

SONET Triggers

Document ID: 62622

Contents

Introduction

Prerequisites

- Requirements
- Components Used
- Conventions

Events that Bring Down a POS Interface

- Section and Line Level Triggers
- Path Triggers
- Summary of POS Triggers CLI Behavior
- Debouncing of SONET Alarms
- Defect Handling

Triggers in Action

Why Use Triggers?

SLAs and POS Triggers

- Theorem
- Postulates

Deployment of SONET Triggers

- Protected SONET Network: No APS on the Routers
- Internally Unprotected SONET Network
- Protected or Unprotected SONET Network
- Protected DWDM Network
- Unprotected DWDM Network
- Routers Connected Back-to-Back
- Remote Notification Based on Signal Quality

Related Information

Introduction

A trigger is any event that fulfills the role of *cause* in the cause-and-effect relationship in a Synchronous Optical Network (SONET) interface in IOS. Sometimes, you can use the **pos delay triggers** command. At other times, Cisco recommends that you do not use the **pos delay triggers** command, especially when you attempt to meet tight Service Level Agreements (SLAs). Service providers sell differentiated levels of service based on certain agreements. The agreements deal with how the network internally routes, protects, or prioritizes customer traffic. These commands assist providers to tune networks to meet service agreements.

This document examines the triggers that relate to interface up and down events. This document also explains how to deploy Packet Over SONET (POS), and considers SLAs and convergence times at Layer 3.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

This document is not restricted to specific software and hardware versions.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Events that Bring Down a POS Interface

This section describes the events that bring down a POS interface, and lists the related commands.

Section and Line Level Triggers

The list of triggers in this section refers to the GR-253-CORE *Synchronous Optical Network (SONET) Transport Systems: Common Generic Criteria* specification:

- Section Loss of Signal (SLOS) The specification indicates that you must detect no less than 2.5us, and no greater than 100us (6.2.1.1.1).
- Section Loss of Frame (SLOF) The specification indicates that you must detect this in a minimum of 3ms (or 24 consecutive errored framing patters) (6.2.1.1.2).
- Alarm Indication Signal – Line (AIS–L) AIS–L must be sent out when appropriate, within 125usec of detection. A device must detect the receipt of AIS–L if the device sees 5 consecutive frames where bits 6,7, and 8 of K2 are set to 111 (6.2.1.2.1).
- Signal Degrade Bit Error Rate (SD–BER) SD–BER is a trigger only on interfaces with Automatic Protection Switching (APS) (tied to B2 BER calculation).
- Signal Failure Bit Error Rate (SF–BER) SF–BER is a trigger for both APS and non–APS interfaces (tied to B2 BER calculation).
- Remote Defect Indication – Line (RDI–L) RDI–L is not a trigger for POS or APS. (However, RDI–L is a trigger for MPLS FRR) (section 5.3.3.1).

For more information on the sections mentioned in this list, see the Telcordia Information SuperStore [web site](#).

Related Commands

The **pos delay triggers line n** command holds off LOS/LOF/AIS for n ms before the command triggers the line down:

If you configure the command without any numeric value, the delay time is 100ms by default. You can use Line triggers on any non–APS POS interface. You cannot use Line triggers on interfaces that participate in APS, because Line triggers interfere with APS operation. The **pos delay triggers line n** command does not allow the line to go down on the brief LOS that comes from internally protected Dense Wavelength–Division Multiplexing (DWDM) gear, from the time an internal DWDM protection switch occurs. If the defect clears

during the holdoff period, it is like the defect never occurred.

The **pos delay triggers line** command holds off any action based on the defect (except to increment the defect counter) until the specified holdoff period ends.

If you do not enable this command, APS and link down from the above SONET defects are triggered immediately in the Route Processor (RP).

Path Triggers

These specific PATH level defects initiate a state change only if you have enabled **pos delay triggers path** on the interface:

- **AIS–P** This defect must be raised within 125usec from the detection of the defect that results in the AIS–P. The Path Terminating Equipment (PTE) must detect this defect when the H1 and H2 bytes for an STS path contain all 1s for 3 consecutive frames. Concatenated paths need to observe only the first H1 and H2 bytes. For more information, see section 6.2.1.2.2 of R6–175 and R6–176.
- **RDI–P** If RDI–P is present, the defect must be detected within 10 frames. See 6.2.1.3.2 of R6–221.
- **B3–TCA (Threshold Crossing Alarms) for B3** This alarm is tied to the B3 Binary Synchronous Communications (Bisync) IP (BIP) calculation.
- **LOP–P (Path Loss of Pointer)** (if the IOS version includes CSCdx58021) See section 6.2.1.1.3 of GR–253.

