Chapter 2

Configuring the M120 Internet Router

This chapter contains configuration information for M120 router software features that are not covered in the JUNOS software documentation for other existing routers or routing platforms. This chapter contains the following sections:

Configuring FPC to FEB Connectivity

The M120 router supports six FEBs and six FPCs. The supported FPCs include:

- Two compact FPCs:
  - OC192 compact FPC (supported only on the D4 chip-based compact FPC)
  - 10-Gigabit Ethernet compact FPC
- Up to four Type 1, Type 2, or Type 3 FPCs

On an M120 router, you can map a connection between any Flexible PIC Concentrator (FPC) and any Forwarding Engine Board (FEB). This allows you to configure resources for a chassis that contains empty slots, supporting configurations where the FPC and FEB pairs are not in slot order. You do not have to populate every empty slot position, but you must configure an FEB for every FPC.

If you do not map a connection between an FPC and an FEB, you must explicitly configure the FPC not to connect to an FEB. To do this, include the none option at the [edit chassis fpc-feb-connectivity fpc *number* feb] hierarchy level.

A maximum of two Type 1 FPCs and one Type 2 or Type 3 compact FPC can be mapped per FEB.

The following restrictions apply when configuring FPC and FEB connectivity:

- When an FPC is configured not to connect to any FEB, interfaces on that FPC are not created.
- If a PIC comes online, but the FEB to which the FPC is configured to connect to is not online, the physical interfaces for the PIC will not be created. For example, PIC1 on FPC 2 is coming online. The configuration specifies that FPC 2 connects to FEB 3. If FEB 3 is not online at the time PIC 1 comes online, the physical interfaces corresponding to PIC1 on FPC2 are not created. If FEB 3 subsequently comes online, the physical interfaces are created.
- FPCs and FEBs can reboot following a change in the FPC and FEB connectivity configuration. If an FPC connects to a different FEB as a result of the configuration change, it is rebooted following the commit. As a result of the reboot, interfaces on the FPC will be deleted.
- If an FEB connects to a different FPC or set of FPCs after a connectivity configuration change, it is rebooted. The exception is if the same FPC is connected to the FEB in both the old and the new configuration.
- Graceful Routing Engine switchover (GRES) is not supported.
- If connectivity is not mapped between an FPC and an FEB, connectivity is automatically assigned in the following order: FPC 1 to FEB 1, FPC 2 to FEB 2, and so on.

To configure a connection between an FPC and an FEB, include the fpc-feb-connectivity statement at the [edit chassis] hierarchy level:

```
[edit chassis]
fpc-feb-connectivity {
    fpc number feb (number | none);
    }
}
```

For fpc *number*, enter a value from 0 through 5. For feb *number*, enter a value from 0 through 5 or none. The none option disconnects the FPC from the FEB.

Example: Configuring FPC to FEB Connectivity

The following configuration assumes that FPC 3 also connects to FEB 3:

```
[edit chassis]
fpc-feb-connectivity {
    fpc 2 feb 3;
    }
}
```

However, this configuration results in a mismatch between the FPC type and the FEB type. For example, FPC 3 is not a Type-1 FPC. You can use the **fpc-feb-connectivity** statement to explicitly disconnect FPC 3. To do this, include the **none** option at the [**edit chassis fpc-feb-connectivity fpc *number* feb**] hierarchy level:

```
[edit chassis]
fpc-feb-connectivity {
      fpc 2 feb 3;
      fpc 3 feb none;
      }
}
```

Configuring LAN PHY or WAN PHY Mode

The 10-Gigabit Ethernet Compact FPC on the M120 router provides one external interface running at 10 Gbps that operates in one of two modes:

- 10GBASE-R, LAN physical layer device (LAN PHY)
- 10GBASE-W, WAN physical layer device (WAN PHY)

The 10-Gigabit Ethernet Compact FPC uses interface type xe-*fpc*/*pic*/*port*. On this single-port PIC, the port number is always zero.

When the external interface is running in LAN PHY mode, it bypasses the WAN Interface sublayer (WIS) to directly stream block-encoded Ethernet frames on a 10-Gigabit Ethernet serial interface. 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.

