Chapter 15

# Point-to-Point Protocol (PPP)

# Introduction

This chapter describes the main features of the Point-to-Point Protocol (PPP), support for it on the router, and how to configure network interfaces on the router to use the PPP.

PPP allows one device such as a host, router, or switch to connect to another single device via a WAN or LAN. It is the most common protocol for linking a host PC to an ISP. PPP runs over physical interfaces, including Ethernet ports, VLANs (over Ethernet switch ports), synchronous ports, ISDN and asynchronous ports, depending on the individual device.

The Point-to-Point Protocol was developed by the Internet Engineering Task Force (IETF) as a means of transmitting data for more than one network protocol over the same point-to-point serial link in a standard, vendor-independent way. It can carry IP, Novell IPX, AppleTalk, and DECnet traffic. Traffic can also be encrypted or compressed.

This implementation complies with RFC 1994.

Some interface and port types mentioned in this chapter may not be supported on your router. The interface and port types that are available vary depending on your product's model, and whether an expansion unit (PIC, NSM) is installed. For more information, see the Hardware Reference.

# The Point-to-Point Protocol

The Point-to-Point Protocol consists of the following main components:

- A method for encapsulating datagrams over serial links.
- A Link Control Protocol (LCP) for establishing, configuring, and testing the data-link connection.
- A family of Network Control Protocols (NCPs) for establishing and configuring different network-layer protocols.

The mechanism that PPP uses to carry network traffic is to open a link with a short exchange of packets. Once the link is open, network traffic is carried with very little overhead. Frames are sent as unnumbered information frames, meaning that no data link acknowledgement is required and no retransmissions are carried out. Once the link is established, PPP acts as a straight data pipe for protocols.

## Encapsulation

At the lowest level, the Point-to-Point Protocol is an example of the HDLC protocol and has the following features:

- Data comes in frames, delimited by special characters called flags.
- When a frame is not being sent, the sender transmits flags continually. This means that there is constant activity on any synchronous line that is running properly.
- The first four bytes of a PPP frame comprise a 1 octet address field that is always set to 0xFF, a 1 octet control field that is always set to 0x03 ("unnumbered information"), and a 2-octet protocol field.
- The data that follows the address and control fields is interpreted by the device receiving the frame depending on the encapsulation type.

Link Control Protocol (LCP) exists to bring up the PPP link before any other protocols can begin transmission. Each protocol carried over PPP has an associated Network Control Protocol (NCP) that negotiates options for the protocol and brings up the link for that protocol (Table 15-1 on page 15-4).

Table 15-1: Supported Network protocols and Network Control Protocols for the Point-to-Point Protocol

| Protocol | PPP Type (hexadecimal) |
|---|---|
| LCP | 0xC021 |
| IP | 0x0021 |
| IPCP | 0x8021 |
| TCP/IP Comp | 0x002D |
| TCP/IP Uncomp | 0x002F |
| IPX | 0x002B |
| IPXCP | 0x802B |
| IPv6 | 0x0057 |
| IPv6CP | 0x8057 |
| DECnet | 0x0027 |
| DECnetCP | 0x8027 |
| AppleTalk | 0x0029 |
| ATCP | 0x8029 |
| Multilink | 0x003D |
| Individual Link Compression | 0x00FB |
| ILCCP | 0x80FB |
| Compression | 0x00FD |
| CCP | 0x80FD |
| Encryption | 0x0053 |
| ECP | 0x8053 |
| Bridging | 0x0031 |
| Bridge Spanning Tree | 0x0201 |
| BCP | 0x8031 |
| Bandwidth Allocation Protocol | 0xC02D |
| BACP | 0xC02B |
| Link Quality Report (LQR) | 0xC025 |
| Password Authentication Protocol (PAP) | 0xC023 |
| Challenge-Handshake Authentication Protocol (CHAP) | 0xC223 |

The TCP/IP Comp and TCP/IP Uncomp protocols provide direct support for Van Jacobson's header compression. For more information on Van Jacobson's header compression see Chapter 22, Internet Protocol (IP).

# Control Protocols

Control protocols are protocols run by PPP between the two stations at either end of a link to allow the link to be used to carry a particular type of traffic. The Link Control Protocol (LCP) must run before any other control protocol in order to allow the link to be used at all.

The local and remote stations negotiate the configuration options to be used on the link. A *configure request packet* is sent first containing configuration options. The remote station responds with a packet confirming that the options are okay, suggesting different options or rejecting the options. This exchange takes place in both directions and when a station has sent and received an acknowledge packet the link is declared open.

Once the link has been opened by the LCP, any authentication that is required is performed. When authentication has been completed successfully, or if no authentication is required, then a Network Control Protocol (NCP) is run for each network layer protocol using the link. The NCPs operate in a similar way to the LCP, negotiating configuration options specific to the network layer protocol. No NCPs can use the PPP link until the LCP has opened the link, and no data packets can be exchanged unless the appropriate NCP is open.

Control protocols consist of states, events, and packets. Events cause the state of a link to change (Table 15-2). Two important events are *open* and *close*. They can be caused either by a management command or internally, for example, when the router powers up. An *open* event causes the control protocol to try to establish a link and a *close* event terminates a link. Other events are the hardware becoming available (up) or unavailable (down), timeouts, and the arrival of packets.

Table 15-2: States for control protocols of the Point-to-Point Protocol

| State | Meaning |
|---|---|
| initial | Startup state; no open event has occurred and the hardware is down. |
| starting | An open event has occurred and the hardware is down. |
| closed | The hardware is up and no open event has occurred. |
| stopped | The hardware is up and a down or timeout event has occurred. |
| closing | The link has been up and a close event has occurred; trying to close link. |
| stopping | The link has been open and the remote station is trying to close the link. |
| req sent | A configure request has been sent; waiting for a reply. |
| ACK rcvd | A configure request has been sent, and an acknowledge received. |
| ACK sent | A configure request has been received, and an acknowledge sent. |
| opened | An acknowledge has been sent and received. |

The state of a PPP link (LCP) and the NCPs running on that link can be displayed with the command:

```
show ppp
```

For information about configuring the LCP terminate and the configure request counters, see "Configuring PPP Control Protocols" on page 15-30.

# LCP Options

The LCP attempts to negotiate the following options:

- Maximum Receive Unit (MRU).
- Endpoint Discriminator.
- Link Discriminator, as defined in RFC 2125.
- Authentication Protocol.
- Link Quality Reporting (LQR).
- Magic Number.
- Asynchronous Control Character Map (ACCM).
- Maximum Received Reconstructed Unit (MRRU).

All other options are set to the defaults specified in the relevant RFC.

## Endpoint Discriminator Option

The Endpoint Discriminator Option is defined in RFC 1990 and is required for PPP to form multilink bundles from dynamic PPP calls. The Endpoint Discriminator provides a mechanism for identifying the physical location of the peer at the remote end of a PPP link. When two or more dynamic PPP calls are made from the same peer, with the same authentication information, they can be bundled together to form a multilink interface if they have the same Endpoint Discriminator. The router uses its MAC address to identify itself.

If an Endpoint Discriminator is received during LCP negotiation on a newly activated link in a static PPP interface with more than one link, and that Endpoint Discriminator value is different from the Endpoint Discriminators received during negotiation on the other active links in the interface, then the new link with the invalid Endpoint Discriminator is deactivated.

## Link Discriminator Option

The Link Discriminator Option is defined in RFC 2125 and is required for the operation of BAP. During LCP negotiation it is used to declare a unique identifier for the link over which negotiation occurs. BAP uses this unique identifier to differentiate the various links in a multilink bundle.

# Configuring PPP

The router supports PPP over the following, separately and as members of a multilink bundle:

- Ethernet (but not multilinked with other PPPoE interfaces)
- VLAN interfaces on Ethernet switch ports (but not multilinked with other PPPoE interfaces)
- synchronous links (see Chapter 9, Interfaces) (routers with PIC bay only)
- ISDN calls (see Chapter 11, Integrated Services Digital Network (ISDN)) (routers with PIC bay only)
- ACC calls (see Chapter 19, Asynchronous Call Control)
- MIOX calls (see Chapter 13, X.25)(routers with PIC bay only)
- L2TP calls (see Chapter 20, Layer Two Tunnelling Protocol (L2TP))
- TDM groups (see *Chapter 12, Time Division Multiplexing (TDM)*) (routers with PIC bay only)

PPP can be used on the router to carry:

- IP
- IPX
- DECnet
- AppleTalk routing protocols
- bridged protocols
- compressed data
- encrypted data

Some interface and port types mentioned in this chapter may not be supported on your router. The interface and port types that are available vary depending on your product's model, and whether an expansion unit (PIC, NSM) is installed. For more information, see the Hardware Reference for the router.

To create a PPP interface, use the command:

```
create ppp=ppp-interface over=physical-interface [other-
   options...]
```

To remove an entire PPP interface, use the command:

```
destroy ppp=ppp-interface
```

To add physical interfaces to the PPP interface to form a multilink bundle, use the command:

```
add ppp=ppp-interface over=physical-interface [other-
   options...]
```

To modify interface parameters after the interface has been created, use the command:

```
set ppp=ppp-interface [over=physical-interface] [other-
   options...]
```

If an ISDN call is being added as the physical interface, multiple physical interfaces can be added using the NUMBER parameter. For example, the following command adds two identical ISDN calls (named "HeadOffice") as physical interfaces to PPP interface 0:

```
add ppp=0 over=isdn-headoffice num=2
```

To selectively delete members of a multilink bundle, use the command:

```
delete ppp=ppp-interface over=physical-interface
```

To disable an entire PPP interface, use the command:

```
disable ppp=interface
```

Disabling a dynamic PPP interface destroys it. To re-enable or reset a static PPP interface, use the commands:

```
enable ppp=interface
```

```
reset ppp=interface
```

One of the features of PPP is the negotiation of options for each protocol using the link. All options have a default to which the option is set if either end of the PPP link does not wish the option to be different from the default. The LCP attempts to negotiate the Maximum Receive Unit (MRU), Authentication Protocol, Link Quality Reporting (LQR), Magic Number, Asynchronous Control Character Map (ACCM), and Maximum Received Reconstructed Unit (MRRU) options. All other possible options are set to the defaults specified in the relevant RFC.

To negotiate Van Jacobson's TCP/IP header compression, use the command:

```
add ip interface=interface ipaddress={ipadd|dhcp} vjc=on
   [other-options...]
```

For more information about turning on Van Jacobson's TCP/IP header compression, see Chapter 22, Internet Protocol (IP).

**Important** Van Jacobson's TCP/IP header compression should not be enabled on a multilink PPP interface.

To display information about a PPP interface, use the command:

```
show ppp[=ppp-interface] [configuration|count|idletimer|
   multilink]
```

If **configuration** is specified, the settings of configuration parameters such as LQR and restart timers are displayed. If **count** is specified, counters from the interface MIB and counters for the users of the interface are displayed. If **idletimer** is specified, the configured and current values of the idle timer are displayed. If **multilink** is specified, information about the multilink bundle associated with the interface is displayed. The display includes the number of links in the bundle, the number of packets fragmented, the number of packets or fragments in the multilink receive queue, and information about the sequence numbers on the multilink bundle. If no optional parameters are specified, a summary of the configured PPP interfaces, the physical interfaces used and the Network Control Protocols (NCPs) in use is displayed.

# Link Quality Management

Link quality management determines the quality of a PPP link. A Link Quality Report (LQR) packet is transmitted down the link by the router at regular intervals. This LQR packet contains information that determines how many packets are being lost on the link. The interval between transmissions of LQR packets is determined by the LQR timer value obtained from the peer during the negotiation of the LQR LCP option. To configure this timer at the peer, which defines how often the peer expects to see an LQR packet, use the commands:

```
create ppp=ppp-interface over=physical-interface lqr=period

add ppp=ppp-interface over=physical-interface lqr=on

set ppp=ppp-interface over=physical-interface lqr=period
```

If an LQR packet is not seen by the peer within the configured timer value the link is deemed to have failed and is reset.

Each LQR packet also contains the magic number determined during the LCP negotiation process. If the magic number in an incoming LQR packet is the same as the local magic number, then the link is deemed to be in loopback mode and is reset.

# Multilink PPP

PPP provides a mechanism for combining a number of PPP links into a single bundle of links, whose bandwidth is the sum of the bandwidths of the individual links. This mechanism is known as multilink PPP (MP) and is described in RFC 1990. A PPPoE link cannot be multilinked with other PPPoE links.

**Important** Van Jacobson's TCP/IP header compression should not be enabled on a multilink PPP interface.

When a packet is transmitted over a multilink bundle it is encapsulated by a multilink header that includes information to allow the packets sent over the links in the bundle to be sequenced. This gives the multilink bundle the same properties as a single PPP link. This encapsulation also includes information that allows large packets to be fragmented, spreading the data across a number of links and giving better packet throughput in some circumstances.

When a packet is about to be transmitted across a PPP multilink bundle, a decision is made as to which link to use to transmit the packet. all link speeds in the multilink bundle are the same, and packets are being transmitted at a rate so that each packet has been transmitted before the next packet arrives for transmission, a round-robin scheme is used to choose between links. If there is a choice between two or more equally desirable links, the packet is sent on the link that has been least recently used. Rotating traffic in this way prevents links from remaining idle for long periods of time and reduces the number of null fragments that must be transmitted during idle periods.

Both static and dynamic PPP interfaces can be multilinked. The Endpoint Discriminator LCP option ("Endpoint Discriminator Option" on page 15-6) lets a single dynamic PPP interface accept and bundle more than one call. If two or more dynamic PPP calls are made from the same peer with the same authentication information, they are bundled together to form a multilink interface.

# Bandwidth Allocation Protocol

The Bandwidth Allocation Protocol (BAP), defined in RFC 2125, provides a mechanism for two PPP peers to manage the bandwidth available to the protocols using a multilink PPP bundle by negotiating gracefully to add and remove links from the multilink bundle. The negotiation process allows each peer to choose the algorithm used to determine when to add or remove links in the multilink bundle.

The Bandwidth Allocation Control Protocol (BACP), defined in RFC 2125, is a standard PPP NCP protocol used to negotiate the use of BAP on a multilink PPP interface. BACP is negotiated once per multilink bundle. If BACP is negotiated on any of the links in a multilink bundle, it is opened for all of the links in the bundle. BACP must be successfully negotiated before BAP can be used.

The Favoured Peer Option is the only option defined for BACP and determines which peer is favoured when both peers simultaneously transmit the same BAP request. Each peer negotiates a 4-octet magic number, which is successfully negotiated when the two magic numbers are different. The favoured peer is the peer with the lowest magic number.

After BACP reaches the opened state, either peer can request that another link be added to the bundle by sending a BAP *Call-Request* or *Callback-Request* packet. A *Call-Request* packet is sent if the peer wishes to originate the call for the new link, and a *Callback-Request* packet is sent if the peer wishes its remote peer to originate the call for the new link.

A peer can also request that a link be dropped from the bundle. A BAP *Link-Drop-Query-Request* packet is sent to the remote peer to negotiate dropping a link. The link remains active as long as the remote peer considers the link necessary and rejects the *Link-Drop-Query-Request*. A peer can force the dropping of a link without negotiation by sending an LCP *Terminate-Request* packet on the link.

To configure BAP when a PPP interface is created, use the command:

```
create ppp=ppp-interface over=physical-interface bap={on|off}
   bapmode={call|callback}
```

To modify an existing PPP interface, use the command:

```
set ppp=ppp-interface bap={on|off} bapmode={call|callback}
```

By default, BAP is enabled (**on**). If **bap** is disabled, PPP uses the **uprate**, **uptime**, **downrate** and **downtime** parameters to manage bandwidth on demand (see "Bandwidth on Demand" on page 15-13).

# Dial-On-Demand

A PPP interface can be configured for dial-on-demand over:

- an ISDN call (routers with PIC bay only)
- an ACC call (routers with PIC bay only)
- a synchronous port controlling a modem connected to the PSTN (routers with PIC bay only)
- PPPoE (on the host only)
- L2TP

To configure dial-on-demand operation use the commands:

```
create ppp=ppp-interface over=physical-interface
   idle={on|seconds} [other-options...]

set ppp=ppp-interface over=physical-interface
   idle={on|seconds} [other-options...]
```

The call is activated when there is traffic to transmit over the PPP interface. The call is disconnected when the link has been idle for the specified time. If the **idle** parameter is set to **off**, the dial-on-demand feature is disabled. The default for the **idle** parameter is **off**, and **on** sets the idle time to 60 seconds. To display the configured and current timer values, use the command:

```
show ppp idletimer
```

# Link Backup

A PPP link can be configured as a backup to provide redundancy for another PPP link.

In theory, any kind of PPP link can be configured to back up any other PPP link. A common application is an on-demand call configured as a backup to a permanent line. Figure 15-1 on page 15-12 shows this configuration for an ISDN call as backup to a leased line (routers with PIC bay only).

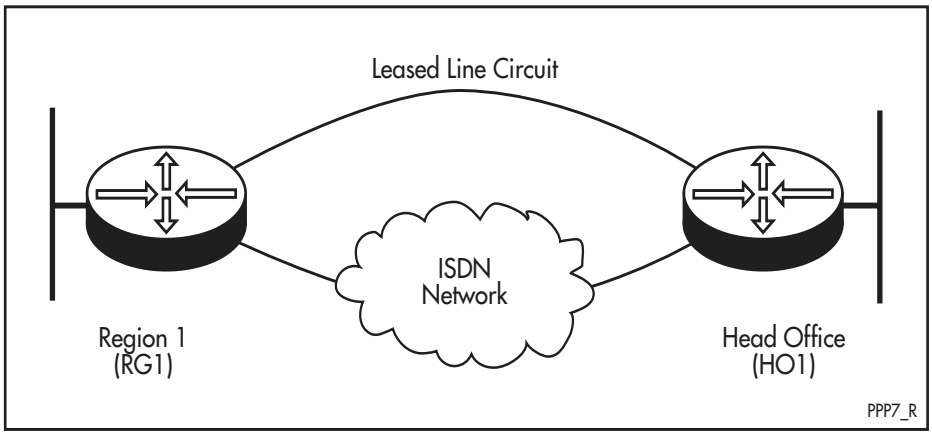
On-demand links that can be configured on the router include:

- ISDN links (routers with PIC bay only)
- ACC links (routers with PIC bay only)
- synchronous ports controlling a modem connected to the PSTN (routers with PIC bay only)
- PPPoE links
- MIOX circuits (routers with PIC bay only)
- L2TP links

Permanent connections that can be configured on the router include:

- permanent synchronous links (routers with PIC bay only)
- TDM links (routers with PIC bay only)
- MIOX circuits (routers with PIC bay only)

Figure 15-1: Example network configuration for leased line backup




To configure redundancy, create, or modify the PPP interface for the permanent link, use one of the commands:

```
create ppp=ppp-interface over=physical-interface type=primary
   configure=value [restart=seconds] [other-options...]

set ppp=ppp-interface [over=physical-interface] type=primary
   configure=value [restart=seconds] [other-options...]
```

To create or modify the PPP interface for the backup link, use the commands:

```
create ppp=ppp-interface over=physical-interface
   type=secondary [other-options...]

set ppp=ppp-interface [over=physical-interface] type=primary
   configure=value [restart=seconds] [other-options...]
```

When the primary link (**type=primary**) fails, Link Quality Monitoring (LQM) or ECHO detects the failure and resets the link. This causes configure requests to be transmitted. The interval between successive retransmissions of configure request packets is set by using the **restart** parameter and is 3 seconds by default. If the primary link fails to open after the number of requests specified by the **configure** parameter, the backup call is activated and traffic is redirected over the backup link. PPP continually attempts to reopen the primary link, and when the primary link is restored, the backup call is deactivated and traffic is redirected over the primary link again.

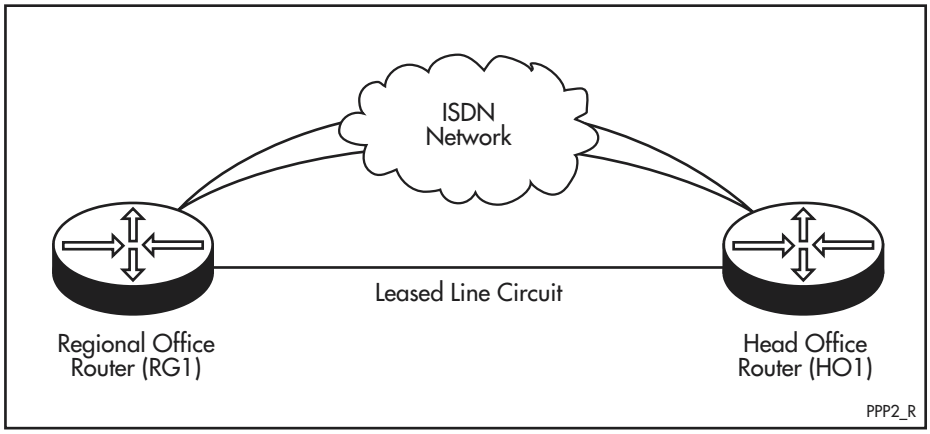
When the permanent synchronous, or TDM link enters loopback mode, LQR detects the loopback and resets the link. When the link tries to re-open, the negotiation of the Magic Number LCP option fails and the backup calls are activated. When the link leaves loopback mode, the negotiation succeeds, the permanent link reopens, the backup call is deactivated, and traffic is redirected over the permanent link.

# Bandwidth on Demand

A PPP interface can be configured to provide bandwidth on demand (Figure 15-2 on page 15-13) over a number of:

- ISDN channels (routers with PIC bay only)
- ACC calls
- PPPoE connections
- L2TP calls

Figure 15-2: Example network configuration for bandwidth on demand




One application of bandwidth on demand is the use of ISDN calls to provide additional bandwidth to a leased line during peak load periods. This application is best suited to a network connection that has a fairly constant load most of the time, but is overloaded during peak periods. A leased line with sufficient capacity to handle the normal loading is supplemented by a connection to an ISDN service. This avoids the high cost of a leased line capable of handling peak loads but which is under-utilised most of the time.

Bandwidth on demand can be configured to use multiple on-demand connections, instead of a leased line, to provide the bandwidth required at any one time. This application is best suited to a network connection that has a variable and irregular load.

To configure bandwidth on demand, create a PPP interface, then add other physical interfaces to it. There are two options:

- Configure all physical interfaces to open on demand. In this case no channels are available when there is no traffic. To create the PPP interface, use the command:

```
create ppp=ppp-interface over=physical-interface-1
    type=demand [downrate=seconds] [downtime=seconds]
    [uprate=seconds] [uptime=seconds] [other-options...]
```

- Configure one physical interface as a primary interface, and all others to open on demand. In this case a call is always available, even when demand falls off completely. If a primary link is required, create the PPP interface by using the command:

```
create ppp=ppp-interface over=physical-interface-1
    [type=primary] idle={on|seconds} [other-options...]
```

In both cases, add further demand links to the same PPP interface by using the command:

```
add ppp=ppp-interface over=physical-interface-n type=demand
    [downrate=seconds] [downtime=seconds] [uprate=seconds]
    [uptime=seconds] [other-options...]
```

To trigger the addition and removal of calls, the total utilisation of the PPP interface as a percentage of the maximum bandwidth of the PPP interface is measured every second. Each time the utilisation remains above the threshold specified by the **uprate** parameter for a time longer than that specified by the **uptime** parameter, a new call is made, increasing the bandwidth of the PPP interface. When each new link is added, the total utilisation of the interface decreases. However, this decrease is momentary when the rate of utilisation is increasing. When the rate of utilisation decreases again, each time the utilisation drops below the threshold specified by the **downrate** parameter for a time longer than that specified by **downtime** parameter, a call is disconnected and the total bandwidth of the PPP interface is decreased.

For **primary** interfaces, use the **idle** parameter to determine when the link closes when there is no traffic for the specified time. The default for the **idle** parameter is **off**, and **on** sets the idle time to 60 seconds.