For more information on the sections mentioned in this list, see the Telcordia Information SuperStore [web site](#).

Related Command

The **pos delay triggers path** *<msec>* command enables link–down triggering on AIS–P, RDI–P, and excessive B3 errors. By default, link–down triggering for path errors is disabled.

The command also specifies a holdoff time in the range of 0 to 511 ms (the default is 100ms). Path trigger defects (AIS–P, RDI–P) that clear before the end of the holdoff period do not cause triggering. When you have not explicitly configured this command on a POS interface, no action results if the PATH level defects are processed. Unlike the Line triggers, APS interfaces allow Path triggers, because Path triggers do not interfere with the line level activity of APS. Path triggers were not allowed to be configured with APS in versions earlier than Cisco IOS® Software Release 12.0(28)S. Path triggers were added in order to speed up the link up/down behavior of POS interfaces when connected to SONET networks. This allowed quicker Layer 3 convergence in the presence of remote errors.

Summary of POS Triggers CLI Behavior

This table lists the POS triggers conditions and the associated results:

Condition	Result
If you have configured nothing explicitly related to POS triggers.	Line level triggers are processed immediately.
If you have configured the pos delay triggers line command.	Line level triggers are processed after a delay of 100ms.

If you have configured the pos delay triggers line x command.	Line level triggers are processed after x msecs, where x is between 0 and 511.
If you have configured nothing explicitly related to Path triggers.	Path triggers are not processed and will not cause any action to be taken.
If you have configured the pos delay triggers path command.	Path level triggers are processed after a delay of 100ms.
If you have configured the pos delay triggers path x command.	Path level triggers are processed after x msecs, where x is between 0 and 511.

Debouncing of SONET Alarms

SONET alarms that result from defects are held for 10 seconds (10.5 \pm .5) after the defect clears.

Defect Handling

In IOS, the POS cards change their LINE state due to different triggers, through two general means for defect processing. While this depends on the specific configuration of the interface (APS or non-APS), in general there are two types of failures:

- Managed
- Unmanaged

You must understand the terms specific to alarm-handling that this document uses:

- Defect The failure condition that the hardware recognizes.
- Failure A defect that has been soaked for the required ~ 2.5 sec, and then is reported through the SONET-4-ALARM messages. Any defect that is a trigger does not get soaked.
- Unmanaged failures Events such as LOS, LOF, etc. They are detected by the SONET framer by a defined set of parameters, and require no calculation. There is either a defect present and asserted by the hardware, or there is no defect. Hard failures such as these, in general, are handled through interrupts. LOS, LOF, AIS-L, and in special cases, AIS-P and RDI-P get asserted immediately. These are dependent on the framer and the defined rules to detect each of these defects. The effect of these defects is immediate. However, you can instruct the router to delay assertion of this defect as a failure. There are two timers that determine the delay value, **pos delay triggers [path | line]** and carrier delay. These are addressed later in the document.
- Managed alarms Events such as TCAs and SD/SF-BER calculations. These require some calculation to determine if they are present, are on the increase or decrease, etc. For example, you cannot have an LOS that increases its LOS-ness from the perspective of the router. However, you can have BER that is on the increase or decrease; the action taken may be different. Soft failures, like BER and TCA, need some calculation, because they depend on a number of factors, for example, thresholds that a user can configure, bit rate, and maximum number of BIP CVs (because they are different for B1, B2, and B3). These failures also take longer to be detected, because the hardware is polled for the BIP counters, and also because these types of defects are gradual in nature and accumulated over time. It is also true that in general you do not go from 0 BIP straight to an signal degrade (SD) or signal failure (SF) without some other type of hard failure present in the network. These defects are slower to occur when compared to the hard failures.

