This chapter describes how to configure remote network monitoring (RMON) on the ML1000-2, ML100T-12, ML100X-8, and ML-MR-10 cards for the ONS 15454 SONET/SDH.

RMON is a standard monitoring specification that defines a set of statistics and functions that can be exchanged between RMON-compliant console systems and network probes. RMON provides you with comprehensive network-fault diagnosis, planning, and performance-tuning information. The ML-Series card features RMON and is designed to work with a network management system (NMS).

**Note:** For complete syntax and usage information for the commands used in this chapter, see the "System Management Commands" section in the *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2*.

**Note:** For general information about using Cisco IOS to manage RMON, refer to the "Configuring RMON Support" chapter of the *Cisco IOS Configuration Fundamentals Configuration Guide*.

**Note:** The ML-MR-10 card does not support cyclic redundancy check (CRC) threshold monitoring or Cisco proprietary resilient packet ring (RPR).

This chapter consists of these sections:



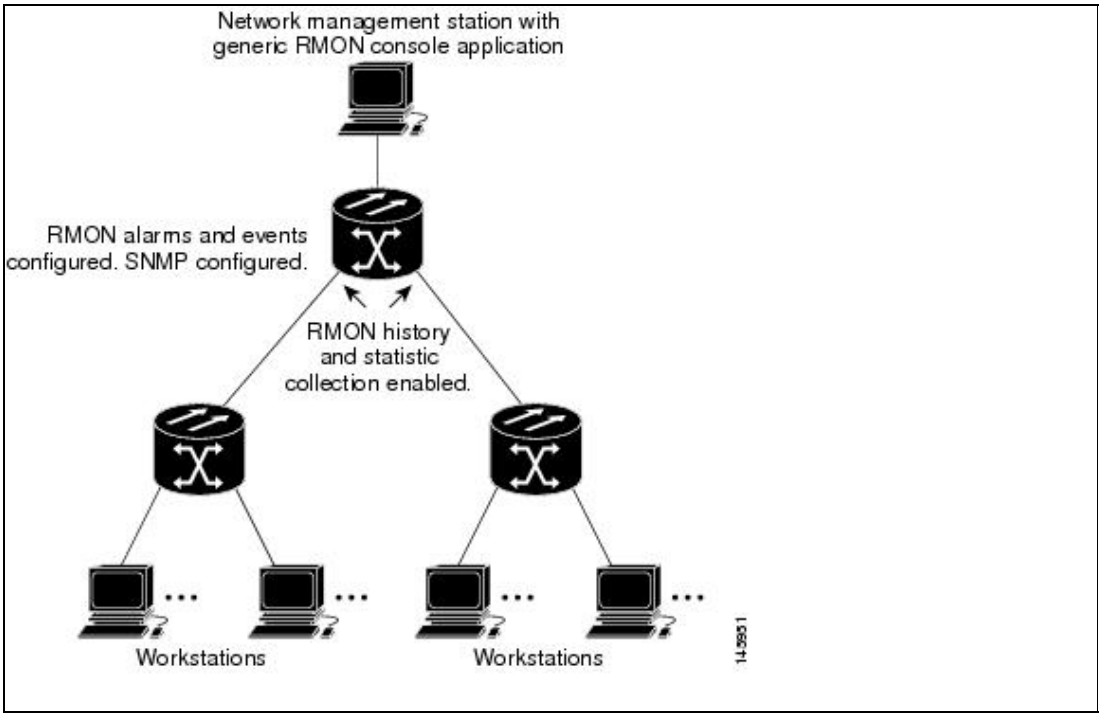
# Contents

# Understanding RMON

RMON is an Internet Engineering Task Force (IETF) standard monitoring specification that allows various network agents and console systems to exchange network monitoring data. You can use the RMON feature with the Simple Network Management Protocol (SNMP) agent to monitor all the traffic flowing among ML-Series card and other switches on all connected LAN segments. Figure 23-1 illustrates RMON.

For information on the MIBs supported by the ML-Series card, see the Supported MIBs.

**Figure 23-1: Remote Monitoring Example**




# Configuring RMON

These sections describe how to configure RMON on your ML-Series card:

- Default RMON Configuration
- Configuring RMON Alarms and Events (required)
- Collecting Group History Statistics on an Interface (optional)
- Collecting Group Ethernet Statistics on an Interface (optional)

## Default RMON Configuration

RMON is disabled by default; no alarms or events are configured.

## Configuring RMON Alarms and Events

You can configure your ML-Series card for RMON by using the command-line interface (CLI) or an SNMP-compatible NMS. For the ML-MR-10 card, RMON can be configured using the Cisco Transport Controller (CTC) interface, as well. We recommend that you use a generic RMON console application on the NMS to take advantage of RMON network management capabilities. You must also configure SNMP on the ML-Series card to access RMON MIB objects. For more information about configuring SNMP, see Configuring SNMP. For information on configuring RMON using CTC, refer to the Cisco ONS 15454 Procedure Guide." or the "Cisco ONS 15454 SDH Procedure Guide."

Beginning in privileged EXEC mode, follow these steps to enable RMON alarms and events. This procedure is required.

| Step | Command | Purpose |
|---|---|---|
| 1 | `Router# configure terminal` | Enter global configuration mode. |
| 2 | `Router (config)# rmon event number [description string] [log] [owner string] [trap community]` | Add an event in the RMON event table that is associated with an RMON event number.<br>• For *number*, assign an event number. The range is 1 to 65535.<br>• (Optional) For **description** *string*, specify a description of the event.<br>• (Optional) Use the **log** keyword to generate an RMON log entry when the event is triggered.<br>• (Optional) For **owner** *string*, specify the owner of this event.<br>• (Optional) For **trap** *community*, enter the SNMP community string used for this trap. |
| 3 | `Router (config)#rmon alarm number variable interval {absolute \| delta} rising-threshold value [event-number] falling-threshold value [event-number] [owner string]` | Set an alarm on a MIB object.<br>• For *number*, specify the alarm number. The range is 1 to 65535.<br>• For *variable*, specify the MIB object to monitor.<br>• For *interval*, specify the time in seconds that the alarm monitors the MIB variable. The range is 1 to 2147483647 seconds.<br>• Specify the **absolute** keyword to test each MIB variable directly. Specify the **delta** keyword to test the change between samples of a MIB variable.<br>• For *value*, specify a number at which the alarm is triggered and a |

| | | |
|---|---|---|
| | | number at which the alarm is reset. The range for the rising threshold and falling threshold values is -2147483648 to 2147483647.<br>• (Optional) For *event-number*, specify the event number to trigger when the rising or falling threshold exceeds its limit.<br>• (Optional) For **owner** *string*, specify the owner of the alarm. |
| 4 | `Router (config)# end` | Return to privileged EXEC mode. |

To disable an alarm, use the **no rmon alarm** *number* global configuration command on each alarm you configured. You cannot disable all the alarms that you configured by not specifying a specific number. You must disable each alarm separately. To disable an event, use the **no rmon event** *number* global configuration command. To learn more about alarms and events and how they interact with each other, see RFC 1757.

