FIBERROAD

Media Redundancy Protocol(MRP) Configuration Guide

Application Note

Rev. APPL-2022.03
2022-03-31

Table of Contents

1. Introduction

This document explains how to configure Media Redundancy Protocol (MRP - not to be confused with IEEE's Multiple Registration Protocol) features .

MRP is defined by International Electrotechnical Commission (IEC) in the 62439-2 standard, edition 2.0, 2016-03. In the following, this is referred to as "the standard".

MRP is a recovery protocol based on a ring topology, designed to react deterministically on a single failure of an inter-switch link or switch in the network, under the control of a dedicated media redundancy manager node.

Unlike another well-known ring protocol like ERPS, MRP blocks or unblocks the entire ring port rather than only a set of VLAN IDs.

2. MRP Rings

The following shows an example of an MRP ring with four switches (interchangeably called nodes in this guide).

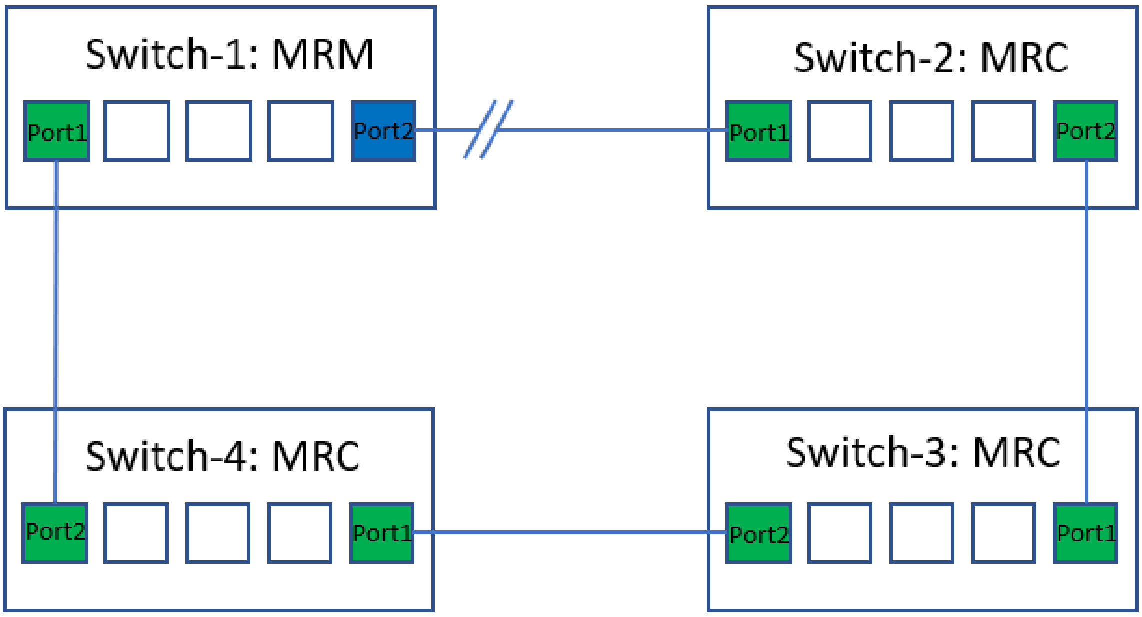

Figure 1. Example of a closed MRP ring

The elements of an MRP ring is outlined in the following sections.

2.1. Ring Instances

The MRP protocol can be quite CPU consuming as more than 2000 PDUs per second may have to be both transmitted, received and processed by each node in a single MRP ring, and even more if the ring also interconnects with another ring.

Therefore, each node can handle between 1 and N separate rings at a time. The upper limit is defined by the actual switch chip and its hardware capabilities. The more hardware offload features the switch chip supports, the more ring instances can be created simultaneously:

Some switch chips do not have any hardware offload and are limited to creating only one single ring instance, since everything is handled by software.

Other chips have hardware support for automatic frame transmission and can support a few more ring instances.

Yet, others have support for both automatic frame transmission and reception, so the burden on the CPU is minimal. These chips have support for up to X/2 rings, where X is the number of ports on the platform on which the chip resides.

2.2. Ring Ports

Each ring instance has two dedicated ports known as ring ports. These are often referred to as `Port1` and `Port2`. Any port on the switch can be assigned as a ring port, but the standard mandates the link speed be at least 100 Mbps, full duplex.

> **NOTE** The IStaX MRP implementation does not check for for actual link speed and duplex, so it is up to the end user to ensure proper port configuration.

2.3. Media Redundancy Manager (MRM)

A ring must have one Media Redundancy Manager (MRM), which is responsible for controlling the ring state.

Initially, the MRM blocks one ring port and unblocks the other. When a ring port is blocked, no frames can ingress and egress that port except for certain CPU-destined and -generated frames. The unblocked (forwarding) ring port is called the primary ring port. The other is called the secondary ring port.

The MRM sends MRP_Test PDUs at a configured time period in both directions of the ring and if it does not receive its own PDUs after a certain time, it unblocks the secondary ring port and transmits an MRP_TopologyChange PDU, which contains an MRP_Interval parameter that indicates when the MRCs must flush their forwarding database (FDB).

NOTE: Only one node in the ring may be configured as an MRM. If you have multiple nodes configured as MRM, one node will block for the other node's MRP_Test PDUs. As a consequence, each MRM will unblock both its ring ports, but still not pass MRP_Test PDUs through from one ring port to the other. This is likely to cause loops in the network.

The ring can be in one of two states: Closed and Open. Figure 1 shows an example of a closed ring, where the MRM has blocked the secondary ring port.

In Figure 2, the ring is in open state, because of a signal fail (SF) on one of the ring's ring ports. Here, the MRM has unblocked its secondary ring port.

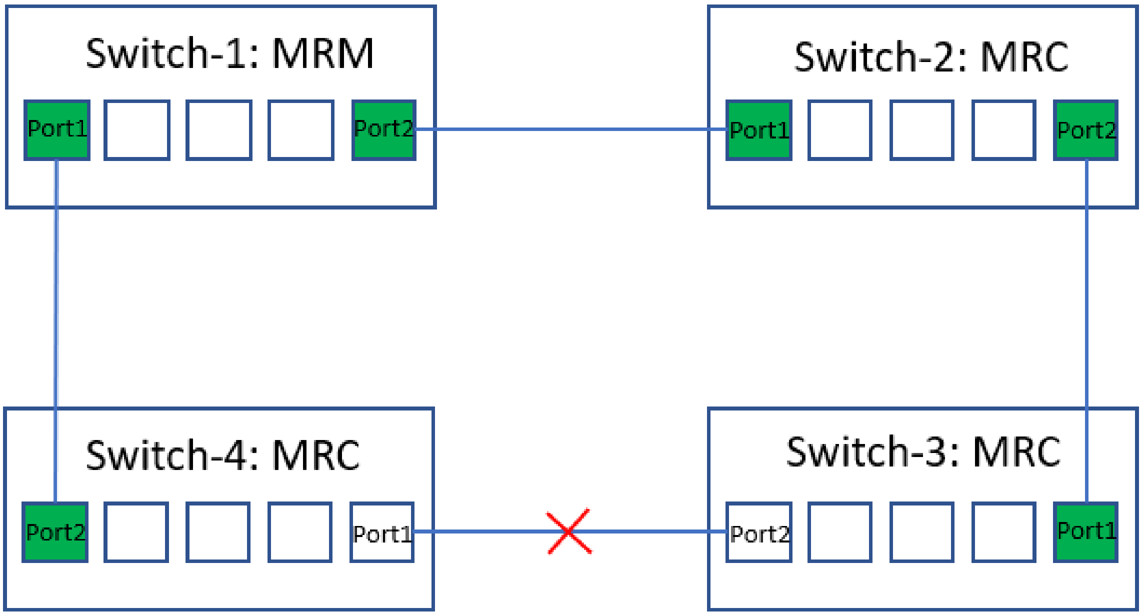

Figure 2. Example of an open MRP ring

2.4. Media Redundancy Client (MRC)

The remaining nodes of an MRP ring must be Media Redundancy Clients (MRCs).

