Chapter 6
Configure Logical Interface Properties

For a physical interface device to function, you must configure at least one logical interface on that device. For each logical interface, you must at a minimum specify the protocol family that the interface supports. You can also configure other logical interface properties. These vary by PIC and encapsulation type, but include the IP address of the interface, whether the interface does not support multicast traffic, DLCIs, VCIs and VPIs, and traffic shaping.

To configure logical interface properties, you include the unit statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
unit logical-unit-number {
    disable;
    dlci dlci-identifier;
    encapsulation type;
    inverse-arp;
    multicast-dlci dlci-identifier;
    multicast-vci vpi-identifier.vci-identifier;
    multipoint;
    no-traps;
    oam-liveness {
        up-count cells;
        down-count cells;
    }
    oam-period seconds;
    point-to-point;
    shaping {
        (cbr rate | vbr peak rate sustained rate burst length);
        queue-length number;
    }
    tunnel {
        source address;
        destination address;
        ttl number;
    }
    vci vpi-identifier.vci-identifier;
    vlan-id number;
}
```

This chapter describes the logical interface properties that you can configure.

Specify the Logical Interface Number

Each logical interface must have a logical unit number. The logical unit number corresponds to the logical unit part of the interface name. For more information, see “Configure the Interface Name” on page 28.

PPP and Cisco HDLC encapsulations support only a single logical interface, whose logical unit number must be 0. Frame Relay and ATM encapsulations support multiple logical interfaces, so you can configure one or more logical unit numbers.

You specify the logical unit number in the unit statement the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces]
interface-name {
    unit 0 {
        ...
    }
}
interface-name {
    unit logical-unit-number {
        ...
    }
}
```

The logical unit number can range from 0 through 16384.

Configure a Point-to-Point Connection

By default, all interfaces are assumed to be point-to-point connections. You must ensure that the MTU sizes on both sides of the connection are the same.

Optionally, you can explicitly configure an interface to be a point-to-point connection by including the point-to-point statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
point-to-point;
```

Configure a Multipoint Connection

By default, all interfaces are assumed to be point-to-point connections. To configure an interface to be a multipoint connection, include the multipoint statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
multipoint;
```

Configure Frame Relay DLCIs

When you are using Frame Relay encapsulation on an interface, each logical interface corresponds to one or more permanent virtual circuits (PVCs) or switched virtual circuits (SVCs). For each PVC or SVC, you must configure one data-link connection identifier (DLCI).

A Frame Relay interface can be a point-to-point interface or a point-to-multipoint (also called a multipoint nonbroadcast multi-access [NBMA]) connection.

To configure Frame Relay DLCIs, you can do the following:

Configure a Point-to-Point Frame Relay Connection on page 67

Configure a Point-to-Multipoint Frame Relay Connection on page 67

Configure a Multicast-Capable Frame Relay Connection on page 68

Configure a Point-to-Point Frame Relay Connection

To configure a point-to-point Frame Relay connection, include the dlci statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
dlci dlci-identifier;
```

dlci-identifier is the DLCI identifier, which is a number from 1 through 1022. A point-to-point interface can have one DLCI.

When you are configuring point-to-point connections, the MTU sizes on both sides of the connection must be the same.

Configure a Point-to-Multipoint Frame Relay Connection

To configure a point-to-multipoint Frame Relay connection (also called a multipoint NBMA connection), include the multipoint-destination statement within the address statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
address address {
    multipoint-destination destination-address dlci dlci-identifier;
}
```

address is the interface's address.

For each destination, include one multipoint-destination statement. *destination-address* is the address of the remote side of the connection, and *dlci-identifier* is the DLCI identifier for the connection.

When you are configuring point-to-multipoint connections, all interfaces in the subnet must use the same MTU size.

If keepalives are enabled, causing the interface to send LMI messages during idle times, the number of possible DLCI configurations is limited by the MTU selected for the interface. For information about disabling keepalives, see "Configure Keepalives" on page 34.

