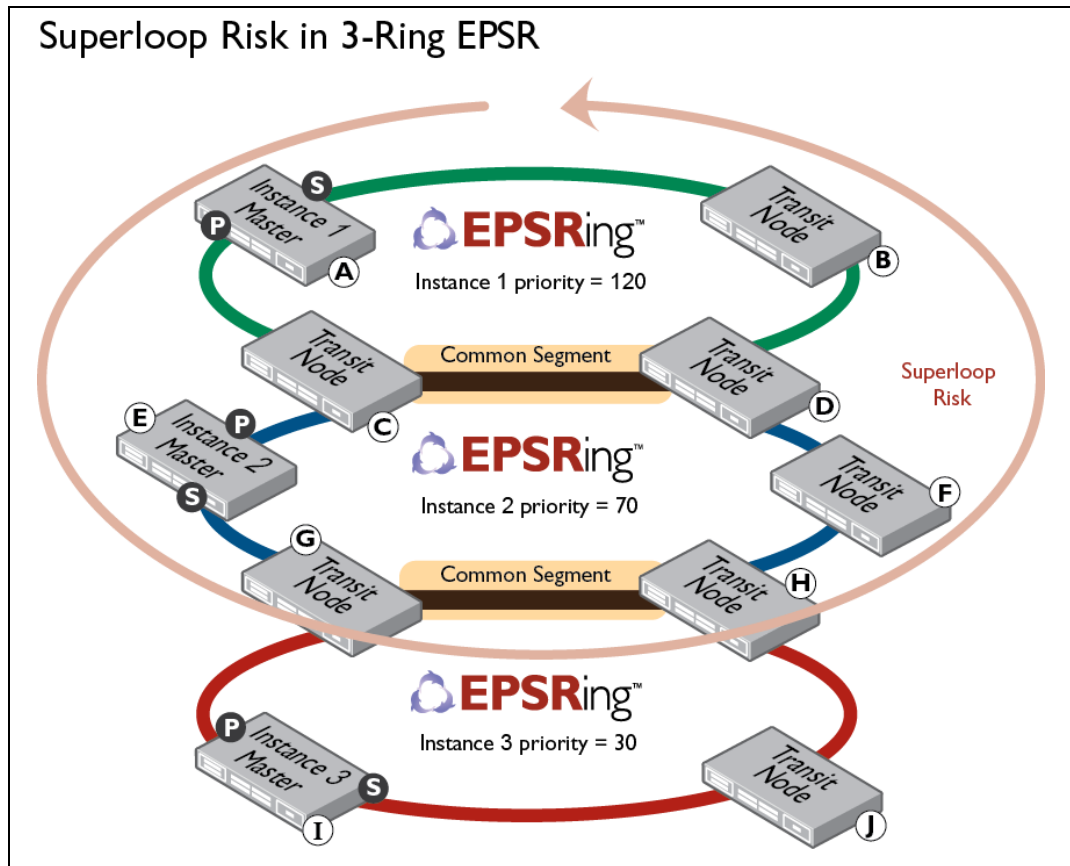
### Caution about Enhanced Recovery with EPSR-SLP topologies with 3 or more rings

Any EPSR-SLP topology that includes three or more rings with two or more common segments (i.e. ladder topology) must not have Enhanced Recovery enabled on the common segment nodes. Referring to the diagram below, the problem with having Enhanced Recovery enabled is that a SuperLoop will form if the following sequence of events occurs:

1. Both common segments go down (i.e. the common segments between switches C and D; G and H).
2. The common segment between switches G and H becomes available again, but the other common segment (between switches C and D) remains down.

Let's look at the sequence of events that will cause a SuperLoop to form in this scenario.