In normal configuration, the MRC unblocks both its ring ports and allows MRP_Test PDUs from the MRM to pass through. Upon link failure, these PDUs cannot pass through and the MRM detects the failure and unblocks its blocked ring port.

On link change, the MRC transmits an MRP_LinkChange PDU, which can take the form of an MRP_LinkUp or MRP_LinkDown PDU.

An MRC is responsible for forwarding all MRP PDUs from one ring port to the other - if possible.

NOTE If all nodes in the ring are configured as MRCs, there is no MRM to block one of its ring ports, and since an MRC by default unblocks both of its ring ports, it is likely to cause loops in the network.

2.5. Media Redundancy Automanager (MRA)

Instead of configuring a node as an MRM or an MRC, it may be configured as a Media Redundancy Automanager (MRA). MRAs select one MRM among each other by using a voting protocol based on MRP_Test PDUs and MRP_TestMgrNAck PDUs.

Each MRP_Test PDU contains a priority field, which is provisioned by the end user. A lower numerical value indicates a higher priority.

Whenever a new MRA joins the ring, it starts sending MRP_Test PDUs with its own priority. All MRAs receive these MRP_test PDUs and compare the priority field to their own priority.

If an MRA’s own priority is higher (numerically lower), it returns an MRA_MgrTestNAck PDU to the sender and remains an MRM. The recepient of an MRA_Test_MgrNAck PDU ceases its MRM role and becomes an MRC.

If an MRA’s own priority is lower (numerically higher), it becomes an MRC.

If an MRA’s own priority is equal to the received priority, it’s the node’s MAC address that determines the behavior: The lower MAC address the higher priority.

MRAs operating as MRCs continuously monitor MRP_Test PDUs to see if they need to return to the MRM role in case of missing MRP_Test PDUs or in case it has higher priority than what it receives.

NOTE: It must be ensured that a given ring does not contain nodes configured as MRMs as well as nodes configured as MRAs. The reason is that a node configured as an MRM doesn’t have the ability to send MRP_TestMgrNAck PDUs back to a node configured as MRA (even though the standard specifies that the priority of an MRM must be numerically lower than the priority of an MRA). An MRA will therefore think that it is the manager of the ring, causing the ring to have multiple MRMs, which may cause loops as outlined in the note under the MRM definition. So in short, the priority of a node is only relevant for MRAs and not MRMs.

2.6. Recovery Profiles

The MRM and MRC contain a set of parameters that specify the maximum recovery time of a ring. Table 59 and Table 60 of the standard outline four consistent sets of such maximum recovery times for MRM and MRC, respectively. These are 10 ms, 30 ms, 200 ms, and 500 ms. In this configuration guide, one set of parameters is called a recovery profile.

NOTE

Not all four recovery profiles are available on all chips. Only the two slowest (200 and 500 ms) are available on software-based MRP implementations, whereas also the two fastest (10 ms and 30 ms profiles) are available on hardware-based MRP implementations.

NOTE

It is very important that all nodes in a ring run the same recovery profile. If not, MRAs may repeatedly turn from MRC to MRM and back to MRC, making the ring unstable.

2.7. Signal Fail Trigger

The IStaX-based MRP implementation supports either the physical link state on the ring ports to detect signal fail and recovery or the use of CFM MEPs.

If the ring ports are connected back-to-back with its partner node (which is a normal case, because otherwise MRP_Xxx PDUs will flood the network unless otherwise handled by the intermediate switches, e.g. through VLAN configuration), you may rely on using link state.

NOTE Not all IStaX-based switches support the recommended CCM rates of 3.3 ms or 10 ms, so on such switches, it makes sense to only use the link state and not CFM MEPs as signal fail trigger.

3. Interconnected Rings

The standard provides a method for interconnecting rings. To redundantly connect two MRP rings, two nodes of each ring are assigned additional roles, besides being operating as either MRM or MRC.

One of the nodes is assigned the role of a Media redundancy Interconnection Manager (MIM). The remaining three nodes (one in the MIM’s ring and the two of the other ring) are assigned the role of Media redundancy Interconnection Clients (MICs).

The MIM connects to a MIC in the other ring, and the two remaining MICs connect to each other. In this way, a third ring - the interconnection ring - is formed between the four I/C nodes.