Configure a Multicast-Capable Frame Relay Connection

By default, Frame Relay connections assume unicast traffic. If your Frame Relay switch performs multicast replication, you can configure the connection to support multicast traffic by including the multicast-dlci statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
multicast-dlci dlci-identifier;
```

dlci-identifier is the DLCI identifier, which is a number from 1 through 1022 that defines the Frame Relay DLCI over which the switch is expecting to receive multicast packets for replication.

You can configure multicast support only on point-to-multipoint Frame Relay connections.

If keepalives are enabled, causing the interface to send LMI messages during idle times, the number of possible DLCI configurations is limited by the MTU selected for the interface. For information about disabling keepalives, see "Configure Keepalives" on page 34.

Configure ATM Virtual Circuits

When you are using ATM encapsulation on an interface, you must map each logical interface to a virtual circuit identifier (VCI) and optionally to a virtual path identifier (VPI).

An ATM interface can be a point-to-point interface or a point-to-multipoint (also called a multipoint nonbroadcast multi-access [NBMA]) connection.

When configuring ATM virtual circuits, you can do the following:

Configure the Maximum Number of VCs on a VP

When you are configuring ATM VCs, you must configure the maximum number of virtual circuits (VCs) allowed on a virtual path (VP). You configure this value so that sufficient memory on the ATM PIC can be allocated for each VC.

To configure the largest numbered VCs on a VP, include the atm-options statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
atm-options {
    vpi vpi-identifier max-vcs maximum-vcs;
}
```

The VP identifier can be a value from 0 through 255. The maximum number of VCs you can configure per ATM interface is 4090. The largest numbered VC value you can configure is 4089.

Configure a Point-to-Point ATM Connection

To configure a VCI and a VPI on a point-to-point ATM interface, include the vpi statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
vci vpi-identifier.vci-identifier;
```

For each VCI, you configure the VCI and VPI identifiers. The default VPI identifier is 0. The VCI identifier cannot exceed the highest numbered VC configured for the interface with the vpi statement, as described in “Configure ATM Physical Interface Properties” on page 38.

When you are configuring point-to-point connections, the MTU sizes on both sides of the connections must be the same.

Configure a Point-to-Multipoint ATM Connection

To configure a point-to-multipoint (NBMA) ATM connection, include the multipoint-destination statement within the address statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *inet* address *address*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet]
address address {
    multipoint-destination destination-address vci vpi-identifier.vci-identifier;
}
```

address is the interface’s address. The address must include the destination prefix (for example, /24).

For each destination, include one multipoint-destination statement. *destination-address* is the address of the remote side of the connection, and *vci-identifier* and *vpi-identifier* are the VCI and optional VPI identifiers for the connection.

When you are configuring point-to-multipoint connections, all interfaces in the subnet must use the same MTU size.

Configure a Multicast-Capable ATM Connection

By default, ATM connections assume unicast traffic. If your ATM switch performs multicast replication, you can configure the connection to support multicast traffic by including the multicast-vci statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
multicast-vci vpi-identifier.vci-identifier;
```

vci-identifier and *vpi-identifier* are the VCI and VPI identifiers, which define the ATM VCI over which the switch is expecting to receive multicast packets for replication.

You can configure multicast support only on point-to-multipoint ATM connections.

Configure ATM Inverse ARP

You can configure ATM interfaces to support inverse ATM ARP, as described in RFC 2225. When inverse ATM ARP in enabled, the router responds to received Inverse ATM ARP requests by providing IP address information to the requesting ATM device.

The router does not initiate inverse ATM ARP requests.

By default, inverse ATM ARP is disabled. To configure a VC to respond to inverse ATM ARP requests, include the inverse-arp statement at the [edit interfaces *interface-name* unit *logical-unit-number*] or [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* multipoint-destination *destination*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
inverse-arp;
```

Define the ATM Traffic-Shaping Profile

When you are using an ATM encapsulation, you can configure a traffic-shaping profile that defines the following:

- Bandwidth utilization, which consists of either a constant rate, or a peak cell rate with sustained cell rate and burst tolerance
- Maximum queue length

These values are used in the ATM generic cell-rate algorithm, which is a leaky bucket algorithm that defines the short-term burst rate for ATM cells, the maximum number of cells that can be included in a burst, and the long-term sustained ATM cell traffic rate. Each individual VC has its own independent shaping parameters.

