Configuring SONET/SDH Physical Interface Properties

To configure SONET/SDH physical interface properties, include the sonet-options statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces so-fpc/pic/port]
[Unresolved xref] (sdh | sonet);
sonet-options {
    [Unresolved xref] asx;
    aps {
        [Unresolved xref] milliseconds;
        [Unresolved xref]-b
        authentication-key key;
        force;
        [Unresolved xref] milliseconds;
        lockout;
        neighbor address;
        paired-group group-name;
        protect-circuit group-name;
        request;
        revert-time seconds;
        switching-mode (bidirectional | unidirectional);
        working-circuit group-name;
    }
    bytes {
        e1-quiet value;
        f1 value;
        f2 value;
        s1 value;
        z3 value;
        z4 value;
    }
    fcs (16 | 32);
    [Unresolved xref] (local | remote);
    mpls {
        pop-all-labels {
            required-depth number;
        }
    }
    path-trace trace-string;
    (payload-scrambler | no-payload-scrambler);
    rfc-2615;
    trigger {
        defect ignore;
        defect [Unresolved xref] up milliseconds down milliseconds;
    }
}
vtmapping (itu-t | klm);
(z0-increment | no-z0-increment);
[Unresolved xref] (oc3 | oc12 | oc48);
```

Note that when you configure SONET/SDH OC48 interfaces for channelized (multiplexed) mode (by including the no-concatenate statement at the [edit chassis fpc *slot-number* pic *pic-number*] hierarchy level), the bytes f1 statement has no effect. Currently, the bytes e1-quiet statement is ignored if you include it in the configuration. The bytes f2, bytes z3, bytes z4, and path-trace options work correctly on channel 0 and work in the transmit direction only on channels 1, 2, and 3. When using no-concatenate, you must specify a channel. For more information, see the *JUNOS System Basics Configuration Guide*.

For DS3 channels on a channelized OC12 interface, the bytes f1, bytes f2, bytes z3, and bytes z4 options have no effect. The bytes s1 option is supported only for channel 0; it is ignored if configured on channels 1 through 11. The bytes s1 value configured on channel 0 applies to all channels on the interface.

You can also include some of the statements in the sonet-options statement to set SONET/SDH parameters on ATM interfaces.

You can configure the following SONET/SDH physical interface properties:

Configuring SONET/SDH Framing

The 4-port OC48 PIC with SFP installed, the next-generation SONET/SDH PICs with SFP, and the 4-port OC192 PIC on M-series, MX-series, and T-series routing platforms, support SONET or SDH framing on a per-port basis. This functionality allows you to mix SONET and SDH modes on interfaces on a single PIC. You can use the framing statement to configure incoming SDH links from Europe and outgoing SONET links to the US on the same PIC. Traffic flowing through other ports of the same PIC will not be affected.

When you change SONET/SDH mode on a port, only the port's framing type is changed. The PIC does not go offline.

To configure framing on a per-port basis, include the framing (sdh | sonet) statement at the [edit interfaces so-*fpc/pic/port*] hierarchy level:

```
[edit interfaces]
so-fpc/pic/port {
    [Unresolved xref] (sdh | sonet);
}
```

NOTE: Per-port framing configuration is applicable for SONET interfaces in concatenated mode (default mode) only. When you configure a PIC to operate in nonconcatenated mode, the individual channels inherit framing configuration from the [edit chassis fpc *number* pic *number* framing (sonet | sdh)] hierarchy level.

NOTE: Automatic Protection Switching (APS) is used by SONET add/drop multiplexers (ADMs) to protect against circuit failures. If APS is configured, and you do not change the SONET/SDH mode on both the working and protection port, APS support will not function properly. Both the working and protection ports must have the same mode configuration.

To view interface information, use the operational mode command show interfaces so-*fpc/pic/port*.

Configuring SONET/SDH Interface Speed

You can configure the speed of SONET/SDH interfaces on next-generation SONET/SDH Type 1 and Type 2 PICs with SFP. The speed you select is dependent upon whether the PIC is in concatenated or nonconcatenated mode. In concatenated mode, the bandwidth of the interface is in a single channel. In nonconcatenated mode, the PIC operates in channelized (multiplexed) mode.

Table 1 shows the mode combinations for the next-generation SONET/SDH Type 1 PICs with SFP.

Table 1: Type 1 PIC Mode Combinations

PIC	Mode	Speed Configuration	Default Mode
2-port OC3	2xOC3 concatenated	*fpc/pic*/0 speed oc3	Concatenated
4-port OC3	1xOC12 concatenated	*fpc/pic*/0 speed oc12	
	1xOC12 nonconcatenated	*fpc/pic*/0:0 speed oc3	Nonconcatenated
	4xOC3 concatenated	*fpc/pic/port* speed oc3	Concatenated

Table 1: Type 1 PIC Mode Combinations *(continued)*

PIC	Mode	Speed Configuration	Default Mode
1-port OC12	1xOC12 concatenated	*fpc/pic*/0 speed oc12	Concatenated
	1xOC12 nonconcatenated	*fpc/pic*/0:0 speed oc3	Nonconcatenated
	1xOC3 concatenated	*fpc/pic*/0 speed oc3	

Table 2 shows the mode combinations for the next-generation SONET/SDH Type 2 PICs with SFP.

Table 2: Type 2 PIC Mode Combinations

PIC	Mode	Speed Configuration	Default Mode
1-port OC48	1xOC48 concatenated	*fpc/pic*/0 speed oc48	Concatenated
	1xOC48 nonconcatenated	*fpc/pic*/0:0 speed oc12	Nonconcatenated
	1xOC12 concatenated	*fpc/pic*/0 speed oc12	
	1xOC12 nonconcatenated	*fpc/pic*/0 0 speed oc3	
	1xOC3 concatenated	*fpc/pic*/0 speed oc3	
4-port OC12	1xOC48 concatenated	*fpc/pic*/0 speed oc48	
	1xOC48 nonconcatenated	*fpc/pic*/0:0 speed	Nonconcatenated
	1xOC12 nonconcatenated	*fpc/pic*/0 speed oc3	
	4xOC12 concatenated	*fpc/pic/port* speed oc3|oc12	Concatenated
4-port OC3	1xOC12 concatenated	*fpc/pic*/0 speed oc12	
	1xOC12 nonconcatenated	*fpc/pic*/0:0 speed oc3	Nonconcatenated
	4xOC3 concatenated	*fpc/pic/port* speed oc3	Concatenated

By default, SONET/SDH PICs operate in concatenated mode. To specify interface speed in concatenated mode, include the **speed** statement with options at the [edit interfaces so-*fpc*/*pic*/*port*] hierarchy level:

```
[edit interfaces so-fpc/pic/port
[Unresolved xref] (oc3 | oc12 | oc48);
```

For example, each port of 4-port OC12 PIC can be configured to be in OC3 or OC12 speed independently when this PIC is in 4xOC12 concatenated mode.

To specify interface speed in nonconcatenated mode, include the **speed** statement at the [edit interfaces so-*fpc/pic/port.channel*] hierarchy level:

```
[edit interfaces so-fpc/pic/port.channel]
[Unresolved xref] (oc3 | oc12);
```

To configure the PIC to operate in channelized (multiplexed) mode, include the **no-concatenate** statement at the [edit chassis fpc *slot-number* pic *pic-number*] hierarchy level.

For more information about using the **non-concatenate** statement, see the *JUNOS System Basics Configuration Guide*.

Configuring SONET/SDH Header Byte Values

To configure values in SONET/SDH header bytes, include the **bytes** statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
bytes {
    c2 value;
    e1-quiet value;
    f1 value;
    f2 value;
    s1 value;
    z3 value;
    z4 value;
}
```

You can configure the following SONET/SDH header bytes:

- **c2**—Path signal label SONET/SDH overhead byte. SONET/SDH frames use the C2 byte to indicate the contents of the payload inside the frame. SONET/SDH interfaces use the C2 byte to indicate whether the payload is scrambled. For the c2 byte, *value* can be from **0** through **255**. The default value is **0xCF**.
- **e1-quiet**—Default idle byte sent on the orderwire SONET/SDH overhead bytes. The routing platform does not support the orderwire channel, and hence sends this byte continuously.
- **f1**, **f2**, **z3**, **z4**—SONET/SDH overhead bytes. For these bytes, *value* can be from **0** through **255**. The default value is **0x00**.
- **s1**—Synchronization message SONET/SDH overhead byte. This byte is normally controlled as a side effect of the system reference clock configuration and the state of the external clock coming from an interface if the system reference clocks have been configured to use an external reference. For the s1 byte, *value* can be from **0** through **255**.

Table 3 displays JUNOS software framing bytes for several specific speeds.

Table 3: SONET/SDH Framing Bytes for Specific Speeds

Overhead Bytes	STM4	STM16	STM64	OC12	OC48	OC192
A1	F6	F6	F6	F6	F6	F6
A2	28	28	28	28	28	28
C1	—	—	—	1..12	1..48	1..192
H1/H2	6A0A	6A0A	6A0A	620A	620A	620A
Z0	01/CC	01/CC	01/CC	—	—	—
Concatenated mode	93FF	93FF	93FF	93FF	93FF	93FF

When you configure SONET/SDH header bytes, note the following:

- The C2 byte is the path signal label. If the C2 byte value on an interface does not match the C2 byte value on the remote interface, the path label mismatch (PLM-P) or unequipped (UNEQ-P) alarm might occur.
- When you configure SONET/SDH OC48 interfaces for channelized (multiplexed) mode (by including the no-concatenate statement at the [edit chassis fpc *slot-number* pic *pic-number*] hierarchy level), the bytes f1 statement has no effect.
- Currently, the bytes e1-quiet statement is ignored if you include it in the configuration.
- The bytes f2, bytes z3, bytes z4, and path-trace options work correctly on channel 0 and work in the transmit direction only on channels 1, 2, and 3.
- For DS3 channels on a channelized OC12 interface, the bytes f1, bytes f2, bytes z3, and bytes z4 options have no effect.
- The bytes s1 option is supported only for channel 0; it is ignored if configured on channels 1 through 11. The bytes s1 value configured on channel 0 applies to all channels on the interface.
- Embedded operations channel (EOC) D1, D2, and D3 bytes are not supported.
- For channelized OC12 IQE and channelized OC48 IQE PICs with SFPs:
  - Only C2 (Path signal label) and S1 byte setting is supported.
  - Following header bytes are not supported. The router will syslog an INFO message if a command for an unsupported header byte is received.

    F1—Section user channel byte

    F2—Path user channel byte

    Z3, Z4—SONET/SDH overhead bytes

    E1—quiet default idle byte

Configuring an Incrementing STM ID

When configured in SDH framing mode, SONET/SDH interfaces on a Juniper Networks routing platform might not interoperate with some older versions of ADMs or regenerators that require an incrementing STM ID.

Current SDH standards specify a set of 3**n* overhead bytes in an STM*n* that includes the J0 section trace byte. The rest are essentially unused (spare Z0) and contain hexadecimal values (0x01, 0xCC, 0xCC ... 0xCC). The older version of the standard specified that the same set of bytes should contain an incrementing sequence: 1, 2, 3, ..., 3**n*. Their use was still unspecified although they might have been used to assist in frame alignment. You can configure an incrementing STM ID to enable your Juniper Networks routing platform to interoperate with older equipment that relies on these bytes for frame alignment.

The STM identifier has a precise definition in the SDH specifications. In ITU-T Recommendation G.707, *Network node interface for the synchronous digital hierarchy (SDH)* (03/96), Section 9.2.2.2.

You can explicitly configure an incrementing STM ID rather than a static one in the SDH overhead by including the z0-increment statement at the [edit interfaces *interface-name* sonet-options] hierarchy level. You should include this statement only for SDH mode; do not use it for SONET mode.

```
[edit interfaces so-fpc/pic/port sonet-options]
z0-increment;
```

To explicitly disable incrementing of the STM ID, include the following statement:

```
[edit interfaces so-fpc/pic/port sonet-options]
no-z0-increment;
```

Configuring the SONET/SDH Frame Checksum

By default, SONET/SDH interfaces use a 16-bit frame checksum. You can configure a 32-bit checksum, which provides more reliable packet verification. However, some older equipment might not support 32-bit checksums.

To configure a 32-bit checksum, include the fcs statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
fcs 32;
```

To return to the default 16-bit frame checksum, delete the fcs 32 statement from the configuration:

```
[edit]
user@host# delete interfaces so-fpc/pic/port sonet-options fcs 32
```

To explicitly configure a 16-bit checksum, include the fcs statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
fcs 16;
```

On a channelized OC12 interface, the **sonet-options fcs** statement is not supported. To configure the frame checksum sequence (FCS) on each DS3 channel, you must include the **t3-options fcs** statement in the configuration for each channel.