1. When the common link between switches G and H is repaired, they send LinkForwardRequest messages to their highest priority Master, which is switch E.
2. Because the link between C and D is still down, the healthcheck packets that switch E sends do not arrive back on its secondary port. So, switch E sees the ring as down, and therefore permits switches G and H to transition their ports on the common segment to Forwarding.
3. At this point, because the secondary ports of Master switches E and A are still Forwarding, a SuperLoop forms around the path A->B->D->F->H->G->E->C->A



#### Precaution rule for 3-or-more rings EPSR-SLP topologies

To avoid this SuperLoop storm, do not enable Enhanced Recovery on common segment nodes for EPSR topologies that:

- Involve three or more rings, and
- Include two or more common segments.

## Cases where manual intervention may be required

The guiding principle in EPSR protocol design is to avoid loops. This principle means that in some cases the automatic recovery from ring failure will be very slow, and manual intervention is required to achieve faster recovery.

### High priority master reboot when ring is down

In some situations, unexpected Secondary blocking can occur.

When an EPSRing is broken, the highest priority master node's secondary port enters the Failed state. The master node's secondary port must receive a Link Down message before the Failover timer expires, in order to re-enter the Forwarding state.

If the master node has rebooted while the ring was in the Failed state, then after this reboot the master node cannot receive new Link Down messages. As such, it enters the Blocking state. The same situation occurs when the secondary port has its state toggled.

Furthermore, after a reboot, the master node cannot judge whether it is safe to allow its secondary port to forward. For example, it does not know if:

- It is the highest priority Master
- Any other Master in the multi-ring topology is already forwarding

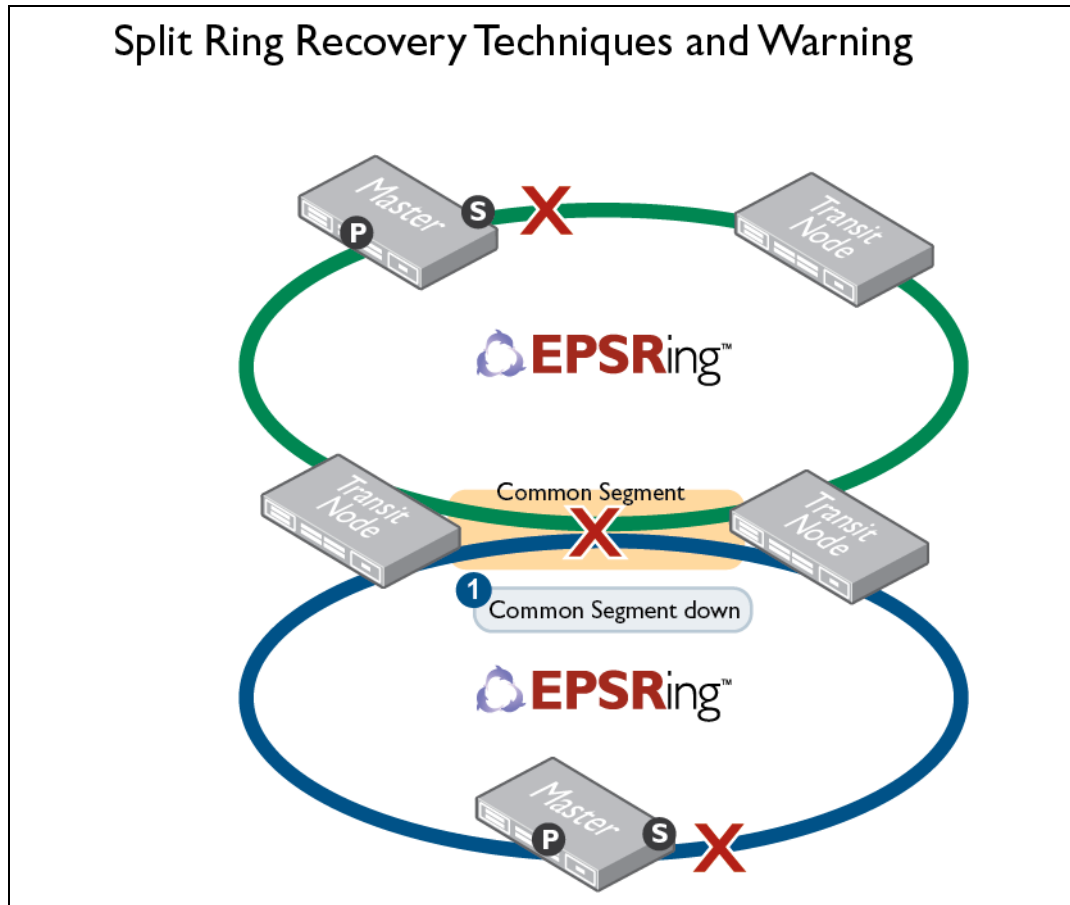
In some cases this can lead to a split ring. If you cannot quickly repair the common segment, you can manually intervene, using the following techniques.