# Always On/Dynamic ISDN (AODI)

A PPP interface can be configured to provide AODI (*Always On/Dynamic ISDN*) by specifying a MIOX circuit as the primary link and an ISDN call as the demand link in the **add ppp** and **create ppp** commands:

```
create ppp=0 over=miox3-aodi idle=40000000

add ppp=0 over=isdn-aodi type=demand num=2
```

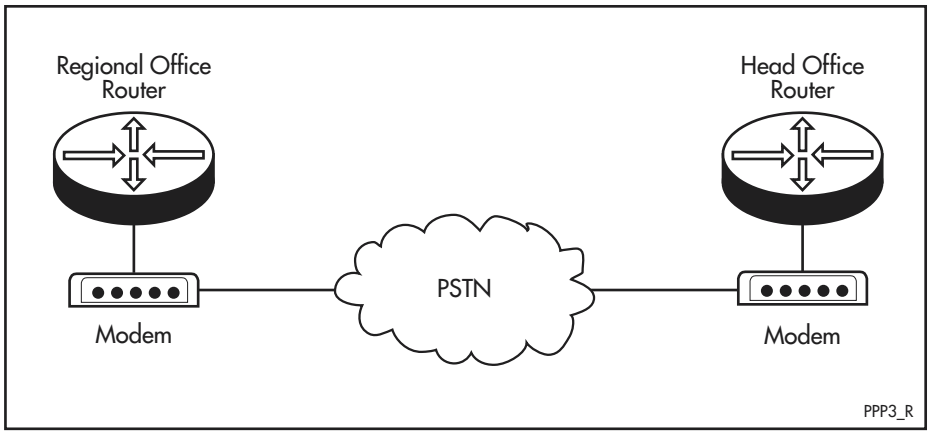
See "Always On/Dynamic ISDN (AODI)" on page 11-45 of Chapter 11, Integrated Services Digital Network (ISDN) for more information about configuring AODI.

# Synchronous Dialling

A PPP interface over a synchronous link can be configured to control a modem connected to the PSTN for router-to-router connections (Figure 15-3 on page 15-15). The functionality that uses V.25bis DTR support enables synchronous dialling to be used as an alternative to ISDN or ACC calls for dial on demand and leased line backup applications.

Synchronous links are available only on routers with PIC bays only.

Figure 15-3: Example network configuration for dial-on-demand using modems




**Important** Synchronous dialling is slower at setting up and taking down calls than ISDN, so the response time for handling events, such as bringing up additional links in a dial on demand application, is longer.

To configure the router to control a modem connected to a synchronous port, use the commands:

```
create ppp=ppp-interface over=syn0 modem=on [idle=on] [other-
    options...]

set ppp=ppp-interface over=syn0 modem=on [idle=on] [other-
    options...]
```

The router controls the modem using the modem's DTR input signal. For bidirectional calling each modem must be configured with a number to dial when the DTR signal from the router is raised. Each modem must also be configured to raise the DSR signal to the router when a call has been received and answered.

A modem makes a call when the router to which it is connected asserts DTR. The modem at the remote end of the link answers the call when its DTR signal is high. A modem that answers a call raises its DSR signal to the attached router, which responds by activating the PPP interface. A call is disconnected when the DTR signal of one of the modems is driven low. These conditions mean that DTR must be idle high so the modem is always ready to answer a call.

To make a call, the router drives DTR low and then high, leaving it ready to drop the call. To drop a call, the router drives DTR low and then high leaving it ready to answer a call. This has the side effect of causing the modem that drops the call to make another call. However, it encounters an engaged signal because the modem at the remote end of the link is still off-hook, and it then hangs up.

A potential problem with a bidirectional link is call collisions resulting from both modems attempting to dial each other at the same time. In dial on demand applications, a call collision may occur when traffic to be sent appears at both ends of the link at the same time. In leased line backup applications, a call collision may occur if the failure of the primary link is detected at the routers at each end of the link at the same time. The solution is a simple random backoff scheme. When a call collision occurs, each modem detects that the remote modem is engaged and hangs up after 15 seconds. The router detects that the call has failed after 10 seconds and toggles DTR a random time later. This time varies between 15 and 45 seconds. If another collision occurs, the routers back off and try again until one of the modems establishes a call. This may take several minutes at most.

Another potential problem occurs when the routers are turned on. DTR is asserted causing the connected modem to initiate a call. If both routers are powered on at the same time, a call collision occurs. In this case the random backoff scheme is not used, and the modems hang up after 15 seconds and do not try again. However, if one router is powered up or reset, the call is successful as long as the modem and router at the remote end of the link are switched on. To avoid this extraneous call on power up, it is advisable to turn on each modem after the router to which it is connected has been powered up.

# PPP Over Ethernet

PPP over Ethernet, defined in RFC 2516, *A Method of Transmitting PPP Over Ethernet*, provides the ability to connect a network of hosts over a single bridging access device to a remote *Access Concentrator*. An Access Concentrator may offer multiple services. A PPP over Ethernet link is a point-to-point connection between a host and a single service on an Access Concentrator. Typically, the bridging access device is a DSL or cable modem to which local hosts are connected via Ethernet, and the remote Access Concentrator is a server at an ISP. It is possible to tunnel a PPP over Ethernet session through an L2TP tunnel. For details, refer to Chapter 20, Layer Two Tunnelling Protocol (L2TP). PPP over Ethernet enables multiple hosts at a remote site to share the same access device, while providing the access control and billing functionality of dial-up PPP connections.

The router can be configured as an Access Concentrator. Remote devices can access services configured on the router. The router can also be configured as a host, creating PPP links over Ethernet to services on remote Access Concentrators.

PPP over Ethernet has two distinct stages. In the *Discovery Stage*, the host discovers all the available Access Concentrators that offer the required service and then selects one. The host broadcasts an *Initiation* packet specifying the name of the required service or indicating that any service is acceptable. If a service name is specified, Access Concentrators that support the requested service respond with *Offer* packets that specify the Access Concentrator's unicast Ethernet Address.

If the Initiation packet indicated that any service was acceptable, all Access Concentrators that have services available respond with an *Offer* packet that specifies the Access Concentrator's unicast Ethernet Address. The host then selects an Access Concentrator and sends a *Request* packet specifying the name of the required service. The Access Concentrator responds with a *Session Confirmation* packet. When the Discovery Stage is complete, the host and the selected Access Concentrator have the information they need to create the point-to-point connection over Ethernet. In the *Session Stage* the host and the Access Concentrator exchange PPP packets.

# PPP over Ethernet Client Mode

To configure the switch as a PPPoE client, create a PPP interface over an Ethernet service by using the command:

```
create ppp=ppp-interface over=physical-interface
    [other-ppp-options]...
```

where *ppp-interface* is the PPP interface number and *physical-interface* is the name of the physical interface in the format ETH*n-servicename*. To specify that any service name is acceptable, use the special service name ANY. Service names may be up to 18 characters long and are usually supplied by the ISP providing the service.

A PPPoE link cannot be multilinked to another PPPoE link, but can be multilinked to other PPP calls by using the command:

```
add ppp=ppp-interface over=physical-interface
    [other-ppp-options]...
```

PPPoE can be configured over ATM by first creating a virtual Ethernet interface over an ATM channel, then configuring PPP over the virtual Eth interface. See the **create eth** command on page 9-26 of Chapter 9, Interfaces.

PPPoE can be configured on VLAN interfaces in both Client and Access Concentrator modes. To configure PPPoE in Client Mode, the physical-interface parameter VLAN*n-servicename* has been added, where *servicename* is 1 to 18 characters long, and for a PPPoE client is usually supplied by the ISP providing the service. To specify that any service name is acceptable, you can use the special service name ANY.

# PPP over Ethernet Access Concentrator Mode

To configure the router as a PPPoE Access Concentrator, add one or more access concentrator services, and enable Access Concentrator mode.

A PPPoE Access Concentrator service uses a PPP template to determine link parameters used for incoming PPPoE traffic. See "Templates" on page 15-18 for more information about creating PPP templates. PPPoE AC services cannot use the default template. To create a template, use the command:

```
create ppp template=template [other-options...]
```

Then create a service, giving it a name up to 18 characters long by using the command:

```
add ppp acservice=service-name template=ppp-template
    [acradius={off|on}] [maxsessions=1..512] [vlan={none|
    0..4094}]
```

To enable the router to provide these Access Concentrator services, use the command:

```
enable ppp accessconcentrator
```

The router does not respond to PPPoE discovery initiation packets unless Access Concentrator mode has been enabled with this command, even if it has the requested service defined.

# Templates

Templates are used whenever a call is created dynamically. The template is specified when defining the call. Dynamic PPP interfaces are created in response to a request from a lower layer to create a new PPP interface. PPP templates enable a range of configuration options available on static PPP interfaces to be applied to dynamic PPP interfaces.

A template is a blueprint for the configuration of dynamic PPP interfaces, specifying the link parameters. To create a new template, use the command:

```
create ppp template=template [copy=template]
   [authentication={chap|either|pap|none}] [bap={on|off}]
   [bapmode={call|callback}] [cbdelay=1..100]
   [cbmode={accept|off|request}] [cbnumber=e164number]
   [cboperation={e164number|userauth}]
   [compalgorithm={predictor|staclzs}] [compression={on|off|
   link}] [debugmaxbytes=16..256] [description=description]
   [echo={on|off|period}] [encryption={on|off}]
   [fragment={on|off}] [fragoverhead=0..100] [idle={on|off|
   time}] [indatalimit={none|1..65535}] [ippool={pool-name|
   none}] [iprequest={on|off}] [login={all|radius|tacacs|
   user}] [lqr={on|off|period}] [magic={on|off}]
   [maxlinks=1..64] [mru={on|off|256..1656}] [mtu=256..1500|
   256..1492] [multilink={on|off}] [nullfragtimer=time]
   [onlinelimit={none|1..65535}] [outdatalimit={none|
   1..65535}] [password=password] [predcheck={crc16|
   crcccitt}] [rechallenge={on|off|360..3600}] [restart=time]
   [staccheck={lcb|sequence}] [starentity=1..255]
   [totaldatalimit={none|1..65535}] [username=username]
```

Once a template has been created, it can be associated with the required dynamic link by using the commands:

```
add acc call=name asyn=port-number ppptemplate=template

set acc call=name ppptemplate=template

add isdn call=name number=number precedence={in|out}
   ppptemplate=template

set isdn call=name ppptemplate=template

add l2tp ip=ipadd-ipadd ppptemplate=template

add ppp acservice=service-name template=ppp-template
   [acradius={off|on}] [maxsessions=1..512] [vlan={none|
   0..4094}]

set ppp acservice=service-name [acradius={off|on}]
   [maxsessions=1..512] [template=ppp-template] [vlan={none|
   0..4094}]
```

When the lower layer activates a call that creates a dynamic PPP interface, PPP uses the associated template to create and configure the dynamic PPP interface. If no PPP template is specified in these commands, the default template is used (except for PPPoE AC services, for which a template must be created). The default template has defaults for all template parameters.

Configuration templates cannot be associated with TDM or SYN interfaces.

To modify or delete an existing template, use the commands:

```
set ppp template=template [options...]

destroy ppp template=template
```

To display the list of currently defined templates, including the default template (*pppT33*), use the command:

```
show ppp template
```

To display the configuration of a specific template, use the command:

```
show ppp template=template
```

To enable or disable the full range of PPP debugging options on a PPP template, use the commands:

```
enable ppp template=template debug={all|auth|bapstate|
   callback|demand|enco|lcp|ncp|pkt|utilisation}[,...]
   [asyn=port-number] [timeout={none|1..4000000000}]
   [numpkts={cont|1..4000000000}]

disable ppp template=template debug={all|auth|bapstate|
   callback|demand|enco|lcp|ncp|pkt|utilisation}[,...]
```

Table 15-11 on page 15-78 lists the debugging options and their meanings. Any dynamic PPP interface created from a template that has debugging enabled displays the requested debug information. Debugging ceases when the dynamic PPP interface is destroyed.

# PPP Callback

The PPP callback feature allows a PPP link to be configured to accept callback requests or to make callback requests. A callback request is made during the LCP negotiation using the LCP callback option that is defined in RFC 1570 as an LCP extension. This option contains a callback operation that specifies how the peer determines the number to use when making the call back, and contains a message field whose contents are dependent on the operation being used.

Dynamic PPP interfaces can support PPP callback provided the dynamic PPP interface is created using a PPP template in which PPP callback has been configured.

A PPP link is configured to make callback requests with the commands:

```
create ppp=ppp-interface over=physical-interface
   cbmode=request

set ppp=ppp-interface over=physical-interface cbmode=request

create ppp template=template [description=description]
   cbmode=request

set ppp template=template cbmode=request
```

A PPP link is configured to accept callback requests with the commands:

```
create ppp=ppp-interface over=physical-interface
   cbmode=accept authentication={chap|pap}

set ppp=ppp-interface over=physical-interface cbmode=accept

create ppp template=template [description=description]
   cbmode=accept authentication={chap|pap}

set ppp template=template cbmode=accept
```

For static PPP links, callback is supported only over ISDN calls.

Two types of callback request operations are supported by the router—user authentication and E.164 number. The user authentication callback operation specifies that the number to call back is contained in the User Authentication Database and is obtained during authentication just prior to the call being brought down. To configure user authentication callback, use the command:

```
create ppp=ppp-interface over=physical-interface
   cboperation=userauth

set ppp=ppp-interface over=physical-interface
   cboperation=userauth

create ppp template=template [description=description]
   cboperation=userauth

set ppp template=template cboperation=userauth
```

The E.164 number operation specifies that the callback number is contained in the message field of the callback option. When the E.164 number operation is configured for requesting a callback, the E.164 number must also be provided. To configure E.164 number callback, use the command:

```
create ppp=ppp-interface over=physical-interface
   cboperation=e164number cbnumber=e164number

set ppp=ppp-interface over=physical-interface
   cboperation=e164number cbnumber=e164number

create ppp template=template [description=description]
   cboperation=e164number cbnumber=e164number

set ppp template=template cboperation=e164number
   cbnumber=e164number
```

The **cboperation** parameter is valid when the callback mode is set to request callback.

A PPP link that is configured to accept callback requests must also be configured to request authentication. This is necessary to prevent unauthorised peers from requesting a callback.

When a callback request is accepted, and authentication succeeds, the call is brought down and a call is made back to the peer making the request. If authentication fails the link is brought down and no call back is made. A delay between bringing down the call and attempting to make the call back can be configured in order to cope with any variable delays in bringing down the call (for example, due to any differences in ISDN switches). The units of this delay are tenths of seconds. To configure it, use the command:

```
create ppp=ppp-interface cbdelay=1..100 cbmode=request

set ppp=ppp-interface cbdelay=1..100

create ppp template=template cbdelay=1..100 cbmode=request

set ppp template=template cbdelay=1..100
```

The **cbdelay** parameter is valid when the callback mode is set to accept callback requests.

# Magic Number

The magic number option is used for loopback detection. A PPP interface that is looped back does not enter an opened state when the magic number option is enabled. The magic number option is enabled with the **magic** parameter of the **add ppp** command on page 15-50, the **create ppp** command on page 15-56, the **set ppp** command on page 15-83, the **create ppp template** command on page 15-63 and the **set ppp template** command on page 15-92.

# MSS Clamping

Maximum Segment Size (MSS) clamping functionality allows you to control and prevent IP packet fragmentation. Configuration options enable you to:

- Apply a fixed value for the MSS header space (MSSH), via the command line interface.
- Set an MSSH value of 40 to 200 bytes.

## Overview

MSS clamping reserves a set amount of space within a TCP packet for the header, which in turn limits the amount of space that may be occupied by the data (payload). The purpose of MSS clamping is to set the header space value to a level that prevents fragmentation from occurring.

**Maximum Transmission Unit and Maximum Segment Size**

The Maximum Transmission Unit (MTU) is the maximum number of bytes per packet that may be transmitted by the network interface. If a single packet exceeds the MTU, it is divided into smaller packets before being transmitted.

For a TCP packet, the MTU can be illustrated by the following equation:

```
MTU = Header Size + Maximum Segment Size
```

where:

- Header Size is the size of the packet header. More accurately, it is the size of the combined headers that are applied to the MSS (payload) by the TCP, IP and PPPoE protocols.
- Maximum Segment Size is the largest amount of TCP data, in bytes, that the router can transmit or receive in one single data packet.

The MTU of the underlying network protocol, e.g. PPPoE, can be manually set using the **set interface mtu** command on page 9-48 of Chapter 9, Interfaces.

**Data Transmission and MSS clamping**

Each TCP device uses its MSS value to let other devices know what is the highest allowable amount of data it can receive in a single packet. Although devices in a TCP/IP connection calculate the amount of data to send in a packet based on variables, such as the current window size and various algorithms, the amount of actual data in a single packet can never exceed the MSS of the device the packet is being sent to.

Various protocols are applied to data as it passes through a computer network. Each of these protocols adds its own header, which encapsulates the information. This encapsulation increases the size of the packet being transmitted, potentially exceeding the MTU of devices on the TCP/IP/PPPoE link.

When the packet exceeds the defined MTU for an interface, IP will fragment (or split) the packet. Packet fragmentation can be costly for the following reasons:

- decreased throughput, the amount of data transferred or processed in a specified amount of time.
- networks that are explicitly set to drop fragmented packets suffer communication loss.

Setting the MSS clamping value at an appropriate limit prevents fragmentation by reserving a set amount of header space within a TCP packet. This in turn limits the amount of space that can be consumed by the data payload.

### Example

The typical MTU value of a PPPoE interface is approximately 1495 bytes, and you estimate that your combined TCP, IP and PPPoE headers will be no greater than 70 bytes. You therefore decide to set your **mssheader** value to 70.

```
set ppp=0 mssheader=70
```

By setting the **mssheader** parameter to 70 bytes, you are reserving this amount of space for the headers. For the MTU of 1495, this leaves space within the packet for 1425 bytes of data.

## Authentication Protocols

The router supports two authentication protocols: the Password Authentication Protocol (PAP) and the Challenge-Handshake Authentication Protocol (CHAP). These protocols are typically used with PCs and hosts that connect to the router through on-demand calls (for example, ISDN calls or modems attached to the asynchronous or synchronous ports of the router) but may also be applied to network connections that use dedicated leased lines.

The PPP Link Control Protocol (LCP) establishes, configures, and tests data link connections between devices. Part of the process of configuring a link is the negotiation of various options, including an authentication protocol, which is performed before allowing Network Layer protocols to transmit data over the link.

The Authentication phase is optional. It takes place after the PPP link has been established (the Link Establishment phase) and before proceeding to the Network-Layer Protocol phase, if authentication has been negotiated by the router at either end of the link.

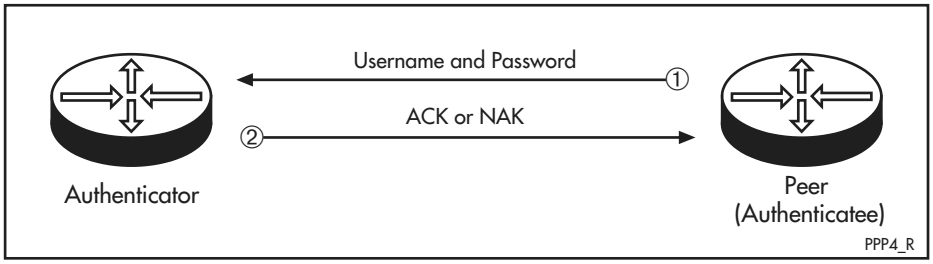
#### Device Definitions

The following sections on PAP and CHAP authentication use the definitions in RFC 1334 and RFC 1994 to describe the roles of each device. These definitions are as follows: the device performing the authentication is known as the *authenticator*; the device being authenticated is known as the *peer*. A router can be configured to be **both** an *authenticator* and a *peer*.

# Password Authentication Protocol (PAP)

The Password Authentication Protocol (PAP) is a relatively simple authentication protocol that allows a peer to establish its identity by repeatedly transmitting a user name/password pair to an authenticator until the authenticator acknowledges the peer or terminates the link. The peer requesting authentication controls the process; the authenticator simply responds to requests (Figure 15-4).

Figure 15-4: The Password Authentication Protocol (PAP) authentication process




When a peer requests authentication, it transmits a login name and password that the router compares against entries in the User Authentication Database and any defined TACACS or RADIUS servers.

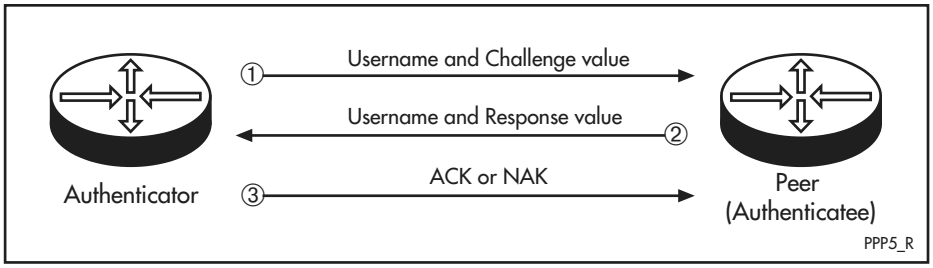
Transmitted passwords are not encrypted, and since the peer always uses the same user name/password pair there is no protection from playback or repeated trial-and-error attacks. PAP provides a similar level of security to a normal remote login.

# Challenge-Handshake Authentication Protocol (CHAP)

The Challenge-Handshake Authentication Protocol (CHAP) is a more robust protocol that provides both authentication during the Link Establishment phase and verification at random intervals during the Network-Layer Protocol phase. Either the router's User Authentication Database or a connected RADIUS server can be used to authenticate peers using CHAP.

CHAP is controlled by the authenticator, which sends a *challenge* message containing an identifier and a unique challenge value to the peer. The peer responds with a user name and value calculated by applying a *one-way hash function* (MD5) to a string created by concatenating the identifier, the password for the user name and the challenge value. The authenticator compares the response against its own computation of the function, using the user name to look up the password in the User Authentication Database. If the values match, the authentication is acknowledged, otherwise the link is terminated (Figure 15-5).

Figure 15-5: The Challenge Handshake Authentication Protocol (CHAP) authentication process



The challenge is repeated at intervals during the Network-Layer Protocol phase to ensure that there has been no change to the link. Each challenge uses a different identifier and challenge value. The identifier value changes in a predictable way (typically the value of a regularly incremented counter), but the challenge value is a unique and random value. The interval between challenges varies randomly between 5 and 15 minutes.

The CHAP rechallenge function is controlled by the **rechallenge** parameter of the **create ppp** or **set ppp** command. This parameter specifies if rechallenge is on or off, or specifies a maximum rechallenge period. PPP calculates a random period for the CHAP rechallenge between the maximum rechallenge period and a minimum of 5 minutes. If **on** is specified, the CHAP rechallenges take place with a maximum rechallenge period of 15 minutes. If **off** is specified, then CHAP rechallenges do not take place because the CHAP rechallenge is disabled. If a *time* is specified, then this time is the upper limit on the rechallenge time calculation. The default is **on**.

The repeated challenges and changing identifier and challenge values provide protection against both playback and trial-and-error attacks. The uniqueness and random nature of the challenge value prevents an attacker from tricking a peer into responding to a challenge then using the response to masquerade as the peer to an authenticator. CHAP relies on the password being known to both the authenticator and the peer, although the password is not transmitted over the link.

In the case of a PC using PPP to connect to an authenticating router, the PC responds to the first challenge by sending the user name and the value derived using the one-way hash function to the router. The router compares the response to its own calculation of the value, using the user name to retrieve the password from either its User Authentication Database or from a connected RADIUS server. The PC continues to store the user name and password in its connection software to use for rechallenges by the router.

A router acting as a peer must have its user name and password set in the PPP interface. These are defined using the **username** and **password** parameters in these commands:

- **create ppp** command on page 15-56
- **set ppp** command on page 15-83
- **create ppp template** command on page 15-63
- **set ppp template** command on page 15-92

If the user name is not set in the PPP interface, it will default to the router system name, which can be set using the **set system name** command on page 4-31 of Chapter 4, Configuring and Monitoring the System.

A router acting as an authenticator will need the user names and passwords of its peers held on either a connected RADIUS server or on its User Authentication Database. To enter peer user name and password details onto a router's database, see Chapter 41, User Authentication. If the authenticator router also acts as a peer router, it will need to have its user name and password defined in the PPP interface, as described above. If the router does not have to act as a peer, then the password and user name parameters do not need to be defined.

