

## Configuring Frame Relay Interface Encapsulation

---

Point-to-Point Protocol (PPP) encapsulation is the default encapsulation type for physical interfaces. You need not configure encapsulation for any physical interfaces that support PPP encapsulation. If you do not configure encapsulation, PPP is used by default. For physical interfaces that do not support PPP encapsulation, 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.

For more information, see the following sections:

- Configuring the Frame Relay Encapsulation on a Physical Interface on page 1
- Configuring the Frame Relay Encapsulation on a Logical Interface on page 4

### Configuring the Frame Relay Encapsulation on a Physical Interface

For Frame Relay interfaces, configure Frame Relay encapsulation on the physical interface. This encapsulation is defined in RFC 1490, *Multiprotocol Interconnect over Frame Relay*. SONET/SDH and T3 interfaces can use Frame Relay encapsulation.

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

```
[edit interfaces interface-name]  
[Unresolved xref] type;
```

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.

The encapsulation type can be one of the following:

- Flexible Frame Relay (**flexible-frame-relay**)—IQ interfaces can use flexible Frame Relay encapsulation. You use flexible Frame Relay encapsulation when you want to configure multiple per-unit Frame Relay encapsulations. This encapsulation type allows you to configure any combination of TCC, CCC, and standard Frame Relay encapsulations on a single physical port. Also, each logical interface can have any DLCI value from 1 through 1022.
- Frame Relay (**frame-relay**)—Defined in RFC 1490. E1, E3, link services, SONET/SDH, T1, T3, and voice services interfaces can use Frame Relay encapsulation. Five related versions are supported:
  - Circuit cross-connect (CCC) version (**frame-relay-ccc**)—The same as standard Frame Relay for DLCIs 0 through 511. DLCIs 512 through 1022 are dedicated to CCC. The logical interface must also have **frame-relay-ccc** encapsulation. When you use this encapsulation type, you can configure the **ccc** family only.
  - Translational cross-connect (TCC) version (**frame-relay-tcc**)—Similar to Frame Relay CCC and has the same configuration restrictions, but used for circuits with different media on either side of the connection.

- Extended CCC version (**extended-frame-relay-ccc**)—This encapsulation type allows you to dedicate DLCIs 1 through 1022 to CCC. The logical interface must have **frame-relay-ccc** encapsulation. When you use this encapsulation type, you can configure the **ccc** family only.
- Extended TCC version (**extended-frame-relay-tcc**)—Similar to extended Frame Relay CCC, this encapsulation type allows you to dedicate DLCIs 1 through 1022 to TCC, which is used for circuits with different media on either side of the connection.
- Port CCC version (**frame-relay-port-ccc**)—Defined in the Internet Engineering Task Force (IETF) document, *Frame Relay Encapsulation over Pseudo-Wires* (expired December 2002). This encapsulation type allows you to transparently carry all the DLCIs between two customer edge (CE) routers without explicitly configuring each DLCI on the two provider edge (PE) routers with Frame Relay transport. The connection between the two CE routers can be either user-to-network interface (UNI) or network-to-network interface (NNI); this is completely transparent to the PE routers. The logical interface does not require an encapsulation statement. When you use this encapsulation type, you can configure the **ccc** family only.
- Frame Relay Ether Type (**frame-relay-ether-type**)—Physical interfaces can use Frame Relay ether type encapsulation for compatibility with Cisco Frame Relay. IETF frame relay encapsulation identifies the payload format using NLPID and SNAP formats. Cisco-compatible Frame Relay encapsulation uses the Ethernet type to identify the type of payload. Two related versions are supported:
  - TCC version (**frame-relay-ether-type-tcc**)—Cisco-compatible Frame Relay for DLCIs 0 through 511. DLCIs 512 through 1022 are dedicated to TCC. This numbering restriction does not apply to IQ interfaces. This encapsulation is used for circuits with different media on either side of the connection.
  - Extended TCC version (**extended-frame-relay-ether-type-tcc**)—This encapsulation allows you to dedicate Cisco-compatible Frame Relay TCC for DLCIs 1 through 1022. This encapsulation is used for circuits with different media on either side of the connection. Extended Frame Relay ether type TCC encapsulation is supported on the same PICs as extended Frame Relay TCC encapsulation.



**NOTE:** When the encapsulation type is set to Cisco-compatible Frame Relay encapsulation, ensure that the LMI type is set to ANSI or Q933-A.

Support for extended Frame Relay and flexible Frame Relay differs by PIC type, as shown in Table 1.

**Table 1: PIC Support for Enhanced Frame Relay Encapsulation Types**

PIC Type	Extended Frame Relay CCC	Extended Frame Relay TCC	Flexible Frame Relay
<b>Intelligent Queuing</b>			
1-port Channelized CHOC12 IQ	Yes	Yes	Yes

**Table 1: PIC Support for Enhanced Frame Relay Encapsulation Types** *(continued)*

PIC Type	Extended Frame Relay CCC	Extended Frame Relay TCC	Flexible Frame Relay
4-port Channelized DS3 IQ	Yes	Yes	Yes
10-port Channelized E1 IQ	Yes	Yes	Yes
4-port E3 IQ	Yes	Yes	Yes
1-port Channelized STM1 IQ	Yes	Yes	Yes
<b>SONET/SDH</b>			
1-port OC12	Yes	Yes	No
2-port OC3	Yes	Yes	No
1-port OC48	Yes	Yes	No
1-port OC192	Yes	Yes	No
1-port STM16 SDH, SMSR	Yes	Yes	No
<b>Others</b>			
4-port E1	No	No	No
4-port T1	No	No	No
4-port T3	No	No	No
10-port Channelized E1	No	No	No
2-port Channelized DS3	No	No	No
1-port Channelized OC12, SMIR	No	No	No
4-port Channelized DS3	No	No	No
1-port Channelized STM1, SMIR	No	No	No
2-port Serial	No	No	No

### Example: Configuring the Encapsulation on a Physical Interface

Configure Frame Relay encapsulation on a SONET/SDH interface. The second and third family statements allow Intermediate System-to-Intermediate System (IS-IS) and Multiprotocol Label Switching (MPLS) to run on the interface.

```
[edit interfaces]
so-7/0/0 {
  [Unresolved xref] frame-relay;
  unit 0 {
    point-to-point;
    family inet {
```

```

        address 192.168.1.113/32 {
            destination 192.168.1.114;
        }
    }
    family iso;
    family mpls;
}

```

## Configuring the Frame Relay Encapsulation on a Logical Interface

Generally, you configure an interface's encapsulation at the [edit interfaces *interface-name*] hierarchy level. However, for Frame Relay encapsulation, you can also configure the encapsulation type that is used inside the Frame Relay packet itself. To do this, include the **encapsulation** statement, specifying the **frame-relay-ccc**, **frame-relay-ppp**, **frame-relay-tcc**, **frame-relay-ether-type**, or **frame-relay-ether-type-tcc** option:

```

[Unresolved xref] (frame-relay-ccc | frame-relay-ppp | frame-relay-tcc |
    frame-relay-ether-type | frame-relay-ether-type-tcc);

```

You can include this statement at the following hierarchy levels:

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