By default, the bandwidth utilization is unlimited; that is, unspecified bit rate (UBR) is used. Also, by default, buffer usage by VCs is unregulated. To define limits to bandwidth utilization on a point-to-point interface or to limit buffer use, include the shaping statement. For point-to-point interfaces, include the shaping statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name]
unit logical-unit-number {
    vci vpi-identifier.vci-identifier;
    shaping {
        (cbr rate | vbr peak rate sustained rate burst length);
        queue-length number;
    }
}
```

For virtual circuits that are part of a point-to-multipoint interface, include the shaping statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family family]
address address {
    multipoint-destination destination-address {
        vci vpi-identifier.vci-identifier;
        shaping {
            (cbr rate | vbr peak rate sustained rate burst length);
            queue-length number;
        }
    }
}
```

Configure CBR

For traffic that does not require the ability to periodically burst to a higher rate, you can configure a constant bit rate (CBR) by including the cbr statement at the [edit interfaces *interface-name* unit *logical-unit-number*] or [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address*] hierarchy level:

```
shaping {
    cbr rate ;
}
```

Configure VBR

To define variable bandwidth rate (VBR) utilization, include the vbr statement at the [edit interfaces *interface-name* unit *logical-unit-number*] or [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address*] hierarchy level:

```
vbr peak rate sustained rate burst length;
```

You can define the following VBR traffic-shaping properties:

Peak rate—Top rate at which traffic can burst.

Sustained rate—Normal traffic rate averaged over time.

Burst length—Maximum number of cells that a burst of traffic can contain. It can be a value from 1 through 255 cells.

Specify the rates in bits per second (bps) or cells per second (cps). For OC-3 interfaces, the highest rate is 135631698 bps (353207.55 cps), which corresponds to 100 percent of the available line rate. For OC-12 interfaces, the highest rate is 271263396 bps (706415.09 cps), which corresponds to 50 percent of the available line rate. Table 5 lists some of the other rates you can specify. If you specify a rate that is not listed, it is rounded to the nearest rate.

In general, the actual packet rate on the interface is calculated with the following formula:

actual-rate = (128 * *line-rate*) / (trunc ((128 * *line-rate*) / *desired-rate*))

line-rate is the maximum available rate on the interface (in bits per second) after factoring out the overhead for SONET and ATM (per-cell) overheads. For OC-3 interfaces, the line rate is calculated as follows:

line-rate = 155,520,000 bps x (26/27) x (48/53) = 135,631,698.1 bps

For OC-12 interfaces, the line rate is calculated as follows:

line-rate = 622,080,000 bps x (26/27) x (48/53) = 542,526,792.45 bps

desired-rate is the rate you enter in the vbr statement, in bits per second.

The trunc operator indicates that all digits to the right of the decimal point should be dropped.

For OC-3 interfaces, the maximum available rate is 100 percent of *line-rate*, or 135,631,698 bps. For OC-12 interfaces, the maximum available rate is 50 percent of *line-rate*, or 271,263,396 bps.

The following example shows the calculations for determining the actual rate when the desired rate is 80 percent of the maximum rate:

OC-3:

135,631,968 bps * 0.8 = 108,505,358 bps
actual-rate = (128 * 135,631,698.1) / (trunc ((128 * 135,631,698.1) / 101,723,773.5))
actual-rate = 17,360,857,344 / (trunc ((17,360,857,344) / 101,723,773.5)
actual-rate = 17,360,857,344 / 160
actual-rate = 108,505,358 bps

OC-12:

271,263,396 bps * 0.8 = 217,010,716.8 bps
actual-rate = (128 * 542,526,792.45) / (trunc ((128 * 542,526,792.45)/217,010,716.8))
actual-rate = 69,443,429,434 / (trunc ((69,443,429,434) / 217,010,716.8))
actual-rate = 69,443,429,434 / 320
actual-rate = 217,010,717 bps

Table 5: Traffic-Shaping Rates

Interface Type	Line Rate (bps)	Line Rate (cps)	Percentage of Total Line Rate
OC-3			
	135631698	353207.55	100.00
	134580290	350469.50	99.22
	133545057	347773.58	98.46
	132525629	345118.82	97.71
	131521647	342504.29	96.97
	130532762	339929.07	96.24
	129558637	337392.28	95.52
	128598943	334893.08	94.81
	127653363	332430.63	94.12
	126721587	330004.13	93.43
OC-12			
	271263396	706415.09	50.00
	270207897	703666.40	49.81
	269160579	700939.01	49.61
	268121349	698232.68	49.42
	267090113	695547.17	49.23
	266066779	692882.24	49.04
	265051257	690237.65	48.85
	264043458	687613.17	48.67
	263043293	685008.58	48.48
	262050677	682423.64	48.30