Configuring Channelized IQ and IQE SONET/SDH Loop Timing

By default, internal clocking (line timing) is used on channelized IQ and IQE interfaces. To configure SONET/SDH or DS3-level clocking, include the **loop-timing** statement:

```
loop-timing;
```

You can include this statement at the following hierarchy levels:

- [edit interfaces ct3-*fpc/pic/port* t3-options]
- [edit interfaces stm1-*fpc/pic/port* sonet-options]

To explicitly configure the default line timing, include the **no-loop-timing** statement in the configuration:

```
no-loop-timing;
```

The **loop-timing** and **no-loop-timing** statements apply only to E1 and T1 interfaces you configure on channelized IQ and IQE PICs. If you attempt to include these statements on any other interface type, they are ignored.

For all channelized IQ and IQE PICs, the **clocking** statement is supported on all channels. To configure clocking on individual interfaces, include the **clocking** statement at the [edit interfaces *type-fpc/pic/port:channel*] hierarchy level. If you do not include the **clocking** statement, the individual interfaces use internal clocking by default.

For more information, see Configuring the Clock Source and [Unresolved xref].

Configuring SONET/SDH Loopback Capability

To configure loopback capability on a SONET/SDH interface, include the **loopback** statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
[Unresolved xref] (local | remote);
```

To exchange BERT patterns between a local routing platform and a remote routing platform, include the **loopback remote** statement in the interface configuration at the remote end of the link. From the local routing platform, issue the **test interface** command.

For more information about configuring BERT, see Interface Diagnostics. For more information about using operational mode commands to test interfaces, see the *JUNOS System Basics and Services Command Reference*.

To turn off the loopback capability, remove the loopback statement from the configuration:

```
[edit]
user@host# delete interfaces so-fpc/pic/port sonet-options loopback
```

For channel 0 on channelized interfaces only, you can include the loopback statement at the [edit interfaces *interface-name interface-type*-options] hierarchy level. The loopback setting configured for channel 0 applies to all channels on the channelized interface. The loopback statement is ignored if you include it at this hierarchy level in the configuration of other channels. To configure loopbacks on individual channels, you must include the *channel-type*-options loopback statement in the configuration for each channel. This allows each channel to be put in loopback mode independently.

For example, for DS3 channels on a channelized OC12 interface, the sonet-options loopback statement is supported only for channel 0; it is ignored if included in the configuration for channels 1 through 11. The SONET/SDH loopback configured for channel 0 applies to all 12 channels equally. To configure loopbacks on the individual DS3 channels, you must include the t3-options loopback statement in the configuration for each channel. This allows each DS3 channel can be put in loopback mode independently.

You can determine whether there is an internal problem or an external problem by checking the error counters in the output of the show interface *interface-name* extensive command:

```
user@host> show interfaces so-fpc/pic/port extensive
```

Example: Configuring SONET/SDH Loopback Capability

To determine whether a problem is internal or external, loop packets on both the local and the remote routing platform. To do this, include the no-keepalives and encapsulation cisco-hdlc statements at the [edit interfaces *interface-name*] hierarchy level, and the loopback local statement at the [edit interfaces *interface-name* sonet-options] hierarchy level. With this configuration, the link stays up, so you can loop ping packets to a remote routing platform. The loopback local statement causes the interface to loop within the PIC just before the data reaches the transceiver.

```
[edit interfaces]
so-1/0/0 {
    no-keepalives;
    encapsulation cisco-hdlc;
    sonet-options {
        loopback local;
    }
    unit 0 {
        family inet {
            address 10.100.100.1/24;
        }
    }
}
```

Configuring the SONET/SDH Path Trace Identifier

The SONET/SDH *path trace identifier* is a text string that identifies the circuit. If the string contains spaces, enclose it in quotation marks.

By default, the JUNOS software uses the router and interface names for the path trace identifier. Depending on the router and interface names, the default path trace identifier might be longer then 16 bytes. The SDH standards define a maximum 16-byte path trace. For this reason, the default path trace identifier might be truncated in SDH mode. You can prevent the path trace identifier from being truncated in SDH mode by configuring a path trace identifier that is under 16-bytes long. In SONET mode, a path trace identifier can be up to 64-bytes long.

For DS3 channels on a channelized OC12 interface, you can configure a unique path trace for each of the 12 channels. Each path trace can be up to 16 bytes. For channels on a channelized OC12 intelligent queuing (IQ and IQE) interface, each path trace can be up to 64 bytes.

To configure a path trace identifier, include the path-trace statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces interface-name sonet-options]
path-trace trace-string;
```

A common convention is to use the circuit identifier as the path trace identifier.

To display the local router's path trace identifier, issue the show interfaces command on the remote router.

Configuring SONET/SDH HDLC Payload Scrambling

SONET/SDH HDLC payload scrambling, which is enabled by default, provides better link stability. Both sides of a connection must either use or not use scrambling.

NOTE: HDLC payload scrambling conflicts with traffic shaping configured using leaky bucket properties. If you configure leaky bucket properties, you must disable payload scrambling, because the JUNOS software rejects configurations that have both features enabled. For more information, see Configuring Receive and Transmit Leaky Bucket Properties on SONET/SDH Interfaces.

On a channelized OC12 interface, the sonet-options payload-scrambler statement is ignored. To configure scrambling on the DS3 channels on the interface, include the t3-options payload-scrambler statement in the configuration for each DS3 channel.

To disable HDLC payload scrambling, include the no-payload-scrambler statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
no-payload-scrambler;
```

To return to the default, that is, to re-enable payload scrambling, delete the no-payload-scrambler statement from the configuration:

```
[edit]
user@host# delete interfaces so-fpc/pic/port sonet-options no-payload-scrambler
```

To explicitly enable payload scrambling, include the payload-scrambler statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
payload-scrambler;
```

Configuring SONET/SDH RFC 2615 Support

RFC 2615, *PPP over SONET/SDH*, requires certain C2 header byte and FCS settings that vary from the default values configured in accordance with RFC 1619 (the previous version of RFC 2615). The newer values are optimized for stronger error detection, especially when combined with payload scrambling at higher bit rate links.

Table 4 shows the older (RFC 1619) and newer (RFC 2615) values, together with the Juniper Networks default values.