Here is a generalized approach to basic calculations that describes how to calculate BER:

After each restart of the calculations and until BER_Period reaches Required_BER_Period (the integration window is not completely deployed), the algorithm functions strictly as an integrating or averaging one:

- $BER_Period = BER_Period + 1 \text{ sec.}$
- $Current_BIP = Current_BIP + BIP_new.$
- $Current_BER = Current_BIP/BER_Period.$

After BER_Period reaches Required_BER_Period (the integration window was completely deployed and starts to slide), the algorithm functions as a leaky bucket one:

- $BER_Period = Required_BER_Period.$
- $Current_BIP = Current_BIP + BIP_new - Current_BER * 1 \text{ sec.}$
- $Current_BER = Current_BIP/BER_Period.$

The Required_BER_Period is determined based on only the line rate and the configured BER threshold, following the standards (See figure 5-5, Switch Initiation Time Criteria, GR-253). However, it is lower limited to 1 second, our sampling rate.

Thus, the BER_Period (integration window) moves with each poll, and a new BER gets calculated with each poll. If the Current_BER is ever over a defined limit, we raise the appropriate defect immediately during that same poll or calculation interval, and keep the response minimal. We repeat these calculations every second, and check to see if one of three events has occurred:

- BER still falls within that same range. There is no new action.
- BER has increased again, and crossed an SD or SF threshold (for B2). Raise a new alarm.
- BER has decreased below a BER threshold. Clear the alarm.

For the assertion of a TCA or SD/SF, you need to wait only until you have crossed a limit at that respective poll interval. At the time of the calculation, check whether the Current_BER has crossed a threshold, and if it has, you can go ahead and assert the alarm immediately through software.

This is valid because, if the Current_BER is big enough to trigger the alarm initially, the condition is still true at the end of the BER_Period. This is based on how the values are defined and compared in relation to the calculation window.

When you clear an alarm, you need to wait until the end of the BER_Period calculation window. This is to ensure that no new BIPs are accumulated during the last portion of the window that might keep you above a threshold.

Note: According to GR-253, SD-BER and SF-BER are both tied strictly to the B2 BIP count. The current default thresholds are:

- BER thresholds SF = $10e-3$ SD = $10e-6$
- TCA thresholds B1 = $10e-6$ B2 = $10e-6$ B3 = $10e-6$

Note: Engine2 OC-48 cards have these default thresholds:

- BER thresholds SF = $10e-4$ SD = $10e-6$
- TCA thresholds B1 = $10e-6$ B2 = $10e-6$ B3 = $10e-6$

If you want to have B3 TCA Path trigger act similar to SF, the B3 threshold must be set down to the same threshold, $10e-3$. You can do so through the **pos threshold b3-tca 3** command at the `router(config-if)#` prompt.

Note: As the polling interval is one second, that is the minimum time in which we will ever notice and raise TCA or SD/SF defect. Additionally, due to the accumulated nature of TCA/SD/SF, these types of failures are accompanied by some other failure when they occur quickly in typical failures. This maintains a balance between router processor utilization and performance. The polling interval cannot be configured.

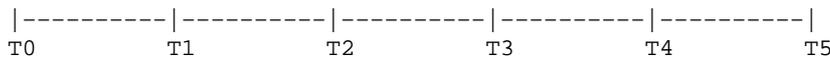
Triggers in Action

This section provides some background information to examine the interaction of some of the various user tunable knobs in IOS:

The **pos delay triggers [line | path]** command briefly delays the reporting and action of a defect.

POS delay trigger line is the hold time before reacting to a line alarm. The default is immediate reaction, which means **pos delay trigger line 0**. If you directly configure **pos delay trigger line** without any value, then the default value of 100ms is taken into account. This allows for an immediate or delayed response, based on the desired effect. With either of these configured, the defect does not show up as an active alarm until the holdoff period is over.

Timeline:



Here:

- t0 Time when the defect occurs.
- t1 Time when the hardware detects the defect.
- t2 Time when the defect gets reported as a failure.
- t2–t3 Time that is held off for any configured triggers.
- t3–t4 Time for which you wait due to carrier delay.
- t4 Time when the interface actually comes down in IOS.
- t5 Time at which any adjacency for a routing protocol comes down.

Examine the timeline to observe how to tweak the different knobs to achieve various results.