# Router configuration

The router may be configured in one of the following modes:

- as a peer in a one-way authentication scheme
- as an authenticator in a one-way authentication scheme
- as both a peer and an authenticator in a two-way authentication scheme

Table 15-3: Parameters of the **create ppp** command for authentication

| To configure a device as | To use PAP specify AUTHENTICATION= | To use CHAP specify AUTHENTICATION= | Specify USERNAME and PASSWORD |
|---|---|---|---|
| Authenticator | **pap** (or **either** if CHAP is never available) | **chap** or **either** | No |
| Peer | **none** (the default) | **none** (the default) | Yes |
| Both Authenticator and Peer | **pap** (or **either** if CHAP is never available) | **chap** or **either** | Yes |

An explanation of the configuration for each of these modes is described next.

## Configuring the Router as a Peer

The router can be configured as the peer in a one-way authentication scheme. When the router makes a call to a remote device (e.g. another router configured as an authenticator), the username and password that have been configured on the peer are used by the remote device to authenticate the peer. The remote device determines whether the username and password are valid, and accepts or rejects the connection. This is the most common PPP authentication configuration. A typical example is a router configured to dial into a remote ISP. The ISP provides clients with a username and password. The ISP's connection server expects the client's router to supply the username and password when it makes a call to the ISP.

To configure the router as a peer, create the PPP interface by using the command:

```
create ppp=ppp-interface over=physical-interface
   username=username password=password
```

To modify an existing PPP interface, use the command:

```
set ppp=ppp-interface username=username password=password
```

where *username* and *password* are supplied by the administrator of the authenticator (for example, the ISP).

Neither **pap** or **chap** have been explicitly configured in this example so the router does not request authentication from remote devices during LCP negotiation. This is appropriate for the example above. Most ISPs configure their connection servers to request authentication only, and not to respond to authentication requests. If you configure the client router to request authentication but the ISP's connection server is not configured to respond to authentication requests (as is typical), the ISP's connection server refuses the connection from the client router and all connection attempts fail.

If a username is not set using these PPP commands, the peer router's system name is used as the username. The system name is set using the **set system name** command on page 4-31 of Chapter 4, Configuring and Monitoring the System. The password can be set using the PPP commands.

## Configuring the Router as an Authenticator

The router can be configured as the authenticator in a one-way authentication scheme. When the router receives a call from a remote device (e.g. a user dialling in via a modem or another device configured as a peer), it requests authentication from the remote device. The remote device supplies a username and password that the router validates before accepting or rejecting the connection. A typical example is an ISP configuring a router to accept dial-in connections from users or other remote devices (routers). The ISP's router requests authentication from the user. The user is expected to reply with the username and password supplied by the ISP when the user signed up for the service.

Each peer that wants to connect to the PPP interface on the authenticator must have a username and password configured that matches one of those stored in the User Authentication Database in the authenticator or in a defined TACACS or RADIUS server.

### To configure the router as an authenticator

1. Create the PPP interface and specify the required authentication type by using the command:

   ```
   create ppp=ppp-interface over=physical-interface
      authentication={chap|pap|either}
   ```

   To modify an existing PPP interface, use the command:

   ```
   set ppp=ppp-interface over=physical-interface
      authentication={chap|pap|either}
   ```

   The **either** option uses the PPP option negotiation process to request CHAP authentication. If the peer supports CHAP, **chap** is used. If the peer does not support CHAP but does support PAP, **pap** is used. If the peer supports neither authentication protocol, then the link is terminated.

2. If you are using the router's User Authentication Database, add a username and password to it for each user PC (or peer) that is allowed to dial in to the router by using the command:

   ```
   add user=username password=password login=no
   ```

   Alternatively, configure a TACACS (for PAP only) or RADIUS server and add the user to it.

**Caution** CHAP is not compatible with TACACS because the password is transmitted as plaintext between the TACACS server and the router.

## Configuring the Router as an Authenticator and a Peer

The router can be configured as both an authenticator and a peer in a two-way authentication scheme. When the router makes a call to a remote device (e.g. another router configured as an authenticator), the username and password that have been configured on the router are used by the remote device to authenticate the router. The remote device determines whether the username and password are valid, and accepts or rejects the connection. When the router receives a call from a remote device (e.g. a user dialling in via a modem or another device configured as a peer), it requests authentication from the remote device. The remote device supplies a username and password that the router validates before accepting or rejecting the connection. A typical example is two routers configured to communicate via ISDN, a synchronous dial-up connection or an asynchronous dial-up connection.

**To configure the router as both a peer and an authenticator**

1. Create the PPP interface by using the command:

   ```
   create ppp=ppp-interface over=physical-interface
      authentication={pap|chap|either} username=username
      password=password
   ```

   To modify an existing PPP interface, use the command:

   ```
   set ppp=ppp-interface over=physical-interface
      authentication={chap|pap|either} username=username
      password=password
   ```

   The **either** option of the **authentication** parameter uses the PPP option negotiation process to request CHAP authentication. If the peer supports CHAP, **chap** is used. If the peer does not support CHAP but does support PAP, **pap** is used. If the peer supports neither authentication protocol, then the link is terminated.

2. If you are using the router's User Authentication Database, add a username and password to it for each user PC (or peer) that is allowed to dial in to the router by using the command:

   ```
   add user=name password=password login=no
   ```

   Alternatively, configure a TACACS (for PAP only) or RADIUS server and add the user to it.

**Caution** CHAP is not compatible with TACACS because the password is transmitted as plaintext between the TACACS server and the router.

This configuration is a combination of the configurations described in "Configuring the Router as a Peer" on page 15-25 and "Configuring the Router as an Authenticator" on page 15-26.

The **authmode** parameter can be used in the **create ppp** or **set ppp** commands to control when authentication is to be requested. If **authmode** is set to **inout** authentication is requested for both incoming and outgoing calls. Some devices do not accept calls when the calling router also requests authentication from the called router. In this case **authmode** can be set to **in** so that only incoming calls result in authentication requests.

# Assigning IP Addresses

The router supports multiple methods for assigning IP addresses to dynamic dial-in calls. The following procedure is used to select the IP address assigned to a dial-in call:

1. If the PPP interface has been added to the IP module by using the **add ip interface** command on page 22-80 of Chapter 22, Internet Protocol (IP), then the IP address, network mask, and MTU are as defined for that IP interface.
2. If the user is authenticated via RADIUS, and the RADIUS response supplies an IP address, then that IP address is used.
3. If the user is authenticated using TACACS and an ACC domain name has been specified with the **add acc domainname** command on page 19-20 of Chapter 19, Asynchronous Call Control then the domain name is appended to the login name and a Domain Name Service (DNS) request is issued to resolve the name to an IP address.
4. If the user is authenticated using TACACS and an ISDN domain name has been specified with the **add isdn domainname** command on page 11-70 of

Chapter 11, Integrated Services Digital Network (ISDN), then the domain name is appended to the login name and a Domain Name Service (DNS) request is issued to resolve the name to an IP address.

5. If the user is authenticated by the User Authentication Database and an IP address and MTU are associated with the user's login name, then they are used for the interface.
6. If the PPP call has an IP pool set, and the request to the IP pool is successful, then that IP address is used. See "IP Address Pools" on page 22-52 of Chapter 22, Internet Protocol (IP) for more information about creating IP address pools.
7. If all of the above steps fail to provide the necessary information then a message is displayed and the call is dropped.

See "IP Address Pools" on page 22-52 of Chapter 22, Internet Protocol (IP) for more information about creating IP address pools.

To associate an IP address pool with a PPP interface so that connections using that interface use IP addresses from the IP address pool, use either of the commands:

```
create ppp=ppp-interface over=physical-interface
   ippool=pool-name [other-ppp-options...]

set ppp=ppp-interface ippool=pool-name [other-ppp-options...]
```

To disassociate an IP address pool from a PPP interface so that connections using that interface no longer use IP addresses from the IP address pool, use the command:

```
set ppp=ppp-interface ippool=none
```

To associate an IP address pool with a PPP interface so that dynamic PPP interfaces created using the PPP template use IP addresses from the IP address pool, use either of the commands:

```
create ppp template=template ippool=pool-name
   [other-template-options...]

set ppp template=template ippool=pool-name
   [other-template-options...]
```

To disassociate an IP address pool from a PPP template so that dynamic PPP interfaces created using the PPP template no longer use IP addresses from the IP address pool, use the command:

```
set ppp template=template ippool=none
```

Once the IP address has been assigned, the router communicates it to the peer using the IP NCP. The IP Address option in IP NCP is used to inform each end of the link what the IP address of the other end of the link is by passing the address to the peer inside the option.

If a PPP interface is created with an IP address of 0.0.0.0, and remote IP address assignment is enabled, during the IP control protocol (IPCP) negotiation process the router allows the remote PPP peer to set the IP address of the local PPP interface. When the peer has an IP address to allocate, it passes the address

to the requesting router in a IPCP Configure Nak packet. To configure the router to request an IP address using the IP address option, use the commands:

```
set ppp=ppp-interface iprequest=on

set ip int=ppp-interface ipaddress=0.0.0.0

enable ip remoteassign

reset ip
```

The IP NCP also provides a number of options for requesting name server addresses from the peer. These name server addresses consist of primary and secondary DNS and WINS (*Windows Internet Name Service*) server addresses. The router requests the primary DNS address from a peer, but supplies the peer with primary and secondary DNS and WINS server addresses when a request is made. To set values to be supplied to the peer, use the command:

```
set ppp [dnsprimary=ipadd] [dnssecondary=ipadd]
   [winsprimary=ipadd] [winssecondary=ipadd]
```

# PPP Link Management

Link management allows users to limit the connection time and data throughput on a PPP interface to thresholds they choose. For example, a user with an Internet connection via an Internet Service Provider (ISP) can limit the connection time or the amount of data transmitted over the PPP interface. By resetting the PPP link counters at the beginning of each billing period, they can keep their ISP bills within chosen limits.

Counters record cumulative up-time and input and output data throughput for each PPP link. The user can set thresholds for these parameters. If any of the thresholds are exceeded, the PPP link is closed. The link cannot be reopened until the counters are reset, or the threshold limits are increased or disabled.

The router writes the accumulated counters to flash memory every five minutes and every time the PPP link is closed. If the router is restarted, the counters are restored from flash memory to their previous values.

To create a new PPP interface with uptime and data throughput thresholds, use the command:

```
create ppp=ppp-interface [over=physical-interface]
   [onlinelimit={none|1..65535}]
   [indatalimit={none|1..65535}]
   [outdatalimit={none|1..65535}]
   [totaldatalimit={none|1..65535}] [other-options]...
```

To specify up-time and data throughput thresholds for an existing PPP interface, use the command:

```
set ppp=ppp-interface [onlinelimit={none|1..65535}]
   [indatalimit={none|1..65535}]
   [outdatalimit={none|1..65535}]
   [totaldatalimit={none|1..65535}]
```

Similarly, a PPP template with thresholds is created with the command:

```
create ppp template=template [onlinelimit={none|1..65535}]
   [indatalimit={none|1..65535}]
   [outdatalimit={none|1..65535}]
   [totaldatalimit={none|1..65535}] [other-options]...
```

To set thresholds on an existing PPP template, use the command:

```
set ppp template=template [onlinelimit={none|1..65535}]
   [indatalimit={none|1..65535}]
   [outdatalimit={none|1..65535}]
   [totaldatalimit={none|1..65535}]
```

To reset the counters that record cumulative uptime and data throughput for a PPP interface to zero (0), use the command:

```
reset ppp=ppp-interface [counter]
   [linkcounter={online|indata|outdata|totaldata|all}]
```

To display the cumulative counters, thresholds and remaining time or data throughput available on the interface, use the command:

```
show ppp[=ppp-interface] limits
```

# Configuring PPP Control Protocols

The router uses a number of counters and timers to control the LCP and NCPs. The timers control the retransmission of *Configure-Request* and *Terminate-Request* control protocol packets. If the correct acknowledgement is not seen in the timeout period, another packet is transmitted. Counters control the number of times the packets can be sent. The *Configure* counter records retransmissions of *Configure-Requests*. If this counter exceeds the value set for it, the LCP resets the interface and starts again. The *Terminate* counter records retransmissions of *Terminate-Requests*. If this counter exceeds the value set for it, the link is assumed to be down. The *Failure* counter controls the number of attempts to reach an agreeable set of values for options being negotiated by an NCP. This counter is not used in the router.

To set values for the counters, use the commands:

```
create ppp=ppp-interface over=physical-interface
   [configure={value|continuous}] [terminate={value|
   continuous}]

add ppp=ppp-interface over=physical-interface
   [configure={value|continuous}] [terminate={value|
   continuous}]

set ppp=ppp-interface over=physical-interface
   [configure={value|continuous}] [terminate={value|
   continuous}]
```

The **configure** parameter sets the number of configure requests sent before some action is taken. For the LCP the action is to reset the hardware and start again. For all other protocols the action is to give up. The default is **continuous**, which means that requests are sent continuously.

The **terminate** parameter sets the number of terminate requests sent when trying to close a link before it is assumed the link is down. The default is 2. The **continuous** option specifies that requests be sent continuously.

# Debugging PPP Links

If a PPP link fails to function correctly, software bugs could be the problem. PPP debugging commands help you establish the problem and locate the source. You can then fix or bypass the bug, allowing the PPP link to function correctly.

To enable debugging on a PPP interface, use the command:

```
enable ppp=ppp-interface debug={all|auth|bapstate|callback|
   ctrlpkt|datapkt|decode|demand|enco|lcp|lqr|ncp|packet|
   pkt|pppoe|utilisation} [,...] [asyn=port-number]
   [timeout={none|1..4000000000}] [numpkts={cont|
   1..4000000000}]
```

To enable the debugging option for dynamic PPP interfaces created using a specified PPP template, use the command:

```
enable ppp template=template debug={all|auth|bapstate|
   callback|ctrlpkt|datapkt|decode|demand|enco|lcp|lqr|ncp|
   packet|pkt|pppoe|utilisation}[,...] [asyn=port-number]
   [timeout={none|1..4000000000}] [numpkts={cont|
   1..4000000000}]
```

Enabling all debug options with **enable ppp debug=all** may generate enormous amounts of output, causing the router to lock up. Use the **timeout** or **numpkts** options to limit the amount of output generated.

To disable debugging on a PPP interface, use the command:

```
disable ppp=ppp-interface debug={all|auth||bapstate|callback|
   ctrlpkt|datapkt|decode|demand|enco|lcp|lqr|ncp|packet|
   pkt|pppoe|utilisation} [,...]
```

To disable the debugging option for dynamic PPP interfaces created using a PPP template, use the command:

```
disable ppp template=template debug={all|auth|bapstate|
   callback|ctrlpkt|datapkt|decode|demand|enco|lcp|lqr|ncp|
   packet|pkt|pppoe|utilisation}[,...]
```

PPP debugging is disabled by default.

The debugging options and their meanings are listed in Table 15-4. Output is sent to the specified asynchronous port or the terminal from which the command was entered.

Table 15-4: Point-to-Point Protocol (PPP) debugging options

| Option | Description |
|---|---|
| all | All debug options. |
| auth | PPP authentication. If LCP opens on a link but the network protocols remain in the closed state, the most likely cause is an authentication failure. |
| bapstate | BAP state machine transitions. |
| callback | Callback state machine transitions. |
| ctrlpkt | Hexadecimal dump of control packets received and transmitted on the PPP interface. |
| datapkt | Hexadecimal dump of data packets received and transmitted on the PPP interface. |
| decode | Decoded control packets. Disabling or enabling this option also automatically disable or enable AUTH debug. |

Table 15-4: Point-to-Point Protocol (PPP) debugging options (cont.)

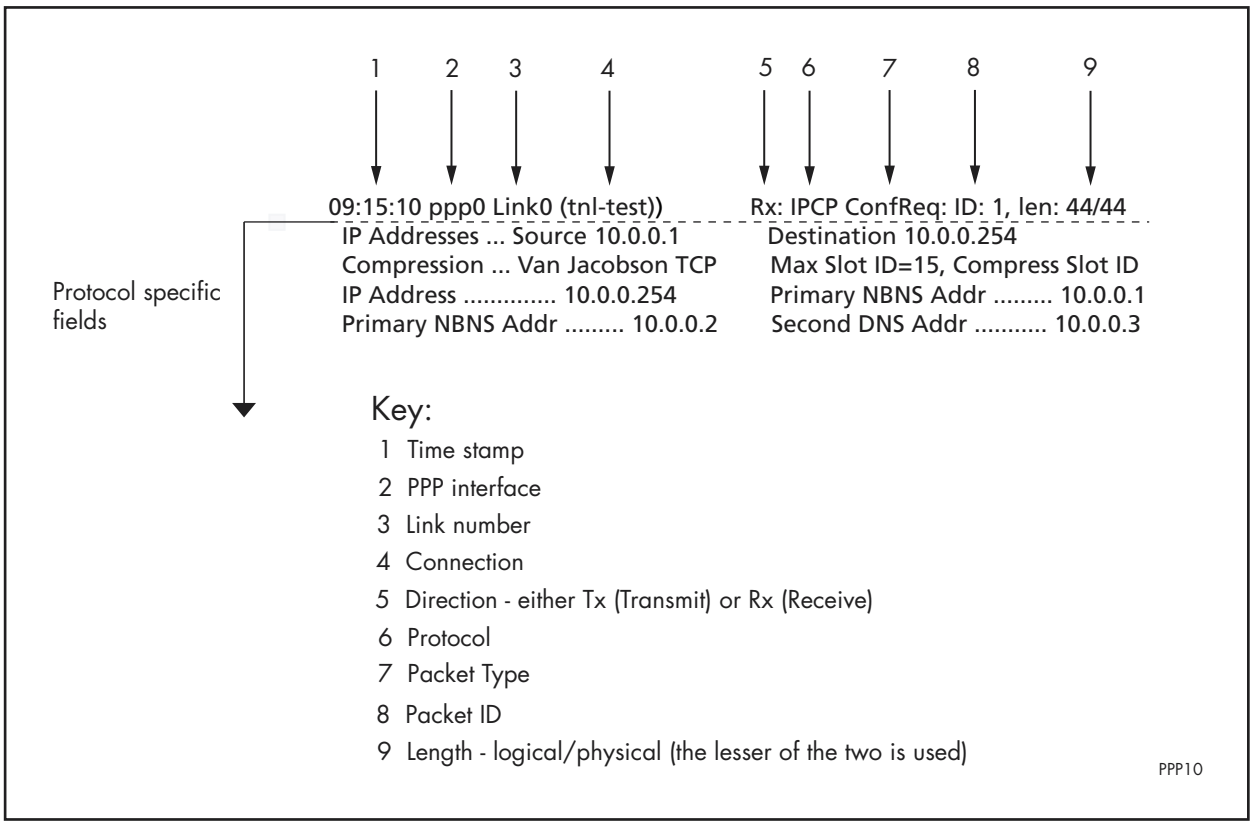
| Option | Description |
|---|---|
| demand | Packets that cause on-demand links to be activated. |
| enco | ENCO state machine used to control attachment to and detachment from the ENCO (encryption/compression) module. |
| lcp | LCP state machine transitions. |
| ncp | NCP state machine transitions. |
| lqr | Decoded LQR packets. |
| packet<br>pkt | Hexadecimal dump of all packets received and transmitted on the PPP interface. This option has the same effect as CTRLPKT and DATAPKT specified at the same time |
| pppoe | PPPoE discovery packets received and transmitted, and PPPoE state transitions. |
| utilisation | Utilisation measurements for each lower layer interface and the overall utilisation. |

The configuration information contained in Link Control Protocol (LCP) and Network Control Protocol (NCP) control packets is displayed in a readable format by using the **decode** debug option. For example, to enable PPP debug, use the command:

```
enable ppp=0 debug=decode
```

Then activate the PPP connection to this interface. Output is produced similar to that in Figure 15-6 on page 15-32.

Figure 15-6: Example PPP packet decode debug output



For comprehensive information about a specific protocol refer to the relevant RFC. See Table 15-5 on page 15-33 for a list of protocols and RFC number(s).

Table 15-5: Protocol and relevant RFC number(s)

| Protocol | RFC number(s) |
|---|---|
| AppleTalk Control Protocol (ATCP) | RFC 2023 |
| Bandwidth Allocation Control (BACP) | RFC 2125 |
| Bandwidth Allocation Protocol (BAP) | .RFC 1638 |
| Banyan VINES Control Protocol (BVCP) | RFC 1763 |
| Compression Control Protocol (CCP) | RFC 1962, 1978, 1974, 2118, 1993, 1977, 1967, 1975, 1979 |
| Encryption Control Protocol (ECP) | RFC 1968, 1969 |
| IP Control Protocol (IPCP) | RFC 1332 |
| IP Version 6 Control Protocol (IPv6CP) | RFC 2472 |
| Novel IPX Control Protocol (IPX) | RFC 1552 |
| Link Control Protocol (LCP) | RFC 1661, 1662, 1570, 1663, 1990, 1976, 1934, 2125 |
| NetBIOS Frames Control Protocol (NBFCP) | RFC 2097 |
| OSI Network Layer Control Protocol (OSINLCP) | RFC 1377 |
| Serial Data Control Protocol (SDCP) | RFC 1963 |

### If the link regularly disconnects

If the device at the other end of the PPP link is not an Allied Telesis router or switch but is supplied by another vendor turn LQR (Link Quality Reporting) off on PPP links (**lqr=off**) and instead use LCP Echo Request and Echo Reply messages to determine link quality (**echo=on**). Enter the command:

```
set ppp=ppp-interface echo=on lqr=off
```

# Configuration Examples

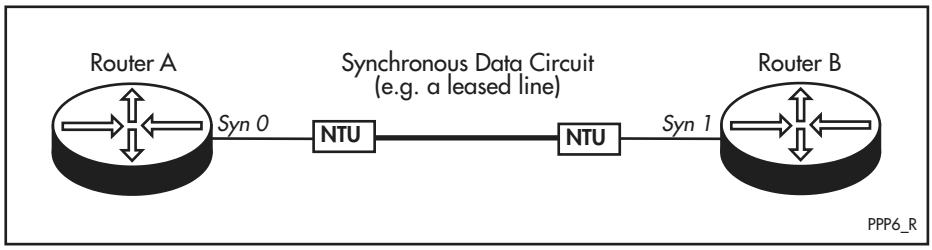
The examples in this section show how to configure PPP interfaces to provide a range of network services.

## Configuring a PPP link

In this example, a Point-to-Point Protocol (PPP) link is set up between two routers (Figure 15-7 on page 15-34). The function of PPP is to maintain a channel between the routers, over which data can be exchanged. To exchange data, the relevant routing module(s) must be assigned to use a PPP link.

Some interface and port types mentioned in this example may not be supported on your router. The interface and port types that are available vary depending on your product's model, and whether an expansion unit (PIC, NSM) is installed. For more information, see the Hardware Reference for the router.

Figure 15-7: Example network for configuring a PPP link




### To configure a PPP link

1. **Connect the routers to the Data Circuit**

   Ensure that the NTUs or modems are correctly installed on the data circuit—try a remote loop back.

   Using the correct cables, connect the synchronous interface on each router (synchronous interface 0 on Router A, synchronous interface 1 on Router B) to the local NTU or modem. Contact your authorised distributor or reseller if you are unsure which cables to use.

2. **Create the PPP interface.**

   On Router A, create a PPP interface numbered 0 over synchronous port 0:

   ```
   create ppp=0 over=syn0
   ```

   On Router B, create a PPP interface numbered 1 over synchronous port 1:

   ```
   create ppp=1 over=syn1
   ```

   The PPP interface is enabled by default when it is created.

   To configure additional PPP links, repeat the above commands for each additional PPP link. Each PPP interface on a router has a unique number and can run over different synchronous interfaces. For example, PPP7 could run over SYN3. However, keeping the interface numbers the same as the physical port numbers wherever possible makes management easier.

**3. Add an on-demand call.**

Additional physical interfaces can be added to the PPP interface to form a multilink bundle, using the **add ppp** command on page 15-50. For example, a PPP interface may also be created to use an ISDN call as a physical interface. The ISDN call must have been defined previously by using the command:

```
add isdn call=demand num=23432 prec=in
```

To add the ISDN call "demand" as a physical interface to the PPP interface created above, on Router A use the command:

```
add ppp=0 over=isdn-demand
```

On Router B, use the command:

```
add ppp=1 over=isdn-demand
```

The PPP interface may be configured for dial on demand operation by adding the **idle=on** option to the CREATE commands, or the option may be set at a later date with the SET command:

```
set ppp=0 idle=on
```

**4. Enable routing modules to use the interface.**

Once a PPP interface has been defined and configured, routing modules can be configured to use the interface. The procedures for achieving this are described in the chapter for the particular routing module.