Table 4: SONET/SDH Default Settings

Value	RFC 1619	Default	RFC 2615
SONET/SDH C2 header byte	0XCF	0XCF	0X16
Frame checksum (bit)	16	16	32
Payload scrambling	n/a	Enabled	Enabled

To enable support for the RFC 2615 features, include the rfc-2615 statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
rfc-2615;
```

Configuring SONET/SDH Defect Triggers to Be Ignored

A trigger is a defect alarm that causes a physical interface to be marked down. By default, all defects are honored with no hold time. For SONET/SDH and ATM over SONET/SDH interfaces only, you can configure individual triggers to ignore a defect, honor a defect, and apply up and down hold timers to the defect.

Table 5 lists the defects you can configure.

Table 5: SONET/SDH and ATM Active Alarms and Defects

Alarm	Description
Physical	
pll	Phase-locked loop out of lock
lol	Loss of light
Section	
lof	Loss of frame
los	Loss of signal
Line	
ais-l	Alarm indication signal—line
rfi-l	Remote failure indication—line
ber-sd	Bit error rate defect-signal degrade
ber-sf	Bit error rate fault-signal fail
Path	
ais-p	Alarm indication signal—path
locd (ATM only)	Loss of cell delineation
lop-p	Loss of pointer—path
plm-p	Payload label mismatch
rfi-p	Remote failure indication—path
uneq-p	Path unequipped

To configure defects to be ignored, include the trigger statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces interface-name sonet-options]
trigger {
    defect ignore;
}
```

If you configure a defect to be ignored, that defect does not contribute to the interface being marked down or up.

After you configure a defect to be ignored, the JUNOS software reevaluates the state of the defect on the interface. If the defect is outstanding and has caused the interface to be marked down, the interface is marked up.

When you configure a trigger on a low-level defect—for example, an LOS—only the low-level defect is affected. Higher-level defects that might result from the lower-level

defect are not affected by the low-level trigger configuration. Therefore, you must configure higher-level defects as well.

Configuring SONET/SDH Defect Hold Times

By default, an interface is marked down as soon as a defect is detected, and is marked up as soon as the defect is absent. You might want to apply hold times to defects for the following reasons:

- To prevent route flaps from happening before a defect has been outstanding for a longer period than would be expected for an Automatic Protection Switching (APS) cutover
- To reduce the number of interface transitions

NOTE: On M-series and T-series platforms with Channelized SONET IQ PICs and Channelized SONET IQE PICs, the SONET defect alarm trigger hold-time statement is not supported.

When you apply a “down” hold time to a defect, the defect must be present for at least the hold-time period before the interface is marked down. When you apply an “up” hold time to a defect, the defect must remain absent for at least the hold-time period before the interface is marked up, assuming no other defect is outstanding.

When you configure hold timers and the interface goes from up to down, the interface transition is not advertised to the rest of the system until the interface has remained down for the hold-time period. Similarly, when an interface goes from down to up, the interface transition is not advertised until the interface has remained up for the hold-time period.

To configure hold timers, include the hold-time statement at the [edit interfaces *interface-name* sonet-options trigger *defect*] hierarchy level:

```
[edit interfaces interface-name sonet-options trigger defect]
[Unresolved xref] up milliseconds down milliseconds;
```

The time can be a value from 1 through 65,534 milliseconds.

When you configure defect hold times, you should note the following:

- You can configure an up hold time, a down hold time, or both.
- Each interface on a SONET/SDH PIC controls certain aspects of the SONET/SDH overhead. For example, when you configure an OC48 PIC to be nonconcatenated, four interfaces are created. Each interface has its own path overhead. However, all four path interfaces share the same physical, section, and line overhead. This means the following:
  - Each interface’s path trigger configuration is honored.
  - The physical, section, and line trigger configuration for the primary interface (so-*fpc/pic/slot*:0) is applied to all four interfaces.

Therefore, if you configure the so-*fpc*/*pic*/*slot*:0 interface to have a hold time for the LOS trigger, when an LOS event occurs, all four interfaces remain up until the trigger expires, and then all four interfaces are marked down.

- The hold timers on the SONET/SDH defects are applied in addition to any other hold timers you configure on the interface. For example, if an interface is up and you configure a SONET/SDH trigger down hold time of 100 milliseconds and an interface down hold time of 250 milliseconds, when the SONET/SDH defect occurs, the SONET/SDH trigger timer starts. After 100 milliseconds, assuming the defect is still present, the SONET/SDH defect starts the 250 millisecond down timer. After this has expired and again assuming the defect is still outstanding, the interface will be marked down. For more information about interface hold timers, see Damping Interface Transitions.
- Some defects are reported through a periodic poll (once every second). For these defects, there could be up to one second lost before the defect is detected and the hold timer is started. The hold timer expires in precisely the amount of time configured. At that point, the existence of the defect is checked again and the interface is marked up or down accordingly. These defects are as follows:
  - lol
  - pll
  - ber-sf
  - ber-sd
- We recommend the following settings:
  - Configure SONET/SDH defect timers on no more than 64 interfaces per FPC.
  - Configure a combined up hold time and down hold time for a SONET/SDH defect to be at least 100 milliseconds.

Example: Configuring SONET/SDH Defects to Be Ignored

Prevent an LOS from bringing down an interface. An LOS can lead to the following defects:

- AIS-L
- LOF
- PLL
- RFI-L
- RFI-P

```
[edit interfaces sonet-options trigger]
ais-l ignore;
lof ignore;
los ignore;
pll ignore;
rfi-l ignore;
rfi-p ignore;
```

Configuring Virtual Tributary Mapping

You can configure virtual tributary mapping to use KLM mode or ITU-T mode. By default, virtual tributary mapping uses KLM mode.

For the Channelized STM1 IQ and IQE PICs, you can configure virtual tributary mapping by including the **vtmapping** statement at the **[edit interfaces cau4-*fpc/pic/port* sonet-options]** hierarchy level:

```
[edit interfaces cau4-fpc/pic/port sonet-options]
vtmapping (klm | itu-t);
```

For the STM1 PIC, you can configure virtual tributary mapping by including the **vtmapping** statement at the **[edit chassis fpc *slot-number* pic *pic-number*]** hierarchy level:

```
[edit chassis fpc slot-number pic pic-number]
vtmapping (klm | itu-t);
```

Configuring Channelized STM1 Interfaces lists the KLM mappings used by the Channelized STM1-to-E1 PIC interfaces.

Configuring APS and MSP

Automatic Protection Switching (APS) is used by SONET add/drop multiplexers (ADMs) to protect against circuit failures. The JUNOS implementation of APS allows you to protect against circuit failures between an ADM and one or more routing platforms, and between multiple interfaces in the same routing platform. When a circuit or routing platform fails, a backup immediately takes over.

NOTE: For SDH interfaces, the JUNOS software supports multiplex section protection (MSP). You configure MSP with the same CLI statements you use to configure APS.