When the external interface is running in WAN PHY mode, some SONET options are supported. For more information, see “Configuring SONET Options for 10-Gigabit Ethernet Interfaces” on page 20.The xe-*fpc*/*pic*/*port* interface inherits all the configuration commands that are used for Gigabit Ethernet (ge-*fpc*/*pic*/*port*) interfaces.

To configure LAN PHY or WAN PHY operating mode, include the **lan-phy** or **wan-phy** statement at the [**edit interfaces xe-*fpc*/*pic*/0 framing**] hierarchy level:

```
[edit interfaces]
xe-0/0/0 {
      framing {
         (lan-phy | wan-phy);
      }
}
```

To display interface information, use the operational mode command **show interfaces xe-*fpc*/*pic*/*port* *extensive***. For more information about this command, see “show interfaces (10-Gigabit Ethernet)” on page 43.

Configuring SONET Options for 10-Gigabit Ethernet Interfaces

The M120 router supports the 10-Gigabit Ethernet Compact FPC. This FPC 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.

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

```
[edit interfaces]
xe-0/0/0 {
    sonet-options {
        loopback (local | remote);
        path-trace trace-string;
        trigger {
            defect (ignore | hold-time up milliseconds down milliseconds);
            }
        }
    }
}
```

For information about using the loopback statement, see “Configuring SONET Loopback Capability” on page 20. For information about using the path-trace statement, see “Configuring the SONET Path Trace Identifier” on page 20. For information about using the trigger statement, see “Configuring SONET Defect Triggers” on page 21.

Configuring SONET Loopback Capability

To configure loopback capability on a 10-Gigabit Ethernet interface in WAN PHY mode, include the loopback statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces xe-fpc/pic/port sonet-options]
loopback (local | remote);
```

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

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

Configuring the SONET Path Trace Identifier

The SONET 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 interface names for the path trace identifier. The M120 router uses a 16-byte trace. The trace contains 15 configurable bytes and 1 contain cyclic redundancy check (CRC) byte used as an identifier.

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 Defect Triggers

A trigger is a defect alarm that causes a physical interface to be marked down. By default, all defect alarms are honored with no hold time. For 10-Gigabit Ethernet interfaces operating in WAN PHY mode, 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 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
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 affect whether the interface is marked down or up.

Configuring SONET 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

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]
hold-time up milliseconds down milliseconds;
```

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

Example: Configuring SONET Defects

The following example ignores the alarm defect LOF and applies a hold time on the alarm defect LOS on the xe-1/0/0 interface:

```
[edit interfaces xe-1/0/0 sonet-options trigger]
    lof ignore;
    los hold-time up 20 down 10;
}
```

Configuring the Clock Source

For 10-Gigabit Ethernet interfaces operating in WAN PHY mode, you can configure the source of the transmit clock on each interface. The source can be internal (also called line timing or normal) or external (also called loop timing). The default source is internal, which means that each interface uses the routing platform's internal stratum 3 clock.

To configure loop timing on an interface, include the clocking external statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
clocking external;
```

To explicitly configure line timing on an interface, include the clocking internal statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
clocking internal;
```

Configuring Single-Rate Tricolor Marking

Single-rate tricolor marking (TCM) (as defined in RFC 2697, *A Single Rate Three Color Marker*) extends the functionality of class-of-service (CoS) traffic policing by providing ingress policing of a service, where only the length—not the peak rate—determines service eligibility.

Single-rate TCM is automatically enabled. You do not have to include the tri-color statement at the [edit class-of-service] hierarchy level (as you do when configuring two-rate TCM). TCM functionality is always enabled.