The **post delay triggers** command affects the duration between t2 and t3, and in effect, hides the defect from IOS, until the holdoff period is over. Of course, if the defect is cleared before you reach t3, nothing occurs, and it is as if nothing happened. The default value for both line and Path triggers is 100ms, and the range is 0 to 511 ms. Path triggers are not enabled (in other words, they do not take any action) unless **pos delay triggers path** is first configured. **pos delay trigger path** is the hold time before reacting to a path alarm. The default is no reaction. If you directly configure **pos delay trigger path** without any value, then the default value 100ms will be assigned automatically. This includes AIS–P, RDI–P, and B3–TCA. This functionality was added through CSCds82814 (around 12.0(15.5)S/ST).

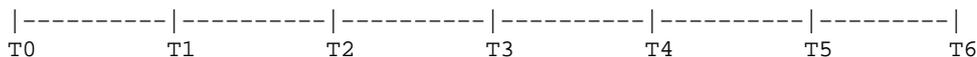
Carrier–delay is the hold time between the end of the POS delay hold time and it will bring down the IOS interface. The default is 2000 msec. Carrier delay is the time between t3 (when IOS becomes aware of a failure) and t4 (when the interface goes down). By default, this is set to 2 seconds, and can be configured for msec values. As the timeline indicates, it is an additive function on top of the SONET level holdoff timers. It behaves in the same way as the POS triggers – if the alarm clears before the end of the holdoff period, the interface is not brought down. However, there is a conundrum here. The SONET debounce timer does not clear the defect before the carrier delay activates, unless carrier delay is large (well over 10 seconds). This results in a situation where carrier delay is almost always activated, and therefore must be considered to be rather small when deployed with POS interfaces. Carrier delay is also added after the alarm is cleared, before the interface is declared up as well. Hence, you can count the value of carrier delay twice before the interface

comes back up.

With some interfaces and physical media this is helpful. However, with POS interfaces there are a number of triggers and timers that you can use, and combined to create the desired effect, without carrier delay taking such a major role. A carrier delay value of 0–8 msec is a good starting point for customers to consider when they test these knobs on their own. In general, a good strategy is to use the **pos delay triggers** command to absorb any problems, and provide the desired holdoff effect. Carrier delay can be kept small to minimize its impact.

The SONET debounce timer mentioned above is set at 10 seconds (+/- .5sec), and is required by GR–253 to ensure that a flap period less than 10 seconds does not occur. The timer starts after the defect is cleared. The timer is reset if another defect event occurs before the timer window has expired.

Timeline:



Here:

- t0 Defect clears.
- t0 Debounce timer starts.
- t4 t0 + 10sec (hence, the failure must clear if no new defects occur between t0 and t4).

If an event occurs before t4, (say) at t2 (it could be another defect, or a reoccurrence of the same type of defect), the timer is stopped until this new defect is cleared. At t3, the timer starts again, when there are no active defects, and counts for the ~10 seconds. If no new events are encountered, clear the alarm at t5, and then start the carrier delay timer. When carrier delay has been cleared at t6, bring up the interface again.

This information should allow the customer to understand more clearly how the POS interfaces react to various SONET/SDH conditions. This allows the equipment to be configured more precisely according to the customers intended behavior.

Why Use Triggers?

This section explains when you must use the **pos delay triggers [line | path]** command, and when you must not use it.

Here are the scenarios when you must not use **pos delay triggers**. There are several scenarios:

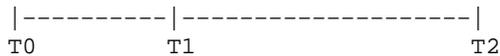
- You cannot use line triggers with APS–configured interfaces. Versions earlier than Cisco IOS Software Release 12.0(28)S did not allow even the use of Path triggers.
- When you explicitly do not want PATH level defects to bring down the interface, you cannot use these triggers.
- When you want line level triggers to bring down the interface with no delay, you cannot use this command.

Here are the scenarios when you can use **pos delay triggers** command:

- When you want to hold off the effect of a line level defect temporarily.
- To enable the ability for PATH level defects to bring down the interface immediately.
- To enable PATH level defects to bring down the interface, but with some holdoff included.

SLAs and POS Triggers

Examine this timeline:



- Time $t=0$ (t_0) When the defect is detected.
- Time t_2 The required SLA restoration time.
- Time t_1 Any holdoff from the **pos delay triggers** command that is configured (the default for LINE is 0 and the default for PATH is not enabled).
- X is the holdoff value (so $X =$ the value of t_1).
- Y is the time it will take Layer 3 to restore service.