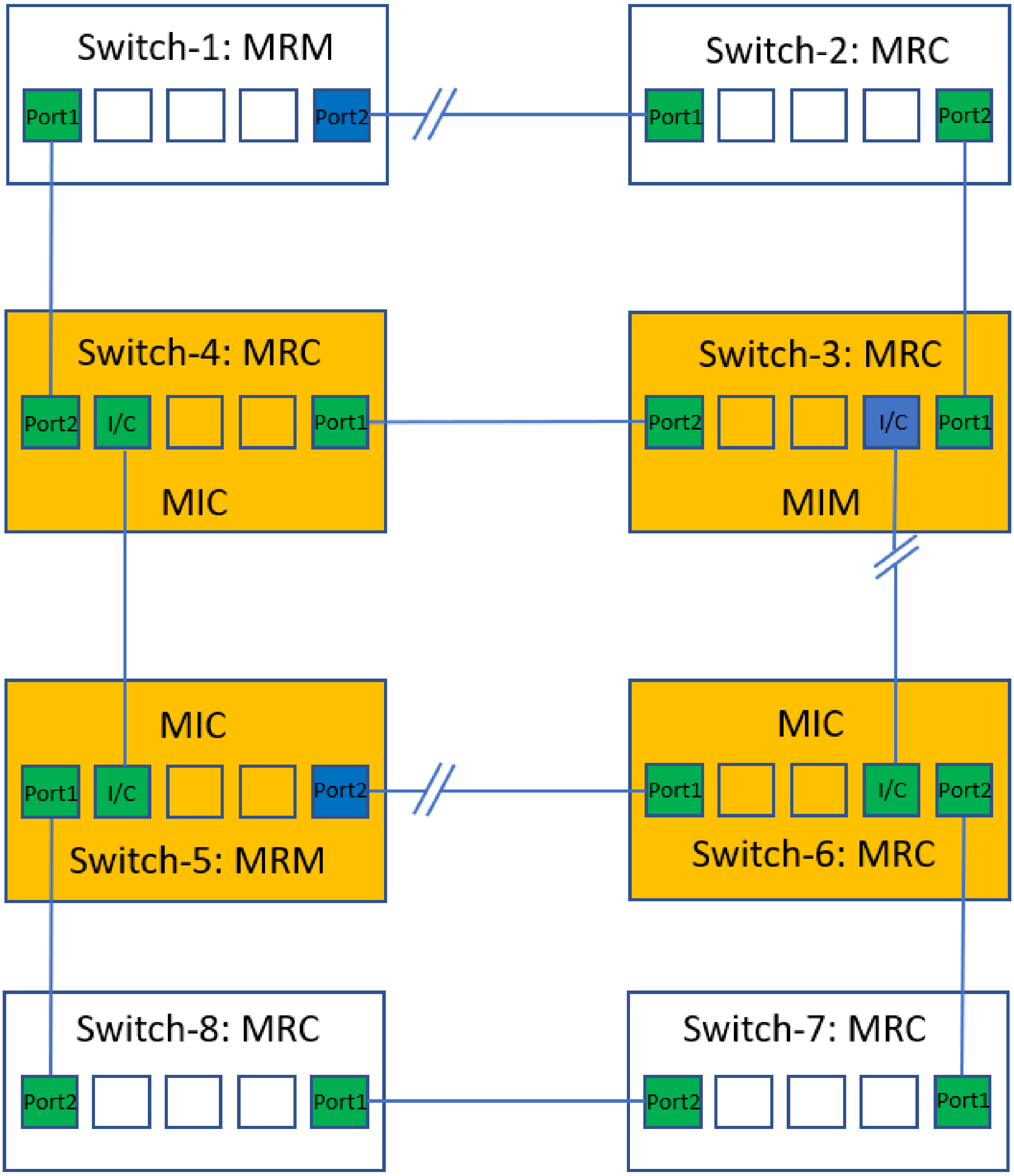

Figure 3. Example of two interconnected rings

It is possible to interconnect more than two rings in a ladder topology. To interconnect N rings, N-1 interconnection rings are required.

Each interconnection ring must have its own, unique ID, called the interconnection ID, or IID in short.

3.1. Media redundancy Interconnection Manager (MIM)

The function of the MIM is to observe and control the redundant I/C topology.

It does so by blocking and unblocking the I/C port as it sees fit.

The MIM operates in one of two modes:

Link Check

In link check (LC) mode, the MIM starts by sending MRP_InLinkStatusPoll PDUs to ask for the MICs' current link status. The MICs in turn reply with MRP_InLinkUp if their I/C port has link or MRP_InLinkDown PDUs if their I/C port has signal fail.
A MIC autonomously sends MRP_InLinkUp or MRP_InLinkDown PDUs whenever the I/C port changes its link status, and the MIM blocks or unblocks its I/C port.

Ring Check

In ring check (RC) mode, the MIM continuously transmits MRP_InTest PDUs on both its ring ports and its I/C port. If these PDUs come back to the MIM, the interconnection ring is considered closed, and the MIM blocks its I/C port. If they don't, the MIM considers the interconnection ring open, and the MIM unblocks its I/C port.

No matter the interconnection mode, the MIM transmits MRP_InTopologyChange PDUs whenever it changes its I/C port's forwarding state. These PDUs ask the three MICs to flush their FDB at a certain point in time.

An MRM picks up these PDUs and transforms then into MRP_TopologyChange PDU and transmits them on both its ring ports, asking the MRCs to also flush their FDB.

NOTE

The IStaX-based switches support both interconnection modes, but if more than two rings are to be interconnected, the Link Check mode must be used in all interconnection rings.
The reason is that it is hardware that forwards the MRP_InTest PDUs between the ring and I/C ports on MICs and non-MIC/MIM nodes, and hardware cannot do so based on the IID inside the MRP_InTest PDUs.
On the other hand, software takes care of forwarding all other MRP_InXxx PDUs, and can therefore distinguish between different IIDs and forward them appropriately.

3.2. Media redundancy Interconnection Client (MIC)

The MIC is responsible for forwarding MRP_InXxx PDUs between its ring and I/C ports and react on MRP_InTopologyChange and MRP_InLinkStatusPoll PDUs from the interconnection's MIM.

The MIC is also configured with either a Link Check or Ring Check mode. The main difference is that the MIC only replies to MRP_InLinkStatusPoll PDUs from the MIM in Link Check mode and that it only forwards MRP_InTest PDUs between ring and I/C ports in Ring Check mode.

3.3. Recovery Profiles

The MIM and MIC contain a set of parameters that specify the maximum recovery time of an interconnection. Table 61 and Table 62 of the standard outline two consistent sets of such maximum recovery times for MIM and MIC, respectively. These are 200 ms, and 500 ms. In this configuration guide, one set of parameters is called a recovery profile.

NOTE It is important that all MIM and MIC nodes in the same interconnection run the same recovery profile.

3.4. Signal Fail Trigger

The standard mandates CFM MEP instances be instantiated in both ends of the interconnection link when the interconnection topology is in LC-mode. It *may* use MEPs when the interconnection is in RC-mode.

NOTE This implementation allows for using link state as signal fail trigger in both LC and RC mode. It is up to the end user to determine a suitable mode.

See also description of this under Signal Fail Trigger for ring ports.

NOTE The standard suggests a very specific coding of a MEP’s Maintenance Association ID (MAID). This coding suggests a dynamic change of the Maintenance Domain Name depending on the state of MRP.
Since CFM and MRP are decoupled entities in the IStaX software solution, it is not possible to have MRP dynamically change the Maintenance Domain Name.
The end user may configure the MEPs in any way the CFM module supports, one of which may be as close as possible to what the MRP standard suggests.

4. Configuration

With all the terms and definitions in place, let’s see how to configure MRP on IStaX-based switches using CLI.

The syntax for creating an MRP instance at the global level is:

```
# configure terminal
(config)# media-redundancy <inst>
```

where:

```
inst      MRP instance number, which is a number between 1 and the number of
          supported instances on the switch.
```

Similarly, the syntax for deleting an MRP instance at the global level is:

```
# configure terminal
(config)# no media-redundancy {<inst> | all}
```

where:

```
inst      MRP instance number to delete.
all       Deletes all MRP instances.
```

Once an instance is selected, CLI enters `config-media-redundancy` mode, where the following commands are available:

```
[no] role {mrc | mrm | mra}
[no] name <string1-240>
[no] uuid <string36-36>
[no] oui {default | siemens | custom <oui>}
[no] port1 interface <port_type_id>
[no] port2 interface <port_type_id>
[no] port1 sf-trigger {link | {mep domain <kword1-15> service <kword1-15> mep-id
<1-8191>}}
[no] port2 sf-trigger {link | {mep domain <kword1-15> service <kword1-15> mep-id
<1-8191>}}
[no] control-vlan <vlan_id>
[no] recovery-profile {10ms | 30ms | 200ms | 500ms}

[no] mrm priority <uint>
[no] mrm react-on-link-change

[no] interconnection role {mic | mim | none}
[no] interconnection mode {link-check | ring-check}
[no] interconnection id <uint16>
[no] interconnection name <string1-240>
[no] interconnection interface <port_type_id>
[no] interconnection sf-trigger {link | {mep domain <kword1-15> service <kword1-15>
mep-id <1-8191>}}
[no] interconnection control-vlan <vlan_id>
[no] interconnection recovery-profile {200ms | 500ms}

admin-state {enable | disable}
```