You can set an alarm on any MIB object. The following example configures RMON alarm number 10 by using the **rmon alarm** command.

```
Router(config)# rmon alarm 10 ifInErrors.65539 20 delta rising 15 1 fall
0
```

The alarm monitors the MIB variable *ifEntry.20.1* once every 20 seconds to check the change in the variable's rise or fall until the alarm is disabled. If the *ifEntry.20.1* value shows a MIB counter increase of 15 or more, such as from 100000 to 100015, the alarm is triggered. The alarm in turn triggers event number 1, which is configured with the **rmon event** command. Possible events can include a log entry or an SNMP trap. If the *ifEntry.20.1* value changes by 0, the alarm is reset and can be triggered again.

**Note:** The example does not trigger an optional event when the falling-threshold is 0.

Where 65539 is the SNMP IfIndex for interface POS 0. You can get the SNMP ifIndex for a given port with an SNMP get. In the example output, you can see that the SNMP ifIndex for POS0 is 65539:

```
tuvoks-view:128> getmany -v2c 10.92.56.97 tcc@1 ifDescr

ifDescr.65536 = GigabitEthernet0

ifDescr.65537 = GigabitEthernet1

ifDescr.65538 = Null0

ifDescr.65539 = POS0

ifDescr.65540 = POS1

ifDescr.65541 = SPR1

tuvoks-view:129>
```

The following example creates RMON event number 1 by using the **rmon event** command. The event is defined as *High ifOutErrors* and generates a log entry when the event is triggered by the alarm. The user *jjones* owns the row that is created in the event table by this command. This example also generates an

SNMP trap when the event is triggered.

```
Router(config)# rmon event 1 log trap eventtrap description "High
ifOutErrors" owner jjones
```

## Collecting Group History Statistics on an Interface

You must first configure RMON alarms and events to display collection information.

Beginning in privileged EXEC mode, follow these steps to collect group history statistics on an interface. This procedure is optional.

| Step | Command | Purpose |
|---|---|---|
| 1 | `Router# configure terminal` | Enter global configuration mode. |
| 2 | `Router (config)# interface interface-id` | Specify the interface on which to collect history, and enter interface configuration mode.<br><br>**Note:** Group history statistics do not work on packet-over-SONET/SDH (POS) interfaces, only on Ethernet interfaces. |
| 3 | `Router (config)# rmon collection history` *index* [**buckets** *bucket-number*] [**interval** *seconds*] [**owner** *ownername*] | Enable history collection for the specified number of buckets and time period.<br><br>• For *index*, identify the RMON group of statistics. The range is 1 to 65535.<br>• (Optional) For **buckets** *bucket-number,* specify the maximum number of buckets desired for the RMON collection history group of statistics. The range is 1 to 65535. The default is 50 buckets.<br>• (Optional) For **interval** *seconds*, specify the number of seconds in each polling cycle. The range is 1 to 3600. The default is 1800 seconds.<br>• (Optional) For **owner** *ownername*, enter the name of the owner of the RMON group of statistics. |
| 4 | `Router (config)# end` | Return to privileged EXEC mode. |
| 5 | `Router# show rmon history` | Display the contents of the ML-Series card history table. |

To disable history collection, use the **no rmon collection history** *index* interface configuration command.

This example shows how to collect and show RMON history for the owner *root*:

```
Router(config)# interface gigabitethernet1

Router(config-if)# rmon collection history 2 owner root

Router(config-if)# end

Router# show rmon history

Entry 2 is active, and owned by root
```

```
Monitors ifIndex.393217 every 1800 second(s)

Requested # of time intervals, ie buckets, is 50,
```

### Collecting Group Ethernet Statistics on an Interface

Beginning in privileged EXEC mode, follow these steps to collect group Ethernet statistics on an interface. This procedure is optional.

| Step | Command | Purpose |
|---|---|---|
| 1 | `Router# configure terminal` | Enter global configuration mode. |
| 2 | **interface** *interface-id* | Specify the interface on which to collect statistics, and enter interface configuration mode. |
| 3 | `Router (config-if)# rmon collection stats` *index* [**owner** *ownername*] | Enable RMON statistic collection on the interface.<br>• For *index*, specify the RMON group of statistics. The range is from 1 to 65535.<br>• (Optional) For **owner** *ownername*, enter the name of the owner of the RMON group of statistics. |
| 4 | `Router (config)# end` | Return to privileged EXEC mode. |
| 5 | `Router# show rmon statistics` | Display the contents of the ML-Series card statistics table. |

To disable the collection of group Ethernet statistics, use the **no rmon collection stats** *index* interface configuration command.

This example shows how to collect RMON statistics for the owner *root*:

```
Router(config)# interface gigabitethernet1

Router(config-if)# rmon collection stats 2 owner root
```

## Understanding ML-Series Card CRC Error Threshold

**Note:** This section does not apply to the ML-MR-10 card.

The POS ports on the ML-Series card report alarms for SONET/SDH defects and generic framing proedure (GFP) defects, including signal fail (SF) and signal degrade (SD) alarms. In most circumstances, these alarms alert the user to problems that also cause excessive CRC errors on the POS port. However, there are situations where excessive CRC errors will occur on the POS port, but the link will not have any SONET defects or GFP defects to report. Examples of this situation include an ML-Series card at the other end of the link sending out packets with CRC errors or a bit error rate too low to trigger SF or SD defect, but high enough to cause a meaningful CRC packet error rate.