To configure single-rate TCM, include the following statements (shown in bold) at the [edit class-of-service] hierarchy level of the configuration:

```
[edit class-of-service]
classifiers {
    (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) classifier-name {
        import (classifier-name | default);
        forwarding-class class-name {
            loss-priority (low | medium-low | medium-high | high) {
                code-points [ aliases ] [ 6-bit-patterns ];
            }
        }
    }
}
rewrite-rules {
    (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) rewrite-name {
        import (rewrite-name | default);
        forwarding-class class-name {
            loss-priority (low | medium-low | medium-high | high)
                        code-point (alias | bits);
        }
    }
}
schedulers {
    scheduler-name {
        drop-profile-map loss-priority (any | low | medium-low | medium-high | high)
                        protocol any drop-profile profile-name;
    }
}
[edit firewall]
policer name {
    then loss-priority (low | medium-low | medium-high | high);
```

```
    }
    three-color-policer name {
        single-rate {
            (color-aware | color-blind);
            committed-information-rate bps;
            committed-burst-size bytes;
            excess-burst-size bps;
        }
    }

    filter filter-name {
        <family family> {
            term rule-name {
                then {
                    three-color-policer {
                        single-rate policer-name;
                    }
                }
            }
        }
    }
```

This section discusses the following topics:

- How Single-Rate Tricolor Marking Works on page 24
- Configuring a Single-Rate Tricolor Marking Policer on page 25
- Applying a Tricolor Marking Policer to a Firewall Filter on page 26

How Single-Rate Tricolor Marking Works

With single-rate TCM, packets are marked based on a single rate and two burst sizes. The single-rate TCM meters an IP packet stream and marks its packets either green, yellow, or red. The single-rate TCM is useful, for example, for ingress policing of a service, where only the length, not the peak rate, of the burst determines service eligibility.

Marking is based on a committed information rate (CIR) and two associated burst sizes, a committed burst size (CBS) and an excess burst size (EBS). The CIR is the guaranteed bandwidth under normal line conditions and the average rate up to which packets are marked green. The CBS is the maximum number of bytes allowed for incoming packets to burst above the CIR and still be marked green. The EBS is the maximum number of bytes allowed for incoming packets to burst above the CIR and still be marked yellow. When the EBS is exceeded, packets are marked red, as shown in Table 6.

Table 6: Single-Rate TCM Color-to-PLP Mapping

Color	PLP	Assigned If...
Red	high	Packet rate exceeds both the CBS and the EBS.
Yellow	medium-high	Packet rate exceeds the CBS, but not the EBS.
Green	low	Packet rate does not exceed the CBS.

With color-blind policing, all packets are evaluated by the CBS. If a packet exceeds the CBS, it is evaluated by the EBS.

With color-aware policing, the metering treatment that the packet receives varies by the classification. Metering can increase a packet's assigned PLP, but cannot decrease it.

Packets belonging to the green class have already been marked by a classifier with low PLP. The marking policer can potentially leave the packet's PLP unchanged or increase the PLP to medium-high or high. Therefore, these packets are metered against both the CBS and the EBS.

Packets belonging to the yellow class have already been marked by a classifier with medium-high PLP. The marking policer can potentially leave the packet's PLP unchanged or increase the PLP to high. Therefore, these packets are metered against the EBS only.

Packets belonging to the red class have already been marked by a classifier with high PLP. The marking policer can only leave the packet's PLP unchanged. Therefore, these packets are not metered against the CBS or the EBS.

The configuration for single-rate TCM is similar to the configuration for two-rate TCM.

Configuring a Single-Rate Tricolor Marking Policer

A tricolor marking policer polices traffic on the basis of metering rates, including the CIR, the associated burst sizes, and any policing actions configured for the traffic. To configure a tricolor marking policer, include the following statements at the [edit firewall] hierarchy level:

```
[edit firewall]
three-color-policer name {
    single-rate {
        (color-aware | color-blind);
        committed-information-rate bps;
        committed-burst-size bytes;
        excess-burst-size bytes;
    }
}
```

You can specify the values for *bps* and *bytes* either as complete decimal numbers or as decimal numbers followed by the abbreviation k (1000), m (1,000,000), or g (1,000,000,000).

The single-rate tricolor policer implicitly marks packets as loss priority low, medium-high, or high. For more information, see "How Single-Rate Tricolor Marking Works" on page 24.

Table 7 describes the configurable statements.