The JUNOS software supports APS 1 + 1 switching, either revertive or nonrevertive mode, and bidirectional mode only (although you can configure interoperation with line-terminating equipment [LTE] provisioned for unidirectional mode). The JUNOS software does not transmit identical data on the working and protect circuits, as the APS specification requires for 1 + 1 switching, but this causes no operational impact.

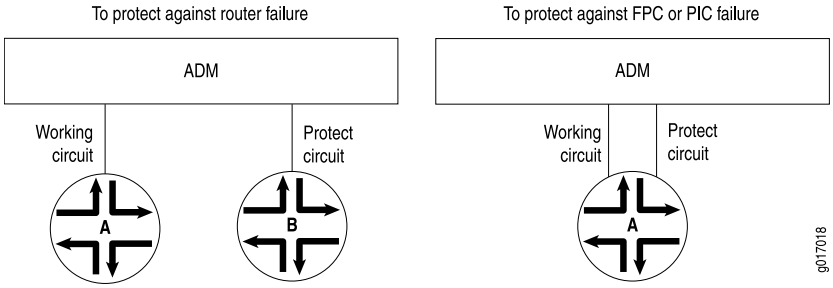
For DS3 channels on a channelized OC12 interface, you can configure APS on channel 0 only. If you configure APS on channels 1 through 11, it is ignored.

With APS and MSP, you configure two circuits, a *working circuit* and a *protect circuit*. Normally, traffic is carried on the working circuit (that is, the working circuit is the active circuit), and the protect circuit is disabled. If the working circuit fails or degrades, or if the working router fails, the ADM and the protect router switch the traffic to the protect circuit, and the protect circuit becomes the active circuit.

To configure APS or MSP, you configure a working and a protect circuit, as shown in Figure 1. To protect against a routing platform failure, you connect two routing

platforms to the ADM, configuring one of them as the working router and the second as the protect router. To protect against a PIC or FPC failure, you connect one routing platform to the ADM through both the working and protect circuits, configuring one of the PICs or FPCs as the working circuit and the second as the protect circuit.

Figure 1: APS/MSP Configuration Topologies



To configure APS or MSP, include the **aps** statement at the [**edit interfaces** *interface-name* **sonet-options**] hierarchy level:

```
[edit interfaces interface-name sonet-options]
aps {
    [Unresolved xref] milliseconds;
    [Unresolved xref]-b
    authentication-key key;
    force;
    [Unresolved xref] milliseconds;
    lockout;
    neighbor address;
    paired-group group-name;
    protect-circuit group-name;
    request;
    revert-time seconds;
    switching-mode (bidirectional | unidirectional);
    working-circuit group-name;
}
```

This section includes the following topics:

- Configuring APS Load Sharing Between Circuit Pairs on page 26
- Example: Configuring APS Load Sharing Between Circuit Pairs on page 28

NOTE: This implementation of APS is not supported on Layer 2 circuits. For Layer 2 circuits, configure APS by including the protect-interface statement. You can include this statement at the following hierarchy levels:

- [edit logical-systems *logical-system-name* protocols l2circuit neighbor *neighbor-id* interface *interface-name*]
- [edit protocols l2circuit neighbor *neighbor-id* interface *interface-name*]

For more information and a configuration example, see the *JUNOS VPNs Configuration Guide*.

When configuring the APS annex-b option, the APS options *must* be configured as follows:

- switching-mode *cannot* be uni-directional
- revert-time *cannot* be configured
- manual-request *cannot* be configured
- excercise-request *cannot* be configured
- lockout-request *cannot* be configured
- wait-to-restore-time is allowed *only* when Annex-B is configured
- protect-circuit *must* be configured
- working-circuit *must* be configured

Configuring Basic APS Support

To set up a basic APS configuration, configure one interface to be the working circuit and a second to be the protect circuit. If you are using APS to protect against routing platform failure, configure one interface on each routing platform. If you are using APS to protect against FPC failure, configure two interfaces on the routing platform, one on each FPC.

For each working–protect circuit pair, configure the following:

- Group name—Creates the association between the two circuits. Configure the same group name for both the working and protect routers.
- Authentication key—You configure this on both interfaces. Configure the same key for both the working and protect routers.
- Address of the other interface on the other routing platform—If you are configuring one routing platform to be the working router and a second to be the protect router, you must configure the address of the remote interface. You configure this on one or both of the interfaces.

The address you specify for the neighbor must never be routed through the interface on which APS is configured, or instability will result. APS neighbor only applies to inter-routing platform configurations. We strongly recommend that you directly connect the working and protect routers and that you configure the interface address of this shared network as the neighbor address.

The working and protect configurations on the routing platforms must match the circuit configurations on the ADM; that is, the working router must be connected to the ADM's working circuit and the protect router must be connected to the protect circuit.

To set up a basic APS configuration, include the following statements at the [edit interfaces *interface-name* sonet-options] hierarchy level:

On the Working Circuit

```
[edit interfaces so-fpc/pic/port sonet-options]
aps {
   working-circuit group-name;
   authentication-key key;
   neighbor address; # Include if protect circuit is on a different routing platform
}
```

On the Protect Circuit

```
aps {
   protect-circuit group-name;
   authentication-key key;
   neighbor address; # Include if working circuit is on a different routing platform
}
```

Example: Configuring Basic APS Support

Configure Router A to be the working router and Router B to be the protect router.

On Router A (the Working Router)

```
[edit interfaces so-6/1/1 sonet-options]
aps {
   working-circuit San-Jose;
   authentication-key " $9$B2612345" ;
}
```

On Router B (the Protect Circuit)

```
[edit interfaces so-0/0/0 sonet-options]
aps {
   protect-circuit San-Jose;
   authentication-key " $9$B2612345" ;
   neighbor 192.168.1.2;# Address of Router A on the link between A and B
}
```

On a Single Platform, One Interface as the Working Circuit and Another Interface as the Protect Circuit

```
[edit interfaces so-2/1/1 sonet-options]
aps {
   working-circuit bayward;
   authentication-key blarney;
}
[edit interfaces so-3/0/2 sonet-options]
aps {
   protect-circuit bayward;
   authentication-key blarney;
```

}

Configuring Container Interfaces

JUNOS supports container interfaces for APS on SONET links. Physical interfaces and logical interfaces remain up on switchover, and their APS parameters are auto-copied from the container interface to the member links. See [Unresolved xref] for more information.

Container interfaces support the following features:

- Cisco HDLC or PPP encapsulation methods.
- Unpaired groups.
- Bidirectional APS.
- Non-container and container-based APS on the same system.
- Use of any combination of (nonchannelized) SONET interfaces installed on the same router.