In general, commands that contain the parameter **interface**= can refer to a PPP interface by name. The form of the name is "ppp*n*", where *n* is the interface number for the PPP module.

Examples of commands that can refer to a PPP interface include:

```
add ip interface=pppn...
add ipx circuit interface=pppn...
add decnet interface=pppn...
add apple port=pppn...
```

As an example, the IP routing module is to use the PPP interface just configured. The RIP routing protocol is to be used, so the PPP link has to be assigned its own an IP subnet. Use of OSPF as the routing protocol would mean that the PPP link could be set up as a unnumbered link. The subnet assigned to the PPP link is 172.16.254.0, with 255.255.255.0 as the subnet mask. The local (Router A) end of the link has address 172.16.254.1, and the remote (Router B) end has address 172.16.254.2. RIP is to be enabled to the remote end of the link. Router A already has an IP interface for the Ethernet interface, with an IP address of 172.16.9.59. The commands for Router A are:

```
enable ip
add ip int=ppp0 ip=172.16.254.1 mask=255.255.255.0
add ip rip int=ppp0
```

**5. Test that the link is active.**

The PPP interface can be checked with the command:

```
show ppp
```

which produces a display like Figure 15-8 on page 15-36. For each control protocol (listed in the *CP* field), the corresponding *State* field should be set to open.

Figure 15-8: Example output from the **show ppp** command for a PPP link

```
Name             Enabled  ifIndex  Over                  CP           State
-------------------------------------------------------------------------------
ppp0               YES    4                              IPCP         OPENED
                                   syn0                  LCP          OPENED
                                   isdn-demand           LCP          OPENED
-------------------------------------------------------------------------------
```

If the LCP has a state that is not open, check the configuration of the physical interfaces used by the PPP interface. Check that the cables connecting the synchronous ports to the local NTUs or modems are the correct type. Contact your authorised distributor or reseller for assistance.

Check the NTUs or modems are correctly installed. Perform a remote loop back from each end alternately. Contact the Telecom supplier or your authorised distributor or reseller if this fails.

Some modems or NTUs may require signals that are not provided by the router directly. These can usually be 'strapped' internal to the NTU or modem by the Telecom supplier or external wire jumpers can be added to the cable. Check with the Telecom supplier or your authorised distributor or reseller.

Check that the synchronous ports have been configured correctly for the network to which they are connected. Check the synchronous interface counters for high link error rates by using the command:

```
show syn=0 counter
```

The counters that usually increase with high link error are the *Aborts*, *CRCErrors* and *UnderlengthFrames* counters. See Chapter 9, Interfaces for a detailed description of these counters. The error counters should be low in relation to the good frame counters. The *seconds* counter gives the length of time that the counters have been active and can be used to assess the quality of the link.

It is normal for most serial wide area links to have a low error rate. Check with the Telecom supplier for an estimate of what they regard as acceptable.

To try to resolve this situation, consider the following possibilities:

1. This circuit may be faulty. Ask the Telecom supplier to test it.
2. This could be caused by poor quality or overlong cables, especially at higher link speeds. This is probably not a factor when below 1–2Mbps link speeds. Contact your authorised distributor or reseller for assistance.

For a PPP interface that is using an ISDN call as the physical interface, check that the calls have been properly defined and are active on the routers at each end of the link.

If the routing protocol is not in an open state, check the configuration of the routing module. As a first step, the IP configuration can be checked with the command:

```
show ip interface
```

which produces a display like Figure 15-9 on page 15-37.

Figure 15-9: Example output from the **show ip interface** command for a PPP link configured for use by the IP routing module

```
Interface     Type     IP Address       Bcast PArp  Filt RIP Met.   SAMode  IPSc
Pri. Filt     Pol.Filt Network Mask     MTU   VJC   GRE  OSPF Met.  DBcast  Mul.
-------------------------------------------------------------------------------
LOCAL         -        Not Set          -     -     ---  -          -       --
---           ---      -                -     -     ---  -          -       ---
eth0          Static   202.36.163.36    1     On    ---  01         Pass    --
---           ---      255.255.255.0    1500  -     ---  0000000001 None    ---
ppp0          Static   192.168.1.1      1     -     ---  01         Pass    --
---           ---      255.255.255.0    1500  Off   ---  0000000001 None    ---
ppp1          Static   192.168.2.1      1     -     ---  01         Pass    --
---           ---      255.255.255.0    1500  Off   ---  0000000001 None    ---
-------------------------------------------------------------------------------
```

# Multilink Aggregation

Traffic can be sent over multiple physical interfaces using PPP multilink. A PPP interface is created and configured to use more than one physical interface (e.g. ISDN B channels or synchronous ports). Any combination of physical interface types may be used (e.g. two ISDN B channels, or one ISDN B channel and a synchronous port, or three synchronous ports). This example expands on "A Basic ISDN Setup" on page 11-49 of Chapter 11, Integrated Services Digital Network (ISDN), by aggregating traffic on two ISDN B channels between router HO1 and RG1.

Some interface and port types mentioned in this example may not be supported on your router. The interface and port types that are available vary depending on your product's model, and whether an expansion unit (PIC, NSM) is installed. For more information, see the Hardware Reference for the router.

### To configure channel aggregation on a PPP interface

1. **Set up the ISDN call.**

   Create an ISDN call between routers HO1 and RG1 as in "A Basic ISDN Setup" on page 11-49 of Chapter 11, Integrated Services Digital Network (ISDN).

2. **Create a PPP interface to use the ISDN call.**

   Create a PPP interface to use the ISDN call Region1 twice (i.e. activate two calls using the same call definition). On the Head Office router, create PPP0 to use ISDN call Region1:

   ```
   create ppp=0 over=isdn-region1 num=2 idle=on
   ```

   On the Region 1 router, create PPP0 to use the ISDN call HeadOffice twice:

   ```
   create ppp=0 over=isdn-region1 num=2 idle=on
   ```

3. **Configure routing modules to use the PPP interface.**

   Configure one or more routing modules to use the PPP interface. See "A Basic ISDN Setup" on page 11-49 of Chapter 11, Integrated Services Digital Network (ISDN).

4. **Test the configuration.**

Check the PPP configuration by using the command:

```
show ppp
```

The expected output is shown in Figure 15-10 on page 15-38. All control protocols should have their State set to open. If either PPP LCP is not in an open state, check that the ISDN calls are active on both routers. If any routing control protocols (in this case IPCP) are not open, check the configuration of the routing module on both routers.

Figure 15-10: Example output from a **show ppp** command for a PPP interface aggregated over two ISDN B channels

```
Name             Enabled  ifIndex  Over                  CP           State
-------------------------------------------------------------------------------
ppp0               YES    4                              IPCP         OPENED
                                   isdn-Region1          LCP          OPENED
                                   isdn-Region1          LCP          OPENED
-------------------------------------------------------------------------------
```

Check the ISDN calls by using the command:

```
show isdn call
```

The expected output is shown in Figure 15-11 on page 15-38. There should be two active calls with the State field set to 'ON'. If not, the calls can be attempted again either by deactivating and then reactivating them, or by resetting the interface.

Figure 15-11: Example output from the **show isdn call** command for a PPP interface aggregated over two ISDN B channels

```
ISDN call details
Name             Number             Remote call      State    Precedence
-------------------------------------------------------------------------
Region1          043332345          -                IN & OUT    OUT
-------------------------------------------------------------------------

ISDN active calls
Index   Name              User     State  Prec
-------------------------------------------------------------
   0    Region1           03-00    ON     Yes
   1    Region1           03-36    ON     Yes
-------------------------------------------------------------
```

For the HO1 router the commands are:

```
deactivate isdn call=region1

activate isdn call=region1
```

or:

```
reset ppp=0
```

The **deactivate isdn call** command deactivates **all** calls with the specified name. In this example, PPP has been configured to make two Region1 calls. The **deactivate isdn call** command deactivates (hang up) both calls. The **activate isdn call** command makes a single call based on the specified call definition. To reactivate both calls for this example, the **activate isdn call** command must be used twice.

# Dial-on-Demand Links

A PPP interface can be configured so that it brings the link up only when there is traffic to send. This feature is useful on switched interfaces (e.g. ISDN) because the physical layer is available all the time for other types. This feature is sometimes called “dial on demand”. The link is disconnected when there has been no traffic for a specified period of time. This feature is disabled by default. The following examples assume PPP interface 0 has been configured as in “A Basic ISDN Setup” on page 11-49 of Chapter 11, Integrated Services Digital Network (ISDN).

To enable dial-on-demand and use the default disconnect timer (60 seconds), use the command:

```
set ppp=0 idle=on
```

To enable dial-on-demand with the disconnect timer set to 20 seconds, use the command:

```
set ppp=0 idle=20
```

To disable dial-on-demand, use the command:

```
set ppp=0 idle=off
```

To check the configuration, use the command:

```
show ppp=0 conf
```

# Link Quality Monitoring

Link quality monitoring is used to measure the quality of a link. The protocol used is an option negotiated when the link is brought up. There is one protocol for this, Link Quality Report (LQR). Packet and octets loss count, and link failure can be determined using LQR. The negotiation process determines how often a router should receive an LQR packet on a PPP interface. When a router does not receive two consecutive LQR packets within the specified time frame it resets the link. When using an ISDN call with the PPP interface, this disconnects the call when it is connected and tries to reconnect it.

The LQR counters can be displayed with the command:

```
show ppp=0 count
```

and the network manager can decide whether the level of packet and octet loss is good or bad. In a multilink configuration, LQR can be configured differently on each physical interface in the multilink bundle.

The following examples assume PPP interface 0 has been configured as in “A Basic ISDN Setup” on page 11-49 of Chapter 11, Integrated Services Digital Network (ISDN).

To enable LQR with the default timer (60 seconds), use the command:

```
set ppp=0 over=isdn-headoffice lqr=on
```

To enable LQR with the timer set to 20 seconds, use the command:

```
set ppp=0 over=isdn-headoffice lqr=20
```

To disable LQR, use the command:

```
set ppp=0 over=isdn-headoffice lqr=off
```

To check the configuration, use the command:

```
show ppp=0 conf
```

# Compression and Encryption

PPP interfaces can be configured to use hardware resources (e.g. a MAC card) to provide compression and/or encryption over wide area links. See Chapter 42, Compression and Encryption Services for more information. Compression must be configured on per-interface basis, on the routers at both ends of the PPP link.

For PPP multilink interfaces, the data may be compressed before the packets are forwarded to the multilinking process (**comp=on**). This means that all packets on all member links of the multilink carry compressed data, or the data may be compressed after the packets are forwarded to the multilinking process (**comp=link**), in which case only packets on the specified member link of the multilink carry compressed data. For multilink bundled interfaces with fragmentation enabled (**frag=on**), the data must not be compressed before the packets are forwarded to the multilinking process (**comp=on**). However, the data may be compressed after the packets are forwarded to the multilinking process (**comp=link**).

The following example commands illustrate some of the options for enabling compression.

Some interface and port types mentioned in this example may not be supported on your router. The interface and port types that are available vary depending on your product's model, and whether an expansion unit (PIC, NSM) is installed. For more information, see the *Hardware Reference for the* router.

To create a PPP interface aggregating synchronous port 0 and synchronous port 1 with compression enabled only on synchronous port 1, use the commands:

```
create ppp=0 over=syn0
add ppp=0 over=syn1 comp=link
```

To change the compression from synchronous port 1 to synchronous port 0, use the commands:

```
set ppp=0 over=syn1 comp=off
set ppp=0 over=syn0 comp=link
reset ppp=0
```

To enable compression before aggregation, use the commands:

```
set ppp=0 comp=on
reset ppp=0
```

To disable compression, use the commands:

```
set ppp=0 comp=off
reset ppp=0
```

To check any of these configurations, use the command:

```
show ppp config
```

Encryption must be configured on per-interface basis, on the routers at both ends of the PPP link. A star entity must be associated with the PPP interface, and specifies the encryption algorithm to use. See "Star Key Management" on page 50-5 of Chapter 50, Link Compression and Encryption for more information about creating star entities. Enable encryption when a PPP

interface is created or by modifying the configuration of an existing PPP interface by using the commands:

```
create ppp=ppp-interface over=physical-interface
   encryption=on starentity=1..255

set ppp=ppp-interface over=physical-interface encryption=on
   starentity=1..255
```

For PPP interfaces created over L2TP calls, data cannot be compressed before the packets are forwarded to the multilinking process (**comp=on**). The Encryption Control Protocol (ECP) defined in RFC 1968 is used to negotiate encryption options with the remote peer. During ECP negotiation, ECP option 0 is used to offer the peer the encryption algorithm configured in the associated star entity. If the star entity associated with the peer is not configured with the same encryption algorithm, the option is rejected, the negotiation fails, and the link is closed.

## Leased Line Backup

A PPP interface can be configured to use an ISDN call to back up a synchronous (leased) line. When a link is added to a PPP interface it can be assigned a channel type of **primary** or **secondary**. The default channel type is **primary**. To perform leased line backup the PPP link using the synchronous line must be assigned a channel type of **primary** and the PPP link using the ISDN must be assigned a channel type of **secondary**. When a primary link failure is detected (by LQR), it is reset and an attempt is made to reopen the link. If the LCP of the primary link fails to reach an open state after sending a number of configure requests, the secondary link is activated. The primary link continually tries to reopen and closes the secondary link when it succeeds. The **configure** parameter specifies the number of configure requests required before the secondary link is activated, and defaults to **continuous**. The **restart** parameter controls how often configure requests are transmitted and defaults to 3 seconds. To enable lease line backup, the **configure** parameter needs to be set to a value other than **continuous**.

This example illustrates how to configure leased line backup between two routers (Figure 15-12 on page 15-41, Table 15-6 on page 15-42).

Some interface and port types mentioned in this example may not be supported on your router. The interface and port types that are available vary depending on your product's model, and whether an expansion unit (PIC, NSM) is installed. For more information, see the Hardware Reference for the router.

Figure 15-12: Example configuration for leased line backup

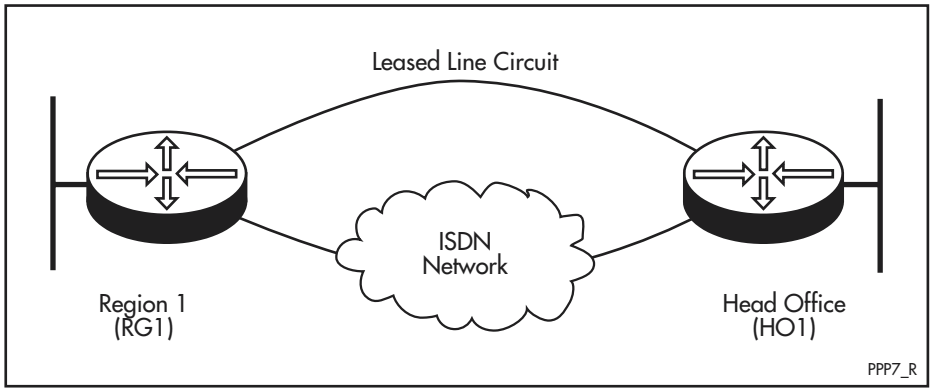


Table 15-6: Example configuration parameters for leased line backup

| Site | Region 1 | Head Office |
|---|---|---|
| Router Name | RG1 | HO1 |
| ISDN Number | 1234567 | 9876543 |
| IP Address for PPP0 | 192.168.35.114 | 192.168.35.113 |
| IP Address for Eth0 | 192.168.35.110 | 192.168.35.45 |
| Subnet Mask | 255.255.255.240 | 255.255.255.240 |

### To configure leased line backup

1. **Create the ISDN calls.**

   An ISDN call must be defined on each router so that either router may initiate a call to transfer data. For a more detailed example of creating ISDN calls see "A Basic ISDN Setup" on page 11-49 of Chapter 11, Integrated Services Digital Network (ISDN).

   Set the ISDN call profile appropriate for the ISDN service provider. The default profile is the ETSI specification for European Union (EU) countries (ETB for Basic Rate interfaces or ETP for Primary Rate interfaces). To use the Australian Telecom profile, for example, on Basic Rate interface BRI 0 for router HO1 and RG1, use the following command on each router:

   ```
   set q931=bri0 profile=aus
   ```

   On the Head Office router, create a call to the Region 1 router:

   ```
   add isdn call=region1 prec=in outsub=local searchsub=local
      number=1234567
   ```

   On the Region 1 router create a call to the Head Office router:

   ```
   add isdn call=region1 prec=out outsub=local
      searchsub=local number=9876543
   ```

2. **Create a PPP interface over the synchronous interface.**

   Create the PPP interface, setting the **configure** parameter to 5 and the LQR timer to 10 seconds. This speeds up link failure detection as the default is 60 seconds. The default link type is **primary**. On the Head Office router create a PPP interface:

   ```
   create ppp=0 over=syn0 conf=5 lqr=10
   ```

   On the Region 1 router create a PPP interface:

   ```
   create ppp=0 over=syn0 conf=5 lqr=10
   ```

3. **Add the ISDN calls to the PPP interface.**

   Add the ISDN calls, specifying a link type of **secondary**. On the Head Office router add an ISDN call:

   ```
   add ppp=0 over=isdn-region1 type=secondary
   ```

   On the Region 1 router add an ISDN call:

   ```
   add ppp=0 over=isdn-region1 type=secondary
   ```

4. **Configure IP.**

   Configure a routing module to use the PPP interfaces. It could be IPX, DECnet, AppleTalk or bridging but for this example IP is used. Configure IP at the Head Office router:

   ```
   enable ip
   add ip int=ppp0 ip=192.168.35.113 mask=255.255.255.240
   ```

   Configure IP at the Region 1 router:

   ```
   enable ip
   add ip int=ppp0 ip=192.168.35.114 mask=255.255.255.240
   ```

   For a more detailed example of configuring IP see "Configuration Examples" on page 22-54 of Chapter 22, Internet Protocol (IP).

# Bandwidth on Demand

A PPP interface can be configured to use up to two B channels on a Basic Rate ISDN interface, or up to 30 B channels on a Primary Rate ISDN interface, to provide bandwidth on demand. PPP activates channels when the used bandwidth exceeds an upper threshold and deactivates channels when the bandwidth drops below a lower threshold. To configure bandwidth on demand, the ISDN channels are assigned a **type** of **demand** when added to the PPP interface. Assigning one channel a type of **primary** and other channels a **type** of **demand** ensures that there is always one channel available. If all channels are assigned a **type** of **demand**, then there are no channels active when there is no traffic; some traffic causes one channel to be activated and continuous traffic causes other channels to be activated. If there is one channel remaining opened, then the IDLE timer determines when this should be closed. In this case, the IDLE timer should not be set to **off**.

This example illustrates how to configure bandwidth on demand between two routers (Figure 15-13 on page 15-43, Table 15-7 on page 15-44).

Some interface and port types mentioned in this example may not be supported on your router. The interface and port types that are available vary depending on your product's model, and whether an expansion unit (PIC, NSM) is installed. For more information, see the Hardware Reference for the router.

Figure 15-13: Example configuration for bandwidth on demand

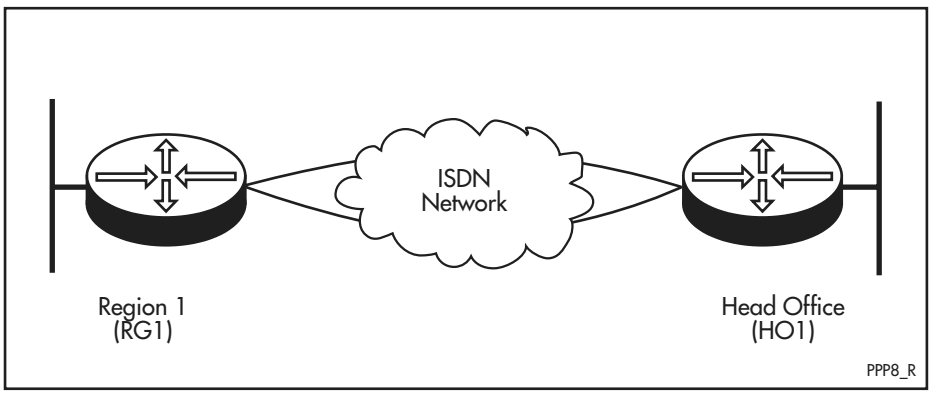


Table 15-7: Example configuration parameters for bandwidth on demand

| Site | Region 1 | Head Office |
|---|---|---|
| Router Name | RG1 | HO1 |
| ISDN Number | 1234567 | 9876543 |
| IP Address for PPP0 | 192.168.35.114 | 192.168.35.113 |
| IP Address for Eth0 | 192.168.35.110 | 192.168.35.45 |
| Subnet Mask | 255.255.255.240 | 255.255.255.240 |

### To configure PPP for bandwidth on demand

1. **Create the ISDN calls.**

   An ISDN call must be defined on each router so that either router may initiate a call to transfer data. For a more detailed example of creating ISDN calls see "A Basic ISDN Setup" on page 11-49 of Chapter 11, Integrated Services Digital Network (ISDN).

   Set the ISDN call profile appropriate for the ISDN service provider. The default profile is the ETSI specification for European Union (EU) countries (ETB for Basic Rate interfaces or ETP for Primary Rate interfaces). To use the Australian Telecom profile, for example, on Basic Rate interface BRI 0 for router HO1 and RG1, use the following command on each router:

   ```
   set q931=bri0 profile=aus
   ```

   On the Head Office router, create calls to the Region 1 router:

   ```
   add isdn call=region1 prec=in outsub=local searchsub=local
      number=1234567
   add isdn call=demand prec=in outsub=local searchsub=local
      number=1234567
   ```

   On the Region 1 router create calls to the Head Office router:

   ```
   add isdn call=region1 prec=out outsub=local
      searchsub=local number=9876543
   add isdn call=demand prec=out outsub=local searchsub=local
      number=9876543
   ```

2. **Create a PPP interface to use the ISDN calls.**

   Create the PPP interface with one primary channel and one demand channel. The primary channel is created with the **idle** parameter **on** (defaults to 60 seconds). The demand channel is added with the **type** parameter set to **demand**. On the Head Office router create a PPP interface:

   ```
   create ppp=0 over=isdn-region1 idle=on
   add ppp=0 over=isdn-demand type=demand
   ```

   On the Region 1 router create a PPP interface:

   ```
   create ppp=0 over=isdn-region1 idle=on
   add ppp=0 over=isdn-demand type=demand
   ```

3. **Configure IP.**

   Configure a routing module to use the PPP interfaces. It could be IPX, DECnet or bridging but for this example IP is used. Static routes must be defined with on-demand links because a routing protocol would keep a link up continuously. Configure IP at the Head Office router:

```
enable ip

add ip int=ppp0 ip=192.168.35.113 mask=255.255.255.240

add ip route=192.168.35.96 int=ppp0 next=192.168.35.114
   met=2
```

   Configure IP at the Region 1 router:

```
enable ip

add ip int=ppp0 ip=192.168.35.114 mask=255.255.255.240

add ip route=192.168.35.0 int=ppp0 next=192.168.35.113
   met=2 mask=255.255.255.0

add ip route=0.0.0.0 int=ppp0 next=192.168.35.113 met=3
```

   For a more detailed example of configuring IP see "Configuration Examples" on page 22-54 of Chapter 22, Internet Protocol (IP).

# Bandwidth on Demand with Leased Line Circuits and ISDN

A PPP interface can be configured to use a number of ISDN channels to provide bandwidth on demand. PPP activates channels when the used bandwidth exceeds an upper threshold, and deactivates channels when the bandwidth drops below a lower threshold. To configure bandwidth on demand, the ISDN channels are assigned a **type** of **demand** when added to the PPP interface. Assigning one channel a type of **primary** and other channels a **type** of **demand** ensures that there is always one channel available. If all channels are assigned a **type** of **demand**, then there are no channels active when there is no traffic; some traffic causes one channel to be activated and continuous traffic causes other channels to be activated. If there is one channel remaining open, then the IDLE timer determines when this should be closed. In this case, the IDLE timer should not be set to **off**.

