- Point-to-Point Protocol -

WAN Encapsulation

Recall that WAN technologies operate at both **Physical** and **Data-link** layers of the OSI models, and that higher-layer protocols such as IP are **encapsulated** when sent across the WAN link.

A WAN is usually terminated on a Cisco device's serial interface. Serial interfaces support a wide variety of **WAN encapsulation types**, which must be manually specified.

By default, a serial interface will utilize **HDLC** for encapsulation. Other supported encapsulation types include:

- SDLC
- PPP
- LAPB
- Frame-Relay
- X.25
- ATM

Regardless of the WAN encapsulation used, it must **identical** on both sides of a point-to-point link.

HDLC Encapsulation

High-Level Data-link Control (HDLC) is a WAN encapsulation protocol used on dedicated point-to-point serial lines.

Though HDLC is technically an ISO standard protocol, Cisco's implementation of HDLC is proprietary, and will not work with other routers.

HDLC is also Cisco's **default encapsulation** type for serial point-to-point links. HDLC provides *no* authentication mechanism.

* * *

PPP Encapsulation

Point-to-Point Protocol (PPP) is a standardized WAN encapsulation protocol that can be used on a wide variety of WAN technologies, including:

- Dedicated point-to-point serial lines
- Asynchronous dial-up links
- ISDN

PPP has four components:

- **Physical** – standard for physical serial communication (such as EIA/TIA-232-C, V.35, ISDN, etc.).
- **HDLC** – for encapsulating packets into frames over serial lines.
- **LCP** – for establishing, maintaining, and terminating point-to-point links.
- **NCP** – allows multiple Layer-3 protocols (such as IP and IPX) to be encapsulated into frames.

PPP supports several features that standalone HDLC does not:

- **Authentication** – secures the communication by forcing the sending/receiving devices to identify themselves with a username and password. PPP supports two forms of authentication – **PAP** and **CHAP.**
- **Compression** – improves efficiency on slow links. PPP supports two forms of compression: **Stac** and **Predictor**.
- **Multilink** – allows multiple channels to be *bundled* or *trunked* together to combine the bandwidth. The bundled channels are treated as one logical channel.
- **Callback** – provides security and billing services. Allows a client to first *dial* a PPP server, *disconnect*, and then have the PPP server *call* the client *back*.
- **Error Control**

* * *

Configuring Basic PPP

To configure a serial interface for PPP encapsulation:

```
Router(config)# int s0/0
Router(config-if)#  encapsulation ppp
```

Recall that PPP supports two methods of authentication, **PAP** and **CHAP**. PAP (**Password Authentication Protocol**) sends passwords in clear text, and thus does not provide much security. CHAP (**Challenge Handshake Authentication Protocol**) uses MD5 to apply an irreversible hash.

To configure PPP authentication:

```
Router(config)# hostname Router1
Router(config)# username Router2 password PASSWORD

Router(config)# int s0/0
Router(config-if)#  ppp authentication chap
```

The first line sets the hostname of the router. The second line sets the username and password used for PPP authentication. The username must be the hostname of the *remote* router, and the password must be the same on both routers.

The above configuration sets the authentication to *chap*. To instead configure *pap* authentication:

```
Router(config)#  int s0/0
Router(config-if)#  ppp authentication pap
```

To view the encapsulation configured on the interface:

```
Router#  show interface s0/0
```

To troubleshoot PPP authentication between two routers:

```
Router#  debug ppp authentication
```

More advanced PPP configuration is discussed in the ISDN guide.

* * *