To configure a container interface, you must first create the number of container devices that you require. You can create up to a maximum of 128 container interfaces per router using the device-count statement at the [edit chassis container-devices] hierarchy level. You can create more container interfaces later if required, up to 128 (total). The resulting container interfaces are designated sequentially from ci0 up to a maximum of ci127, depending on the device-count *number* specified. SONET interfaces can be assigned to any container interface ci*n*.

To configure each container interface, you must assign two SONET interfaces (so-*fpc*/*pic*/*port*) using the container-list ci*n* statement, and specify the member-interface-speed *speed* and container-options for each SONET interface.

Within each of the two SONET interfaces' container options, you must set one container-type as primary (corresponding to an APS working circuit) and the other as standby (corresponding to an APS protect circuit). For each SONET interface, you can also use the allow-configuration-override statement to allow the physical configuration of a member link to override the container configuration.

The following configuration steps are required:

1. Specify the total number of container interfaces (up to 128) to create using the device-count *number* statement at the [edit chassis container-devices] hierarchy level:

   ```
   [edit chassis container-devices]
   user@host# set device-count number
   ```

2. Configure the container interface parameters for a specified container ci*n* as follows:

   a. Specify the container interface using the numbered identifier ci*n*:

      ```
      [edit interfaces]
      ```

```
user@host# edit ci n
```

b. Specify the container interface encapsulation as cisco-hdlc or ppp:

```
[edit interfaces ci n]
user@host# set encapsulation (cisco-hdlc | ppp)
```

c. Specify the container options container-type as aps; a SONET interface is required for APS selection:

```
[edit interfaces ci n]
user@host# set container-options container-type aps
```

d. Specify the container interface member-interface type as sonet:

```
[edit interfaces ci n]
user@host# set interfaces ci n container-options member-interface-type
  sonet
```

e. Specify the container member-interface-speed *speed* to match the specified installed SONET interface links; the available values are *OC3, OC12, OC48, OC192, OC768,* or *mixed*. The member-interface-speed *speed* statement setting applies to all SONET member interfaces of the specified container ci*n*.

```
[edit interfaces ci n]
user@host# set interfaces ci n container-options member-interface-type
  sonet member-interface-speed speed
```

f. Specify the container interface's unit number, family, IP address, and mask:

```
[edit interfaces ci n]
user@host# set interfaces ci n unit number family inet address
  ip-address/mask
```

3. Configure each of the required two SONET interfaces as follows:

a. Specify the SONET interfaces and their container options; including the container-list, identified by its ci*n*.

b. Specify the container-type as *primary* (corresponding to an APS working-circuit) or *standby* (corresponding to an APS protect-circuit).

For example, setting so-0/0/0 as the primary and so-0/0/1 as the standby SONET interfaces for container interface ci0:

```
[edit]
user@host#edit interfaces so-0/0/0 # Enter config mode for interface
  so-0/0/0
[edit interfaces so-0/0/0]
user@host# set container-options container-list ci0 primary # Set so-0/0/0
  as APS primary interface
[edit interfaces so-0/0/0]
user@host# top
[edit]
user@host#edit interfaces so-0/0/1 # Enter config mode for interface
  so-0/0/1
```

```
[edit interfaces so-0/0/1]
user@host# set container-options container-list ci0 standby # Set so-0/0/1
   as APS standby interface
```

Optionally, you can set the allow-configuration-override statement to allow the physical configuration of a member link to override the container configuration:

```
[edit interfaces so-0/0/1]
user@host# set container-options container-list ci0 standby
   allow-configuration-override
```

Example Container Interface Configuration

The following is a sample container interface configuration:

```
[edit chassis]
container-devices {
   device-count 1;
}
[edit interfaces]
so-1/0/2 {
   container-options {
      container-list ci0;
      primary;
   }
}
so-1/0/3 {
   container-options {
      container-list ci0;
      standby;
   }
}
ci0 {
   encapsulation cisco-hdlc;
   container-options {
      container-type aps {
         member-interface-type sonet {
            member-interface-speed mixed;
         }
      }
   }
   unit 0 {
      family inet {
         address 192.168.11.1/24;
      }
   }
}
```

You can run the show aps command to display the APS container interface configuration, as follows:

```
user@host> show aps
Interface    Group                          Circuit    Intf state
ci0           CONTAINER_ci0                 Container  enabled, up
```

```
so-1/2/2      MEMBER_OF_ci0                 Working    enabled, up
so-1/2/3      MEMBER_OF_ci0                 Protect    disabled, up
```

Configuring Switching Between the Working and Protect Circuits

When there are multiple reasons to switch between the working and protect circuits, a priority scheme is used to decide which circuit to use. The routing platforms and the ADM might automatically switch traffic between the working and protect circuits because of circuit and routing platform failures. You can also choose to switch traffic manually between the working and protect circuits.

When an ATM2 PIC is configured for APS, and the protect circuit comes online for the first time, there are no open VCs and the PIC discards the input traffic received on the protect circuit. The show interface extensive or show monitor interface traffic commands display the statistics as zero since the PIC drops the packets at the VC.

When the APS switches from the working circuit to the protect circuit, VCs are created on the protect circuit to accept traffic. However, the VCs on the working circuit remain open to support any future APS switches even though the interface is down or disabled. The input traffic received on the working circuit (current backup) is accepted by the PIC but discarded in the PFE. The show interface extensive or show monitor interface traffic commands displays live statistics for the traffic since it is accepted by the PIC.

When APS switches from the protect circuit to the working circuit again, the VCs on the protect circuit remain open to support a future APS switch even though the interface is down or disabled. The input traffic received on the current backup protect circuit is accepted by the PIC but discarded in the PFE. The show interface extensive or the show monitor interface traffic command displays live statistics for this traffic since it is accepted by the PIC.

There are three priority levels of manual configuration, listed here in order from lowest to highest priority:

- Request (also known as manual switch)—Overridden by signal failures, signal degradations, or any higher-priority reasons.
- Force (also known as forced switch)—Overrides manual switches, signal failures, and signal degradation.
- Lockout (also known as lockout of protection)—Do not switch between the working and protect circuits.

NOTE: Do not use the disable statement at the [edit interfaces *interface-name* aps] hierarchy level to switch between interface working and protect circuits; it can cause loss of traffic on the disabled interface. Use only the request statement or the force statement at the [edit interfaces interface-name aps] hierarchy level to modify interface status.

A routing platform failure is considered to be equivalent to a signal failure on a circuit.