Buffers are shared among all VCs, and by default, there is no limit to the buffer size for a VC. If a VC is particularly slow, it might use all the buffer resources. To limit the queue size of a particular VC, include the queue-length statement when configuring the VC at the [edit interfaces *interface-name* unit *logical-unit-number*] or [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address*] hierarchy level:

```
queue-length number;
```

The length can range from 1 through 16383 packets. The default is 16383 packets.

Define the ATM OAM F5 Loopback Cell Period

When you are using an ATM encapsulation, you can configure the OAM F5 loopback cell period on virtual circuits, which is the interval at which OAM F5 loopback cells are transmitted.

By default, no OAM F5 loopback cells are sent. To send OAM F5 loopback cells on a point-to-point interface, include the oam-period statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name]
unit logical-unit-number {
    vci vpi-identifier.vci-identifier;
    oam-period seconds;
}
```

To send OAM F5 loopback cells on a virtual circuit that is part of a point-to-multipoint interface, include the oam-period statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family family]
address address {
    multipoint-destination destination-address {
        vci vpi-identifier.vci-identifier;
        oam-period seconds;
    }
}
```

The period can range from 1 through 900 seconds.

OAM VC-AIS (alarm indication signal) and VC-RDI (remote defect indication) defect indication cells are used for identifying and reporting VC defects end-to-end. When a physical link or interface failure occurs, intermediate nodes insert OAM AIS cells into all the downstream VCs affected by the failure. Upon receiving an AIS cell on a VC, the router marks the logical interface down and sends an RDI cell on the same VC to let the remote end know the error status. When an RDI cell is received on a VC, the router sets the logical interface status to down. When no AIS or RDI cells are received for 3 seconds, the router sets the logical interface status to up. You do not need to configure anything to enable defect indication.

Configure the ATM OAM F5 Loopback Cell Threshold

When you are using an ATM encapsulation, you can configure the OAM F5 loopback cell threshold on VCs, which is the minimum number of consecutive OAM F5 loopback cells received before declaring that a VC is up or lost before declaring that a VC is down.

By default, when five consecutive OAM F5 loopback cells are received, the VC is considered to be up, and when five consecutive cells are lost, the VC is considered to be down. To modify these values on a point-to-point interface, include the oam-liveness statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name]
unit logical-unit-number {
    vci vpi-identifier.vci-identifier;
    oam-liveness {
        up-count cells;
        down-count cells;
    }
```

```
}
```

To modify the OAM F5 loopback cell count threshold on a virtual circuit that is part of a point-to-multipoint interface, include the oam-liveness statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family family]
address address {
    multipoint-destination destination-address {
        vci vpi-identifier.vci-identifier;
        oam-liveness {
            up-count cells;
            down-count cells;
        }
    }
}
```

The cell count can be a value from 1 through 255 cells.

Configure 802.1Q VLAN IDs

For Ethernet, Fast Ethernet, and Gigabit Ethernet interfaces, the software supports a subset of the IEEE 802.1Q standard for channelizing an Ethernet interface into multiple logical interfaces, allowing many hosts to be connected to the same Gigabit Ethernet switch, but preventing them from being in the same routing or broadcast domain.

Gigabit Ethernet interfaces can be partitioned; up to 1024 logical interfaces are allowed, one for each VLAN. To bind a VLAN ID to a logical interface, include the vlan-id statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
vlan-id number;
```

Disable SNMP Notifications on Logical Interfaces

By default, SNMP notifications are sent when the state of an interface or a connection changes. To disable these notifications on the logical interface, include the no-traps statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
no-traps;
```

Configure Interface Encapsulation

For each physical interface, you must configure an encapsulation to use for packets transmitted on the interface. You can optionally configure an encapsulation on a logical interface, which is the encapsulation used within certain packet types.

Configure the Encapsulation on a Physical Interface

The physical interface encapsulation can be one of the following:

Frame Relay—This encapsulation is defined in RFC 1490, *Multiprotocol Interconnect over Frame Relay*. E3, SONET, and T3 interfaces can use Frame Relay encapsulation.