Theorem

Sometimes, you can use the **pos delay triggers** command, while at other times, you cannot, especially when you attempt to meet tight Service Level Agreements (SLAs).

Postulates

- If $Y > (t_2 - t_1)$ for any value of t_1 , a holdoff is not a good idea because, you cannot meet your SLA if you configure any holdoff.
- If $Y \leq (t_2 - t_1)$, you can consider the implementation of a holdoff. If the duration of the failure is less than $(t_1 - t_0)$, you can hold off because, you do not have to utilize router resources, and you can meet the desired SLA. If the defect persists past time t_1 , you can still meet the SLA, even though you lose some time before you initiate restoration at the IP level.

You must have some knowledge about the underlying transport network, and the convergence times of the Layer 3 network, in order to know the values that you can use in these formulas. You also need to perform some testing.

Here is how the triggers work:

- The **pos delay triggers line** n command holds off LOS/LOF/AIS for n ms before the command triggers line down. The default value is 100ms. You can use this command on any non-APS POS interface. The **pos delay triggers line** n command does not allow the line to go down on the brief LOS that comes from internally-protected DWDM gear, from the time an internal DWDM protection switch occurs. If the defect clears during the holdoff period, it is like the defect never occurred.
- The **pos delay triggers line** command will hold off any action based on the defect (except to increment the defect counter), until the specified holdoff period ends.

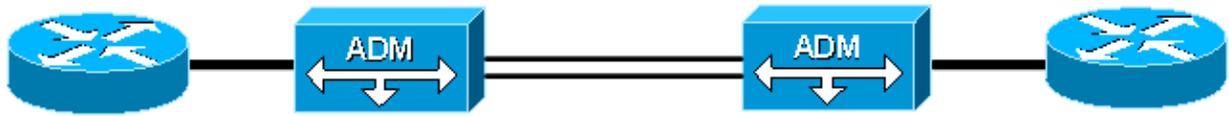
If you do not enable this command, APS and link down are triggered immediately in the RP.

Deployment of SONET Triggers

This section describes the deployment of SONET triggers.

Protected SONET Network: No APS on the Routers

Figure 1 Internally Protected SONET Network



The SONET network has internal protection, which means that a failure inside the SONET network triggers some protection switch to restore service very fast. Therefore, you need to consider whether you want to bring down the interface and notify Layer 3. In most cases, when a protection switch occurs inside the SONET network, the routers see a brief line or path AIS while the network takes restorative action. However, this occurs only if the failure is one hop away from either router. The SONET network can possibly be several NEs in diameter, either router sees LINE failures only as PATH failures. In this case, consider path and line level triggers if you want a holdoff.

To make this decision, you need to understand the associated cost with both approaches. As a network operator, you must consider these questions:

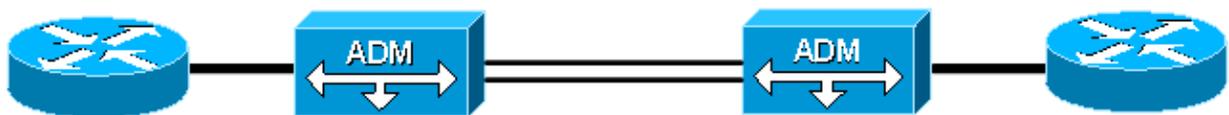
- Does the network converge quickly enough? If not, this approach is not suitable.
- What is the impact of routing around such a failure? Is the impact so great on the router that the performance drops below an acceptable level?

Ultimately, you need to decide whether you can ignore a potential ~60msec hit, or whether you prefer to route around such an event. If you can ignore the hit, you must identify how much of a fudge factor to add in because, you do not want to hold off on this defect only to wait several milliseconds too few, and thereby delay corrective action.

In this scenario, **pos delay triggers line** and **path** are probably sufficient. In addition, consider values of at least 60msec if a holdoff is warranted. If the network is wide enough, and you want to take immediate action on both line and path level defects, you need not configure line level triggers. However, you need to configure **pos delay triggers path** with a value of 0 to enable immediate processing of PATH level defects.