where

Command	**Purpose**

`[no] role`	Set the role of this node for this ring instance. The no-form sets the default, which is mra.
`[no] name`	Set the name of this instance. This is only used for easy identification and has no effect on how the MRP instance operates. The no-form defaults the name to an empty string.
`[no] uuid`	Set the UUID/DomainID for this instance. The format is "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx", where x is a hexadecimal digit. The UUID/DomainID is used in MRP PDUs (encoded as 16 bytes) for identification of which ring a given PDU belongs to. It is recommended - but not required - that the DomainID/UUID is the same on all nodes in a given ring. The no-form sets the UUID/DomainID to all-Fs.
`[no] oui`	An OUI is included in MRP_Test PDUs if configured role is MRA and in MRP_TestMgrNAck and MRP_TestPropagate PDUs. By default, the OUI is the three most significant bytes of the switch’s MAC address. Due to a bug in Wireshark’s dissector of MRP PDUs, which manifests itself in the fact that it only can dissect PDUs with the Siemens OUI (08-00-06), it is possible to change the OUI of these three PDU types to that of Siemens. And there is an option for changing to a custom OUI. The OUI is as such not used for anything but identification of switchmanufacturer/MRP protocol implementator. The no-form sets it to 'default'.
`[no] port1 interface`	Selects the port/interface on the switch that represents MRP port1. The no-form dis-associates a port number. If attempting to enable the MRP instance without an assigned port1 interface, an error will be shown.
`[no] port2 interface`	See 'port1 interface'.
`[no] port1 sf-trigger`	Selects how to determine Signal Fail. By default, the port’s link state determines whether or not the connection to the link partner is up or down. Alternatively, a reference to a CFM MEP running on that port can be specified. If the specified MEP for one or the other reason is not working, the MRP implementation will issue an operational warning and use the link state instead. The no-form causes link state to be used.
`[no] port2 sf-trigger`	See "port1 sf-trigger".

`[no] control-vlan`	By default, all MRP_Xxx PDUs are transmitted untagged. With this command, a VLAN tag with the provided VLAN ID can be inserted into the PDUs. The end user must ensure that all ring ports are members of this VLAN ID. The TPID of the VLAN tag follows the ring ports' TPID, and the PCP will always be 7. The standard mandates that reception of both tagged - with any VLAN ID - and untagged PDUs must be processed by the MRP implementation, so there is no check for actual VLAN ID. However, an operational warning may be issued if, for instance, a ring port is not a member of the given VLAN ID. The no-form causes transmission of untagged PDUs.
`[no] recovery-profile`	This selects the recovery profile. Not all switches support all recovery profiles. The unsupported profiles will not be filtered available on the command line. The no-form defaults the recovery profile to '500ms'.
`[no] mrm priority`	Sets the priority of the MRA/MRM. Used when operational role is MRM. The lower numerical value, the higher priority. As discussed in the last paragraph of the MRA section, the priority is really not relevant when the configured role is MRM. The standard, however, mandates the priority of nodes configured as MRMs to be 0x0000, 0x1000-0x7000, or 0x8000. And for nodes configured as MRAs it needs to be0x9000-0xF000 or 0xFFFF. The no-form sets it to 0xA000, assuming the configured role is MRA.
`[no] mrm react-on-link-change`	Instructs the node operating as an MRM to react immediately on MRP_LinkDown PDUs in the closed ring state by unblocking the secondary ring port and send MRP_TopologyChange PDUs out on both ring ports. The no-form instructs it to wait until it finds its own MRP_Test PDUs to be missing. Default is the latter.
`[no] interconnection role`	When also using this switch as an interconnection node, set the role to being a MIM or a MIC. The no-form causes it not to behave as an interconnection node (default).
`[no] interconnection mode`	The interconnection can operate in either link-check or ring-check as described in the MIM section. The no-form is also the default, which is link-check.

`[no] interconnection id`	The interconnection ID (IID) is a 16-bit number that uniquely identifies the four nodes making up this interconnection domain. All four nodes must use the same IID. The no-form is also the default, which is 0.
`[no] interconnection name`	Set the name of this interconnection domain. This is only used for easy identification and has no effect on how the MRP instance operates. The no-form defaults the name to an empty string.
`[no] interconnection interface`	Selects the port/interface on the switch that represents the interconnection port. The no-form dis-associates a port number.
`[no] interconnection sf-trigger`	See "port1 sf-trigger".
`[no] interconnection control-vlan`	By default, all MRP_InXxx PDUs are transmitted untagged. With this command, a VLAN tag with the provided VLAN ID can be inserted into the PDUs. The end user must ensure that both ring ports and the interconnection port are members of this VLAN ID. The TPID of the VLAN tag follows the egress port's TPID, and the PCP will always be 7. The standard mandates that reception of both tagged - with any VLAN ID - and untagged PDUs must be processed by the MRP implementation, so there is no check for actual VLAN ID. However, an operational warning may be issued if, for instance, one of the three ports is not a member of the given VLAN ID. The no-form causes transmission of untagged PDUs.
`[no] interconnection recovery-profile`	This selects the interconnection's recovery profile. The no-form defaults the recovery profile to '500ms'.
`admin-state`	Enable the MRP intance once all configuration parameters are in place. When disabling an MRP instance, all configuration is removed from hardware and statistics and status is cleared. Be aware that disabling an MRP instance may cause loops in the network if not properly guarded by other means (e.g. Spanning Tree Protocol (STP)).

5. Configuration Examples

The next sections outline a couple of simple examples. The first shows how to configure one MRP ring, and the other shows how to configure two interconnected MRP rings.

5.1. Example 1: Simple MRP Ring

Utilizing Figure 1 as an example, and assuming `port1` is `interface GigabitEthernet 1/3` and `port2` is `interface GigabitEthernet 1/4`, we can put together the configuration of the four switches.

5.1.1. Example 1-1. Simple MRP Ring Made up of MRAs

First, let us start with the simplest possible configuration, which is to configure all four switches with the default rule, MRA, and using link-state as signal fail trigger.

Switch-1, Switch-2, Switch-3, and Switch-4 all have the same configuration:

Command	Purpose
`Switch-X# configure terminal`	Enter configuration mode.
`Switch-X(config)# media-redundancy 1`	Enter configuration mode for MRP instance number 1.
`Switch-X(config-media-redundancy)# port1 interface GigabitEthernet 1/3` `Switch-X(config-media-redundancy)# port2 interface GigabitEthernet 1/4`	Set the two ring ports to the desired interfaces.
`Switch-X(config-media-redundancy)# admin-state enable`	Enable the MRP instance.
`Switch-X(config-media-redundancy)# interface GigabitEthernet 1/3,4`	Select the two ring ports.
`Switch-X(config-if)# no spanning-tree`	Disable spanning tree.
`Switch-X(config-if)# end` `Switch-X#`	We are now done with configuration of Switch-X.

NOTE

The next-to-last step disables spanning tree (STP) on the two ring ports. This is required in order to avoid STP blocking a port that MRP has set to forwarding and vice versa.
If the end user forgets to disable STP, an operational warning will be issued.

To summarize the configuration, here’s a list of commands, where the default commands are omitted.

```
# show running-config feature media-redundancy
Building configuration...
media-redundancy 1
 port1 interface GigabitEthernet 1/3
 port2 interface GigabitEthernet 1/4
 admin-state enable
!
end
```

The entire MRP configuration that includes defaults can be seen with the following command:

```
# show running-config feature media-redundancy all-defaults
Building configuration...
media-redundancy 1
 role mra
 no name
 uuid "FFFFFFFF-FFFF-FFFF-FFFF-FFFFFFFFFFFF"
 oui default
 port1 interface GigabitEthernet 1/3
 port2 interface GigabitEthernet 1/4
 port1 sf-trigger link
 port2 sf-trigger link
 no control-vlan
 recovery-profile 500ms
 mrm priority 0xa000
 no mrm react-on-link-change
 interconnection role none
 interconnection mode link-check
 interconnection id 0
 no interconnection name
 no interconnection interface
 interconnection sf-trigger link
 no interconnection control-vlan
 interconnection recovery-profile 500ms
 admin-state enable
!
end
```

With this configuration, there is no guarantee that Switch-1 becomes the MRM. The switch that becomes the MRM when all switches' priorities are identical will be the one with the lowest MAC address.