In these situations with a default ML-Series card Cisco proprietary RPR implementation and no reported SONET/SDH or GFP defects, the POS interface remains in the up state as a member of the shared packet ring (SPR) interface. Traffic is lost quietly and does not trigger any alarms or action.

The frame check sequence (FCS) threshold configuration and detection feature fixes this problem. The user can now configure the ML-Series card to raise an alarm if the percentage of packet loss due to CRC errors

crosses a configureable threshold. The alarm raised is the CRC Threshold Crossing Alarm (CRC-ALARM), which is a service-affecting (SA) SONET/SDH alarm with a Major (MA) severity. Reported SONET/SDH alarms can be viewed under the Alarms tab of CTC.

The user can also configure the CRC-ALARM to trigger a link state down on the port and to wrap a Cisco proprietary RPR. By default, the CRC-ALARM is disabled. When the alarm is configured, the link down and wrap actions are still disabled by default. This feature is also supported on the ML-Series card Ethernet ports.

## Threshold and Triggered Actions

**Note:** This section does not apply to the ML-MR-10 card.

The configureable threshold is not set with a bit error rate (BER), since variable frame lengths and varying percentages of bandwidth can impair the usefulness of this measure. Instead, the users configure a more relevant measure using CRC error rate as a percentage of the traffic. The available triggering thresholds are:

- 10e-2 or 1 percent traffic (1 CRC error in 100 packets)
- 10e-3 or 0.1 percent traffic (1 CRC error in 1000 packets) (default)
- 10e-4 or 0.01 percent traffic (1 CRC error in 10000 packets)

The default threshold is a CRC error rate of 0.1 percent of the traffic. For voice and video traffic, an error rate of 1 percent is typically a critical issue and 0.1 percent is a major issue. Voice and video needs to trigger a wrap if the error rate is higher than 0.1 percent (1 error every 1000 packets). For normal data traffic, an error rate of 10 percent traffic is a critical issue, requiring an immediate fix, and 1 percent traffic is a minor issue.

The following actions occur after the detection of excessive CRC errors:

1. The Cisco proprietary RPR wraps if this option is configured.
2. The link shuts down if this option is configured.
3. If the link shuts down, a path defect indication (PDI) is sent to the far-end ML-Series card port. This ensures that the remote end wraps.
4. A CRC-ALARM is raised against the local end ML-Series card port. (If the remote end is also receiving excessive CRC errors, a CRC-ALARM is also raised against the far end ML-Series card port.

## SONET/GFP Suppression of CRC-ALARM

**Note:** This section does not apply to the ML-MR-10 card.

This detection of excessive CRC errors is independent of SONET/GFP defects. A problem may have the potential to trigger both the SONET/GFP defects and the CRC-ALARM. In this scenario, the SONET/GFP defect will trigger before the CRC-WRAP alarm because CRC error threshold detection is a slower process. If the SONET/GFP defect causes the link to go down, this link-down happens before the CRC-ALARM is detected, and it suppresses the CRC-ALARM. If the SONET/GFP defect that causes CRC-ALARM is not a link-down trigger and the CRC-ALARM is configured to take the link down, the CRC-ALARM will report and trigger the link down.

## Clearing of CRC-ALARM

**Note:** This section does not apply to the ML-MR-10 card.

When the trigger action is disabled (default), the CRC-ALARM automatically clears when the error rate falls below the threshold for a significant time period.

When the trigger action is enabled, a CRC-ALARM requires a manual clear from the user. This is required because the wrap or link down caused by the alarm blocks both traffic and the CRC errors in the traffic from the port. So with no CRC errors, an automatic clear would occur even though the underlying problem, such as dirty fiber or a defective ML-Series card, still exists. Interface flapping can occur in this situation.

Before doing a manual clear, the user needs to determine the root cause of a CRC-ALARM and correct it. After that, the user has several alternative methods to manually clear the alarm:

- Through the Cisco IOS CLI, enter the clear crc alarm interface interface-type interface-number command at the EXEC level.
- Through the Cisco IOS CLI, do an administrative shutdown on the linked ports and then a no shutdown to enable the ports.
- Through CTC or Transaction Language One (TL-1), disable and then re-enable the circuit.
- Through CTC or TL-1, delete the SONET/SDH circuit and create a replacement circuit with the same source and destination.

## Unwrap Synchronization

The software on the ML-Series card raises the CRC-ALARM alarm on the POS interface that sees the errored frames. For unidirectional FCS errors, the user only needs to issue the unwrap command on the POS port at one end of the span, the one which raised the CRC-ALARM alarm. For bidirectional failures, both ends of the span raise the CRC-ALARM alarm and the user is required to issue the command once at each end of the span.

Since the POS ports at each end of the link are wrapped, removing the wrap (unwrapping) when the CRC-ALARM is cleared requires coordination. The software must also make sure that other errors that might cause wrapping are absent. The following examples illustrate this process for both unidirectional and bidirectional failures. For simplicity, the examples assume that excessive CRC errors is the only existing condition that might cause wrapping.

### Unidirectional Errors

Figure 23-2 shows a Cisco proprietary RPR wrapped by excessive unidirectional CRC errors on POS Port 0 of Node E, which is also reporting the CRC-ALARM. This caused POS Port 1 on Node E and POS Port 0 on Node D to wrap. The figure captions further explain the process.