Internally Unprotected SONET Network

Figure 2 Internally Unprotected SONET Network

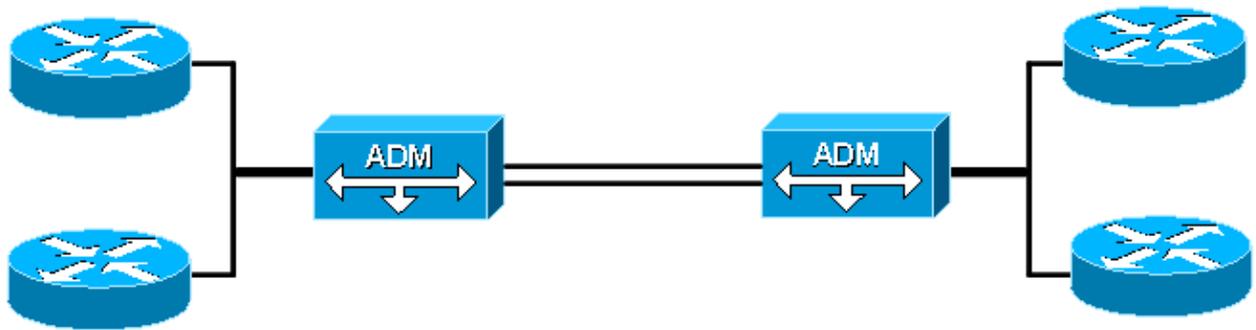


In an unprotected SONET network, you run the same risks as in the first scenario, plus a few more. If the network is large enough, the routers can potentially never see a LINE level defect in the event of a failure, because the defects are all filtered. The routers can see PATH level defects up and down stream. Thus, in some situations, where a failure occurs within the network, the router only sees PATH level events, and there is no end-to-end continuity between the routers. Even worse, no restoration occurs at the SONET level to remedy this situation.

In this scenario, you must configure Path triggers simply to allow the routers at either end to take action when the routers encounter a PATH defect, even if the routers want no holdoff effect. When you have configured Path triggers, as a network operator, you must check whether it is better to hold off or trigger a Layer 3 restoration.

Protected or Unprotected SONET Network

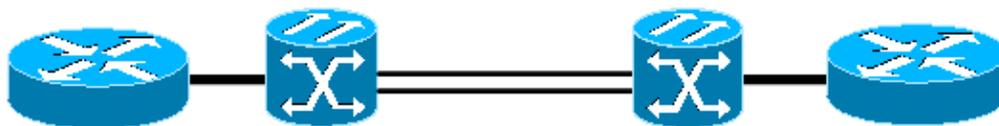
Figure 3 Internally Unprotected SONET Network



In Cisco IOS Software Release 12.0(28)S, you can enable PATH triggers on APS circuits. When you deploy APS on the local or remote routers, an APS switch causes the remote Working and Protect routers to see a brief PATH level defect. With a small trigger value the interfaces go down, and this situation is not desirable. An interface that goes down delays service restoration that is already in progress. A momentary failure that occurs within the cloud can also delay service restoration. However, the occurrence of a persistent PATH level error indicates that the circuit protection (either within the network, or at the far end) has been unable to restore connectivity. In this case, the APS routers must take action, and initiate routing re-convergence. You can configure Path trigger delay values of ≥ 100 ms. With this configuration, when a persistent error occurs either within the SONET network or at the remote end, the routers bring both APS interfaces to a state of link down. Therefore, the routers initiate quicker re-routing and restoration of service.

Protected DWDM Network

Figure 4 Protected DWDM Network



In this scenario, we need not use Path triggers, because the DWDM network does not participate at the SONET protocol level. The router detects any failure at the SECTION or LINE level.

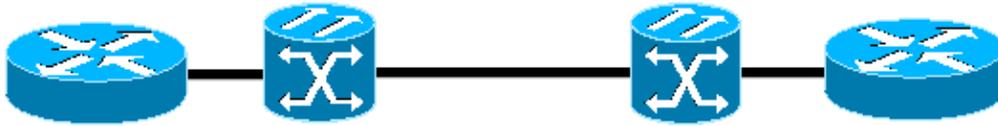
Again, because the DWDM network is internally protected, a failure internal to the network causes restoration to soon occur. The router typically sees a very brief LOS, LOF, or a burst of BIP errors.