### Split ring recovery techniques and warning

A split ring can occur in a two-ring SuperLoop topology with transit nodes on the common segment, and with Enhanced Recovery enabled on all nodes. The split ring occurs when a common segment fails and is then followed by either a Highest Priority master node reboot, or a secondary port state toggle.

The split ring can be described as a 2-ring topology segmented into two isolated sides of the failed common segment. This split-ring is not automatically restored until the common segment comes back up. You can manually fix this, but the resulting configuration is not without risk. See ["Manual fix" on page 81](#).





## Manual fix

You can temporarily allow connectivity by setting the state of the secondary port to Forwarding:

1. disable EPSR
2. disable SuperLoop
3. re-enable EPSR

You should return to your normal configuration before the common segment is repaired, using the following instructions.

### To return to normal configuration

1. On the highest priority Master which needs to forward, disable EPSR using:

```

epsr3(config)# epsr configuration
epsr3(config-epsr)# epsr a state disabled
epsr3(config-epsr)# epsr a priority 0
epsr3(config-epsr)# epsr a state enabled

```

2. Use the **show epsr** command to confirm this action:

```

EPSR Information
-----
Name ..... A
Mode ..... Master
Status ..... Enabled
State ..... Failed
Control VLAN ..... 5
Data VLAN(s) ..... 40
Interface Mode ..... Channel Groups Only
Primary Port ..... sa1
    Status ..... Forwarding
    Is On Common Segment ..... No
    Blocking Control ..... Physical
Secondary Port ..... sa2
    Status ..... Forwarding
    Is On Common Segment ..... No
    Blocking Control ..... Physical
Hello Time ..... 1 s
Failover Time ..... 2 s
Ring Flap Time ..... 0 s
Trap ..... Enabled
Enhanced Recovery ..... Enabled
Priority ..... 0 [SuperLoop prevention disabled]
-----

```

3. Ensure that the fibre repairers notify you when the common segment is close to reconnection. Before it is actually re-connected, you must enable EPSR, and enable SuperLoop at its previous priority setting:

```

epsr3(config)# epsr configuration
epsr3(config-epsr)# epsr A state disabled
epsr3(config-epsr)# epsr A priority 10
epsr3(config-epsr)# epsr A state enabled

```

#### EPSR Information

```
-----
Name ..... A
Mode ..... Master
Status ..... Enabled
State ..... Failed
Control VLAN ..... 5
Data VLAN(s) ..... 40
Interface Mode ..... Channel Groups Only
Primary Port ..... sa1
    Status ..... Forwarding
    Is On Common Segment ..... No
    Blocking Control ..... Physical
Secondary Port ..... sa2
    Status ..... Blocked (for SuperLoop prevention)
    Is On Common Segment ..... No
    Blocking Control ..... Physical
Hello Time ..... 1 s
Failover Time ..... 2 s
Ring Flap Time ..... 0 s
Trap ..... Enabled
Enhanced Recovery ..... Enabled
Priority ..... 10
-----
```

4. As shown above, the EPSR instance's secondary port is Blocked until the common segment is reconnected.

**Note:** It is very important to enable SuperLoop before the common segment is reconnected. Otherwise, the network is at risk of another, possibly longer SuperLoop storm occurring during reconnection.