Chapter 10

Configuring Adaptive Services Interfaces

You can configure basic properties of the adaptive services interface on a global level, including default values for system logging, timeout, and intrusion detection properties. To configure properties for the entire interface, include statements at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
unit logical-unit-number {
    clear-dont-fragment-bit;
    dial-options {
      l2tp-interface-id name;
      (dedicated | shared);
    }
    encapsulation type;
    family inet {
      address address {
      …
      }
      filter {
        group filter-group-number;
        input filter-name;
        output filter-name;
      }
      policer {
        input policer-template-name;
        output policer-template-name;
      }
      service {
        input {
          [ service-set service-set-name <service-filter filter-name> ];
          post-service-filter filter-name;
        }
        output {
          [ service-set service-set-name <service-filter filter-name> ];
        }
      }
    }
    link-layer-overhead percent;
    service-domain (inside | outside);
}
```

```
services-options {
    inactivity-timeout seconds;
    open-timeout seconds;
    syslog {
      host hostname {
        facility-override facility-name;
        log-prefix prefix-number;
        services priority-level;
      }
    }
}
```

This chapter contains the following sections:

Enabling AS PIC Service Packages on page 158

Configuring Service Interface Properties on page 161

Configuring Voice Services on page 164

Applying Filters and Services to an Interface on page 169

Configuring L2TP Dialup Properties on page 170

Example: Configuring a Service Interface on page 171

For detailed information about configuring the Adaptive Services (AS) Physical Interface Card (PIC), see the *JUNOS Services Interf aces Configur ation Guide*.

Enabling AS PIC Service Packages

For AS PICs and the internal Adaptive Services Module (ASM) in the M7i platform, there are two service packages: Layer 2 and Layer 3. Both service packages are supported on all AS PICs, but you can enable only one service package per PIC. However, on a single routing platform, you can enable both service packages by installing two or more AS PICs on the platform.

You enable service packages per PIC, not per port. For example, if you configure the Layer 2 service package, the entire AS PIC uses the configured package. To enable a service package, include the service-package statement at the [edit chassis fpc *slot-number* pic *pic-number* adaptive-services] hierarchy level, and specify layer-2 or layer-3:

```
[edit chassis fpc slot-number pic pic-number adaptive-services]
service-package (layer-2 | layer-3);
```

After you commit a change in the AS PIC service package, the AS PIC is taken offline and then brought back online immediately. You do not need to manually take the PIC offline and online.

NOTE: Changing the AS PIC service package causes all state information associated with the previous service package to be lost. You should change the AS PIC service package only when there is no active traffic going to the AS PIC.

The services supported in each package differ by PIC and platform type. Table 14 on page 159 lists the services supported within each service package for each PIC and platform.

In Table 14 on page 159, the term *standalone* means the service cannot be used with other AS PIC services on the same PIC.

On the AS PIC, *link services* support includes JUNOS software CoS components, link fragment interleaving (LFI) (FRF.12), MLFR UNI NNI (FRF.16), and MLPPP (RFC 1990). For J-series Services Routers only, link services support also includes MLFR end-to-end (FRF.15). For more information, see “Configuring Link Services IQ Interfaces” on page 449.

For detailed information about Layer 3 services, see the *JUNOS Softw are Services Interfaces Configur ation Guide* and the *JUNOS Softw are Feature Guide*.

Table 14: AS PIC Services by Service Package, PIC, and Platform

AS PIC Services	ASM	AS / AS2 PIC	AS / AS2 PIC	AS2 PIC
Layer 2 Service Package	**M7i**	**M7i**	**M10i, M20, and M40e**	**M320, T320, and T640**
Link Services:				
Link services (standalone)	Yes	Yes	Yes	No
Tunnel Services:				
GRE (gr-*fpc*/*pic*/*port*)	Yes	Yes	Yes	Yes
GRE fragmentation (clear-dont-fragment-bit)	No	No	No	No
GRE key	No	No	No	No
IP-IP tunnels (ip-*fpc*/*pic*/*port*)	Yes	Yes	Yes	Yes
Logical tunnels (lt-*fpc*/*pic*/*port*)	Yes	Yes	No	No
Multicast tunnels (mt-*fpc*/*pic*/*port*)	Yes	Yes	Yes	Yes
PIM de-encapsulation (pd-*fpc*/*pic*/*port*)	Yes	Yes	Yes	Yes
PIM encapsulation (pe-*fpc*/*pic*/*port*)	Yes	Yes	Yes	Yes
Virtual tunnels (vt-*fpc*/*pic*/*port*)	Yes	Yes	Yes	Yes
Voice Services:				
CRTP (standalone)	Yes	Yes	Yes	No
CRTP and LFI (standalone)	Yes	Yes	Yes	No

AS PIC Services	ASM	AS/AS2 PIC	AS/AS2 PIC	AS2 PIC
Layer 3 Service Package				
Security Services:				
Stateful firewall	Yes	Yes	Yes	Yes
NAT	Yes	Yes	Yes	Yes
Intrusion detection system (IDS)	Yes	Yes	Yes	Yes
IPSec	Yes	Yes	Yes	Yes
Accounting Services:				
J-Flow	Yes	Yes	Yes	Yes
Tunnel Services:				
GRE (gr-*fpc*/*pic*/*port*)	Yes	Yes	Yes	Yes
GRE fragmentation (clear-dont-fragment-bit)	Yes	Yes	Yes	No
GRE key	Yes	Yes	Yes	Yes
IP-IP tunnels (ip-*fpc*/*pic*/*port*)	Yes	Yes	Yes	Yes
Logical tunnels (lt-*fpc*/*pic*/*port*)	Yes	Yes	No	No
Multicast tunnels (mt-*fpc*/*pic*/*port*)	Yes	Yes	Yes	Yes
PIM de-encapsulation (pd-*fpc*/*pic*/*port*)	Yes	Yes	Yes	Yes
PIM encapsulation (pe-*fpc*/*pic*/*port*)	Yes	Yes	Yes	Yes
Virtual tunnels (vt-*fpc*/*pic*/*port*)	Yes	Yes	Yes	Yes
LNS Services:				
L2TP LNS	Yes	Yes	No	No
Voice Services:				
CRTP (standalone)	Yes	Yes	Yes	No
CRTP and LFI (standalone)	Yes	Yes	Yes	No

Configuring Service Interface Properties

This section describes the following tasks for configuring service sets:

Configuring the Interface Address and Domain on page 161

Configuring Default Timeout Settings on page 162

Configuring Default System Log Properties on page 162

Configuring the Interface Address and Domain

Just as you do for other network interfaces, configure an IP address for a service interface by including the address statement:

```
address address {
...
}
```

You can include this statement at the following hierarchy levels:

[edit interfaces *interface-name* unit *logical-unit-number* family inet]

[edit logical-routers *logical-router-name* interfaces *interface-name* unit *logical-unit-number* family inet]

Assign an IP address to the interface by configuring the *address* value. The AS PIC supports only IPv4 addresses configured using the family inet statement.

For information on other addressing properties you can configure that are not specific to service interfaces, see “Configuring the Interface Address” on page 112.

The service-domain statement specifies whether the interface is used within the network or to communicate with remote devices. The software uses this setting to determine which default stateful firewall rules to apply, and the default direction for service rules. To configure, include the service-domain statement:

```
service-domain (inside | outside);
```

You can include this statement at the following hierarchy levels:

[edit interfaces *interface-name* unit *logical-unit-number*]

[edit logical-routers *logical-router-name* interfaces *interface-name* unit *logical-unit-number*]

Configuring Default Timeout Settings

You can specify global default settings for certain timers that apply to the entire interface. There are two statements of this type:

inactivity-timeout—Sets the inactivity timeout period for established flows, after which they are no longer valid.

open-timeout—Sets the timeout period for Transmission Control Protocol (TCP) session establishment, for use with syn-cookie defenses against network intrusion.

By default, the inactivity timeout is 30 seconds. To configure a setting for the inactivity timeout period, include the inactivity-timeout statement at the [edit interfaces *interface-name* services-options] hierarchy level:

```
[edit interfaces interface-name services-options]
inactivity-timeout seconds;
```

The range of possible values is 4 through 86,400 seconds. Any value you configure in the application protocol definition at the [edit applications] hierarchy level overrides the value specified here.

To configure a setting for the TCP session establishment timeout period, include the open-timeout statement at the [edit interfaces *interface-name* services-options] hierarchy level:

```
[edit interfaces interface-name services-options]
open-timeout seconds;
```

The default value is 30 seconds. Any value you configure in the IDS service definition at the [edit services ids] hierarchy level overrides the value specified here.

Configuring Default System Log Properties

You specify properties that control how system log messages are generated for the interface as a whole. If you configure different values for the same properties at the [edit services service-set *service-set-name*] hierarchy level, the service-set values override the values configured for the interface.

To configure interface-wide default system logging values, include the syslog statement at the [edit interfaces *interface-name* services-options] hierarchy level:

```
[edit interfaces interface-name services-options]
syslog {
    host hostname {
        facility-override facility-name;
        log-prefix prefix-number;
        services priority-level;
    }
}
```

Configure the host statement with a hostname that specifies the system log target server. The hostname local directs system log messages to the Routing Engine.

You can configure one or more facilities with a specified priority level. The supported facilities are any, authorization, change-log, conflict-log, cron, daemon, firewall, interactive-commands, kernel, pfe, and user. The valid priority settings are shown in Table 15.

Table 15: System Log Priority Level Settings

Priority Level	Description
alert	Conditions that should be corrected immediately.
any	Matches any level.
critical	Critical conditions.
emergency	Panic conditions.
error	Error conditions.
info	Informational messages.
notice	Conditions that require special handling.
warning	Warning messages.

To use one particular facility code for all logging to the specified system log host, include the facility-override statement at the [edit interfaces *interface-name* services-options syslog host *hostname*] hierarchy level:

```
[edit interfaces interface-name services-options syslog host hostname]
facility-override facility-name;
```

To specify an address prefix for all logging to this system log host, include the log-prefix statement at the [edit interfaces *interface-name* services-options syslog host *hostname*] hierarchy level:

```
[edit interfaces interface-name services-options syslog host hostname]
log-prefix prefix-number;
```

Configuring Voice Services

The AS PIC supports the compressed Real-time Transport Protocol (RTP) on the interface types lsq-*fpc*/*pic*/*port* and vsp-*fpc*/*pic*/*port*. This enables voice-over-IP (VoIP) traffic to use low-speed links more effectively, by compressing the 40-byte IP/User Datagram Protocol (UDP)/RTP header down to 2 to 4 bytes in most cases.

Voice services do not require a separate service rules configuration.

Voice services interfaces use a bundle configuration. For more information, see “Configuring Link Services and Multilink Interfaces” on page 409 and “Configuring Link Services IQ Interfaces” on page 449. To configure voice services interface properties, include the following statements at the [edit interfaces] hierarchy level:

```
[edit interfaces]
interface-name {
    encapsulation ppp;
    unit logical-unit-number {
        family mlppp {
            bundle interface-name;
        }
    }
}
interface-name {
    unit logical-unit-number {
        encapsulation mlppp;
        family inet {
            address address;
        }
        compression {
            rtp {
                f-max-period number;
                queues [ queue-numbers ];
                port {
                    minimum port-number;
                    maximum port-number;
                }
            }
        }
    }
}
```

This section is organized as follows:

Configuring Voice Services Properties on page 165

Configuring the Bundle Interface on page 168

Example: Configuring Voice Services on page 169

Configuring Voice Services Properties

You define voice service properties such as compression by configuring statements and values for a voice services interface, specified by the physical interface type lsq or vsp. You can configure the following statements:

```
[edit interfaces]
interface-name {
    unit logical-unit-number {
        encapsulation mlppp;
        family inet {
            address address;
        }
        compression {
            rtp {
                f-max-period number;
                queues [ queue-numbers ];
                port {
                    minimum port-number;
                    maximum port-number;
                }
            }
        }
    }
}
```

This section describes the following tasks for configuring voice services properties:

Configuring Logical Interface Encapsulation on page 165

Configuring the Interface Address on page 166

Configuring Compression on page 166

Configuring Logical Interface Encapsulation

Voice services interfaces support only one logical interface encapsulation type, Multilink Point-to-Point Protocol (MLPPP), which is the default encapsulation.

For general information on encapsulation, see “Configuring Interface Encapsulation” on page 73. You can also configure physical interface encapsulation on voice services interfaces.

To configure voice services encapsulation, include the encapsulation statement:

```
encapsulation type;
```

You can include this statement at the following hierarchy levels:

[edit interfaces *interface-name* unit *logical-unit-number*]

[edit logical-routers *logical-router-name* interfaces *interface-name* unit *logical-unit-number*]

You must also configure the T1, E1, or DS3 physical interface with the same encapsulation type.

Configuring the Interface Address

To configure the logical address for the MLPPP bundle, include the address statement:

```
address address {
...
}
```

You can include this statement at the following hierarchy levels:

[edit interfaces *interface-name* unit *logical-unit-number* family inet]

[edit logical-routers *logical-router-name* interfaces *interface-name* unit *logical-unit-number* family inet]

Assign an IP address to the interface by configuring the *address* value. The AS PIC supports only Internet Protocol version 4 (IPv4) addresses configured using the family inet statement.

For information on other addressing properties you can configure that are not specific to service interfaces, see "Configuring the Interface Address" on page 112.

Configuring Compression

You can configure several properties that specify how the interface handles voice traffic compression:

```
compression {
    rtp {
        f-max-period number;
        queues [ queue-numbers ];
        port {
            minimum port-number;
            maximum port-number;
        }
    }
}
```

You can include these statements at the following hierarchy levels:

[edit interfaces *interface-name* unit *logical-unit-number*]

[edit logical-routers *logical-router-name* interfaces *interface-name* unit *logical-unit-number*]

You can configure the following properties at the [edit interfaces *interface-name* unit *logical-unit-number* compression rtp] hierarchy level:

By default, the maximum number of compressed packets inserted between the transmission of full headers is 255 packets. To configure the maximum, include the f-max-period statement at the [edit interfaces *interface-name* unit *logical-unit-number* compression rtp] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number compression rtp]
f-max-period number;
```

To specify the lower and upper boundaries for a range of UDP destination port values on which RTP compression takes effect, include the port statement at the [edit interfaces *interface-name* unit *logical-unit-number* compression rtp] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number compression rtp]
port {
    minimum port-number;
    maximum port-number;
}
```

Values for *port-number* can be from 0 through 65,535. Within the specified range, the router software applies RTP compression to the traffic.

To set the queues on which RTP compression takes effect, include the queues statement at the [edit interfaces *interface-name* unit *logical-unit-number* compression rtp] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number compression rtp]
queues [ queue-numbers ];
```

You can specify q0, q1, q2, and q3 as queue numbers.

The router applies RTP compression on the traffic in the specified queues.

NOTE: If you configure both a port range and one or more queues, compression takes place if either condition is met.

Configuring Delay-Sensitive Packet Interleaving

For M-series platforms (except the M320 router) voice services interfaces with compressed RTP and MLPPP encapsulation, you can configure link fragmentation and interleaving (LFI). LFI reduces excessive delays by fragmenting long packets into smaller packets and interleaving them with real-time frames. This allows real-time and non-real-time data frames to be carried together on lower-speed links without causing excessive delays to the real-time traffic. When the peer interface receives the smaller fragments, it reassembles the fragments into their original packet. For example, short delay-sensitive packets, such as packetized voice, can race ahead of larger delay-insensitive packets, such as common data packets.

By default, LFI is always active when you include the compression rtp statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level. You control the operation of LFI indirectly by setting the fragment-threshold statement on the same logical interface. For example, if you include the fragment-threshold 256 statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level, all IP packets larger than 256 bytes are fragmented.