To perform a manual switch, include the request statement at the [edit interfaces *interface-name* sonet-options aps] hierarchy level. This statement is honored only if there are no higher-priority reasons to switch.

```
[edit interfaces so-fpc/pic/port sonet-options aps]
request (protect | working);
```

When the working circuit is operating in nonrevertive mode, use the request working statement to switch the circuit manually to being the working circuit or to override the revert timer.

To perform a forced switch, include the force statement at the [edit interfaces *interface-name* sonet-options aps] hierarchy level. This statement is honored only if there are no higher-priority reasons to switch. This configuration can be overridden by a signal failure on the protect circuit, thus causing a switch to the working circuit.

```
[edit interfaces so-fpc/pic/port sonet-options aps]
force (protect | working);
```

To configure a lockout of protection, forcing the use of the working circuit and locking out the protect circuit regardless of anything else, include the lockout statement at the [edit interfaces *interface-name* sonet-options aps] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options aps]
lockout;
```

Configuring Revertive Mode

By default, APS is nonrevertive, which means that if the protect circuit becomes active, traffic is not switched back to the working circuit unless the protect circuit fails or you manually configure a switch to the working circuit. In revertive mode, traffic is automatically switched back to the working circuit.

You should configure the ADM and routing platforms consistently with regard to revertive or nonrevertive mode.

To configure revertive mode, include the revert-time statement, specifying the amount of time to wait after the working circuit has again become functional before making the working circuit active again:

```
[edit interfaces so-fpc/pic/port sonet-options aps]
revert-time seconds;
```

If you are using nonrevertive APS, you can use the request working statement to switch the circuit manually to being the working circuit or to override the revert timer (configured with the revert-time statement).

Configuring Unidirectional Switching Mode Support

You can configure interoperation with SONET/SDH Line Terminating Equipment (LTE) that is provisioned for unidirectional linear APS in 1 + 1 architecture on the following interfaces:

- Unchannelized OC3, OC12, and OC48 SONET/SDH interfaces on T-series platforms
- SONET/SDH interfaces on the M40e routing platform
- ATM over SONET interfaces

By default, APS supports only SONET/SDH LTE that is provisioned for bidirectional mode.

In bidirectional switching mode, the working interface switches to the protect interface for both receipt and transmission of data, regardless of whether the signal failure is in the transmit or receive direction.

In true unidirectional mode, the working interface switches to the protect interface only for the direction in which signal failure occurs; for example, if there is a signal failure in the transmit direction, the working interface switches over to the protect interface for transmission but not receipt of data. When the protect interface operates in unidirectional mode, the working and protect interfaces must cooperate to operate the transmit and receive interfaces in a bidirectional fashion.

The JUNOS software does not support true unidirectional mode. Instead the software supports interoperation with SONET/SDH LTE provisioned for unidirectional switching. This means that the SONET/SDH LTE on the router receives and transmits on one interface, even when you configure unidirectional support. The JUNOS implementation of unidirectional mode support allows the router to do the following:

- Accept a unidirectional mode as valid
- Trigger the peer (ADM) selector to switch receive from working interface to protect interface or the reverse
- Not send reverse requests to the far end (ADM)

To configure unidirectional mode support, include the switching-mode unidirectional statement, at the [edit interfaces *interface-name* sonet-options aps] hierarchy level:

```
[edit interfaces interface-name sonet-options aps]
switching-mode unidirectional;
```

NOTE: On interfaces with unidirectional APS support configured, revertive mode and load sharing between circuits are not supported.

To restore the default behavior, include the switching-mode bidirectional statement, at the [edit interfaces *interface-name* sonet-options aps] hierarchy level:

```
[edit interfaces interface-name sonet-options aps]
switching-mode bidirectional;
```

Configuring APS Timers

The protect and working routers periodically send packets to their neighbors to advertise that they are operational. By default, these advertisement packets are sent

every 1000 milliseconds. A routing platform considers its neighbor to be operational for a period, called the hold time, that is, by default, three times the advertisement interval. If the protect router does not receive an advertisement packet from the working router within the hold time configured on the protect router, the protect router assumes that the working router has failed and becomes active.

APS is symmetric; either side of a circuit can time out the other side (for example, when detecting a crash of the other). Under normal circumstances, the failure of the protect router does not cause any changes because the traffic is already moving on the working router. However, if you had configured request protect and the protect router failed, the working router would enable its interface.

To modify the advertisement interval, include the advertise-interval at the [edit interfaces *interface-name* sonet-options aps] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options aps]
[Unresolved xref] milliseconds;
```

To modify the hold time, include the hold-time at the [edit interfaces *interface-name* sonet-options aps] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options aps]
[Unresolved xref] milliseconds;
```

The advertisement intervals and hold times on the protect and working routers can be different.

Configuring Link PIC Redundancy

Link state replication, also called interface preservation, is an addition to the SONET Automatic Protection Switching (APS) functionality that helps promote redundancy of link PICs used in LSQ configurations, providing MLPPP link redundancy at the port level.

Link state replication provides the ability to add two sets of links, one from the active SONET PIC and the other from the standby SONET PIC, to the same bundle. If the active SONET PIC fails, links from the standby PIC are used without link renegotiation. All the negotiated state is replicated from the active links to the standby links to prevent link renegotiation. For more information about LSQ configurations, see the *JUNOS Services Interfaces Configuration Guide*.

To configure link state replication, include the preserve-interface statement at the [edit interfaces *interface-name* sonet-options aps] hierarchy level on the interfaces on both PICs:

```
preserve-interface;
```

APS functionality must be available on the SONET PICs and the interface configurations must be identical on both ends of the link. Any configuration mismatch causes the commit operation to fail.

This feature is supported with SONET APS and the following link PICs:

- Channelized OC3 IQ and IQE PICs

- Channelized OC12 IQ and IQE PICs
- Channelized STM1 IQ and IQE PICs

Link state replication supports MLPPP and PPP over Frame Relay (frame-relay-ppp) encapsulation, and fully supports GRES.

Example: Configuring Link PIC Redundancy

Configure link state replication configuration between the ports coc3-1/0/0 and coc3-2/0/0.