In order to force Switch-1 to become the MRM, adjust its priority to have a lower numerical value than the default:

Command	**Purpose**
`Switch-1# configure terminal`	Enter configuration mode.

`Switch-1(config)# media-redundancy 1`	Enter configuration mode for MRP instance number 1.
`Switch-1(config-media-redundancy)# mrm priority 0x9000`	Raise the priority by lowering the numerical value from default 0xa000 to 0x9000.
`Switch-1(config-if)# end` `Switch-1#`	We are now done with reconfiguration of Switch-1.

5.1.2. Example 1-2. Simple MRP Ring Made up of One MRM and Three MRCs

This is almost as simple as the previous example, but it requires two different configurations: One for the MRM and one for the MRCs, like this (supposing Switch-1 is the MRM):

Switch-1:

Command	**Purpose**
`Switch-1# configure terminal`	Enter configuration mode.
`Switch-1(config)# media-redundancy 1`	Enter configuration mode for MRP instance number 1.
`Switch-1(config-media-redundancy)# role mrm`	Set this switch to become the manager of the ring.
`Switch-1(config-media-redundancy)# port1 interface GigabitEthernet 1/3` `Switch-1(config-media-redundancy)# port2 interface GigabitEthernet 1/4`	Set the two ring ports to the desired interfaces.
`Switch-1(config-media-redundancy)# admin-state enable`	Enable the MRP instance.
`Switch-1(config-media-redundancy)# interface GigabitEthernet 1/3,4`	Select the two ring ports.
`Switch-1(config-if)# no spanning-tree`	Disable spanning tree.
`Switch-1(config-if)# end` `Switch-1#`	We are now done with configuration of the MRM.

Switch-2, Switch-3, and Switch-4:

Command	**Purpose**
`Switch-2,3,4# configure terminal`	Enter configuration mode.
`Switch-2,3,4(config)# media-redundancy 1`	Enter configuration mode for MRP instance number 1.
`Switch-2,3,4(config-media-redundancy)# role mrc`	Set these switches to become clients on the ring.
`Switch-2,3,4(config-media-redundancy)# port1 interface GigabitEthernet 1/3` `Switch-2,3,4(config-media-redundancy)# port2 interface GigabitEthernet 1/4`	Set the two ring ports to the desired interfaces.
`Switch-2,3,4(config-media-redundancy)# admin-state enable`	Enable the MRP instance.
`Switch-2,3,4(config-media-redundancy)# interface GigabitEthernet 1/3,4`	Select the two ring ports.
`Switch-2,3,4(config-if)# no spanning-tree`	Disable spanning tree.
`Switch-2,3,4(config-if)# end` `Switch-2,3,4#`	We are now done with configuration of the MRCs.

5.2. Example 2: Interconnected MRP Rings

With the outset in Figure 3, let us configure the eight switches.

We assume that the I/C port is `GigabitEthernet 1/5` on all four I/C nodes that make up the I/C topology and that we run the interconnection topology in link-check mode.

The standard mandates the use of MEPs on the interconection ports in LC mode, but to emphasize that link state signal fail triggering work just as well, we run two I/C nodes, Switch-3 and Switch-6, with MEPs as signal fail triggering, and the remaining two I/C nodes with link state as signal fail triggering.

All eight switches will be configured as MRAs, so there is no guarantee that the MRMs become those from Figure 3.

Configuration of Switch-1, Switch-2, Switch-7, and Switch-8 is identical to that of Example 1-1, so this will not be repeated here.

In the example, we pick 42 as the interconnection ID.

Configuration of Switch-4 and Switch-5 is identical, since both are MICs.

Switch-4 and Switch-5:

Command	Purpose
`Switch-4,5# configure terminal`	Enter configuration mode.
`Switch-4,5(config)# media-redundancy 1`	Enter configuration mode for MRP instance number 1.
`Switch-4,5(config-media-redundancy)# port1 interface GigabitEthernet 1/3` `Switch-4,5(config-media-redundancy)# port2 interface GigabitEthernet 1/4`	Set the two ring ports to the desired interfaces.
`Switch-4,5(config-media-redundancy)# interconnection role mic`	Let these two switches be MICs.
`Switch-4,5(config-media-redundancy)# interconnection interface GigabitEthernet 1/5`	Set the I/C port to the desired interface.
`Switch-4,5(config-media-redundancy)# interconnection id 42`	Provision the IID.
`Switch-4,5(config-media-redundancy)# admin-state enable`	Enable the MRP instance.
`Switch-4,5(config-media-redundancy)# interface GigabitEthernet 1/3-5`	Select the two ring ports and the I/C port.
`Switch-4,5(config-if)# no spanning-tree`	Disable spanning tree.
`Switch-4,5(config-if)# end` `Switch-4,5#`	Done configuring Switch-4 and Switch-5.

NOTE Also the I/C port needs to have STP disabled.

Now, let’s configure Switch-3 and Switch-6. We set the MEP-ID of Switch-3 to 1 and of Switch-6 to 2.

Command	Purpose
`Switch-3,6# configure terminal`	Enter configuration mode.
`Switch-3,6# cfm domain MyMrpIcDomain`	Start configuring the CFM Domain. Name it arbitrarily.

`Switch-3,6# format string "MRP-Orange-Interconnection"`	The MRP standard suggests the maintenance domain format to be a string. Pick any contents of the string that allows CCM PDUs to be identified on the network as belonging to this interconnection in case they escape the link partner.
`Switch-3,6(config-cfm-dmn)# service MyMrpIcService`	Configure a service under the domain. Name it arbitrarily.
`Switch-3,6(config-cfm-dmn-svc)# format integer 42`	The MRP standard suggests to use a 2-byte integer format for the service and the interconnection ID for the number. Hence using '42'.
`Switch-3,6(config-cfm-dmn-svc)# continuity-check interval 10ms`	The MRP standard suggest 3.3ms or 10ms CCM interval. Let's pick 10ms.
`Switch-3(config-cfm-dmn-svc)# mep 1` `Switch-6(config-cfm-dmn-svc)# mep 2`	Create a MEP with MEP-ID 1 on Switch-3 and a MEP with MEP-ID 2 on Switch-6.
`Switch-3,6(config-cfm-dmn-svc-mep)# interface GigabitEthernet 1/5`	Let the MEP reside on the interconnection port.
`Switch-3(config-cfm-dmn-svc-mep)# remote mep 2` `Switch-6(config-cfm-dmn-svc-mep)# remote mep 1`	Let the two MEPs use each other.
`Switch-3,6(config-cfm-dmn-svc-mep)# continuity-check`	Enable CCM check and generation.
`Switch-3,6(config-cfm-dmn-svc-mep)# admin-state enable`	And finally enable the MEPs.
`Switch-3,6(config-cfm-dmn-svc-mep)# media-redundancy 1`	Enter configuration mode for MRP instance number 1.