This example illustrates how to configure bandwidth on demand between two routers (Figure 15-14 on page 15-46, Table 15-8 on page 15-46).

Some interface and port types mentioned in this example may not be supported on your router. The interface and port types that are available vary depending on your product's model, and whether an expansion unit (PIC, NSM) is installed. For more information, see the *AR400 Series Router Hardware Reference*.

Figure 15-14: Example configuration for bandwidth on demand with leased line circuits and ISDN

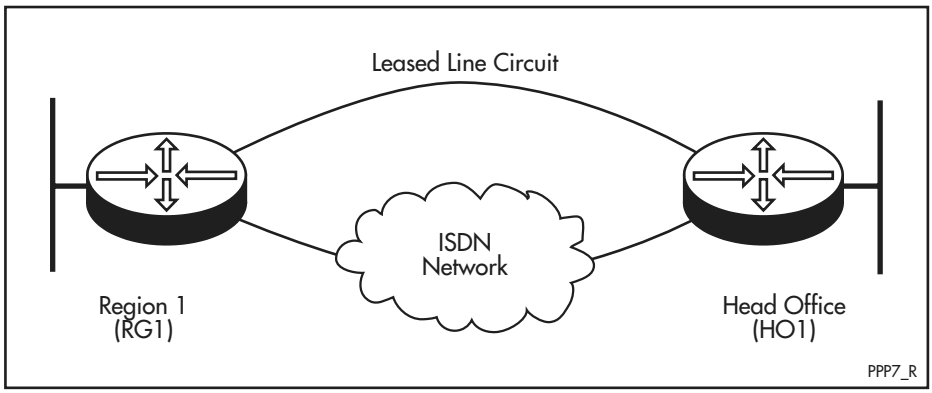



Table 15-8: Example configuration parameters for bandwidth on demand with leased line circuits and ISDN

| Site | Region 1 | Head Office |
|---|---|---|
| Router Name | RG1 | HO1 |
| ISDN Number | 1234567 | 9876543 |
| IP Address for PPP0 | 192.168.35.114 | 192.168.35.113 |
| IP Address for Eth0 | 192.168.35.110 | 192.168.35.45 |
| Subnet Mask | 255.255.255.240 | 255.255.255.240 |

### To configure bandwidth on demand with leased line circuits and ISDN:

**1. Create the ISDN calls.**

An ISDN call must be defined on each router so that either router may initiate a call to transfer data. For a more detailed example of creating ISDN calls see "A Basic ISDN Setup" on page 11-49 of Chapter 11, Integrated Services Digital Network (ISDN).

Set the ISDN call profile appropriate for the ISDN service provider. The default profile is the ETSI specification for European Union (EU) countries (ETB for Basic Rate interfaces or ETP for Primary Rate interfaces). To use the Australian Telecom profile, for example, on Basic Rate interface BRI 0 for router HO1 and RG1, use the following command on each router:

```
set q931=bri0 profile=aus
```

On the Head Office router, create a call to the Region 1 router:

```
add isdn call=region1 outsub=local searchsub=local prec=in
   number=1234567
```

On the Region 1 router create a call to the Head Office router:

```
add isdn call=region1 outsub=local searchsub=local
   prec=out number=9876543
```

**2. Set up the synchronous interfaces.**

The synchronous interfaces on both routers must be set to the same speed as the leased line. For example, for a 9600 baud leased line, the synchronous interfaces must be set to 9600 baud by using the following command on each router:

```
set syn=0 speed=9600
```

The speed of the synchronous interface must be set to correctly match the actual speed of the WAN connection, otherwise the utilisation calculations produce erroneous results.

**3. Create a PPP interface to use the ISDN calls.**

Create the PPP interfaces to use a leased line and one ISDN channel, and set the thresholds for bandwidth on demand. On the Head Office router create a PPP interface:

```
create ppp=0 over=syn0
add ppp=0 over=isdn-region1 type=demand
set ppp=0 uprate=80 downrate=20
```

On the Region 1 router create a PPP interface:

```
create ppp=0 over=syn0
add ppp=0 over=isdn-region1 type=demand
set ppp=0 uprate=80 downrate=20
```

**4. Configure IP.**

Configure a routing module to use the PPP interfaces. It could be IPX, DECnet or bridging but for this example IP is used. Static routes must be defined with on-demand links because a routing protocol would keep a link up continuously. Configure IP at the Head Office router:

```
add ip int=ppp0 ip=192.168.35.113 mask=255.255.255.240
add ip route=192.168.35.96 int=ppp0 next=192.168.35.114
   met=2
```

Configure IP at the Region 1 router:

```
add ip int=ppp0 ip=192.168.35.114 mask=255.255.255.240
add ip route=192.168.35.0 int=ppp0 next=192.168.35.113
   met=2
add ip route=0.0.0.0 int=ppp0 next=192.168.35.113 met=3
```

For a more detailed example of configuring IP see "Configuration Examples" on page 22-54 of Chapter 22, Internet Protocol (IP). The command:

```
enable ppp=0 debug=util
```

can be used to see what bandwidth utilisation is being reported and the average utilisation. All values are expressed in hexadecimal. The utilisation is measured every second.

# Command Reference

This section describes the commands available on the router to configure and manage the Point-to-Point Protocol on the router. The Point-to-Point Protocol (PPP) can be used on:

- Ethernet (ETH ports)
- Ethernet (VLANs)
- Synchronous links (routers with PIC bay only)
- ISDN calls (see Chapter 11, Integrated Services Digital Network (ISDN)) (routers with PIC bay only)
- ACC calls (see Chapter 19, Asynchronous Call Control)
- MIOX circuits (see Chapter 13, X.25)
- L2TP calls (see Chapter 20, Layer Two Tunnelling Protocol (L2TP))
- TDM groups (see Chapter 12, Time Division Multiplexing (TDM)) (routers with PIC bay only)

PPP can be used to carry:

- IP
- compressed data
- encrypted data
- DECnet
- IPX
- AppleTalk routing protocols
- bridged protocols

Some interface and port types mentioned in this chapter may not be supported on your router. The interface and port types that are available vary depending on your product's model, and whether an expansion unit (PIC, NSM) is installed. For more information, see the *AR400 Series Router Hardware Reference*.

See "Conventions" on page lxv of About this Software Reference in the front of this manual for details of the conventions used to describe command syntax. See Appendix A, Messages for a complete list of messages and their meanings.

# activate ppp

**Syntax** `ACTIVATE PPP=`*`ppp-interface`* `RXPKT=`*`hexstring`*

where:

- *ppp-interface* is the PPP interface number, from 0 to 1023.
- *hexstring* is a string of hexadecimal characters.

**Description** This command creates and sends a PPP packet to the specified PPP interface as if the packet had been received from the lower layer interface. This command is intended for debugging purposes only, and should not be used during normal operation.

The **rxpkt** parameter specifies the PPP packet to create and send, as a string of hexadecimal characters. For detailed information about PPP packet formats, see RFC 1661, *The Point-to-Point Protocol (PPP)*.

**Examples** To create and send an LCP packet requesting CHAP authentication to PPP interface 1, use the command:

```
activate ppp=1 rxpkt=ff03c023012100090305c22305
```

**Related Commands**

**disable ppp debug**
**disable ppp template debug**
**enable ppp debug**
**enable ppp template debug**
**show ppp debug**
**show ppp template**

# add ppp

**Syntax**

```
ADD PPP=ppp-interface OVER=physical-interface
    [AUTHENTICATION={CHAP|EITHER|PAP|NONE}] [AUTHMODE={IN|
    OUT|INOUT}] [CBDELAY=1..100] [CBMODE={ACCEPT|OFF|
    REQUEST}] [CBNUMBER=e164number]
    [CBOPERATION={E164NUMBER|USERAUTH}]
    [COMPALGORITHM={PREDICTOR|STACLZS}] [COMPRESSION={LINK|
    OFF}] [CONFIGURE={value|CONTINUOUS}] [ECHO={OFF|ON|
    period}] [LQR={ON|OFF|period}] [MAGIC={ON|OFF}]
    [MODEM={ON|OFF}] [NUMBER=number] [PREDCHECK={CRC16|
    CRCCCITT}] [RECHALLENGE={ON|OFF|360..3600}]
    [RESTART=time] [STACCHECK={LCB|SEQUENCE}]
    [TERMINATE={value|CONTINUOUS}] [TYPE={DEMAND|PRIMARY|
    SECONDARY}]
```

where:

- *ppp-interface* is the PPP interface number, from 0 to 1023.
- *physical-interface* is:
  - ATM (e.g. atm0.1)
  - SYN*n*
  - ISDN-*callname*
  - ACC-*callname*
  - MIOX*n-circuitname*
  - TDM-*groupname*
  - TNL-*callname*
  - ETH*n-servicename*
  - VLAN*n-servicename*
- *servicename* is 1 to 18 characters long and for a PPPoE client, is usually supplied by the ISP providing the service.
- *e164number* is the phone number to dial when performing callback. It may contain digits (0–9) and should be a valid phone number as described in CCITT standard E.164.
- *period* is an integer from 1 to 4294967295.
- *number* is the number of ISDN B channels.

**Description**

This command adds a lower layer interface or link to an existing PPP interface. This configures PPP multilink, which groups links together for increased bandwidth. The following may be added:

- an ATM channel
- a synchronous port
- an ISDN call
- an ACC call
- a MIOX circuit
- TDM group
- an L2TP call
- a PPP over Ethernet service over an ETH interface
- a PPP over Ethernet service over a VLAN interface

The **over** parameter specifies the physical interface over which the PPP interface runs. For PPP over Ethernet and PPP over VLAN links, use the service name provided by your ISP, or the special service name ANY to specify that any service is acceptable.

The **authentication** parameter specifies the authentication protocol to be used on the physical interface or channel. If **chap** is specified, the Challenge-Handshake Authentication Protocol (CHAP) is used. If **pap** is specified, the Password Authentication Protocol (PAP) is used. If **either** is specified, the router uses the option negotiation process to negotiate the authentication protocol to be used with the device at the remote end of the link, specifying **chap** as the first choice. If **none** is specified, no authentication protocol is used. The default is **none**.

The **authmode** parameter specifies how authentication requests to peers are affected by the direction of the call. The **authmode** parameter is valid when the **authentication** parameter is a value other than **none** and the physical interface is a call. When **in** is specified, authentication is requested for incoming calls from peers. When **out** is specified, authentication is requested for outgoing calls to peers. When **inout** is specified, authentication is always requested regardless of the direction of the call. The default is **inout**. This parameter is valid for:

- ISDN calls
- ACC calls

The **cbdelay** parameter specifies the delay in tenths of a second between bringing down a call for callback and actually making the call back to the peer. This parameter handles different timing requirements of various ISDN switches and is valid for PPP links over ISDN calls and when the callback mode is **request**. The default is 1.

The **cbmode** parameter specifies whether a callback request is to be made or accepted during the LCP negotiation. If **request** is specified, a request is made for callback. If **accept** is specified, requests for callback received from the peer are accepted and processed; however, **authentication** must be set to PAP or **chap**. If **off** is specified, no callback requests are made and callback requests are not accepted. The default is **off**.

The **cbnumber** parameter specifies the number to include when requesting a callback with the **cboperation** parameter set to **e164number**. The number specified should be a phone number as specified in the E.164 standard.

The **cboperation** parameter specifies the callback operation to be included in the callback request to specify to the peer how to determine the callback number. If **userauth** is specified, the peer uses the username and password supplied during authentication to look up the callback number. If **e164number** is specified, the callback number specified by the **cbnumber** parameter is included in the callback request. The default is **userauth**.

The **compalgorithm** parameter specifies the compression algorithm to use when compressing and decompressing PPP packets. If **predictor** is specified, the Predictor algorithm is used with type 1 encapsulation as specified in RFC 1978. If **staclzs** is specified, the Stac LZS algorithm is used as specified in RFC 1974. The default is **staclzs**. This parameter does not apply to L2TP calls.

The **compression** parameter enables compression for the physical interface being added. The default is **off**. The **link** option should only be used when compression is required on the interface being added and not on others. For example, if a PPP multilink uses a compressing modem link and a normal dedicated leased line, **compression** should be set to **off** on the physical interface to which the modem is connected, and **link** for the physical interface to which the dedicated leased line is connected. If compression is required on all physical interfaces of a PPP interface, compression should be enabled by setting the **compression** parameter to **on** in the **create ppp** command on page 15-56, the **create ppp template** command on page 15-63, **set ppp** command on page 15-83 or the **set ppp template** command on page 15-92. This parameter does not apply to L2TP calls.

The **configure** parameter sets the number of configure requests sent before some action is taken. For the LCP the action is to reset the hardware and start again. For all other protocols the action is to give up. The default is **continuous**, which means that requests are sent continuously.

The **echo** parameter specifies whether LCP Echo Request and Echo Reply messages determine link quality. When three consecutive Echo Request messages are transmitted without receiving an Echo Reply response, the link is deemed to be down. If **off** is specified, Echo Request messages are not transmitted. If **on** is specified, Echo Request messages are transmitted every 10 seconds. If a period in seconds is specified, Echo Request messages are transmitted at the specified interval. The **lqr** function has precedence over the **echo** function. They cannot both be enabled at the same time. However, both **lqr** and **echo** can be **on** at the same time if **echo** is turned on first, then **lqr**.

The **lqr** parameter sets the LQR timer for link quality management. If **on** is specified, **lqr** is enabled with a default timer of 60 seconds. If a time is specified, **lqr** is enabled with the timer set to the specified time. If **off** is specified, **lqr** is disabled. The LQR function has precedence over the ECHO function. They cannot both be enabled at the same time. However, both **lqr** and **echo** can be **on** at the same time if **echo** is turned on first. The default is **on**.

The **magic** parameter enables or disables negotiation of the magic number option. The default is **on**. The magic number is used to determine whether an interface is looped back. The interface does not reach an open state when there is a loopback.

The **modem** parameter specifies the state of synchronous modem control. The default is **off**. If **on** is specified, the router manipulates the DTR signal to trigger the modem to make a call whenever there is traffic (**idle=on**) or when a backup link is required (**type=secondary**). Raising DTR to the modem triggers the modem to initiate a call to the modem at the other end of the link and enter data transfer mode. The router keeps the DTR signal to the modem raised, and responds to the modem raising the DSR signal by activating the PPP interface. This parameter is valid when the **over** parameter specifies a synchronous interface.

The **number** parameter specifies the number of physical interfaces to be added. This parameter is valid when the **over** parameter specifies an ISDN call as the physical interface. The default is 1.

The **predcheck** parameter specifies the type of CRC to be used for Predictor compression. The Predictor RFC specifies using CRC-16, however some router manufacturers have implemented Predictor with CRC-CCITT that is the CRC specified in RFC 1662, “PPP in HDLC-link Framing”. This value is not negotiated so the same value needs to be configured at both ends of the link for Predictor compression to work correctly. This parameter does not apply to L2TP calls.

The **rechallenge** parameter specifies if the CHAP rechallenge function is **on**, **off**, or the maximum rechallenge period. PPP calculates a random period for the CHAP rechallenge between the maximum rechallenge period and a minimum of 5 minutes. If **on** is specified, then CHAP rechallenges take place with a maximum rechallenge period of 15 minutes. If **off** is specified, then CHAP rechallenges do not take place. If a time is specified, then the time, which is between 360 and 3600 seconds, is used as the upper limit on the rechallenge time calculation. The default is **on**.

The **restart** parameter specifies the time between successive retransmissions of unacknowledged configure requests or terminate requests. The default is 3 seconds.

The **staccheck** parameter specifies the check mode to used for the Stac LZS compression algorithm. If **sequence** is specified an incrementing sequence number is used to determine whether a packet has been lost and therefore whether the compression history needs to be reset. If LCB is specified an LCB value is used to determine if an error has occurred in a packet. The default is **sequence**. This parameter does not apply to L2TP calls.

The **terminate** parameter sets the number of terminate requests sent when trying to close a link before it is assumed the link is down. The default is 2. The **continuous** option specifies that requests be sent continuously.

The **type** parameter specifies the role of the physical interface for bandwidth on demand and link backup. The default is **primary**. If **primary** is specified, the link is open all the time (**idle=off**) or open whenever there is traffic (**idle=on**). If **secondary** is specified, the link is open when the associated primary link fails. If **demand** is specified, the link is open when additional bandwidth is required.

## Examples

To add Ethernet interface 0 as an additional physical interface to PPP interface 1, and enable STAC LZS compression on the synchronous link with a check mode of LCB, use the command:

```
add ppp=1 over=eth0-any comp=link staccheck=lcb
```

To add ISDN call ‘demand” as an additional physical interface to PPP interface 1, and enable STAC LZS compression on the link with a check mode of LCB, use the command:

```
add ppp=1 over=isdn-demand compression=link staccheck=lcb
```

To add a PPPoE interface on VLAN2, using the service name ANY, as an additional physical interface to PPP interface 1, and enable STAC LZS compression of the synchronous link with a check mode of LCB, use the command:

```
add ppp=1 over=vlan2-any compression=link staccheck=lcb
```

## Related Commands

**add ppp acservice**
**create ppp**
**delete ppp**
**delete ppp acservice**
**set ppp**
**show ppp**

# add ppp acservice

**Syntax**

```
ADD PPP ACSERVICE=service-name TEMPLATE=ppp-template
   [ACRADIUS={OFF|ON}] [MAXSESSIONS=1..512]
   [ACINTerface={NONE|interface}]
```

where:

- *service-name* is a character string of up to 18 characters. *service-name* should not be “any”.
- *template* is a number from 0 to 31.
- *interface* is an interface name formed by concatenating an interface type and an interface instance (e.g. eth0). Valid interface types are ETH and VLAN.

**Description**

This command adds a new PPP over Ethernet Access Concentrator service to the router. PPPoE hosts are able to connect to the router using this service.

To allow a PPPoE host to be defined on the router as well as on an Access Concentrator service, the **acinterface** parameter must be used.

The **acservice** parameter specifies the name of the new PPPoE service that is to be added.

The **template** parameter specifies the PPP template to use when creating a dynamic PPP over Ethernet interface. The specified template must exist. See “Templates” on page 15-18 for more information about creating PPP templates.

The **maxsessions** parameter specifies the maximum number of PPPoE sessions that are allowed to simultaneously provide the PPPoE service. The default is 1.

The **acradius** parameter specifies whether PPPoE hosts are authenticated by a RADIUS server using the host’s MAC address. If **off** is specified, the RADIUS authentication is not performed. If on is specified, the RADIUS authentication is performed. The default is **off**.

**Important** RADIUS authentication is performed only if it is specified by the template associated with this PPPoE service.

The **acinterface** parameter specifies the interface to be used by the Access Concentrator service. If **none** is specified, the Access Concentrator service uses all Ethernet interfaces. A service can be offered on several interfaces, but it is necessary to issue one add ppp acservice command for each interface. For example:

```
add ppp acservice=bob template=1 acint=eth0
add ppp acservice=bob template=1 acint=vlan5
```

To offer the service on all the Ethernet interfaces only, there is no need to use the **acinterface** parameter, as it defaults to **none**.

**Examples**

To add a new PPPoE Access Concentrator service named armadillo to be offered on Ethernet interface 0 using template 5 with a maximum of 200 simultaneous sessions and no RADIUS authentication, use the command:

```
add ppp acservice=armadillo template=5 maxsessions=200
   acint=eth0
```

**Related Commands**

create ppp
create ppp template
delete ppp acservice
destroy ppp template
disable ppp accessconcentrator
enable ppp accessconcentrator
set ppp acservice

# create ppp

**Syntax**

```
CREATE PPP=ppp-interface OVER=physical-interface
   [AUTHENTICATION={CHAP|EITHER|PAP|NONE}] [AUTHMODE={IN|
   OUT|INOUT}] [BAP={ON|OFF}] [BAPMODE={CALL|CALLBACK}]
   [CBDELAY=1..100] [CBMODE={ACCEPT|OFF|REQUEST}]
   [CBNUMBER=e164number] [CBOPERATION={E164NUMBER|
   USERAUTH}] [COMPALGORITHM={PREDICTOR|STACLZS}]
   [COMPRESSION={ON|OFF|LINK}] [CONFIGURE={value|
   CONTINUOUS}] [DEBUGMAXBYTES=16..256]
   [DESCRIPTION=description] [DOWNRATE=0..100]
   [DOWNTIME=time] [ECHO={ON|OFF|period}] [ENCRYPTION={ON|
   OFF}] [FRAGMENT={ON|OFF}] [FRAGOVERHEAD=0..100]
   [IDLE={ON|OFF|time}] [INDATALIMIT={NONE|1..65535}]
   [IPPOOL={pool-name|NONE}] [IPREQUEST={ON|OFF}]
   [LQR={ON|OFF|period}] [MAGIC={ON|OFF}] [MODEM={ON|OFF}]
   [MRU={ON|OFF|256..1656}] [MSSheader=40..200]
   [NULLFRAGTIMER=time] [NUMBER=number]
   [ONLINELIMIT={NONE|1..65535}] [OUTDATALIMIT={NONE|
   1..65535}] [PASSWORD=password] [PREDCHECK={CRC16|
   CRCCCITT}] [RECHALLENGE={ON=|OFF|360..3600}]
   [RESTART=time] [STACCHECK={LCB|SEQUENCE}]
   [STARENTITY=1..255] [TERMINATE={value|CONTINUOUS}]
   [TOTALDATALIMIT={NONE|1..65535}] [TYPE={DEMAND|PRIMARY|
   SECONDARY}] [UPRATE=0..100] [UPTIME=time]
   [USERNAME=username]
```

where *ppp-interface* is the PPP interface number, from 0 to 1023

The maximum number of PPP interfaces that can be created depends on the available memory (RAM) on the router. When the number of available buffers falls below buffer level 2, the command fails and an error is displayed. To see the number of available buffers, use the **show buffer** command on page 4-34 of Chapter 4, Configuring and Monitoring the System.

- *physical-interface* is:
  - ATM (e.g. atm0.1)
  - SYN*n*
  - ISDN-*callname*
  - ACC-*callname*
  - MIOX*n-circuitname*
  - TDM-*groupname*
  - TNL-*callname* (L2TP tunnel)
  - ETH*n-servicename*
  - VLAN*n-servicename*
- *e164number* is the phone number to dial when performing callback. It may contain digits (0–9) and should be a valid phone number as described in CCITT standard E.164.
- *value* is a retry threshold.
- *description* is a character string 1 to 70 characters long. Valid characters are any printable character.
- *time* is a timer value in seconds.

- *period* is an integer from 1 to 4294967295.
- *pool-name* is a character string, 1 to 15 characters long. Valid characters are any printable characters. If *pool-name* contains spaces, it must be in double quotes.
- *number* is the number of ISDN B channels.
- *password* is the password to use for authentication 1 to 64 characters long. It may contain any printable character and is case sensitive.
- *username* is the username to use for authentication 1 to 64 characters long. It may contain any printable character and is case sensitive.

**Description** This command requires a user with security officer privilege when the router is in security mode.

This command creates the specified PPP interface running over:

- an ATM channel
- a synchronous port
- an ISDN call
- an ACC call
- a MIOX circuit
- TDM group
- an L2TP call
- a PPP over Ethernet service
- a PPP over Ethernet service over a VLAN interface

For PPP over Ethernet and PPP over VLAN links, use the service name provided by your ISP, or the special service name ANY to specify that any service is acceptable.

PPP interfaces are enabled automatically when they are created.

The **over** parameter specifies the physical interface over which the PPP interface runs. Additional physical interfaces can be added to the PPP interface using the **add ppp** command on page 15-50.

The **authentication** parameter specifies the authentication protocol to be used on the physical interface or channel. If **chap** is specified, the Challenge-Handshake Authentication Protocol (CHAP) is used. If **pap** is specified, the Password Authentication Protocol (PAP) is used. If **either** is specified, the router uses the option negotiation process to negotiate the authentication protocol to be used with the device at the remote end of the link, specifying CHAP as the first choice. If **none** is specified, no authentication protocol is used. The default is **none**.