**Figure 23-2: Wrapped Cisco Proprietary RPR with Unidirectional Excessive CRC Errors**

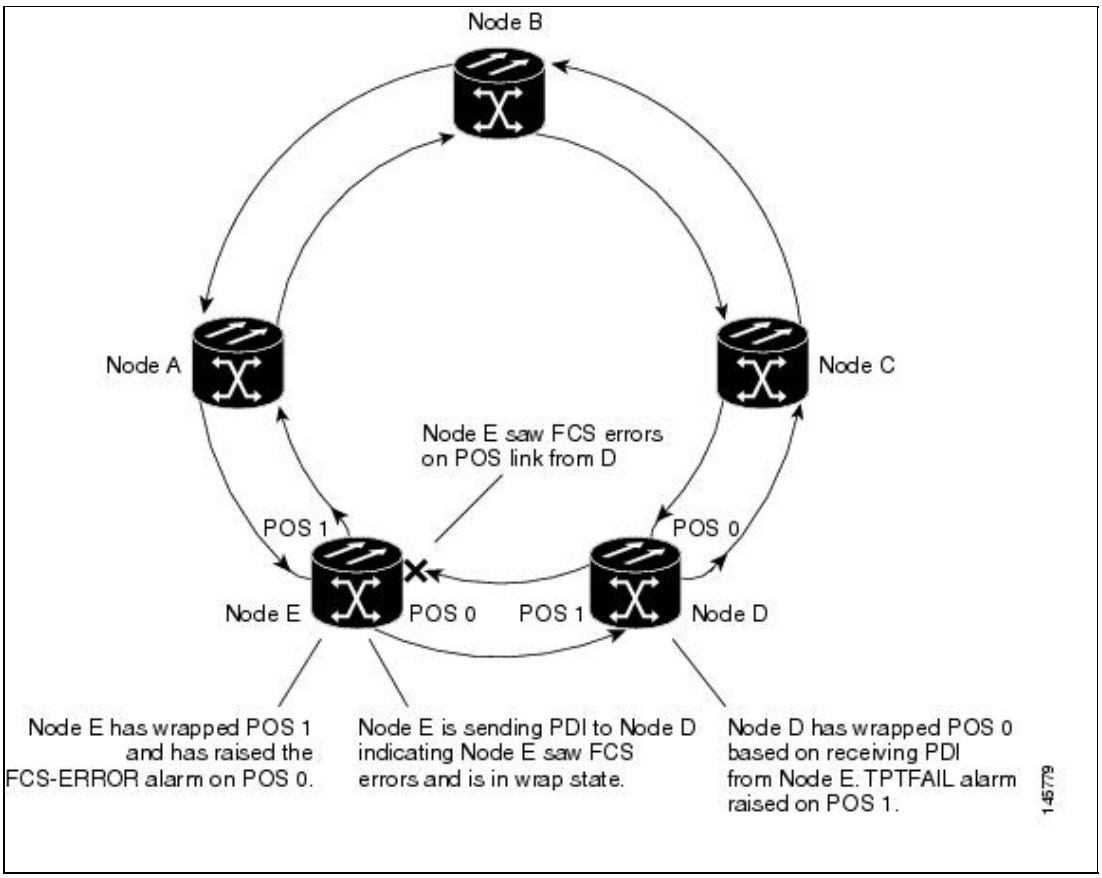



Figure 23-3 illustrates the unwrap sequence for Figure 23-2. The traffic hit for the unwrap is dependent on the soak time required to declare PDI cleared on Node D.

**Figure 23-3: Unwrapped Cisco Proprietary RPR with Unidirectional Excessive CRC Errors**

Output/145780.jpg

### Bidirectional Errors

Figure 23-4 shows a Cisco proprietary RPR wrapped by excessive bidirectional CRC errors. Both ports are reporting CRC-ALARMs. The figure captions further explain the process.