`Switch-3,6(config-media-redundancy)# port1 interface GigabitEthernet 1/3` `Switch-3,6(config-media-redundancy)# port2 interface GigabitEthernet 1/4`	Set the two ring ports to the desired interfaces.
`Switch-3(config-media-redundancy)# interconnection role mim` `Switch-6(config-media-redundancy)# interconnection role mic`	Let Switch-3 be the MIM and Switch-6 be a MIC.
`Switch-3,6(config-media-redundancy)# interconnection interface GigabitEthernet 1/5`	Set the I/C port to the desired interface.
`Switch-3(config-media-redundancy)# interconnection sf-trigger mep domain MyMrpIcDomain service MyMrpIcService mep-id 1` `Switch-6(config-media-redundancy)# interconnection sf-trigger mep domain MyMrpIcDomain service MyMrpIcService mep-id 2`	Use the newly created MEPs for signal fail triggering.
`Switch-3,6(config-media-redundancy)# interconnection id 42`	Provision the IID.
`Switch-3,6(config-media-redundancy)# admin-state enable`	Enable the MRP instance.
`Switch-3,6(config-media-redundancy)# interface GigabitEthernet 1/3-5`	Select the two ring ports and the I/C port.
`Switch-3,6(config-if)# no spanning-tree`	Disable spanning tree.
`Switch-3,6(config-if)# end` `Switch-3,6#`	Done configuring Switch-3 and Switch-6.

This gives the following running-config for CFM on Switch-3:

```
Switch-3# show running-config feature cfm
Building configuration...
!
cfm domain MyMrpIcDomain
 format string "MRP-Orange-Interconnection"
 service MyMrpIcService
  format integer 42
  continuity-check interval 10ms
  mep 1
   interface GigabitEthernet 1/5
   remote mep 2
   continuity-check
   admin-state enable
   exit
!
end
```

And the following MRP configuration:

```
Switch-3# show running-config feature media-redundancy
Building configuration...
media-redundancy 1
 port1 interface GigabitEthernet 1/3
 port2 interface GigabitEthernet 1/4
 interconnection role mim
 interconnection id 42
 interconnection interface GigabitEthernet 1/5
 interconnection sf-trigger mep domain MyMrpIcDomain service MyMrpIcService mep-id 1
 admin-state enable
!
end
```

6. Status and Statistics

Once an MRP instance is administratively enabled, it’s a good idea to look at its status and in particular for operational warnings.

The following command can show both status and statistics and both of them in a brief or a detailed way.

```
# show media-redundancy [<inst_list>] {status | statistics} [details]
```

6.1. Status

Continuing Example 2, let’s look at a few switches' status.

```
Switch-1# show media-redundancy status
                       Ring      Ring      I/C    I/C Ring  Port1 Port2 I/C   Port1   Port2   I/C
Inst Operational State Role      State     Role   State     SF    SF    SF    Blocked Blocked Blocked
---- ----------------- --------- --------- ------ --------- ----- ----- ----- ------- ------- -------
   1 Active            MRM (MRA) Closed    None   -         No    No    -     No      Yes     -
```

```
Switch-3# show media-redundancy status
                      Ring      Ring      I/C    I/C Ring  Port1 Port2 I/C   Port1   Port2   I/C
Inst Operational State Role      State     Role   State     SF    SF    SF    Blocked Blocked Blocked
---- ----------------- --------- --------- ------ --------- ----- ----- ----- ------- ------- -------
   1 Active            MRC (MRA) Undefined MIM-LC Closed    No    No    No    No      No      Yes
```

This command lists one MRP instance per line.

The `Ring Role` shows the current *operational* role. If the operational role is the same as the *configured* role, nothing else is shown. Otherwise, the configured role is shown in parentheses, as in this example.

The `Ring State` shows whether the ring is closed or open. Only the MRM knows the ring state, so MRCs display it as `Undefined`.

The `I/C Role` shows the configured interconnection role. If the ring doesn't participate in the interconnection topology, it shows `None` and the `I/C Ring State` and `I/C Blocked` fields are dashes (`-`). Switch-3 does indeed participate in the I/C topology, so it shows its current role (here `MIM`) and whether is is running in link-check (`LC`) or ring-check (`RC`) mode.

The next three columns shows - with `Yes` or `No` - whether any of the ports have signal fail.

The three last columns shows - also with a `Yes` or `No` - whether the MRP instance has blocked its ports.

Let's modify the MEP of Switch-3 a tiny little bit to provoke a CFM error and see what happens.

```
Switch-3# configure terminal
Switch-3(config)# cfm domain MyMrpIcDomain
Switch-3(config-cfm-dmn)# format string "MRP-Yellow-Interconnection"
Switch-3(config-cfm-dmn)# end
```

```
Switch-3# show media-redundancy status
                      Ring      Ring      I/C    I/C Ring  Port1 Port2 I/C   Port1   Port2   I/C
Inst Operational State Role      State     Role   State     SF    SF    SF    Blocked Blocked Blocked
---- ----------------- --------- --------- ------ --------- ----- ----- ----- ------- ------- -------
   1 Active            MRC (MRA) Undefined MIM-LC Open      No    No    Yes   No      No      Yes
```

The `I/C SF` column now shows `Yes` and to investigate why, have a look at the MEP status:

```
Switch-3# show cfm meps mep-id 1
Defect abbreviations (alarm level in parentheses):
R (1): someRDIdefect (RDI received from at least one remote MEP)
M (2): someMACstatusDefect (received Port Status TLV != psUp or Interface Status TLV != isUp)
C (3): someRMEPCCMdefect (valid CCM is not received within 3.5 times CCM interval from at least one
remote MEP)
E (4): errorCCMdefect (received CCM from an unknown remote MEP-ID or CCM interval mismatch)
X (5): xconCCMdefect (received CCM with an MD/MEG level smaller than configured or wrong MAID/MEGID
(cross-connect))

Domain          Service         MEP-ID Dfcts Operational State
--------------- --------------- ------ ----- ----------------------------------------------
MyMrpIcDomain   MyMrpIcService       1 --C-X Active
```

The MEP shows two defects: `someRMEPCCMdefect` and `xconfCCMdefect`. The latter a.o. means that the remote MEP's MAID is incorrect, which is the case when you change the format of it.

Fix the configuration error and notice that the CFM defects disappear.

6.1.1. Operational Warnings

At times one may forget to disable spanning tree on the ring ports or happen to configure VLANs incorrectly. Such misconfiguration mistakes are captured by the MRP implementation and shown as Operational Warnings.

Let's misconfigure Switch-1 and have a look at the status afterwards.

```
Switch-1# configure terminal
Switch-1(config)# interface GigabitEthernet 1/3
Switch-1(config-if)# spanning-tree
```

```
Switch-1# show media-redundancy status
                       Ring      Ring      I/C    I/C Ring  Port1 Port2 I/C   Port1   Port2   I/C
Inst Operational State Role      State     Role   State     SF    SF    SF    Blocked Blocked Blocked
---- ----------------- --------- --------- ------ --------- ----- ----- ----- ------- ------- -------
   1 Active (warnings) MRM (MRA) Closed    None   -         No    No    -     No      Yes     -
```

The `Operational State` column now shows `Active (warnings)` rather than just `Active`.