The **authmode** parameter specifies how authentication requests to peers are affected by the direction of the call. The **authmode** parameter is valid when the **authentication** parameter is set to a value other than **none** and the physical interface is a call. If **in** is specified, authentication is requested for incoming calls from peers. If **out** is specified, authentication is requested for outgoing calls to peers. If **inout** is specified, authentication is always requested regardless of the direction of the call. The default is **inout**. This parameter is valid for:

- ISDN calls
- ACC calls

The **bap** parameter specifies whether the Bandwidth Allocation Protocol is used for negotiating the activation of demand PPP links. The default is **on**.

The **bapmode** parameter specifies which peer originates another link to add to the multilink bundle. For **callback** mode, the number to call must be configured on the call at the lower layer (ISDN, ACC, MIOX, L2TP). The default is **call**.

The **cbdelay** parameter specifies the delay, in tenths of a second, between bringing down a call for callback and actually making the call back to the peer. This parameter handles different timing requirements of various ISDN switches and is valid for PPP links over ISDN calls and when the callback mode is **request**. The default is 1.

The **cbmode** parameter specifies whether a callback request is to be made or accepted during the LCP negotiation. If **request** is specified, a request is made for callback. If **accept** is specified, requests for callback received from the peer are accepted and processed; however, **authentication** must be set to **pap** or **chap**. If **off** is specified, no callback requests are made and callback requests are not accepted. The default is **off**.

The **cbnumber** parameter specifies the number to include when requesting a callback with the **cboperation** parameter set to **e164number**. The number specified should be a phone number as specified in the E.164 standard.

The **cboperation** parameter specifies the callback operation to be included in the callback request to specify to the peer how to determine the callback number. If **userauth** is specified, the peer uses the username and password supplied during authentication to look up the callback number. If **e164number** is specified, the callback number specified by the **cbnumber** parameter is included in the callback request. The default is **userauth**.

The **compalgorithm** parameter specifies the compression algorithm to use when compressing and decompressing PPP packets. If **predictor** is specified the Predictor algorithm is used with type 1 encapsulation as specified in RFC 1978. If **staclzs** is specified the Stac LZS algorithm is used as specified in RFC 1974. The default is **staclzs**. This parameter does not apply to L2TP calls.

The **compression** parameter enables or disables the use of compression for the interface. When used with multilink, setting **compression** to **on** compresses packets before they are sent to individual links. Setting **compression** to **link** enables compression for the link specified by the **over** parameter. The default is **off**. The **link** option should be used only when compression is required on some physical interfaces and not on others. For example, if a PPP multilink uses a compressing modem link and a normal dedicated leased line, **compression** should be set to **off** on the physical interface to which the modem is connected, and **link** for the physical interface to which the dedicated leased line is connected. If compression is required on all physical interfaces of a PPP interface, the **compression** parameter should be set to **on**. This parameter does not apply to L2TP calls.

The **configure** parameter sets the number of configure requests sent before some action is taken. For the LCP the action is to reset the hardware and start again. For all other protocols the action is to give up. The default is **continuous**, which means that requests are sent continuously.

The **debugmaxbytes** parameter specifies the maximum number of bytes that are displayed for each packet when the **ctrlpkt**, **datapkt**, **packet**, and **pkt** debug options are enabled. The default is 60.

The **description** parameter specifies a user-defined description for the interface, to make it easier to distinguish between a number of PPP interfaces.

The **downtime** parameter specifies the time, in seconds, that the PPP interface must have a total utilisation (as a percentage) below the threshold specified by the **downrate** parameter, before a channel is closed. The default is 60 for **downtime** and 20 for **downrate**. The **uprate**, **uptime**, **downrate** and **downtime** parameters are used with the **type** parameter to configure bandwidth on demand.

The **echo** parameter specifies whether LCP Echo Request and Echo Reply messages are used to determine link quality. If three consecutive Echo Request messages are transmitted without receiving an Echo Reply response, the link is deemed to be down. If **off** is specified, Echo Request messages are not transmitted. If **on** is specified, these messages are transmitted every 10 seconds. If a period in seconds is specified, Echo Request messages are transmitted at the specified interval. The LQR function has precedence over the echo function. Both cannot be enabled at the same time. However, both **lqr** and **echo** can be **on** at the same time if **echo** is turned on first.

The **encryption** parameter enables or disables the use of encryption for the interface. The default is **off**.

The **fragment** parameter applies to a multilink bundle interface and determines whether packets are fragmented. The default is **off**. Fragmentation cannot be enabled if compression or encryption are configured.

The **fragoverhead** parameter specifies the maximum allowable overhead, as a percentage, for fragmenting packets using the variable fragmentation scheme for multilink PPP. If this limit is exceeded for any packet, the packet is fragmented using the fixed fragmentation scheme. The default is 5. The variable fragmentation scheme spreads the packet over all the links in the multilink bundle by splitting the packet into variable sized fragments to match the speed of individual links. Larger fragments are transmitted over faster links, thereby providing an inherent load balancing scheme. The fixed fragmentation scheme spreads the packet over all the links in the multilink bundle by splitting the packet into equal fixed sized fragments. If the number of links is large and the packet is relatively small a fragment is not transmitted over every link.

The **idle** parameter controls the dial-on-demand feature. If **on** is specified, dial-on-demand is enabled with a default timer of 60 seconds. If a time is specified, dial-on-demand is enabled with the timer set to the specified time. If **off** is specified, dial-on-demand is disabled. When the dial-on-demand feature is activated, PPP brings up the link when there is traffic to be sent, and takes down the link when there has been no traffic for the specified timer period. The effect on a PPP interface using an on-demand call is to connect the call when traffic is to be sent and disconnect the call when no traffic has been sent or received for the specified time period. For leased lines, this parameter has no effect, as the links are always connected. The default is **off**.

Note that you can configure the interface so that OSPF keeps on-demand links active regardless of whether the interface is Up or Down. To do this, set the parameter **notifyospfdown=off** in the **add ip interface** or **set ip interface** commands. See "add ip interface" on page 22-80 of Chapter 22, Internet Protocol (IP) for further information.

The **indatalimit** parameter specifies the input data threshold, in megabytes, for the PPP interface. When the interface's cumulative input data counter exceeds this limit, the link is closed and any attempts to open it fail. If the input data counter for the interface is cleared with the **reset ppp** command on page 15-82, the link can be reopened. The default is **none**.

The **ippool** parameter specifies the IP pool to use to allocate IP addresses for the remote end of the PPP connection. If **none** is specified, IP addresses are not allocated from an IP pool. The default is **none**. See "IP Address Pools" on page 22-52 of Chapter 22, Internet Protocol (IP) for more information about creating IP address pools.

The **iprequest** parameter specifies whether to make a request for an IP address to be allocated by the peer during the IPCP negotiation. If **on** is specified, a request is made. If **off** is specified, a request is not made. The default is **off**.

The **lqr** parameter sets the LQR timer for link quality management. If **on** is specified, LQR is enabled with a default timer of 60 seconds. If a time is specified, LQR is enabled with the timer set to the specified time. If **off** is specified, LQR is disabled. The LQR function has precedence over the echo function. Both cannot be enabled at the same time. However, both **lqr** and **echo** can be **on** at the same time if **echo** is turned on first. The default is **on**.

The **magic** parameter enables or disables negotiation of the magic number option. The default is **on**. The magic number is used to determine if a interface is looped back. The interface does not reach an open state when there is a loopback.

The **modem** parameter specifies the state of synchronous modem control. The default is **off**. If **on** is specified, the router manipulates the DTR signal to trigger the modem to make a call whenever there is traffic (**idle=on**) or when a backup link is required (**type=secondary**). Raising DTR to the modem triggers the modem to initiate a call to the modem at the other end of the link and enter data transfer mode. The router keeps the DTR signal to the modem raised, and responds to the modem raising the DSR signal by activating the PPP interface. This parameter is valid when the **over** parameter specifies a synchronous interface.

The **mru** parameter specifies whether the MRU (Maximum Receive Unit) option is included, and what value is sent in LCP configuration requests while bringing up this interface. If **off** is specified, the MRU option is omitted. If a decimal value is specified, the transmitted MRU option is set to this value. If **on** is specified, the MRU is calculated normally. The default is **on**.

The **mssheader** parameter specifies the number of bytes within an MTU that are reserved for packet headers. This amount is subtracted from the MTU of the interface to define its Maximum Segment Size (MSS). The MSS specifies the maximum amount of data that can be contained within each TCP packet. The default is 120 bytes.

The **mssheader** parameter may only be used with an Ethernet or VLAN physical interface (PPPoE).

The **nullfragtimer** parameter specifies the maximum time, in seconds, a link in a multilink bundle may be idle before a NULL fragment is transmitted over the link. NULL fragments are used to keep the last sequence number transmitted over the link up to date. The default is 3.

The **number** parameter specifies the number of physical interfaces to be created. This parameter is valid when the **over** parameter specifies an ISDN call as the physical interface. The default is 1.

The **onlinelimit** parameter specifies the up-time threshold, in hours, for the PPP interface. When the interface's cumulative up-time counter exceeds this limit, the link is closed and any further attempts to open the link fails. If the up-time counter for the interface is cleared with the **reset ppp** command on page 15-82, the link can be reopened. The default is **none**, which sets no threshold.

The **outdatalimit** parameter specifies the output data threshold, in megabytes, for the PPP interface. When the interface's cumulative output data counter exceeds this limit, any further attempts to open the link fail. If the output data counter for the interface is cleared with the **reset ppp** command on page 15-82, the link can be reopened. The default is **none**, which sets no threshold.

The **password** parameter specifies the password to use when the peer requests authentication using either CHAP or PAP. This is normally required for network lines between routers, for which an authentication protocol has been selected with the **authentication** parameter.

The **predcheck** parameter specifies the type of CRC to be used for Predictor compression. The Predictor RFC specifies using CRC-16, however some router manufacturers have implemented Predictor with CRC-CCITT, which is the CRC specified in RFC 1662, "PPP in HDLC-link Framing". This value is not negotiated so the same value needs to be configured at both ends of the link for Predictor compression to work correctly. This parameter does not apply to L2TP calls.

The **rechallenge** parameter specifies if the CHAP rechallenge function is on, off or the maximum rechallenge period. PPP calculates a random period for the CHAP rechallenge between the maximum rechallenge period and a minimum of 5 minutes. If **on** is specified, the CHAP rechallenges take place with a maximum rechallenge period of 15 minutes. If **off** is specified, then CHAP rechallenges do not take place. If a time is specified, then the time, which is between 360 and 3600 seconds, is used as the upper limit on the rechallenge time calculation. The default is **on**.

The **restart** parameter specifies the time between successive retransmissions of unacknowledged configure requests or terminate requests. The default is 3 seconds.

The **staccheck** parameter specifies the check mode to used for the Stac LZS compression algorithm. If **sequence** is specified an incrementing sequence number is used to determine whether a packet has been lost and therefore whether the compression history needs to be reset. If **lcb** is specified an LCB value is used to determine if an error has occurred in a packet. The default is **sequence**. This parameter does not apply to L2TP calls.

The **starentity** parameter specifies the star entity and the encryption algorithm to be used by the encryption channel configured by the PPP interface. This parameter must be specified if PPP encryption is enabled.

The **terminate** parameter sets the number of terminate requests sent when trying to close a link before it is assumed the link is down. The default is 2. The **continuous** option specifies that requests be sent continuously.

The **totaldatalimit** parameter sets the total data throughput threshold, in megabytes, for the PPP interface. When the interface's cumulative total data counter exceeds this limit, the link is closed and any further attempts to open the link fail. If the total data counter for the interface is cleared with the **reset ppp** command on page 15-82, the link can be reopened. The default is **none**, which corresponds to no threshold.

The **type** parameter specifies the role of the physical interface for bandwidth on demand and link backup. The default is **primary**. If **primary** is specified, the link is open all the time (**idle=off**) or open whenever there is traffic (**idle=on**). If **secondary** is specified, the link is open when the associated primary link fails. If **demand** is specified, the link is opened when additional bandwidth is required.

The **uptime** parameter specifies the time, in seconds, that the PPP interface must have a total utilisation (as a percentage) above the threshold specified by the **uprate** parameter, before an additional channel is opened. The default is 30 for **uptime** and 80 for **uprate**. The **uprate**, **uptime**, **downrate** and **downtime** parameters are used with the **type** parameter to configure bandwidth on demand.

The **username** parameter specifies the username to be used when generating PAP authentication requests and when responding to CHAP authentication challenges. If the **username** is not set, the router's system name is used by default.

## Examples

To create PPP interface 0 with two on-demand channels over the ISDN call "ISDN-Region1", use the command:

```
create ppp=0 over=isdn-region1 idle=on num=2 type=demand
```

To create PPP interface 0 over a PPPoE client session over a VLAN interface with ID 2, using a service named "access", use the command:

```
create ppp=0 over=vlan2-access
```

To create a PPPoE interface that has a default MTU of 1492 with a required **mssheader** value of 100, use the command:

```
cre ppp=0 over=eth0-any mssheader=100
```

## Related Commands

**add ppp**
**delete ppp**
**delete ppp acservice**
**disable ppp**
**enable ppp**
**reset ppp**
**set ppp**
**show ppp**
**show ppp config**
**show ppp limits**

# create ppp template

**Syntax**

```
CREATE PPP TEMPLATE=template [COPY=template]
   [AUTHENTICATION={CHAP|EITHER|PAP|NONE}] [BAP={ON|OFF}]
   [BAPMODE={CALL|CALLBACK}] [CBDELAY=1..100]
   [CBMODE={ACCEPT|OFF|REQUEST}] [CBNUMBER=e164number]
   [CBOPERATION={E164NUMBER|USERAUTH}]
   [COMPALGORITHM={PREDICTOR|STACLZS}] [COMPRESSION={ON|
   OFF|LINK}] [DEBUGMAXBYTES=16..256]
   [DESCRIPTION=description] [DOWNRATE=0..100]
   [DOWNTIME=time] [ECHO={ON|OFF|period}] [ENCRYPTION={ON|
   OFF}] [FRAGMENT={ON|OFF}] [FRAGOVERHEAD=0..100]
   [IDLE={ON|OFF|time}] [INDATALIMIT={NONE|1..65535}]
   [IPPOOL={pool-name|NONE}] [IPREQUEST={ON|OFF}]
   [LOGIN={ALL|RADIUS|TACACS|USER}] [LQR={ON|OFF|period}]
   [MAGIC={ON|OFF}] [MAXLINKS=1..64] [MRU={ON|OFF|
   256..1656}] [MSSheader=40..200] [MTU=256..1500|
   256..1492] [MULTILINK={ON|OFF}] [NULLFRAGTIMER=time]
   [ONLINELIMIT={NONE|1..65535}] [OUTDATALIMIT={NONE|
   1..65535}] [PASSWORD=password] [PREDCHECK={CRC16|
   CRCCCITT}] [RECHALLENGE={ON|OFF|360..3600}]
   [RESTART=time] [STACCHECK={LCB|SEQUENCE}]
   [STARENTITY=1..255] [TERMINATE={value|CONTINUOUS}]
   [TOTALDATALIMIT={NONE|1..65535}] [UPRATE=0..100]
   [UPTIME=time] [USERNAME=username] [VJC={ON|OFF}]
```

where:

- *template* is a number from 0 to 31.
- *e164number* is the phone number to dial when performing callback. It may contain digits (0–9) and should be a valid phone number as described in CCITT standard E.164.
- *description* is a character string 1 to 70 characters long. Valid characters are any printable character.
- *period* is an integer from 1 to 4294967295.
- *time* is a timer value in seconds.
- *pool-name* is a character string 1 to 15 characters long. Valid characters are any printable characters. If *pool-name* contains spaces, it must be in double quotes.
- *password* is the password to use for authentication 1 to 64 characters long. It may contain any printable character, and is case sensitive.
- *value* is a retry threshold.
- *username* is the username to use for authentication 1 to 64 characters long. It may contain any printable character, and is case sensitive.

**Description**

This command creates a PPP template that is used to configure dynamic PPP interfaces, and requires a user with Security Officer privilege when the router is in security mode.

The PPP interface is created when one of the following is activated:

- ISDN call
- ACC call
- L2TP call
- PPP over Ethernet service

The **template** parameter specifies the number of the template to create. The specified template must not already exist.

The **authentication** parameter specifies the authentication protocol to be used on the physical interface or channel. If **chap** is specified, the Challenge-Handshake Authentication Protocol (CHAP) is used. If **pap** is specified, the Password Authentication Protocol (PAP) is used. If **either** is specified, the router uses the option negotiation process to negotiate the authentication protocol to be used with the device at the remote end of the link, specifying CHAP as the first choice. If **none** is specified, no authentication protocol is used. The default is **none**.

The **bap** parameter specifies whether the Bandwidth Allocation Protocol is used for negotiating the activation of demand PPP links. The default is **on**.

The **bapmode** parameter specifies which peer originates another link to add to the multilink bundle. For **callback** mode, the number to call must be configured on the call at the lower layer (ISDN, ACC, L2TP). The default is **call**.

The **cbdelay** parameter specifies the delay, in tenths of a second, between bringing down a call for callback and actually making the call back to the peer. This parameter is used to handle the different timing requirements, for example of various ISDN switches, and is valid when the callback mode is REQUEST. The default is 1.

The **cbmode** parameter specifies whether a callback request is to be made or accepted during the LCP negotiation. If **request** is specified, a request is made for callback. If **accept** is specified, requests for callback received from the peer are accepted and processed; however, **authentication** must be set to **pap** or **chap**. If **off** is specified, no callback requests are made and callback requests are not accepted. The default is **off**.

The **cbnumber** parameter specifies the number to include when requesting a callback with the **cboperation** parameter set to **e164number**. The number specified should be a phone number as specified in the E.164 standard.

The **cboperation** parameter specifies the callback operation to be included in the callback request to specify to the peer how to determine the callback number. If **userauth** is specified, the peer uses the username and password supplied during authentication to look up the callback number. If **e164number** is specified, the callback number specified by the **cbnumber** parameter is included in the callback request. The default is **userauth**.

The **compalgorithm** parameter specifies the compression algorithm to use when compressing and decompressing PPP packets. If **predictor** is specified the Predictor algorithm is used with type 1 encapsulation as specified in RFC 1978. If **staclzs** is specified, the Stac LZS algorithm is used as specified in RFC 1974. The default is **staclzs**. This parameter does not apply to L2TP calls.

The **compression** parameter enables or disables the use of compression for the interface. When used with multilink, setting **compression** to **on** compresses the packets before they are sent to the individual links. Setting **compression** to **link** enables compression for the link specified by the **over** parameter. The default is **off**. The **link** option should be used when compression is required on some physical interfaces and not on others. For example, if a PPP multilink uses a compressing modem link and a normal dedicated leased line, **compression** should be set to **off** on the physical interface to which the modem is connected, and **link** for the physical interface to which the dedicated leased line is connected. If compression is required on all physical interfaces of a PPP interface, the **compression** parameter should be set to **on**. This parameter does not apply to L2TP calls.

The **copy** parameter specifies the name of an existing template to copy as the defaults for this template. Any other parameters modify the copy.

The **debugmaxbytes** parameter specifies the maximum number of bytes that are displayed for each packet when the **ctrlpkt, datapkt, packet**, and **pkt** debug options are enabled. The default is **60**.

The **description** parameter specifies a user-defined description for the interface, to make it easier to distinguish between a number of PPP interfaces.

The **downtime** parameter specifies the time, in seconds, that the PPP interface must have a total utilisation (as a percentage) below the threshold specified by the **downrate** parameter, before a channel is closed. The default is **60** for **downtime** and **20** for **downrate**. The **uprate, uptime, downrate** and **downtime** parameters are used with the **type** parameter to configure bandwidth on demand.

The **echo** parameter specifies whether LCP Echo Request and Echo Reply messages are used to determine link quality. If three consecutive Echo Request messages are transmitted without receiving an Echo Reply response, the link is deemed to be down. If **off** is specified, Echo Request messages are not transmitted. If **on** is specified, Echo Request messages are transmitted every 10 seconds. If a period in seconds is specified, Echo Request messages are transmitted at the specified interval. The LQR function has precedence over the ECHO function. They cannot both be enabled at the same time. However, both **lqr** and **echo** can be **on** at the same time if **echo** is turned on first, then **lqr**.

The **encryption** parameter enables or disables the use of encryption for the interface. The default is **off**.

The **fragment** parameter applies to a multilink bundle interface, and determines whether packets are fragmented. The default is **off**. Fragmentation cannot be enabled if compression or encryption are configured.

The **fragoverhead** parameter specifies the maximum allowable overhead, as a percentage, for fragmenting packets using the variable fragmentation scheme for multilink PPP. If this limit is exceeded for any packet, the packet is fragmented using the fixed fragmentation scheme. The default is 5. The variable fragmentation scheme spreads the packet over all the links in the multilink bundle by splitting the packet into variable sized fragments to match the speed of individual links. Larger fragments are transmitted over faster links, thereby providing an inherent load balancing scheme. The fixed fragmentation scheme spreads the packet over all the links in the multilink bundle by splitting the packet into equal fixed sized fragments. If the number of links is large and the packet is relatively small a fragment is not transmitted over every link.

The **idle** parameter controls the dial-on-demand feature. If **on** is specified, dial-on-demand is enabled with a default timer of 60 seconds. If a time is specified, dial-on-demand is enabled with the timer set to the specified time. If **off** is specified, dial-on-demand is disabled. When the dial-on-demand feature is activated, PPP brings up the link when there is traffic to be sent, and takes down the link when there has been no traffic for the specified timer period. The effect on a PPP interface using an on-demand call is to connect the call when traffic is to be sent and disconnect the call when no traffic has been sent or received for the specified timer period. For leased lines, this parameter has no effect, as the links are always connected. The default is **off**.

Note that you can configure the interface so that OSPF keeps on-demand links active regardless of whether the interface is Up or Down. To do this, set the parameter **notifyospfdown=off** in the **add ip interface** or **set ip interface** commands. See "add ip interface" on page 22-80 of Chapter 22, Internet Protocol (IP) for further information.

The **indatalimit** parameter specifies the input data threshold, in megabytes, for the PPP interface. When the interface's cumulative input data counter exceeds this limit, the link is closed and any further attempts to open the link fails. If the input data counter for the interface is cleared with the **reset ppp** command on page 15-82, the link can be reopened. The default is **none**, which sets no threshold.

The **ippool** parameter specifies the IP pool to use to allocate IP addresses for dynamic dial-in PPP connections. If **none** is specified, IP addresses are not allocated from an IP pool. The default is **none**. See "IP Address Pools" on page 22-52 of Chapter 22, Internet Protocol (IP) for more information about creating IP address pools.

The **iprequest** parameter specifies whether a request is to be made for an IP address to be allocated by the peer during the IPCP negotiation. If **on** is specified, a request is made. If **off** is specified, a request is not specified. The default is **off**.

The **login** parameter specifies which login procedure the call creating this dynamic interface must use when it is activated. If **radius** is specified, the router sends requests to the configured RADIUS server(s) to authenticate the call. If **tacacs** is specified, the router sends requests to the configured TACACS server(s) to authenticate the call. If **user** is specified, the router checks the User Authentication Database to authenticate the call. If **all** is specified, the router tries all methods to authenticate the call.

The **lqr** parameter sets the LQR timer for link quality management. If **on** is specified, LQR is enabled with a default timer of 60 seconds. If a time is specified, LQR is enabled with the timer set to the specified time. If **off** is specified, LQR is disabled. The LQR function has precedence over the ECHO function. They cannot both be enabled at the same time. However, both **lqr** and **echo** can be **on** at the same time if **echo** is turned on first, then **lqr**. The default is **on**.

The **magic** parameter enables or disables negotiation of the magic number option. The default is **on**. The magic number determines if an interface is looped back. The interface does not reach an open state when there is a loopback.

The **maxlinks** parameter specifies the maximum number of links allowed in a multilink PPP interface created using this template. The default is 2.

When using the **maxlinks** parameter without BAP, a link is brought up fully before being checked. If the remote station is configured to bring up more links than the local station can accept, links may be brought up and torn down repeatedly. Therefore, to stop this situation occurring, also implement BAP when **maxlinks** is implemented.

The **mru** parameter specifies whether the MRU (Maximum Receive Unit) option is included, and what value is sent in LCP configuration requests while bringing up this interface. If **off** is specified, the MRU option is omitted. If a decimal value is specified, the transmitted MRU option is set to this value. If **on** is specified, the MRU is calculated normally. The default is **on**.