Therefore, you only need to decide whether a holdoff is desirable in this network.

The **pos delay triggers line** command is sufficient in this situation, if you choose a delay.

Unprotected DWDM Network

Figure 5 Unprotected DWDM Network



With an unprotected DWDM network in the transport, you need to address any failure within the routers. In this situation, the default configuration would allow for an immediate response to any failures seen at either router because the DWDM does not participate in the SONET protocol. If you desire this effect, the default configuration of no configured POS triggers is appropriate.

If you require some holdoff, the **pos delay triggers line** command is sufficient to provide this functionality.

Routers Connected Back-to-Back

Figure 6 Routers Connected Back-to-Back



Two routers connected back-to-back between two POS interfaces must operate just like the last scenario. You can see failures immediately at either router, because there is no intermediary equipment that operates on the SONET overhead or terminates any part of the SONET level signal.

An interesting situation is when R1 sees S-LOS, and R2 sees both L-RDI and P-RDI, as R1 is both Line-Terminating Equipment (LTE) and Path-Terminating Equipment (PTE). Since L-RDI explicitly disallows any resultant action to be taken upon receipt, R2 does not drop the interface as a result. This issue can potentially lead to a situation where an interface of R1 is down, but the interface of R2 is still up and forwards traffic. Of course, any Layer 2 keepalive (like High-Level Data Link Control (HDLC) provides) times out and declares the link down, typically in 30 seconds, based on the configured timers. However, a number of operators disable these Layer 2 keepalives, and cannot prevent this situation. In order to address this problem, you can take several approaches, and each approach addresses this from a different perspective, as explained here:

- Turn on Path Triggers As P-RDI brings an interface down with Path triggers enabled, you can use this method to cause a quick response, and drop the interface. The interesting point to note is that L-RDI masks out the P-RDI under normal operation as per GR-253. As the POS triggers are handled at the defect level, the triggers are processed before the alarm masking, and the interface still drops according to the configured delay time.
- Enable Layer 2 Keepalives This option causes the interface on R2 to time out after 3 keepalives are missed. This is typically 30 seconds total (3x10), and Cisco does not generally recommend this option as a tool to tune fast link convergence.
- Enable a Link-State Routing Protocol When the interface on R1 is brought down due to the S-LOS, a link state message is sent immediately. Even though the interface on R2 can still be up, when the link state message is received throughout the area, SPF is run, and the link is removed from the topology because the link fails the two-way connectivity check. This prevents the network from trying to route through that simplex scenario.

Remote Notification Based on Signal Quality

When you connect two routers, either back-to-back, or across a SONET network, the provided OAM architecture covers the detection of a majority of failure scenarios.

Typically, there are local notifications and remote notifications. However when a high number of BIP errors cross a threshold (SD or SF, or B3-TCA), no remote notification is sent to indicate that this condition has occurred. Thus, when you employ Multi Protocol Label Switching (MPLS) Fast Re-Route protection, no trigger activates an immediate protection switch. Traffic continues to be blackholed until sufficient traffic is lost to cause a failure of either Layer 2 keepalives on the link or neighbor relationships among Interior Gateway Protocol (IGP) peers. Sometimes this never occurs, and continues to blackhole the traffic.

To address this scenario, CSCec85117 introduces the **pos action b3-ber prdi** command to the POS and SONET command structure.

This command allows the operator to configure the interface to send a P-RDI when the B3 threshold has been crossed. This option enables you to monitor the link end-to-end optimally, regardless of topology. If **pos delay triggers path** is enabled on the routers, the **pos action b3-ber prdi** command activates the link that comes down (and the corresponding Fast ReRoute (FRR) or routing update). This avoids the black-hole effect on degraded links.

To change the sensitivity of this action, tune the b3-tca as shown here:

```
router(config-if)# pos threshold b3-tca ?
```

The value provided is the exponential component for the BER calculation (for example, **pos threshold b3-tca 3** sets the B3-TCA to be equivalent to a rate of 1×10^{-3}).

Related Information

- [Telcordia Information SuperStore](#) 
- [Technical Support & Documentation – Cisco Systems](#)

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2013 – 2014 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Jul 21, 2005

Document ID: 62622