Table 7: Tricolor Marking Policer Statements

Statement	Meaning	Configurable Values
single-rate	Marking is based on the CIR, CBS, and EBS.	None
color-aware	Metering depends on the packet's preclassification. Metering can increase a packet's assigned PLP, but cannot decrease it. For more information, see "How Single-Rate Tricolor Marking Works" on page 24.	None
color-blind	All packets are evaluated by the CIR. If a packet exceeds the CIR, it is evaluated.	None
committed-information-rate	Guaranteed bandwidth under normal line conditions, and the average rate up to which packets are marked green.	32,000 through 40,000,000,000 bps
committed-burst-size	Maximum number of bytes allowed for incoming packets to burst above the CIR, but still be marked green.	1500 through 100,000,000,000 bytes
excess-burst-size	Minimum number of bytes allowed for incoming packets to exceed the CBS. These packets are marked red.	1500 through 100,000,000,000 bytes

Applying a Tricolor Marking Policer to a Firewall Filter

To rate-limit traffic by attaching a tricolor marking policer to a firewall filter, include the three-color-policer statement:

```
three-color-policer single-rate name;
```

You can include this statement at the following hierarchy levels:

- [edit firewall family *family* filter *filter-name* term *rule-name* then]
- [edit firewall filter *filter-name* term *rule-name* then]

In the family statement, the protocol family can be any, ccc, inet, inet6, mpls, or vpls.

Example: Applying a Tricolor Marking Policer to a Firewall Filter

Apply the srtcm1 policer to a firewall filter:

```
firewall {
    three-color-policer srtcm1 {  # Configure the srtcm1 policer.
        single-rate {
            color-blind;
            committed-information-rate 1048576;
            committed-burst-size 65536;
            excess-burst-size 131072;
        }
    }
    filter fil {   # Configure the fil firewall filter, which attaches the srtcm1 policer.
        term default {
            then {
                three-color-policer {
                    single-rate srtcm1;
                }
            }
```

```
            }
        }
    }
```

For more information about applying policers to firewall filters, see the *JUNOS Policy Framework Configuration Guide*.

Using New MIB Objects

This chapter contains new chassis Management Information Base (MIB) objects that are available for the M120 router. The new MIB objects include:

```
    jnxProductLineM120      OBJECT IDENTIFIER ::= { jnxProductLine      18 }
    jnxProductNameM120      OBJECT IDENTIFIER ::= { jnxProductName      18 }
    jnxProductModelM120     OBJECT IDENTIFIER ::= { jnxProductModel     18 }
    jnxProductVariationM120 OBJECT IDENTIFIER ::= { jnxProductVariation 18 }
    jnxChassisM120          OBJECT IDENTIFIER ::= { jnxChassis          18 }

    jnxSlotM120             OBJECT IDENTIFIER ::= { jnxSlot             18 }
      jnxM120SlotFPC        OBJECT IDENTIFIER ::= { jnxSlotM120 1  }
      jnxM120SlotFEB        OBJECT IDENTIFIER ::= { jnxSlotM120 2  }
      jnxM120SlotHM         OBJECT IDENTIFIER ::= { jnxSlotM120 3  }
      jnxM120SlotPower      OBJECT IDENTIFIER ::= { jnxSlotM120 4  }
      jnxM120SlotFan        OBJECT IDENTIFIER ::= { jnxSlotM120 5  }
      jnxM120SlotCB         OBJECT IDENTIFIER ::= { jnxSlotM120 6  }
      jnxM120SlotFPB        OBJECT IDENTIFIER ::= { jnxSlotM120 7  }

jnxMediaCardSpaceM120      OBJECT IDENTIFIER ::= { jnxMediaCardSpace    18 }
  jnxM120MediaCardSpacePIC OBJECT IDENTIFIER ::= { jnxMediaCardSpaceM120 1 }

jnxMidplaneM120         OBJECT IDENTIFIER ::= { jnxBackplane       18 }

jnxModuleM120           OBJECT IDENTIFIER ::= { jnxModule          18 }
  jnxM120FEB            OBJECT IDENTIFIER ::= { jnxModuleM120   1 }
```