```
interfaces {
    coc3-1/0/0 {
        sonet-options {
            aps {
                preserve-interface;
                working-circuit aps-group-1;
            }
        }
    }
    coc3-2/0/0 {
        sonet-options {
            aps {
                preserve-interface;
                working-circuit aps-group-1;
            }
        }
    }
}
```

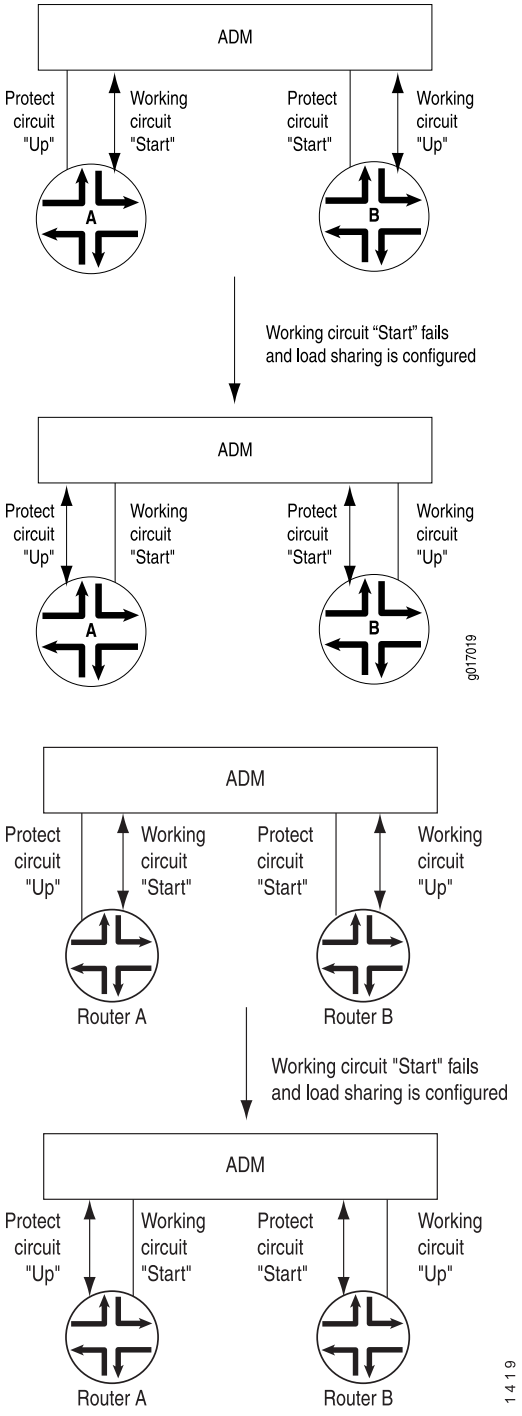
Configuring APS Load Sharing Between Circuit Pairs

When two routing platforms are connected to a single ADM, you can have them back up each other on two different pairs of circuits. This arrangement provides load balancing between the routing platforms if one of the working circuits fails.

Figure 2 illustrates load sharing between circuits on two routing platforms. Router A has a working circuit “Start” and a protect circuit “Up,” and Router B has a working circuit “Up” and a protect circuit “Start.” Under normal circumstances, Router A carries the “Start” circuit traffic and Router B carries the “Up” circuit traffic. If the working circuit “Start” were to fail, Router B would end up carrying all the traffic for both the “Start” and “Up” circuits.

To balance the load between the circuits, you pair the two circuits. In this case, you pair the “Start” and “Up” circuits. Then, if the working circuit “Start” fails, the two routing platforms automatically switch the “Up” traffic from the working to the protect circuit so that each routing platform is still carrying only one circuit’s worth of traffic. That is, the working circuit on Router A would be “Up” and the working circuit on Router B would be “Start.”

Figure 2: APS Load Sharing Between Circuit Pairs



To configure load sharing between two working–protect circuit pairs, include the paired-group statement when configuring one of the circuits on one of the routing platforms. In this statement, the *group-name* is the name of the group you assigned to one of the circuits with the working-circuit and protect-circuit statements. The JUNOS

software automatically configures the remainder of the load-sharing setup based on the group name.

```
[edit interfaces so-fpc/pic/port sonet-options aps]
paired-group group-name;
```

Example: Configuring APS Load Sharing Between Circuit Pairs

Configure APS load sharing to match the configuration shown in Figure 2:

On Router A

```
[edit interfaces so-7/0/0 sonet-options aps]
user@host# set working-circuit start
[edit interfaces so-7/0/0 sonet-options aps]
user@host# set authentication-key linsey
[edit interfaces so-7/0/0 sonet-options aps]
user@host# set paired-group "Router A-Router B"
...
[edit interfaces so-0/0/0 sonet-options aps]
user@host# set protect-circuit up
[edit interfaces so-0/0/0 sonet-options aps]
user@host# set authentication-key woolsey
[edit interfaces so-0/0/0 sonet-options aps]
user@host# set paired-group "Router A-Router B"
```

On Router B

```
[edit interfaces so-1/0/0 sonet-options aps]
user@host# set working-circuit up
[edit interfaces so-1/0/0 sonet-options aps]
user@host# set authentication-key woolsey
[edit interfaces so-1/0/0 sonet-options aps]
user@host# set paired-group "Router A-Router B"
...
[edit interfaces so-6/0/0 sonet-options aps]
user@host# set protect-circuit start
[edit interfaces so-6/0/0 sonet-options aps]
user@host# set authentication-key linsey
[edit interfaces so-6/0/0 sonet-options aps]
user@host# set paired-group "Router A-Router B"
```

Configuring SONET Options for 10-Gigabit Ethernet Interfaces

The 10-Gigabit Ethernet IQ2 and IQ2-E PIC is supported on the M120, M320, and T-series routing platforms. The PIC provides one external interface running at 10 Gbps. The interface operates in either LAN PHY or WAN PHY mode. When the external interface is running in WAN PHY mode, it uses the WIS sublayer to transport 10-Gigabit Ethernet frames in an OC192c SONET payload, and can interoperate with SONET section or line level repeaters. This creates an advantage when the interface is used for long-distance, point-to-point 10-Gigabit Ethernet links.

When the external interface is running in WAN PHY mode, you can configure specific physical SONET options. To configure SONET options, include the loopback, mpls, path-trace, and trigger statements at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces]
xe-0/0/0 {
    sonet-options {
        [Unresolved xref] (local | remote);
        mpls {
            pop-all-labels {
                required-depth number;
            }
        }
        path-trace trace-string;
        trigger {
            defect ignore {
                defect [Unresolved xref] up milliseconds down milliseconds;
            }
        }
    }
}
```

For information about using the **loopback** statement, see “Configuring SONET/SDH Loopback Capability” on page 8. For information about using the **mpls** statement, see Enabling Passive Monitoring on SONET/SDH Interfaces. For information about using the **path-trace** statement, see “Configuring the SONET/SDH Path Trace Identifier” on page 10. For information about using the **trigger** statement, see “Configuring SONET/SDH Defect Triggers to Be Ignored” on page 11.