Frame Relay Circuit Cross Connect (CCC)—This encapsulation is the same as standard Frame Relay for DLCIs 0 through 511. DLCIs 512 through 1022 are dedicated to CCC, and the logical interface must also have frame-relay-ccc encapsulation.

ATM PVC—ATM Permanent Virtual Circuit (PVC) encapsulation is defined in RFC 1483, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*. ATM interfaces can use ATM PVC encapsulation.

Point-to-Point Protocol (PPP)—PPP encapsulation is defined in RFC 1331, *The Point-to-Point Protocol (PPP) for the Transmission of Multiprotocol Datagrams over Point-to-Point Links*. E3, SONET, and T3 interfaces can use PPP encapsulation. There is also a CCC version (ppp-ccc); the logical interfaces do not require an encapsulation statement, but they cannot have families.

Cisco HDLC—E3, SONET, and T3 interfaces can use Cisco HDLC encapsulation.There is also a CCC version (cisco-hdlc-ccc); the logical interfaces do not require an encapsulation statement, but they cannot have families.

VLAN Circuit Cross-Connect (CCC)—Ethernet interfaces with Virtual Local Area Network (VLAN) tagging enabled can use VLAN-CCC encapsulation.

To configure the encapsulation on a physical interface, include the encapsulation statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
encapsulation (atm-pvc | cisco-hdlc | cisco-hdlc-ccc | frame-relay | frame-relay-ccc | ppp | ppp-ccc |
   vlan-ccc);
```

When you configure a point-to-point encapsulation (such as PPP or Cisco HDLC) on a physical interface, the physical interface can have only one logical interface (that is, only one unit statement) associated with it. When you configure a multipoint encapsulation (such as Frame Relay), the physical interface can have multiple logical units and the units can be either point to point or multipoint.

Ethernet interfaces in VLAN mode can have multiple logical interfaces, but in CCC mode VLAN IDs from 0 through 511 are reserved for normal VLANs and VLAN IDs from 512 through 1023 are reserved for CCC VLANs. For more information, see “Configure 802.1Q VLANs” on page 140.

Example: Configure the Encapsulation on a Physical Interface

Configure PPP encapsulation on a SONET interface. The second two family statements allow IS-IS and MPLS to run on the interface.

```
[edit interfaces]
so-7/0/0 {
    encapsulation ppp;
    unit 0 {
        point-to-point;
        family inet {
            address 192.168.1.113/32 {
                destination 192.168.1.114;
            }
        }
        family iso;
        family mpls;
    }
}
```

Configure the Encapsulation on a Logical Interface

Generally, you configure an interface's encapsulation at the [edit interfaces *interface-name*] hierarchy level. However, for some encapsulation types, such as Frame Relay, ATM, and Ethernet VLAN encapsulations, you also can configure the encapsulation type that is used inside the Frame Relay, ATM, or VLAN circuit itself. To do this, include the encapsulation statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
encapsulation (atm-nlpid | atm-cisco-nlpid | atm-snap | atm-vc-mux | atm-ccc-vc-mux |
    frame-relay-ccc | vlan-ccc);
```

Some of the ATM encapsulations are defined in RFC 1483, *Multiprotocol Encapsulation o ver ATM Adaptation Layer 5*.

With the atm-nlpid, atm-cisco-nlpid, and atm-vc-mux encapsulations, you can configure the family inet only. With the circuit cross-connect (CCC) circuit encapsulations, you cannot configure a family on the logical interface. A logical interface cannot have frame-relay-ccc encapsulation unless the physical device also has frame-relay-ccc encapsulation. In addition, you must assign this logical interface a DLCI in the range 512 through 1022 and configure it as point-to-point.

A logical interface cannot have vlan-ccc encapsulation unless the physical device also has vlan-ccc encapsulation. You must also assign this logical interface a VLAN ID in the range 512 through 1023; if the VLAN ID is 511 or lower, it is subject to the normal destination filter lookups in addition to source address filtering.

Disable a Logical Interface

You can unconfigure a logical interface, effectively disabling that interface, without removing the logical interface configuration statements from the configuration. To do this, include the disable statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
disable;
```