**Figure 23-4: Wrapped Cisco Proprietary RPR with Bidirectional Excessive CRC Errors**

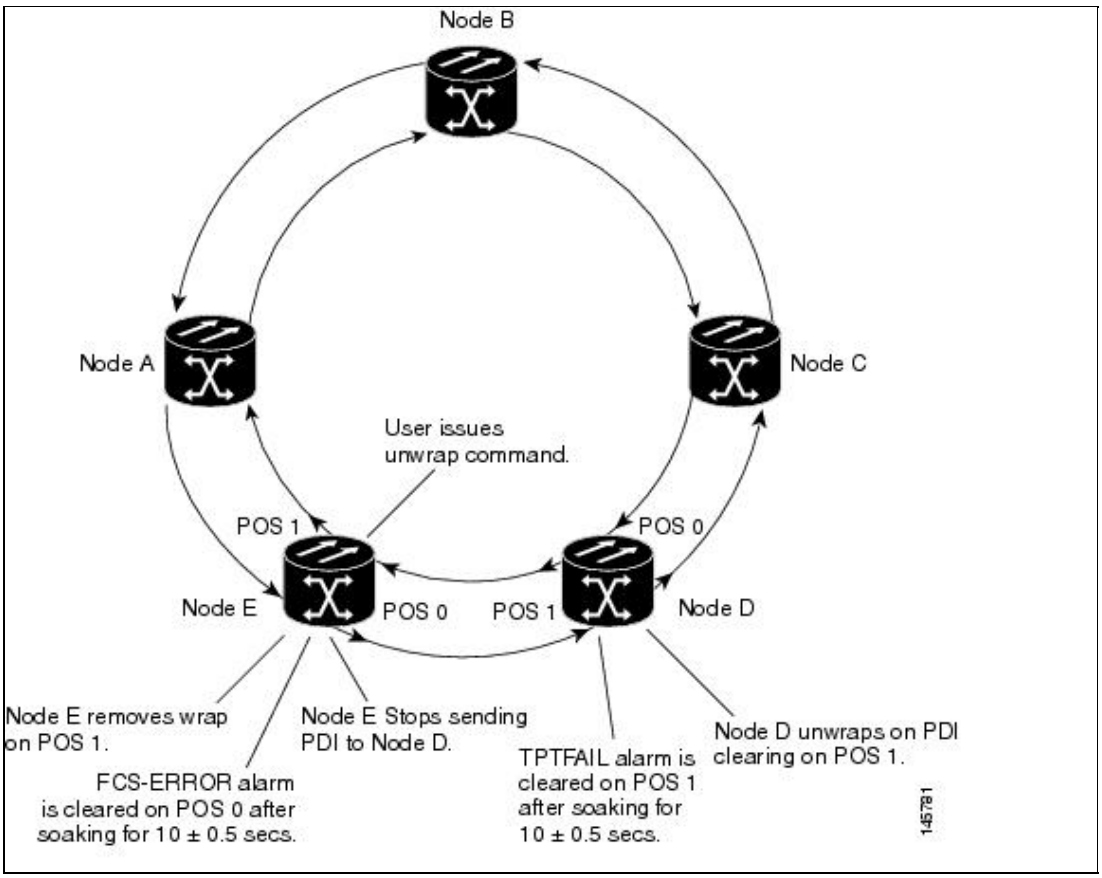
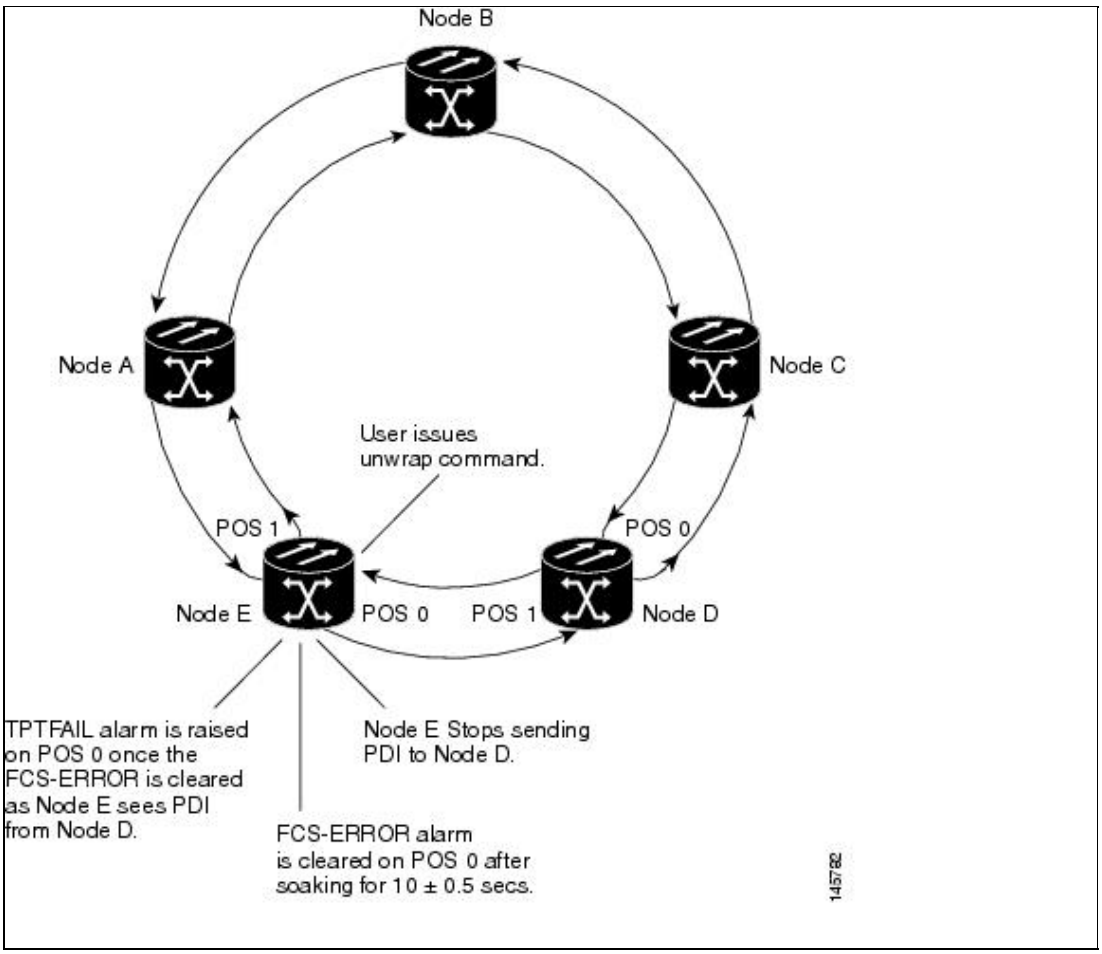


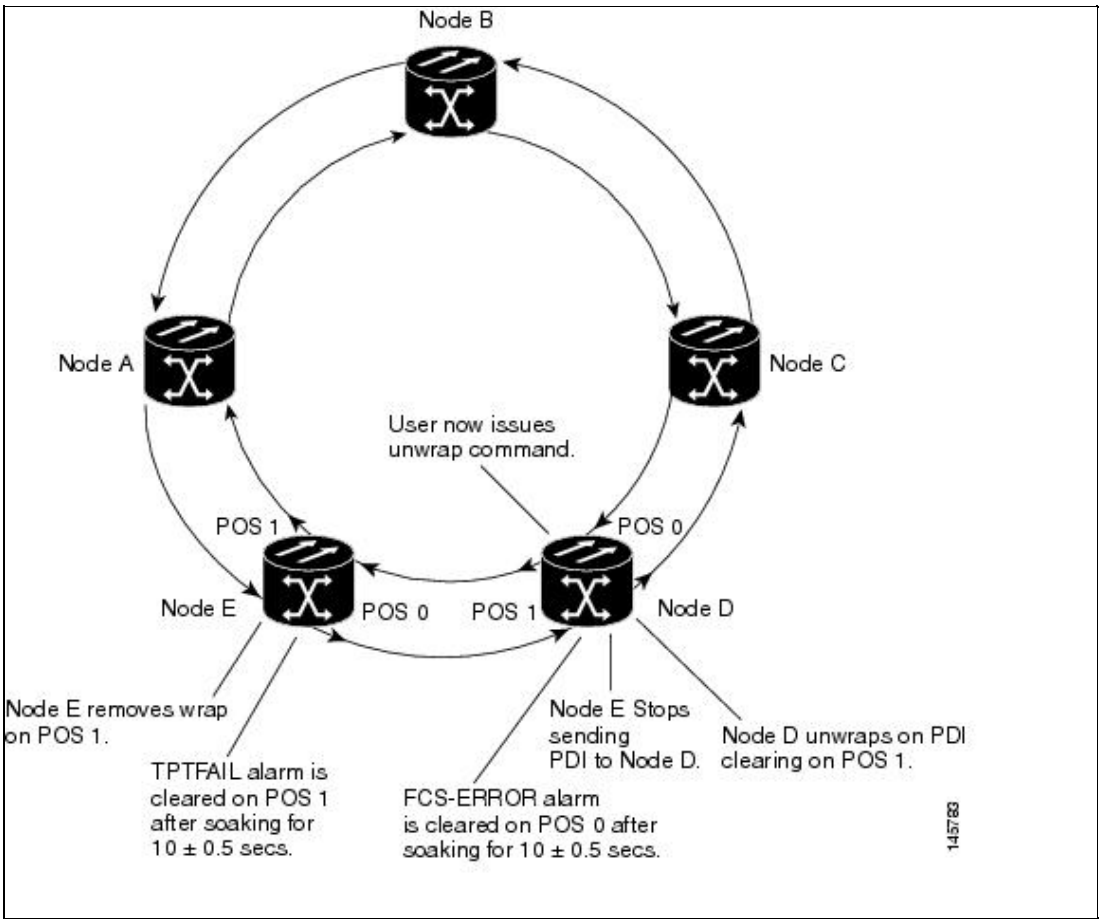

Figure 23-5 illustrates the first part of the unwrap sequence for Figure 23-4. This occurs after the unwrap command is configured on Node E. For unwrap in this bidirectional scenario, the user must configure the command on the POS ports at both ends of the link.