Example: Configuring Compression

Configure compression on a T1 interface with MLPPP encapsulation. Configure fragmentation for all IP packets larger than 128 bytes.

```
[edit interfaces]
t1-1/0/0 {
    unit 0 {
        family mlppp {
            bundle vsp-1/1/0.1;
        }
    }
}
vsp-1/1/0 {
    encapsulation multilink-ppp;
    unit 1 {
        compression {
            rtp {
                port minimum 2000 maximum 64009;
            }
        }
        family inet {
            address 30.1.1.2/24;
        }
        fragment-threshold 128;
    }
}
```

Configuring the Bundle Interface

To complete a voice services interface configuration, you need to configure both the physical interface and the voice services bundle. For voice services interfaces, configure the link bundle as a channel. The physical interface is usually connected to networks capable of supporting MLPPP; the interface types supported for voice traffic are T1, E1, and T3.

To configure a physical interface link for MLPPP, include the following statements at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
unit 0 {
    family mlppp {
        bundle interface-name;
    }
}
```

When you configure family mlppp, no other protocol configuration is allowed. For more information on link bundles, see “Configuring Bundles” on page 428.

Example: Configuring Voice Services

The following is a complete example of a voice services configuration using a T1 physical interface.

```
[edit interfaces]
t1-0/2/0:1 {
    encapsulation ppp;
    unit 0 {
        family mlppp {
            bundle vsp-1/3/0.1;
        }
    }
}
vsp-1/3/0 {
    unit 1 {
        encapsulation multilink-ppp;
        family inet {
            address 10.5.5.2/30;
        }
        compression {
            rtp {
                f-max-period 100;
                queues [ q1 q2 ];
                port {
                    minimum 16384;
                    maximum 32767;
                }
            }
        }
    }
}
```

Applying Filters and Services to an Interface

When you have defined and grouped the service rules by configuring the service-set definition, you need to apply services to one or more interfaces installed on the routing platform. To associate a defined service set with an interface, include the input and output statements:

```
input {
    [ service-set service-set-name <service-filter filter-name> ];
    post-service-filter filter-name;
}
output {
    [ service-set service-set-name <service-filter filter-name> ];
}
```

You can include these statements at the following hierarchy levels:

[edit interfaces *interface-name* unit *logical-unit-number* family inet service]

[edit logical-routers *logical-router-name* interfaces *interface-name* unit *logical-unit-number* family inet service]

You can configure different service sets on the input and output sides of the interface. You can optionally include filters before or after each service set to refine the target and additionally process the traffic. For an example, see “Example: Configuring a Service Interface” on page 171.

Configuring L2TP Dialup Properties

For adaptive services interfaces on the M7i platform only, you can configure the Layer 2 Tunneling Protocol (L2TP). This section describes how to configure dialup properties for L2TP.

For more information about L2TP, see the *JUNOS Services Interfaces Configuration Guide*.

To configure L2TP dialup properties on the logical interface, include the dial-options statement:

```
dial-options {
    l2tp-interface-id name;
    (shared | dedicated);
}
```

You can include this statement at the following hierarchy levels:

```
[edit interfaces interface-name unit logical-unit-number dial-options]

[edit logical-routers logical-router-name interfaces interface-name unit
logical-unit-number dial-options]
```

The dial-options statement includes configuration for the l2tp-interface-id statement and the shared/dedicated flag. The interface identifier associates a user session with a logical interface. Sessions can use either shared or dedicated logical interfaces. To run routing protocols, a session must use a dedicated logical interface. A dedicated logical interface can represent only one session at a time. A shared logical interface can have multiple sessions.

Example: Configuring a Service Interface

The following example applies my-input-service-set on an interface-wide basis. All traffic that is accepted by my-input-filter has my-service-set applied to it. After the service set is applied, additional filtering is done using my-post-service-input-filter.

```
[edit interfaces fe-0/0/0]
unit 0 {
    family inet {
        filter {
            input my-input-filter;
            output my-output-filter;
        }
        service {
            input {
                service-set my-input-service-set;
                post-service-filter my-post-service-input-filter;
            }
            output {
                service-set my-output-service-set;
            }
        }
    }
}
```