The **mssheader** parameter specifies the number of bytes within an MTU that are reserved for packet headers. This amount is subtracted from the MTU of the interface to define its Maximum Segment Size (MSS). The MSS specifies the maximum amount of data that can be contained within each TCP packet. The default is 120 bytes.

The **mssheader** parameter may only be used with an Ethernet or VLAN physical interface (PPPoE).

The **mtu** parameter specifies the value of the maximum transmission unit. The allowable MTU values for the PPP lower layers are shown in Table 15-9.

Table 15-9: Allowable MTU values

| Interface | Minimum MTU | Maximum MTU | Default MTU |
|---|---|---|---|
| PPP (except over Ethernet) | 256 | 1500 | 1500 |
| PPP over Ethernet | 256 | 1492 | 1492 |

We recommend that you leave the default MTU values unchanged for normal operations.

The **multilink** parameter specifies whether incoming dynamic PPP calls using this template are multilinked together. If **on** is specified, incoming dynamic PPP calls from the same peer and with the same authentication information are multilinked together. If **off** is specified, incoming dynamic PPP calls are not multilinked under any circumstances. The default is **on**.

The **nullfragtimer** parameter specifies the maximum time, in seconds, a link in a multilink bundle may be idle before a NULL fragment is transmitted over the link. NULL fragments are used to keep the last sequence number transmitted over the link up to date. The default is **3**.

The **onlinelimit** parameter specifies the up-time threshold, in hours, for the PPP interface. When the interface's cumulative up-time counter exceeds this limit, the link is closed and any further attempts to open the link fail. If the up-time counter for the interface is cleared with the **reset ppp** command on page 15-82, the link can be reopened. The default is **none**, which sets no threshold.

The **outdatalimit** parameter specifies the output data threshold, in megabytes, for the PPP interface. When the interface's cumulative output data counter exceeds this limit, any further attempts to open the link fail. If the output data counter for the interface is cleared with the **reset ppp** command on page 15-82, the link can be reopened. The default is **none**, which sets no threshold.

The **password** parameter specifies the password to use when the peer requests authentication using either CHAP or PAP. This is normally required for network lines between routers, for which an authentication protocol has been selected with the **authentication** parameter.

The **predcheck** parameter specifies the type of CRC to be used for Predictor compression. The Predictor RFC specifies using CRC-16, however some router manufacturers have implemented Predictor with CRC-CCITT, which is the CRC specified in RFC 1662, "PPP in HDLC-link Framing". This value is not negotiated so the same value needs to be configured at both ends of the link for Predictor compression to work correctly. This parameter does not apply to L2TP calls.

The **rechallenge** parameter specifies if the CHAP rechallenge function is on, off or the maximum rechallenge period. PPP calculates a random period for the CHAP rechallenge between the maximum rechallenge period and a minimum of 5 minutes. If **on** is specified, the CHAP rechallenges take place with a maximum rechallenge period of 15 minutes. If **off** is specified, then **chap** rechallenges do not take place. If a time is specified, then the time, which is between 360 and 3600 seconds, is used as the upper limit on the rechallenge time calculation. The default is **on**.

The **restart** parameter specifies the time between successive retransmissions of unacknowledged configure requests or terminate requests. The default is 3 seconds.

The **staccheck** parameter specifies the check mode to used for the Stac LZS compression algorithm. If **sequence** is specified an incrementing sequence number is used to determine whether a packet has been lost and therefore whether the compression history needs to be reset. If **lcb** is specified an LCB value is used to determine if an error has occurred in a packet. The default is **sequence**. This parameter does not apply to L2TP calls.

The **starentity** parameter specifies the star entity and the encryption algorithm to be used by the encryption channel configured by the PPP interface. This parameter must be specified if PPP encryption is enabled.

The **terminate** parameter sets the number of terminate requests sent when trying to close a link before it is assumed the link is down. The default is **2**. The **continuous** option specifies that requests be sent continuously.

The **totaldatalimit** parameter sets the total data throughput threshold, in megabytes, for the PPP interface. When the interface's cumulative total data counter exceeds this limit, the link is closed and any further attempts to open the link fail. If the total data counter for the interface is cleared with the **reset ppp** command on page 15-82, the link can be reopened. The default is **none**, which corresponds to no threshold.

The **uptime** parameter specifies the time, in seconds, that the PPP interface must have a total utilisation (as a percentage) above the threshold specified by the **uprate** parameter, before an additional channel is opened. The default is **30** for **uptime** and **80** for **uprate**. The **uprate**, **uptime**, **downrate** and **downtime** parameters are used with the **type** parameter to configure bandwidth on demand.

The **username** parameter specifies the username to be used when generating PAP authentication requests and when responding to CHAP authentication challenges. If the **username** is not set, the router's system name is used by default.

The **vjc** parameter specifies whether Van Jacobson Header Compression will be negotiated on dynamic PPP interfaces created from the template. The default is **off**.

**Examples** To create a template that creates a dynamic PPP interface with BAP and Predictor compression enabled, use the command:

```
create ppp template=1 bap=on bapmode=call
   description="dynamic ppp with predictor and bap"
   compression=on compalgorithm=predictor
```

To create a template that creates a dynamic PPP interface with BAP and STAC LZS compression enabled, use the command:

```
create ppp template=1 bap=on bapmode=call
   description="dynamic ppp with stac lzs and bap"
   compression=on compalgorithm=staclzs
```

To create PPP template *pppT2* using the factory default settings, use the command:

```
create ppp template=2
```

To create a PPPoE template that uses the default MTU of 1000 and has an MSSH value of 200, use the command:

```
cre ppp temp=1 mssheader=200
```

**Related Commands**

**destroy ppp template**
**disable ppp template debug**
**enable ppp template debug**
**reset ppp**
**set ppp acservice**
**set ppp template**
**show ppp pppoe**
**show ppp limits**
**show ppp template**

# delete ppp

**Syntax**

```
DELETE PPP=ppp-interface OVER=physical-interface
   [NUMBER=number] [TYPE={DEMAND|PRIMARY|SECONDARY}]
```

where:

- *ppp-interface* is the PPP interface number from 0 to 1023.
- *physical-interface* is:
  - ATM (e.g. atm0.1)
  - SYN*n*
  - ISDN-*callname*
  - ACC-*callname*
  - MIOX*n-circuitname*
  - TDM-*groupname*
  - TNL-*callname*
  - ETH*n-servicename*
  - VLAN*n-servicename*
- *number* is the number of ISDN B channels to delete.

**Description**

This command deletes the specified lower layer interface from an existing PPP multilink bundle. The interface may be left with no lower layer interfaces.

The **over** parameter specifies the interface to be deleted.

The **number** parameter specifies the number of ISDN channels to be deleted. This parameter is valid when the **over** parameter specifies an ISDN call as the interface. The default is **1**.

The **type** parameter specifies the role of the interface for bandwidth on demand and link backup. The default is **primary**.

**Examples**

To delete Ethernet interface 2 as a interface from PPP interface 1, use the command:

```
delete ppp=1 over=eth2-any
```

To delete ISDN call “demand” as a interface from PPP interface 1, use the command:

```
delete ppp=1 over=isdn-demand
```

To delete the PPPoE service “any” on vlan2 as a physical interface from PPP interface 1, use the command:

```
delete ppp=1 over=vlan2-any
```

**Related Commands**

**add ppp**
**add ppp acservice**
**delete ppp acservice**
**disable ppp**
**enable ppp**
**reset ppp**
**set ppp**
**show ppp**

# delete ppp acservice

**Syntax**

```
DELete PPP ACservice=service-name [ACINTerface={NONE|
   interface}]
```

where:

- *service-name* is a character string of up to 18 characters. *service-name* should not be "any".
- *interface* is an interface name formed by concatenating an interface type and an interface instance (e.g. eth0). Valid interface types are ETH and VLAN.

**Description**

This command deletes a PPP over Ethernet Access Concentrator service from the router. Note that it is not possible to delete a service that is currently in use.

The **acservice** parameter specifies the name of the PPPoE service that is to be deleted.

The **acinterface** parameter specifies the interface on which the service is offered. This parameter is used to further identify the service to delete, as it is possible to have two or more services with the same name, but which are offered on different interfaces:

- If you specify an interface, it is on that interface that the service with the specified name is deleted.
- If you specify **none,** the service offered on the Ethernet port is deleted if it was added with **acinterface=none** specified in the **add ppp acservice** command.

If multiple interfaces exist for the service, you are prompted to specify an **acinterface**. The default is **none**.

**Examples**

To delete the PPPoE Access Concentrator service named armadillo from Ethernet interface 0, use the command:

```
del ppp ac=armadillo acint=eth0
```

**Related Commands**

**add ppp acservice**
**create ppp**
**create ppp template**
**destroy ppp**
**destroy ppp template**
**set ppp acservice**

# destroy ppp

**Syntax** `DESTROY PPP=`*`ppp-interface`*

where *ppp-interface* is the PPP interface number, from 0 to 1023

**Description** This command destroys the specified PPP interface, as opposed to the **delete ppp** command on page 15-70, which deletes a physical interface used by a PPP interface.

**Examples** To destroy PPP interface 0, use the command:

```
destroy ppp=0
```

**Related Commands**
**add ppp**
**add ppp acservice**
**delete ppp**
**disable ppp**
**enable ppp**
**reset ppp**
**set ppp**
**show ppp**

# destroy ppp template

**Syntax** `DESTROY PPP TEMPLATE=`*`template`*

where *template* is a number from 0 to 31

**Description** This command destroys the specified PPP template and eliminates any call associations. The **template** parameter specifies the number of the template to destroy. The specified template must already exist.

**Examples** To destroy template 1, use the command:

```
destroy ppp template=1
```

**Related Commands**
**create ppp template**
**disable ppp template debug**
**enable ppp template debug**
**set ppp acservice**
**show ppp pppoe**

# disable ppp

**Syntax** `DISABLE PPP=`*`ppp-interface`*

where *ppp-interface* is the PPP interface number, from 0 to 1023

**Description** This command disables the specified PPP interface. The interface must currently be enabled. The interface is not available for use by higher layer network protocols, but the configuration of a static PPP interface is retained, and is restored when the interface is re-enabled. PPP interfaces are enabled automatically when they are created.

**Examples** To disable PPP interface 2, use the command:

```
disable ppp=2
```

**Related Commands**
**add ppp**
**add ppp acservice**
**delete ppp**
**disable ppp**
**enable ppp**
**reset ppp**
**set ppp**
**show ppp**

# disable ppp accessconcentrator

**Syntax** `DISABLE PPP ACCESSCONCENTRATOR`

**Description** This command disables PPPoE Access Concentrator (AC) mode on the router. When AC mode has been disabled, the router does not respond to PPPoE discovery initiation packets, even if it has the requested service defined. PPPoE AC mode cannot be disabled while services are in use.

**Example** To stop the router acting as a PPPoE access concentrator and offering any AC services that have been defined on the router, use the command:

```
disable ppp acessconcentrator
```

**Related Commands**
**add ppp**
**delete ppp**
**disable ppp**
**enable ppp**
**set ppp**
**show ppp**

# disable ppp debug

**Syntax**

```
DISABLE PPP=ppp-interface DEBUG={ALL|AUTH|BAPSTATE|
   CALLBACK|CTRLPKT|DATAPKT|DECODE|DEMAND|ENCO|LCP|LQR|
   NCP|PACKET|PKT|PPPOE|UTILISATION}[,...]
```

where *ppp-interface* is the PPP interface number, from 0 to 1023

**Description**

This command disables the debugging option for the specified PPP interface. The option must currently be enabled. A list of options separated by commas may be specified to disable more than one debugging option at a time. By default PPP debugging is disabled.

The **debug** parameter specifies which debugging options are to be disabled. The value of this parameter is a single item or a comma-separated list of items. The items allowed and the debugging that results from specifying the item are shown in Table 15-10.

Table 15-10: Point-to-Point Protocol (PPP) debugging options

| Option | Description |
|---|---|
| all | All debug options. |
| auth | PPP authentication. If LCP opens on a link but the network protocols remain in the closed state, the most likely cause is an authentication failure. |
| bapstate | BAP state machine transitions. |
| callback | Callback state machine transitions. |
| ctrlpkt | Hexadecimal dump of control packets received and transmitted on the PPP interface. |
| datapkt | Hexadecimal dump of data packets received and transmitted on the PPP interface. |
| decode | Decoded control packets. Disabling this option also automatically disables AUTH debug. |
| demand | Packets that cause on-demand links to be activated. |
| enco | ENCO state machine used to control attachment to and detachment from the ENCO (encryption/compression) module. |
| lcp | LCP state machine transitions. |
| lqr | Decoded LQR packets. |
| ncp | NCP state machine transitions. |
| packet<br>pkt | Hexadecimal dump of all packets received and transmitted on the PPP interface. This option has the same effect as CTRLPKT and DATAPKT specified at the same time |
| pppoe | PPPoE discovery packets received and transmitted, and PPPoE state transitions. |
| utilisation | Utilisation measurements for each lower layer interface and the overall utilisation. |

**Examples**

To disable all debugging options on PPP interface 2, use the command:

```
disable ppp=2 debug=all
```

**Related Commands**

**disable ppp template debug**
**enable ppp debug**
**enable ppp template debug**
**show ppp debug**
**show ppp utilisation**

# disable ppp template debug

**Syntax**

```
DISABLE PPP TEMPLATE=template DEBUG={ALL|AUTH|BAPSTATE|
   CALLBACK|CTRLPKT|DATAPKT|DECODE|DEMAND|ENCO|LCP|LQR|
   NCP|PACKET|PKT|PPPOE|UTILISATION}[,...]
```

where *template* is a number from 0 to 31

**Description**

This command disables the debugging option for dynamic PPP interfaces created using the specified PPP template. A list of options separated by commas may be specified to disable more than one debugging option at a time.

The **template** parameter specifies the number of the template for which debugging is to be disabled. The specified template must already exist.

The **debug** parameter specifies which debugging options are to be disabled. The value of this parameter is a single item or a comma-separated list of items. The items allowed and the debugging that results from specifying the item are shown in Table 15-10 on page 15-74.

**Examples**

To disable the display of debugging information for dial on demand link activation on template 2, use the command:

```
disable ppp template=2 debug=demand
```

**Related Commands**

**create ppp template**
**destroy ppp template**
**enable ppp template debug**
**set ppp acservice**
**show ppp utilisation**

# enable ppp

**Syntax** `ENABLE PPP=`*`ppp-interface`*

where *ppp-interface* is the PPP interface number from 0 to 1023

**Description** This command enables a specific PPP interface. The interface must be a currently disabled static interface. The interface configuration is restored to the settings in existence before the interface was disabled. The interface is made available to network layer protocols to transmit and receive data. PPP interfaces are enabled automatically when they are created.

This command requires a user with security officer privilege when the router is in security mode.

**Examples** To enable PPP interface 2, use the command:

```
enable ppp=2
```

**Related Commands**
**add ppp**
**add ppp acservice**
**delete ppp**
**delete ppp acservice**
**disable ppp**
**reset ppp**
**set ppp**
**show ppp**

# enable ppp accessconcentrator

**Syntax** `ENABLE PPP ACCESSCONCENTRATOR`

**Description** This command enables the router to perform as a PPPoE Access Concentrator (AC). While performing as an AC, the router offers AC services that have been defined with the **add ppp acservice** command. The router does not respond to PPPoE discovery initiation packets, even when it has the requested service defined, unless Access Concentrator mode has been enabled with this command.

**Examples** To define an AC service named 'bob' that uses dynamic PPP interfaces based on PPP template 1 (which already exists) and then enable the service to be offered by the router as an AC, use the commands:

```
add ppp acservice=bob template=1

enable ppp accessconcentrator
```

**Related Commands**
**add ppp**
**add ppp acservice**
**delete ppp**
**delete ppp acservice**
**disable ppp**
**enable ppp**
**set ppp**
**set ppp acservice**
**show ppp**

# enable ppp debug

**Syntax**

```
ENABLE PPP=ppp-interface DEBUG={ALL|AUTH|BAPSTATE|
   CALLBACK|CTRLPKT|DATAPKT|DECODE|DEMAND|ENCO|LCP|LQR|
   NCP|PACKET|PKT|PPPOE|UTILISATION}[,...]
   [ASYN=port-number] [TIMEOUT={NONE|1..4000000000}]
   [NUMPKTS={CONT|1..4000000000}]
```

where:

- *ppp-interface* is the PPP interface number from 0 to 1023.
- *port-number* is the number of an asynchronous port on the router. Port numbers start at zero (0).

**Description**

This command enables the debugging option for a specific PPP interface. Debugging may or may not be enabled already. By default PPP debugging is disabled. This command requires a user with Security Officer privilege when the router is in security mode.

Debugging information is sent to the port or Telnet session from which the command was entered if the **asyn** parameter is not specified; otherwise, it is sent to the specified port. A list of options separated by commas may be specified to enable more than one debugging option at a time. For packet debugging, the number of packets output may be specified. For all other types of debugging, the length of time to continue debugging may be specified.

The **debug** parameter specifies which debugging options are to be enabled. The value of this parameter is a single item or a comma-separated list of items. The items allowed and the debugging that results from specifying the item are shown in Table 15-11.

Table 15-11: Point-to-Point Protocol (PPP) debugging options

| Option | Description |
|---|---|
| all | All debug options. |
| auth | PPP authentication. If LCP opens on a link but the network protocols remain in the closed state, the most likely cause is an authentication failure. |
| bapstate | BAP state machine transitions. |
| callback | Callback state machine transitions. |
| ctrlpkt | Hexadecimal dump of control packets received and transmitted on the PPP interface. |
| datapkt | Hexadecimal dump of data packets received and transmitted on the PPP interface. |
| decode | Decoded control packets. Enabling this option also automatically enables AUTH debug. |
| demand | Packets that cause on-demand links to be activated. |
| enco | ENCO state machine used to control attachment to and detachment from the ENCO (encryption/compression) module. |
| lcp | LCP state machine transitions. |
| lqr | Decoded LQR packets. |
| ncp | NCP state machine transitions. |

Table 15-11: Point-to-Point Protocol (PPP) debugging options (cont.)

| Option | Description |
|---|---|
| packet<br>pkt | Hexadecimal dump of all packets received and transmitted on the PPP interface. This option has the same effect as CTRLPKT and DATAPKT specified at the same time |
| pppoe | PPPoE discovery packets received and transmitted, and PPPoE state transitions. |
| utilisation | Utilisation measurements for each lower layer interface and the overall utilisation. |

**Caution** Enabling all debug options with **enable ppp debug=all** can generate enormous amounts of output and cause the router to lock up. Use the **timeout** or **numpkts** options to limit the amount of output generated.

The **asyn** parameter specifies the asynchronous port to which the debug output is to be sent. This enables debugging to be enabled in a script. The default is to send the output to the terminal or Telnet session from which the command was executed. Each time the **enable ppp debug** command is entered the destination of the debugging output is calculated again using this rule.

The **timeout** parameter specifies a time in seconds after which debugging automatically ceases. If **none** is specified, then debugging must be disabled manually. The timeout applies to debugging modes that do not involve the output of data packets; that is, all debugging modes except for **ctrlpkt, datapkt, decode, lqr**, and **pkt**. The value of the **timeout** parameter the first time an applicable debugging mode is enabled is retained for future **enable ppp template debug** commands. The default is **none**.

The **numpkts** parameter specifies, for **ctrlpkt, datapkt, decode, lqr, packet**, and **pkt** debugging, the number of packets to be displayed before debugging ceases. This option is useful when attempting to debug a very busy link, since the amount of output generated by PKT debugging can easily cause the router to lock up the device to which the debugging output is being sent. The value of this parameter the first time **ctrlpkt, datapkt, decode, lqr, packet**, and **pkt** debugging are enabled is retained for subsequent **enable ppp template debug** commands. If **cont** is specified, packet debugging continues indefinitely and must be disabled manually. The default is **cont**.

## Examples

To enable the display of debugging information for dial on demand link activation on PPP interface 2, use the command:

```
enable ppp=2 debug=demand
```

## Related Commands

**disable ppp debug**
**disable ppp template debug**
**enable ppp template debug**
**show ppp debug**
**show ppp utilisation**

# enable ppp template debug

**Syntax**

```
ENABLE PPP TEMPLATE=template DEBUG={ALL|AUTH|BAPSTATE|
   CALLBACK|CTRLPKT|DATAPKT|DECODE|DEMAND|ENCO|LCP|LQR|
   NCP|PACKET|PKT|PPPOE|UTILISATION}[,...]
   [ASYN=port-number] [TIMEOUT={NONE|1..4000000000}]
   [NUMPKTS={CONT|1..4000000000}]
```

where:

- *template* is a number from 0 to 31.
- *port-number* is the number of an asynchronous port on the router. Ports are numbered starting at zero (0).

**Description**

This command enables the debugging option for dynamic PPP interfaces created using the specified PPP template. Debugging may or may not be enabled already. Debugging information is sent to the port or Telnet session from which the command was entered if the ASYN parameter was not specified; otherwise, it is sent to the specified port. A list of options separated by commas may be specified to enable more than one debugging option at a time. For packet debugging, the number of packets output may be specified. For all other types of debugging, the length of time to continue debugging may be specified.

This command requires a user with Security Officer privilege when the router is in security mode.

The **template** parameter specifies the number of the template for which debugging is to be enabled. The specified template must already exist.

The **debug** parameter specifies which debugging options are to be enabled. The value of this parameter is a single item or a comma-separated list of items. The items allowed and the debugging that results from specifying the item are shown in Table 15-11 on page 15-78.

**Important** Enabling all debug options with **enable ppp template debug=all** can generate enormous amounts of output and cause the router to lock up. Use the **timeout** or **numpkts** options to limit the amount of output generated.

The **asyn** parameter specifies the asynchronous port to which the debug output is to be sent. This enables debugging to be enabled in a script. The default is to send the output to the terminal or Telnet session from which the command was executed. Each time the **enable ppp template debug** command is entered the destination of the debugging output is calculated again using this rule.

The **timeout** parameter specifies a time in seconds after which debugging automatically stops. If **none** is specified, then debugging must be disabled manually. The timeout applies to debugging modes that do not involve the output of data packets; that is, all debugging modes except for **ctrlpkt**, **datapkt, decode, lqr**, and **pkt**. The value of the **timeout** parameter the first time an applicable debugging mode is enabled is retained for future **enable ppp template debug** commands. The default is **none**.

The numpkts parameter specifies, for **ctrlpkt, datapkt, decode, lqr, packet**, and **pkt** debugging, the number of packets to be displayed before debugging ceases. This option is useful when attempting to debug a very busy link, since the amount of output generated by PKT debugging can easily cause the router

to lock up the device to which the debugging output is being sent. The value of this parameter the first time **ctrlpkt, datapkt, decode, lqr, packet**, and **pkt** debugging are enabled is retained for subsequent **enable ppp template debug** commands. If **cont** is specified, packet debugging continues indefinitely and must be disabled manually. The default is **cont**.

**Examples** To enable the display of debugging information for dial on demand link activation on template 2, use the command:

```
enable ppp template=2 debug=demand
```

**Related Commands**
**create ppp template**
**destroy ppp template**
**disable ppp template debug**
**set ppp acservice**
**show ppp pppoe**
**show ppp utilisation**

# purge ppp

**Syntax**
```
PURGE PPP
```

**Description** This command destroys all PPP interfaces and reinitialises the PPP module.

**Examples** To the PPP configuration, use the command:

```
purge ppp
```

**Related Commands**
**delete ppp**
**delete ppp acservice**
**disable ppp**
**enable ppp**

# reset ppp

**Syntax** `RESET PPP=`*`ppp-interface`* `[COUNTER] [LINKCOUNTER={ONLINE|INDATA|OUTDATA|TOTALDATA|ALL}]`

where *ppp-interface* is the PPP interface number, from 0 to 1023

**Description** This command resets the specified PPP interface, the general counters for the specified PPP interface, or the cumulative up-time, and input/output data counters.

If neither the **counter** nor **linkcounter** parameters are specified, the interface is reset if it is a static interface, forcing the interface to renegotiate all protocols and options.