**Figure 23-5: First Stage of Unwrapped Cisco Proprietary RPR with Bidirectional Excessive CRC Errors**




Node E has not unwrapped POS port 1 after the first CRC-ALARM clear command. Since Node D continues to send PDI to Node E, Node E will raise the TPTFAIL alarm once the CRC-ALARM is cleared. At this point, the Cisco proprietary RPR is in a state similar to the unidirectional failure. The unwrap completes after the user issues the second unwrap command, as illustrated by Figure 23-6.

**Figure 23-6: Second Stage of Unwrapped Cisco Proprietary RPR with Bidirectional Excessive CRC Errors**




# Configuring the ML-Series Card CRC Error Threshold

**Note:** This section does not apply to the ML-MR-10 card.

Beginning in privileged EXEC mode, follow these steps to configure the ML-Series card CRC error threshold:

| Step | Command | Purpose |
|---|---|---|
| 1 | `Router# configure terminal` | Enter global configuration mode. |
| 2 | `Router (config)# interface interface-type interface-number` | Enters interface configuration mode. |
| 3 | `Router (config-if)# [no] trigger crc threshold [threshold-value]` | Sets an FCS error level as a percentage of bandwidth to trip the SONET/SDH CRC-ALARM. Values for the threshold-value are:<br>• **2-**10e-2 or 1 percent traffic (1 CRC error in100 packets)<br>• **3-**10e-3 or 0.1 percent traffic (1 CRC error in 1000 packets) (default)<br>• **4-**10e-4 or 0.01 percent traffic (1 CRC error in 10000 packets)<br><br>The no form of the command sets the level back to the default threshold of 3. |

Figure 23-6: Second Stage of Unwrapped Cisco Proprietary RPR with Bidirectional Excessive CRC Errors

| | | |
|---|---|---|
| 4 | `Router (config-if)# **[no] trigger crc action**` | (Optional) Sets the CRC-ALARM to trigger a link down for the reporting port. Set on an Cisco proprietary RPR POS port, this also wraps the Cisco proprietary RPR.<br><br>The no form of the command sets the trigger back to the default off. |
| 5 | `Router(config-if)# **[no] trigger crc delay** soak-time` | (Optional) Sets the minutes of soak time for excessive CRC error detection. The value for soak-time is from 3 minutes to 10 minutes.<br><br>The no form of the command sets the delay back to the default of one minute. |
| 6 | `Router (config)# **end**` | Return to privileged EXEC mode. |

### Clearing the CRC-ALARM Wrap with the Clear CRC Error Command

The Cisco IOS CLI clear crc alarm interface interface-type interface-number command is intended to clear the Cisco proprietary RPR wrap when it occurs due to FCS errors without corresponding SONET/SDH errors. It is not intended to unwrap wraps due to other causes, such as SONET/SDH defects or keep alive (KA) failures. If SONET/SDH or KA defects are present without FCS errors, the software rejects the command with an error message. When FCS errors are present and SONET/SDH or KA defects are present, the command is accepted by the software but the node unwraps only after all the failures have been fixed. In this case, the user does not need to reissue the command after the SONET/SDH or KA defect has cleared.

**Note:** The unwrap does not occur immediately, but after conditions are met.

Beginning in privileged EXEC mode, follow these steps to clear the ML-Series card CRC-ALARM:

| Step | Command | Purpose |
|---|---|---|
| 1 | Router # clear crc alarm interface interface-type interface-number | Clears the SONET/SDH CRC-ALARM and allows the Cisco proprietary RPR to unwrap when conditions are met. |

## Configuring ML-Series Card RMON for CRC Errors

**Note:** This section does not apply to the ML-MR-10 card.

The ML-Series card supports using an NMS for SNMP performance monitoring (PM), including monitoring CRC errors. If the NMS supports periodic polling and programmed threshold values to monitor interface index errors (ifInErrors) for all the ML-Series card interfaces, you can manage and monitor CRC errors by relying on the NMS.

If the NMS does not support polling or if the desired polling frequency uses too much bandwidth, you can configure SNMP traps on the ML-Series card through the Cisco IOS CLI. This method is only for ML-Series cards on the ONS 15454 SONET/SDH.

### Configuration Guidelines for CRC Thresholds on the ML-Series Card

These are the guidelines for determining the interface CRC errors (ifInErrors) threshold values for generating an NMS PM alert:

- SONET/SDH bit errors also create POS CRC errors. There is no alarm suppression hierarchy between the SONET/SDH errors and POS errors, so each set of errors creates separate alerts.

- The actual packet rate of an interface is unpredictable. A high bandwidth interface might forward only a few packets per minute in a particular time period of low data traffic, which means a relatively low number of CRC errors would represent a 100 percent loss. A lower bandwidth interface might forward a high packet count (millions) per minute during a particular time period, and so a relatively few CRC errors would represent an error rate of 10-9. This situation prevents the straightforward determination of a maximum BER, which is often used for non-packet-based PM.
- You can set up the monitoring of ML-Series card CRC errors for either signs of minor trouble or signs of major trouble. For minor trouble monitoring, set a relatively quick and sensitive error rate trigger, such as 10 errors in a 60 second period. This method will likely generate an NMS alert every time an interface goes up or down, a fiber error occurs, or a SONET/SDH protection event occurs (even though protection might occur within 50 ms). To monitor only major trouble and to reduce the number of alerts, set a relatively high threshold, such as 1000 errors in a 300 second period.