To figure out exactly which warnings, we need to see the detailed status:

```
Switch-1# show media-redundancy status details
Instance:                     1
Name:
Operational State:            Active
Operational Warnings:         Port1 has spanning tree enabled
Role (conf/oper):             MRA/MRM
Rec. Profile:                 500ms
MRM Priority:                 0xA000
UUID:                         FFFFFFFF-FFFF-FFFF-FFFF-FFFFFFFFFFFF
Ring State:                   Closed
VLAN:                         Untagged
Interconnection Role:         None
Interconnection Name:
Interconnection Rec. Profile: -
Interconnection State:        -
Interconnection VLAN:         -

           | Port1     | Port2     | Interconnection
-----------|-----------|-----------|----------------
Interface  | Gi 1/3    | Gi 1/4    | -
SF         | No        | No        | -
Blocked    | No        | Yes       | -
```

The detailed status also shows a bit of configuration, but otherwise much the same as the non-detailed.

In this example, the `Operational Warnings` row shows that spanning-tree isn't disabled on Port1 (which happens to be `Gi 1/3`).

More than one operational warning can be active at the same time, in which case there will be a line for each.

The following shows the operational warnings that can be raised by the MRP implementation:

- Port1 is not member of the ring's control VLAN, which is configured for tagged operation.
- Port2 is not member of the ring's control VLAN, which is configured for tagged operation.
- Port1 is not member of the interconnection control VLAN, which is configured for tagged operation.
- Port2 is not member of the interconnection control VLAN, which is configured for tagged operation.
- Interconnection port is not member of the interconnection control VLAN, which is configured for tagged operation.
- Port1 is not member of its own PVID (ring's control VLAN is configured for untagged operation).
- Port2 is not member of its own PVID (ring's control VLAN is configured for untagged operation).
- Port1 and Port2's PVID differ (ring VLAN is configured for untagged operation).
- Port1 is not member of the interconnection port's PVID (interconnection's control VLAN is configured for untagged operation).
- Port2 is not member of the interconnection port's PVID (interconnection's control VLAN is configured for untagged operation).

- Interconnection port is not member of its own PVID (interconnection’s control VLAN is configured for untagged operation).
- Port1 untags ring’s control VLAN, which is configured for tagged operation.
- Port2 untags ring’s control VLAN, which is configured for tagged operation.
- Port1 untags interconnection’s control VLAN, which is configured for tagged operation.
- Port2 untags interconnection’s control VLAN, which is configured for tagged operation.
- Interconnection port untags interconnection’s control VLAN, which is configured for tagged operation.
- Port1 tags its own PVID (ring’s control VLAN is configured for untagged operation).
- Port2 tags its own PVID (ring’s control VLAN is configured for untagged operation).
- Port1 tags the interconnection port’s PVID (interconnection’s control VLAN is configured for untagged operation).
- Port2 tags the interconnection port’s PVID (interconnection’s control VLAN is configured for untagged operation).
- Interconnection port tags itw own PVID (interconnection’s control VLAN is configured for untagged operation).
- Port1 MEP is not found. Using link-state for signal-fail instead.
- Port2 MEP is not found. Using link-state for signal-fail instead.
- Interconnection MEP is not found. Using link-state for signal-fail instead.
- Port1 MEP is administratively disabled. Using link-state for signal-fail instead.
- Port2 MEP is administratively disabled. Using link-state for signal-fail instead.
- Interconnection MEP is administratively disabled. Using link-state for signal-fail instead.
- Port1 MEP is not a Down-MEP. Using link-state for signal-fail instead.
- Port2 MEP is not a Down-MEP. Using link-state for signal-fail instead.
- Interconnection MEP is not a Down-MEP. Using link-state for signal-fail instead.
- Port1 MEP’s residence port is not that of Port1. Using link-state for signal-fail instead.
- Port2 MEP’s residence port is not that of Port2. Using link-state for signal-fail instead.
- Interconnection MEP’s residence port is not that of the interconnection port. Using link-state for signal-fail instead.
- Port1 has spanning tree enabled.
- Port2 has spanning tree enabled.
- Interconnection port has spanning tree enabled.
- Multiple MRMs detected on the ring. This is normal if MRAs are negotiating. Cleared after 10 seconds w/o detection.

- Multiple MIMs with same ID detected on the interconnection ring. Cleared after 10 seconds w/o detection.
- An internal error has occurred. A code update is required. Please check console for details.

A special note about the "Multiple MRMs" and "Multiple MIMs" operational warnings:

Multiple MRMs is detected if an MRM receives MRP_Test PDUs not transmitted by itself. The consequence of multiple MRMs on the ring is discussed in the note in Section 2.3.

Multiple MIMs is detected if a MIM in RC mode receives MRP_InTest PDUs or a MIM in LC mode receives MRP_InTopologyChange or MRP_InLinkStatusPoll PDUs with its own IID but not transmitted by itself. The consequence of multiple MIMs in the interconnection topology is similar to that of multiple MRMs.

These two operational warnings can also result in an SNMP trap or JSON notification.

6.2. Statistics

Continuing Example 2, let’s look at a few switches' statistics.

```
Switch-1# show media-redundancy statistics
             Port1                            Port2                            Interconnection
Inst Flushes Rx PDU      Tx PDU      SF Count Rx PDU      Tx PDU      SF Count Rx PDU      Tx PDU
SF Count
---- ------- ----------- ----------- -------- ----------- ----------- -------- ----------- -----------
--------
   1       4      276832          17        0      276553          17        0           -
-        -
```

```
Switch-3# show media-redundancy statistics
             Port1                            Port2                            Interconnection
Inst Flushes Rx PDU      Tx PDU      SF Count Rx PDU      Tx PDU      SF Count Rx PDU      Tx PDU
SF Count
---- ------- ----------- ----------- -------- ----------- ----------- -------- ----------- -----------
--------
   1      10      179926          25        0      180041          25        0           1
1        1
```

This command lists one MRP instance per line.

The `Flushes` column shows how many FDB flushes have been performed since the last time this instance was (re-)started or statistics were cleared.

The `Rx PDU` columns (one for each port) shows the sum of all MRP PDUs received by the CPU (software) on that port.

> **NOTE**
> In some configurations, the MRP PDUs are hardware-forwarded and will therefore not be sent to the CPU for handling and counting, so the numbers are not necessarily the number of MRP PDUs going through the given port.

The `Tx PDU` columns (one for each port) shows the sum of all MRP PDUs transmitted *by software* on the given port.

> **NOTE**
>
> If hardware is used for transmission of MRP_Test or MRP_InTest PDUs or if hardware forwards MRP PDUs from one port to another, the Tx counters will not count them. This is the case for Switch-1, because we know that this switch is the MRM in the ring, so it is supposed to transmit MRP_Test PDUs at regular intervals, but it has only transmitted a few MRP PDUs.

The `SF Count` columns (also one for each port) shows how many times that port has had signal fails. If the MRP instance is administratively enabled while a ring- or I/C-port has SF, it starts at 1. Otherwise it starts at 0.

Let’s have a look at the detailed statistics for Switch-3.

```
Switch-3# show media-redundancy statistics details
Instance:                                 1
Flushes:                                  10
MRM/MRC transitions:                      1
Ring Open/Closed transitions:             0
I/C  Open/Closed transitions:             4
MRP_Test   round-trip time (min/last/max): N/A
MRP_InTest round-trip time (min/last/max): N/A

Counter          Port1 Rx     Port2 Rx     I/C Rx       Port1 Tx     Port2 Tx     I/C Tx
---------------- ------------ ------------ ------------ ------------ ------------ ------------
Test                   179519       179520            0            2            2            0
TopologyChange              7            7            0            0            0            0
LinkDown                    0            0            0            0            0            0
LinkUp                      0            0            0            0            0            0
TestMgrNAck                 4            4            0            0            0            0
TestPropagate               0            0            0            1            1            0
Option                      0            0            0            0            0            0
InTest                      0            0            0            0            0            0
InTopologyChange            0            4           12            4            4            4
InLinkDown                  0            2            1            0            0            0
InLinkUp                    0          108          100            0            0            0
InLinkStatusPoll            0            0           32           18           18            0
Unknown                     0            0            0
Errors                      0            0            0
Unhandled                   0            0            0
Own                         0            4           44
Others                 179530       179641          101
Signal Fails                0            0            1
```

The `Flushes` row is similar to that from the non-detailed statistics.

The `MRM/MRC transitions` row shows the number of times that this node has gone from being an MRM to becoming an MRC or vice versa. Since Switch-3 is an MRA, it always starts out operating as an MRM. in the "Test" row you can see that it got to transmit two MRP_Test PDUs before it received an MRP_TestMgrNAck PDU, causing it to transition to becoming an MRC.

The `Ring Open/Closed transitions` row shows the number of times the ring has gone from being open to being closed and vice versa. It will always be 0 on nodes operating as MRCs.

The `I/C Open/Closed transitions` row shows the number of times the interconnection topology has gone from being an open to a closed topology. It only counts on the MIM. If neither MIC nor MIM, it will contain a dash (`-`).

The `MRP_Test round-trip time (min/last/max)` row shows the number of milliseconds an MRP_Test PDU transmitted by this node spent in the ring until it came back. The minimum and maximum times are shown along with the time it took for the last MRP_Test PDU to travel the ring.
This field shows `N/A` if the node is currently not an MRM or if using hardware to transmit MRP_Test PDUs. For Switch-3 it is the former that rules, as we shall see in a second.

The `MRP_InTest round-trip time (min/last/max)` row shows the number of milliseconds an MRP_InTest PDU transmitted by this node spent in the interconnection ring until it came back. The minimum and maximum times are shown along with the time it took for the last MRP_InTest PDU to travel the ring.
This field shows `N/A`if the node is currently not a MIM or if using hardware to transmit MRP_InTest PDUs or it is not in ring-check mode. The latter is the case for Switch-3.

The `Counter` table’s first 12 rows list Rx and Tx counters per port and port MRP PDU type.

One can deduct that this is an MRA operating as an MRC, because if it was configured as an MRC, it would not count MRP_Test PDUs, because MRCs are not interested in these PDU types, whereas MRAs are, despite their operating role, in order to take over a possibly failing MRM.

On switches that has hardware support for transmitting MRP_Test and MRP_InTest PDUs, the Tx counters will contain `N/A`, because the actual number of transmitted PDUs cannot be obtained. Switch-3 does not have hardware support for this, since Port1 Tx and Port2 Tx counters show a number.

NOTE	Switches with hardware support for transmitting MRP_Test and MRP_InTest PDUs will only update the PDUs' MRP_Timestamp and MRP_SequenceID fields when one or more of the remaining fields in the PDUs are changed.

The `Unknown` row counts the number of received MRP PDUs that wasn’t recognised as a known MRP_Xxx PDU type.

The `Errors` row counts the number of received erroneous MRP PDUs. An errouneous MRP PDU could e.g. be if the frame is too short to carry all the required fields of a given MRP PDU type.

The `Unhandled` row counts the number of MRP PDUs received by, but not processed by, the node for one or the other reason.

The `Own` row counts the number of received MRP PDUs that were transmitted by the node itself.

The `Others` row counts the number of received MRP PDUs that were NOT originally transmitted by the node itself.

Before we leave this topic, let's see the round-trip time in action. Switch-3 utilizes software to transmit PDUs and it's a MIM but it's not in RC mode. Let's put it into RC mode and see what happens:

```
Switch-3# configure terminal
Switch-3(config)# media-redundancy 1
Switch-3(config-media-redundancy)# interconnection mode ring-check
Switch-3(config-media-redundancy)# end
Switch-3#
```

Show detailed statistics

```
Switch-3# show media-redundancy statistics details
Instance:                                  1
Flushes:                                   0
MRM/MRC transitions:                       1
Ring Open/Closed transitions:              0
I/C  Open/Closed transitions:              2
MRP_Test   round-trip time (min/last/max): N/A
MRP_InTest round-trip time (min/last/max): 2/2/6 msecs (last received 0 seconds ago)
...
```

6.2.1. Clearing Statistics

Statistics of one or all MRP instances can be cleared with the following command:

```
# clear media-redundancy [<inst_list>] statistics
```

Notice that this also may change the MRP PDUs transmitted by the node, because the number of ring and I/C open/closed transitions is also cleared and these numbers are part of some of the MRP PDU types.

7. Debugging MRP

This final section is meant for experts that know the protocol inside out, only.

The IStaX-software contains a number of debug commands that can assist in troubleshooting MRP.

NOTE	All debug commands are subject to change without notice and the use of these are at the user's own risk.

```
# platform debug allow

WARNING: The use of 'debug' commands may negatively impact system behavior.
Do not enable unless instructed to. (Use 'platform debug deny' to disable
debug commands.)

NOTE: 'debug' command syntax, semantics and behavior are subject to change
without notice.

# debug show media-redundancy ?
    capabilities    Show media-redundancy capabilities
    history         Show media-redundancy state change history
    rules           Show current media-redundancy-installed rules
    state           Show current media-redundancy state
    timers          Show current media-redundancy timers
```

Of these, `history` is probably the most valuable. Let's see it in action.

```
Switch-3# debug show media-redundancy history
Now = 19848535 ms
Inst  # Time [ms] Role SM State Ring State InRole InSM State InRing State Prm   Port1  Port2  I/C
Changed by
---- -- --------  ---- -------- ---------- ------ ---------- ----------- ----- ------ ------ ------
-------------------
   1  1 19078999  MRA  Power On Disabled   MIM    Power On   Disabled     Port1 Dn/Fwd Dn/Fwd Dn/Fwd
Init
   1  2 19079006  MRM  AC_STAT1 Open       MIM    Power On   Disabled     Port1 Dn/Blk Dn/Blk Dn/Fwd
MRA PowerOn
   1  3 19079006  MRM  AC_STAT1 Open       MIM    AC_STAT1   Open         Port1 Dn/Blk Dn/Blk Dn/Blk
MIM-RC PowerOn
   1  4 19079011  MRM  PRM_UP   Open       MIM    AC_STAT1   Open         Port1 Up/Fwd Dn/Blk Dn/Blk
MAUTypeChangeInd(P1-Up)
   1  5 19079012  MRM  CHK_RC   Closed     MIM    AC_STAT1   Open         Port1 Up/Fwd Up/Blk Dn/Blk
MAUTypeChangeInd(P2-Up)
   1  6 19079014  MRM  CHK_RC   Closed     MIM    CHK_IC     Closed       Port1 Up/Fwd Up/Blk Up/Blk
MAUTypeChangeInd(In-Up)
   1  7 19079016  MRC  PT_IDLE  Undefined  MIM    CHK_IC     Closed       Port1 Up/Fwd Up/Fwd Up/Blk
TestMgrNAckInd
```

A state change in either the ring state machine or the I/C state machine creates a new row in this list. Only the last 50 rows are saved.

The names and the states are directly from the standard and won't be described further.

The primary purpose of showing this is that if a user cannot figure out why a given node is in a given state, he may consult this history for hints.