## Accessing CRC Errors Through SNMP

CRC errors for each interface are reported in the IF-MIB object ifInErrors (OID 1.3.6.1.2.1.2.2.1.14). Users can check the current value of ifInErrors through SNMP get requests. Each ML-Series card runs a separate instance of SNMP. SNMP requests are relayed to the individual ML-Series card based on the community string. The community string uses the following format:

```
com_str_configured_from_CTC@ml_slot_number
```

## Configuring an SNMP Trap for the CRC Error Threshold Using Cisco IOS

The ML-Series card supports RMON trap functionality in Cisco IOS. You must use the Cisco IOS CLI to configure RMON to monitor ifInErrors and generate a trap to an NMS when a threshold is crossed. The ML-Series card on the ONS 15454 SONET/SDH does not support the configuration of RMON traps through an SNMP set request, which typically initiates an action on a network device.

Beginning in privileged EXEC mode, follow these steps to configure RMON to monitor ifInErrors and generate a trap for an NMS when a threshold is crossed:

| Step | Command | Purpose |
|---|---|---|
| 1 | `Router# configure terminal` | Enter global configuration mode. |
| 2 | `Router (config)# rmon event number [log] [trap community] [description string] [owner string]` | Add an event in the RMON event table that is associated with an RMON event number.<br>• For *number*, assign an event number. The range is 1 to 65535.<br>• (Optional) Use the **log** keyword to generate an RMON log entry when the event is triggered.<br>• (Optional) For **trap** *community*, enter the SNMP community string used for this trap.<br>• (Optional) For **description** *string*, specify a description of the event.<br>• (Optional) For **owner** *string*, specify the owner of this event. |
| 3 | `Router (config)# rmon alarm number ifInErrors.ifIndex-number interval` | Set an alarm on the MIB object. |

| | | |
|---|---|---|
| | {**absolute** \| **delta**} **rising-threshold** *value* [*event-number*] **falling-threshold** *value* [*event-number*] [**owner** *string*] | • For *number*, specify the alarm number. The range is 1 to 65535.<br>• The *ifIndex-number* variable is the ifIndex number of an ML-Series card interface in decimal form. (For information about determining this number, see Determining the ifIndex Number for an ML-Series Card.)<br>• For *interval*, specify the time in seconds the alarm monitors the MIB variable. The range is 1 to 4294967295 seconds.<br>• Specify the **absolute** keyword to test each MIB variable directly. Specify the **delta** keyword to test the change between samples of a MIB variable.<br>• For *value*, specify a number at which the alarm is triggered and a number at which the alarm is reset. The range for the rising threshold and falling threshold values is -2147483648 to 2147483647.<br>• (Optional) For *event-number*, specify the event number to trigger when the rising or falling threshold exceeds its limit.<br>• (Optional) For **owner** *string*, specify the owner of the alarm. |
| 4 | Router (config)# **end** | Return to privileged EXEC mode. |

Below is an example of configuring an SNMP trap for the CRC error threshold.

```
Router # configure terminal

Router(config)# rmon event 10 log trap slot15 owner config

Router(config)# rmon alarm 9 ifInErrors.983043 300 delta rising-threshold
1000 10 falling-threshold 1000 10 owner config

Router(config)# end

Router # show running-config

Router # copy running-config startup-config
```

The ifIndex number of an ML-Series card interface in decimal form used for the rmon alarm command in the example is **ifInErrors.983043**. This variable is the MIB object to monitor combined with the ifIndex number of an ML-Series card interface. For information on determining the ifIndex number for an ML-Series card, see the Determining the ifIndex Number for an ML-Series Card.

The following example shows a rising-threshold trap generated by 1002 ifInErrors crossing a threshold of 1000 in a 5-minute period:

```
2005-03-22 16:25:38 ptlm9-454e56-97.cisco.com [10.92.56.97]:

SNMPv2-MIB:sysUpTime.0 = Wrong Type (should be Timeticks): 43026500

SNMPv2-MIB:snmpTrapOID.0 = OID: RMON-MIB:risingAlarm

RFC1271-MIB:alarmIndex.9 = 9

RFC1271-MIB:alarmVariable.9 = OID: IF-MIB:ifInErrors.983043

RFC1271-MIB:alarmSampleType.9 = deltaValue(2)

RFC1271-MIB:alarmValue.9 = 1002

RFC1271-MIB:alarmRisingThreshold.9 = 1000

SNMPv2-SMI:snmpModules.18.1.3.0 = IpAddress: 10.92.56.97
```

## Determining the ifIndex Number for an ML-Series Card

When an NMS polls an ML-Series card for performance data, the NMS uses ifIndex numbers internally to consolidate interface data from multiple MIBs and associate this data with an interface name. The user can rely on the interface name and does not need to know the actual ifIndex number.

When you use the Cisco IOS CLI to configure the ML-Series card to generate traps directly, you do not have this associated name to use. You must use the actual ifIndex number for each interface being configured with a trap. To determine the actual ifIndex number, you can use an NMS to retrieve the ifIndex number of each ML-Series card interface and VLAN subinterface, or you can calculate the ifIndex number for the interface.

The user can also use a MIB browser (SNMP MIB definition lookup service) to examine the ifDescr for the appropriate ifIndex number. The ifIndex number from the ifDescr must be the ifIndex number for the desired port.

On an ML-Series card, the ifIndex number of Ethernet and POS interfaces is compiled from two pieces of information:

- The chassis slot number of the card-The slot number is the number of the physical space in the shelf that the ML-Series card resides in. It ranges from Slot 1 to Slot 6 or Slot 12 to Slot 17 on an ONS 15454 SONET/SDH shelf. You can find this information in many ways, including through the graphical representation of the shelf slots on CTC, or by looking at the front of the physical shelf.
- A local port number within the card-Port numbers of the ML-Series cards for the ONS 15454 SONET/SDH match the interface numbers for Fast Ethernet and Gigabit Ethernet interfaces. POS port numbers do not match the interface numbers and do not consecutively follow the Ethernet port numbering. A consecutive value is skipped between the last Ethernet port number and the first POS number (POS Port 0). Port numbers for the interfaces are listed in Table 23-1.

**Table 23-1: Port Numbers for ML-Series Card Interfaces**

| ML100T-12 FastEthernet | ML100X-8 FastEthernet | ML1000-2 Gigabit | ML-MR-10 Gigabit | ML100T-12 POS | ML100X-8 POS | ML1000-2 POS | ML-MR-10 RPR |
|---|---|---|---|---|---|---|---|

| Interfaces | Interfaces | Ethernet Interfaces | Ethernet Interfaces | Interfaces | Interfaces | Interfaces | Interfaces |
|---|---|---|---|---|---|---|---|
| FE 0 = Port 0 | FE 0 = Port 0 | GE 0 = Port 0 | GE 0 = Port 0 | POS 0 = Port 13 | POS 0 = Port 9 | POS 0 = Port 3 | RPR West = Port 0 |
| FE 1 = Port 1 | FE 1 = Port 1 | GE 1 = Port 1 | GE 1 = Port 1 | POS 1 = Port 14 | POS 1 = Port 10 | POS 1 = Port 4 | RPR East = Port 1 |
| FE 2 = Port 2 | FE 2 = Port 2 | - | GE 2 = Port 2 | - | - | - | - |
| FE 3 = Port 3 | FE 3 = Port 3 | - | GE 3 = Port 3 | - | - | - | - |
| FE 4 = Port 4 | FE 4 = Port 4 | - | GE 4 = Port 4 | - | - | - | - |
| FE 5 = Port 5 | FE 5 = Port 5 | - | GE 5 = Port 5 | - | - | - | - |
| FE 6 = Port 6 | FE 6 = Port 6 | - | GE 6 = Port 6 | - | - | - | - |
| FE 7 = Port 7 | FE 7 = Port 7 | - | GE 7 = Port 7 | - | - | - | - |
| FE 8 = Port 8 | - | - | GE 8 = Port 8 | - | - | - | - |
| FE 9 = Port 9 | - | - | GE 9 = Port 9 | - | - | - | - |
| FE 10 = Port 10 | - | - | - | - | - | - | - |
| FE 11 = Port 11 | - | - | - | - | - | - | - |

The slot and port are combined to form the ifIndex using the following formula:

ifIndex = (slot * 10000h) + (port)

10000h is the hexadecimal equivalent number of 65536. The resulting ifIndex is a meaningful two-part number in hexadecimal, but seems confusing and arbitrary in decimal. For example, ifIndex E0002h is Slot 14, Port 2. This same number in decimal notation is 917506. The rmon alarm command requires the ifindex number in decimal form.

As an additional reference for calculating the correct ifindex value to use with the rmon alarm command, Table 23-2 lists the base ifindex number for Slots 1 to 17. The desired port number can be added to the slot base number to quickly determine the correct ifIndex number.

**Table 23-2: Port Numbers for the Interfaces of ML-Series Cards**

| Slot Number for the ML-Series Card | Base ifIndex Number in Hexidecimal Format | Base ifIndex Number in Decimal Format |
|---|---|---|
| 1 | 10000h | 65536 |
| 2 | 20000h | 131072 |
| 3 | 30000h | 196608 |
| 4 | 40000h | 262144 |
| 5 | 50000h | 327680 |
| 6 | 60000h | 393216 |

| | | |
|---|---|---|
| 12 | C0000h | 786432 |
| 13 | D0000h | 851968 |
| 14 | E0000h | 917504 |
| 15 | F0000h | 983040 |
| 16 | 100000h | 1048576 |
| 17 | 110000h | 1114112 |

## Manually Checking CRC Errors on the ML-Series Card

Users can also check the current count of ML-Series card CRC errors on an interface by using the show interface command. The example shows ten total input errors, which are all CRC errors, in the last line of the output.

Router# **show interface gigabitEthernet 0** GigabitEthernet0 is up, line protocol is up

Hardware is marvel_port, address is 0019.076c.8436 (bia 0019.076c.8436)

MTU 9600 bytes, BW 1000000 Kbit, DLY 10 usec,

reliability 255/255, txload 1/255, rxload 125/255

Encapsulation: ARPA, loopback not set

Keepalive not set

Full-duplex, 1000Mb/s, media type is SX

output flow-control is unsupported, input flow-control is unsupported

ARP type: ARPA, ARP Timeout 04:00:00

Last input 00:00:00, output 00:00:00, output hang never

Last clearing of "show interface" counters 00:01:50

Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0

Queueing strategy: fifo

Output queue: 0/40 (size/max)

5 minute input rate 490884000 bits/sec, 613608 packets/sec

5 minute output rate 0 bits/sec, 0 packets/sec

200 packets input, 20000 bytes, 0 no buffer

Received 0 broadcasts (0 IP multicast)

0 runs, 0 giants, 0 throttles

10 input errors, 10 CRC, 0 frame, 0 overrun, 0 ignored

0 watchdog, 0 multicast, 0 pause input

2 packets output, 644 bytes, 0 underruns

0 output errors, 0 collisions, 0 interface resets

0 babbles, 0 late collision, 0 deferred

0 lost carrier, 0 no carrier, 0 PAUSE output

0 output buffer failures, 0 output buffers swapped out

# Displaying RMON Status

**Note:** RMON status commands do not work for POS interfaces.

To display the RMON status, use one or more of the privileged EXEC commands in Table 23-3.

**Table 23-3: Commands for Displaying RMON Status**

| Command | Purpose |
|---|---|
| **show rmon** | Displays general RMON statistics. |
| **show rmon alarms** | Displays the RMON alarm table. |
| **show rmon events** | Displays the RMON event table. |
| **show rmon history** | Displays the RMON history table. |
| **show rmon statistics** | Displays the RMON statistics table. |

Example 23-1 shows examples of the commands in Table 23-1.

**Example 23-1: CRC Errors Displayed with show rmon Commands**

```
Router# show rmon alarms
Alarm 9 is active, owned by config
Monitors ifInErrors.983043 every 300 second(s)
Taking delta samples, last value was 0
Rising threshold is 1000, assigned to event 10
Falling threshold is 1000, assigned to event 10
On startup enable rising or falling alarm

Router# show rmon events
Event 10 is active, owned by config
Description is
Event firing causes log and trap to community slot15,
last event fired at  0y3w2d,00:32:39,
Current uptime       0y3w6d,03:03:12
Current log entries:
index  uptime              description
1      0y3w2d,00:32:39
```