If the **counter** parameter is specified, all counters for the interface are reset to zero (0) except for the Link Quality Monitoring (LQM) counters, and cumulative up-time and input/output data counters.

If the **linkcounter** parameter is specified, one or all of the up-time and input/output data counters are reset to zero (0). If **online** is specified the up-time counter is reset to zero. If **indata** is specified the input data counter is reset to zero. If **outdata** is specified the output data counter is reset to zero. If **totaldata** is specified the total data counter is reset to zero. If **all** is specified all four counters are reset to zero.

**Examples** To reset PPP interface 0, use the command:

```
reset ppp=0
```

To reset all counters for PPP interface 0 without resetting the interface itself, use the command:

```
reset ppp=0 counter
```

**Related Commands**

**add ppp acservice**
**delete ppp**
**delete ppp acservice**
**disable ppp**
**enable ppp**
**purge ppp**
**set ppp**
**show ppp limits**

# set ppp

**Syntax**

```
SET PPP [DNSPRIMARY=ipadd] [DNSSECONDARY=ipadd]
   [WINSPRIMARY=ipadd] [WINSSECONDARY=ipadd]

SET PPP=ppp-interface [OVER=physical-interface]
   [AUTHENTICATION={CHAP|EITHER|PAP|NONE}] [AUTHMODE={IN|
   OUT|INOUT}] [BAP={ON|OFF}] [BAPMODE={CALL|CALLBACK}]
   [CBDELAY=1..100] [CBMODE={ACCEPT|OFF|REQUEST}]
   [CBNUMBER=e164number] [CBOPERATION={E164NUMBER|
   USERAUTH}] [COMPALGORITHM={PREDICTOR|STACLZS}]
   [COMPRESSION={ON|OFF|LINK}] [CONFIGURE={value|
   CONTINUOUS}] [DEBUGMAXBYTES=16..256]
   [DESCRIPTION=description] [DOWNRATE=0..100]
   [DOWNTIME=time] [ECHO={ON|OFF|period}] [ENCRYPTION={ON|
   OFF}] [FRAGMENT={ON|OFF}] [FRAGOVERHEAD=0.100]
   [IDLE={ON|OFF|time}] [INDATALIMIT={NONE|1..65535}]
   [IPPOOL={pool-name|NONE}] [IPREQUEST={ON|OFF}]
   [LQR={ON|OFF|period}] [MAGIC={ON|OFF}] [MAXLINKS=1..64]
   [MODEM={ON|OFF}] [MRU={ON|OFF|256..1656}]
   [MSSheader=40..200]  [NULLFRAGTIMER=time]
   [ONLINELIMIT={NONE|1..65535}] [OUTDATALIMIT={NONE|
   1..65535}] [PASSWORD=password] [PREDCHECK={CRC16|
   CRCCCITT}] [RECHALLENGE={ON|OFF|360..3600}]
   [RESTART=time] [STACCHECK={LCB|SEQUENCE}]
   [STARENTITY=1..255] [TERMINATE={value|CONTINUOUS}]
   [TOTALDATALIMIT={NONE|1..65535}] [TYPE={DEMAND|PRIMARY|
   SECONDARY}] [UPRATE=0..100] [UPTIME=time]
   [USERNAME=username]
```

where:

- *ppp-interface* is the PPP interface number, from 0 to 1023.
- *physical-interface* is:
  - ATM (e.g. atm0.1)
  - SYN*n*
  - ISDN-*callname*
  - ACC-*callname*
  - MIOX*n-circuitname*
  - TDM-*groupname*
  - TNL-*callname*
  - ETH*n-servicename*
  - VLAN*n-servicename*
- *e164number* is the phone number to dial when performing callback. It may contain digits (0–9) and should be a valid phone number as described in CCITT standard E.164.
- *value* is a retry threshold.
- *ipadd* is an IP address in dotted decimal notation.
- *description* is a character string 1 to 70 characters long. Valid characters are any printable character.
- *time* is a timer value in seconds.

- *period* is a decimal number from 1 to 4294967295.
- *pool-name* is a character string 1 to 15 characters long. Valid characters are any printable characters. If *pool-name* contains spaces, it must be in double quotes.
- *password* is the password to use for authentication 1 to 64 characters long. It may contain any printable character, and is case sensitive.
- *username* is the username to use for authentication 1 to 64 characters long. It may contain any printable character, and is case sensitive.

**Description** This command requires a user with security officer privilege when the router is in security mode. It is used to change the configuration parameters of a PPP interface running over:

- an ATM channel
- a synchronous port
- an ISDN call
- an ACC call
- a MIOX circuit
- TDM group
- an L2TP call
- a PPP over Ethernet service
- a PPP over Ethernet service over a VLAN interface

For PPP over Ethernet and PPP over VLAN links, use the service name provided by your ISP, or the special service name **any** to specify that any service is acceptable.

If a dynamic interface is modified with this command, the changes are lost when the interface goes down. This command is also used to set global primary and secondary DNS and WINS server addresses. In this case the PPP interface may not be specified. All other options require the PPP interface to be specified.

The **over** parameter specifies the physical interface over which the PPP interface is running. Additional physical interfaces can be added to the PPP interface with the **add ppp** command on page 15-50.

The **authentication** parameter specifies the authentication protocol to be used on the physical interface or channel. If **chap** is specified, the Challenge-Handshake Authentication Protocol (CHAP) is used. If **pap** is specified, the Password Authentication Protocol (PAP) is used. If **either** is specified, the router uses the option negotiation process to negotiate the authentication protocol to be used with the device at the remote end of the link, specifying CHAP as the first choice. If **none** is specified, no authentication protocol is used. The default is **none**.

The **authmode** parameter specifies how authentication requests to peers are affected by the direction of the call. The **authmode** parameter is valid when the **authentication** parameter is set to a value other than **none** and the physical interface is a call. If **in** is specified, authentication is requested for incoming calls from peers. If **out** is specified, authentication is requested for outgoing calls to peers. If **inout** is specified, authentication is always requested regardless of the direction of the call. The default is INOUT. This parameter is valid for:

- ISDN calls
- ACC calls

The **bap** parameter specifies whether the Bandwidth Allocation Protocol is used for negotiating the activation of demand PPP links. The default is **on**.

The **bapmode** parameter specifies which peer originates another link to add to the multilink bundle. For **callback** mode, the number to call must be configured on the call at the lower layer (ISDN, ACC, MIOX, L2TP). The default is **call**.

The **cbdelay** parameter specifies the delay, in tenths of a second, between bringing down a call for callback and actually making the call back to the peer. This parameter handles different timing requirements of various ISDN switches and is valid for PPP links over ISDN calls and when the callback mode is **request**. The default is **1**.

The **cbmode** parameter specifies whether a callback request is made or accepted during the LCP negotiation. If **request** is specified, a request is made for callback. If **accept** is specified, requests for callback received from the peer are accepted and processed, however **authentication** must be set to PAP or CHAP. If **off** is specified, no callback requests are made and callback requests are not accepted. The default is **off**.

The **cbnumber** parameter specifies the number to include when requesting a callback with the **cboperation** parameter set to **e164number**. The number specified should be a phone number as specified in the E.164 standard.

The **cboperation** parameter specifies the callback operation to be included in the callback request to specify to the peer how to determine the callback number. If **userauth** is specified, the peer uses the username and password supplied during authentication to look up the callback number. If **e164number** is specified, the callback number specified by the **cbnumber** parameter is included in the callback request. The default is **userauth**.

The **compalgorithm** parameter specifies the compression algorithm to use when compressing and decompressing PPP packets. If **predictor** is specified the Predictor algorithm is used with type 1 encapsulation as specified in RFC 1978. If **staclzs** is specified the Stac LZS algorithm is used as specified in RFC 1974. The default is **staclzs**. This parameter does not apply to L2TP calls.

The **compression** parameter enables or disables the use of compression for the interface. When used with multilink, setting **compression** to **on** compresses packets before they are sent to individual links. Setting **compression** to **link** enables compression for the link specified by the **over** parameter. The default is **off**. The **link** option should be used only when compression is required on some physical interfaces and not on others. For example, if a PPP multilink uses a compressing modem link and a normal dedicated leased line, **compression** should be set to **off** on the physical interface to which the modem is connected, and **link** for the physical interface to which the dedicated leased line is connected. If compression is required on all physical interfaces of a PPP interface, the **compression** parameter should be set to **on**. This parameter does not apply to L2TP calls.

The **configure** parameter sets the number of configure requests sent before some action is taken. For the LCP the action is to reset the hardware and start again. For all other protocols the action is to give up. The default is **continuous**, which means that requests are sent continuously.

The **debugmaxbytes** parameter specifies the maximum number of bytes that are displayed for each packet when the **ctrlpkt, datapkt, packet**, and **pkt** debug options are enabled. The default is **60**.

The **description** parameter specifies a user-defined description for the interface, to make it easier to distinguish between a number of PPP interfaces.

The **dnsprimary** parameter specifies the IP address to pass to a peer when it requests a primary DNS address using the IPCP primary DNS option. If neither this parameter nor the IP nameserver is configured, and the PPP interface is assigned an IP interface, then IPCP negotiates for a primary DNS address.

The **dnssecondary** parameter specifies the IP address to pass to a peer when it requests a secondary DNS address using the IPCP secondary DNS option. If neither this parameter nor the IP Secondary Nameserver is configured, and the PPP interface is assigned an IP interface, then IPCP negotiates for a secondary DNS address.

The **downtime** parameter specifies the time, in seconds, that the PPP interface must have a total utilisation (as a percentage) below the threshold specified by the **downrate** parameter, before a channel is closed. The default is **60** for **downtime** and **20** for **downrate**. The **uprate**, **uptime**, **downrate** and **downtime** parameters are used with the **type** parameter to configure bandwidth on demand.

The **echo** parameter specifies whether LCP Echo Request and Echo Reply messages are used to determine link quality. If three consecutive Echo Request messages are transmitted without receiving an Echo Reply response, the link is deemed to be down. If **off** is specified, Echo Request messages are not transmitted. If **on** is specified, these messages are transmitted every 10 seconds. If a period in seconds is specified, Echo Request messages are transmitted at the specified interval. The LQR function has precedence over the echo function. Both cannot be enabled at the same time. However, both **lqr** and **echo** can be **on** at the same time if **echo** is turned on first.

The **encryption** parameter enables or disables the use of encryption for the interface. The default is **off**.

The **fragment** parameter applies to a multilink bundle interface, and determines whether packets are fragmented. The default is **off**. Fragmentation cannot be enabled if compression or encryption are configured.

The **fragoverhead** parameter specifies the maximum allowable overhead, as a percentage, for fragmenting packets using the variable fragmentation scheme for multilink PPP. When a packet exceeds this limit, it is fragmented using the fixed fragmentation scheme. The default is 5. The variable fragmentation scheme spreads the packet over all the links in the multilink bundle by splitting the packet into variable sized fragments to match the speed of individual links. Larger fragments are transmitted over faster links, thereby providing an inherent load balancing scheme. The fixed fragmentation scheme spreads the packet over all the links in the multilink bundle by splitting the packet into equal fixed sized fragments. If the number of links is large and the packet is relatively small a fragment is not transmitted over every link.

The **idle** parameter controls the dial-on-demand feature. If **on** is specified, dial-on-demand is enabled with a default timer of 60 seconds. If a time is specified, dial-on-demand is enabled with the timer set to the specified time. If **off** is specified, dial-on-demand is disabled. When the dial-on-demand feature is activated, PPP brings up the link when there is traffic to be sent, and takes down the link when there has been no traffic for the specified timer period. The effect on a PPP interface using an on-demand call is to connect the call when traffic is to be sent and disconnect the call when no traffic has been sent or received for the specified timer period. For leased lines, this parameter has no effect, as the links are always connected. The default is **off**.

Note that you can configure the interface so that OSPF keeps on-demand links active regardless of whether the interface is Up or Down. To do this, set the parameter **notifyospfdown=off** in the **add ip interface** or **set ip interface** commands. See "add ip interface" on page 22-80 of Chapter 22, Internet Protocol (IP) for further information.

The **indatalimit** parameter specifies the input data threshold, in megabytes, for the PPP interface. When the interface's cumulative input data counter exceeds this limit, the link is closed and attempts to open it fail. If the input data counter for the interface is cleared with the **reset ppp** command on page 15-82, the link can be reopened. The default is **none**.

The **ippool** parameter specifies the IP pool to use to allocate IP addresses for the remote end of the PPP connection. If **none** is specified, IP addresses are not allocated from an IP pool. The default is **none**. See "IP Address Pools" on page 22-52 of Chapter 22, Internet Protocol (IP) for more information about creating IP address pools.

The **iprequest** parameter specifies whether a request is to be made for an IP address to be allocated by the peer during the IPCP negotiation. If **on** is specified, a request is made. If **off** is specified, a request is not specified. The default is **off**.

The **lqr** parameter sets the LQR timer for link quality management. If **on** is specified, LQR is enabled with a default timer of 60 seconds. If a time is specified, LQR is enabled with the timer set to the specified time. If **off** is specified, LQR is disabled. The LQR function has precedence over the echo function. Both cannot be enabled at the same time. However, both **lqr** and **echo** can be **on** at the same time if **echo** is turned on first. The default is **on**.

The **magic** parameter enables or disables negotiation of the magic number option. The default is **on**. The magic number determines whether a interface is looped back. The interface does not reach an open state when there is a loopback.

The **maxlinks** parameter specifies the maximum number of links allowed in a multilink PPP interface created using this template. The default is **2**.

When using the **maxlinks** parameter without BAP, a link is brought up fully before being checked. If the remote station is configured to bring up more links than the local station accepts, links may be brought up and torn down repeatedly. Therefore, to stop this situation occurring, also implement BAP when **maxlinks** is implemented.

The **modem** parameter specifies the state of synchronous modem control. The default is **off**. If **on** is specified, the router manipulates the DTR signal to trigger the modem to make a call whenever there is traffic (**idle=on**) or when a backup link is required (**type=secondary**). Raising DTR to the modem triggers the modem to initiate a call to the modem at the other end of the link and enter data transfer mode. The router keeps the DTR signal to the modem raised, and responds to the modem raising the DSR signal by activating the PPP interface. This parameter is valid when the **over** parameter specifies a synchronous interface.

The **mru** parameter specifies whether the **mru** (Maximum Receive Unit) option is included, and what value is sent in LCP configuration requests while bringing up this interface. If **off** is specified, the **mru** option is omitted. If a decimal value is specified, the transmitted MRU option is set to this value. If **on** is specified, the MRU is calculated normally. The default is **on**.

The **mssheader** parameter specifies the number of bytes within an MTU that are reserved for packet headers. This amount is subtracted from the MTU of the interface to define the Maximum Segment Size (MSS). The MSS specifies the maximum amount of data that can be contained within each TCP packet. The default is 120 bytes.

The **mssheader** parameter may only be used with an Ethernet or VLAN physical interface (PPPoE).

The **nullfragtimer** parameter specifies the maximum time, in seconds, a link in a multilink bundle may be idle before a NULL fragment is transmitted over the link. NULL fragments are used to keep the last sequence number transmitted over the link up to date. The default is **3**.

The **onlinelimit** parameter specifies the up-time threshold, in hours, for the PPP interface. When the interface's cumulative up-time counter exceeds this limit, the link is closed and any further attempts to open the link fail. If the up-time counter for the interface is cleared with the **reset ppp** command on page 15-82, the link can be reopened. The default is **none**.

The **outdatalimit** parameter specifies the output data threshold, in megabytes, for the PPP interface. When the interface's cumulative output data counter exceeds this limit, any further attempts to open the link fail. If the output data counter for the interface is cleared with the **reset ppp** command on page 15-82, the link can be reopened. The default is **none**.

The **password** parameter specifies the password to use when the peer requests authentication using either CHAP or PAP. This is normally required for network lines between routers, for which an authentication protocol has been selected with the **authentication** parameter.

The **predcheck** parameter specifies the type of CRC to be used for Predictor compression. The Predictor RFC specifies using CRC-16, however some router manufacturers have implemented Predictor with CRC-CCITT, which is the CRC specified in RFC 1662, "PPP in HDLC-link Framing". This value is not negotiated so the same value needs to be configured at both ends of the link for Predictor compression to work correctly. This parameter does not apply to L2TP calls.

The **rechallenge** parameter specifies whether the CHAP rechallenge function is on, off, or the maximum rechallenge period. PPP calculates a random period for the CHAP rechallenge between the maximum rechallenge period and a minimum of 5 minutes. If **on** is specified, the CHAP rechallenges take place with a maximum rechallenge period of 15 minutes. If **off** is specified, then CHAP rechallenges do not take place. If a time is specified, then the time, which is between 360 and 3600 seconds, is used as the upper limit on the rechallenge time calculation. The default is **on**.

The **restart** parameter specifies the time between successive retransmissions of unacknowledged configure requests or terminate requests. The default is 3 seconds.

The **staccheck** parameter specifies the check mode to used for the Stac LZS compression algorithm. If **sequence** is specified an incrementing sequence number is used to determine whether a packet has been lost and therefore whether the compression history needs to be reset. If LCB is specified an LCB value is used to determine if an error has occurred in a packet. The default is **sequence**. This parameter does not apply to L2TP calls.

The **starentity** parameter specifies the star entity and the encryption algorithm to be used by the encryption channel configured by the PPP interface. This parameter must be specified if PPP encryption is enabled.

The **terminate** parameter sets the number of terminate requests sent when trying to close a link before it is assumed the link is down. The default is **2**. The **continuous** option specifies that requests are sent continuously.

The **totaldatalimit** parameter sets the total data throughput threshold, in megabytes, for the PPP interface. When the interface's cumulative total data counter exceeds this limit, the link is closed and any further attempts to open the link fail. If the total data counter for the interface is cleared with the **reset ppp** command on page 15-82, the link can be reopened. The default is **none**.

The **type** parameter specifies the role of the physical interface for bandwidth on demand and leased line backup. The default is **primary**. If **primary** is specified, the link is open all the time (**idle=off**) or opened whenever there is traffic (**idle=on**). If **secondary** is specified, the link opens when the associated primary link fails. If **demand** is specified, the link opens when additional bandwidth is required.

The **uptime** parameter specifies the time, in seconds, that the PPP interface must have a total utilisation (as a percentage) above the threshold specified by the **uprate** parameter, before an additional channel is opened. The default is **30** for **uptime** and **80** for **uprate**. The **uprate, uptime, downrate**, and **downtime** parameters are used with the **type** parameter to configure bandwidth on demand.

The **username** parameter specifies the username to be used when generating PAP authentication requests and when responding to CHAP authentication challenges. If the **username** is not set, the router's system name is used by default.

The **winsprimary** parameter specifies the IP address to pass to a peer when it requests a primary WINS server address with the IPCP primary WINS server option.

The **winssecondary** parameter specifies the IP address to pass to a peer when it requests a primary WINS server address with the IPCP secondary WINS server option.

## Examples

To disable compression on the Ethernet interface 0 link of PPP interface 1, use the command:

```
set ppp=1 over=eth0 comp=off
```

To set a PPPoE interface that has a default MTU of 1492 to use an MSS value of 1292, use the command:

```
set ppp=0 over=eth0-any mssheader=200
```

## Related Commands

**add ppp**
**add ppp acservice**
**create ppp**
**reset ppp**
**show ppp**
**show ppp config**
**show ppp limits**

# set ppp acservice

**Syntax**

```
SET PPP ACservice=service-name [ACRadius={OFF|ON}]
   [MAXSessions=1...512] [TEMPlate=ppp-template]
   [ACINTerface={NONE|interface}]
```

where:

- *service-name* is a character string of up to 18 characters.
- *ppp-template* is a number from 0 to 31.
- *interface* is an interface name formed by concatenating an interface type and an interface instance (e.g. eth0). Valid interface types are ETH and VLAN.

**Description**

This command sets the parameters associated with the specified PPPoE Access Concentrator service.

The **acservice** parameter specifies the name of the PPPoE service whose parameters are to be changed.

The **acradius** parameter specifies whether PPPoE hosts are authenticated by a RADIUS server using the host's MAC address as the radius parameter calling-station ID. If **off** is specified, the RADIUS authentication is not performed. If **on** is specified, the **radius** authentication is performed. The default is **off**.

Note that radius authentication is performed only if it is specified by the template associated with this PPPoE service.

The **maxsessions** parameter specifies the maximum number of PPPoE sessions that are allowed to simultaneously provide the PPPoE service. The default is 1.

The **template** parameter specifies the PPP template to use when creating a dynamic PPP over Ethernet interface. The specified template must exist. See "Templates" on page 15-18 for more information about creating PPP templates.

The **acinterface** parameter specifies the interface on which the service is offered. This parameter further identifies the service whose parameters are to be changed, as it is possible to have two or more services with the same name but offered on different interfaces. It is not possible to change the interface on which the service is offered.

- If an interface is specified, the service with the specified name on that interface has its parameters changed.
- If **none** is specified, the service offered on the Ethernet ports has its parameters changed.
- If the **acinterface** parameter is omitted, the service is mapped to its corresponding interface (if one exists).

If multiple interfaces exist for the service, you are asked to specify an **acinterface**. The default for this parameter is **none**.

**Examples**

To set a limit of 20 simultaneous sessions for the service named armadillo on the interface Ethernet 0, use the command:

```
set ppp ac=armadillo maxs=20 acint=eth0
```

**Related Commands**

add ppp acservice
create ppp
create ppp template
delete ppp acservice
destroy ppp
destroy ppp template

# set ppp template

**Syntax**

```
SET PPP TEMPLATE=template [AUTHENTICATION={CHAP|EITHER|
   PAP|NONE}] [BAP={ON|OFF}] [BAPMODE={CALL|CALLBACK}]
   [CBDELAY=1..100] [CBMODE={ACCEPT|OFF|REQUEST}]
   [CBNUMBER=e164number] [CBOPERATION={E164NUMBER|
   USERAUTH}] [COMPALGORITHM={PREDICTOR|STACLZS}]
   [COMPRESSION={ON|OFF|LINK}] [DEBUGMAXBYTES=16..256]
   [DESCRIPTION=description] [ECHO={ON|OFF|period}]
   [ENCRYPTION={ON|OFF}] [FRAGMENT={ON|OFF}]
   [FRAGOVERHEAD=0..100] [IDLE={ON|OFF|time}]
   [INDATALIMIT={NONE|1..65535}] [IPPOOL={pool-name|NONE}]
   [IPREQUEST={ON|OFF}] [LOGIN={ALL|RADIUS|TACACS|USER}]
   [LQR={ON|OFF|period}] [MAGIC={ON|OFF}] [MAXLINKS=1..64]
   [MRU={ON|OFF|256..1656}] [MSSheader=40..200]
   [MTU=256..1500|256..1492] [MULTILINK={ON|OFF}]
   [NULLFRAGTIMER=time] [ONLINELIMIT={NONE|1..65535}]
   [OUTDATALIMIT={NONE|1..65535}] [PASSWORD=password]
   [PREDCHECK={CRC16|CRCCCITT}] [RECHALLENGE={ON|OFF|
   360..3600}] [RESTART=time] [STACCHECK={LCB|SEQUENCE}]
   [STARENTITY=1..255] [TOTALDATALIMIT={NONE|1..65535}]
   [USERNAME=username] [VJC={ON|OFF}]
```

where:

- *template* is a number from 0 to 31.
- *e164number* is the phone number to dial when performing callback. It may contain digits (0–9) and should be a valid phone number as described in CCITT standard E.164.
- *description* is a character string 1 to 70 characters long. Valid characters are any printable character.
- *period* is a decimal number from 1 to 4294967295.
- *time* is a timer value in seconds.
- *pool-name* is a character string 1 to 15 characters long. Valid characters are any printable characters. If *pool-name* contains spaces, it must be in double quotes.
- *password* is the password to use for authentication 1 to 64 characters long. It may contain any printable character and is case sensitive.
- *username* is the username to use for authentication 1 to 64 characters long. It may contain any printable character and is case sensitive.

**Description**

This command modifies an existing PPP template that is used to configure dynamic PPP interfaces. It requires a user with Security Officer privilege when the router is in security mode.

The PPP interface is created when one of the following is activated:

- ISDN call
- ACC call
- L2TP call
- PPP over Ethernet service

The **template** parameter specifies the number of the template to modify. The specified template must already exist.

The **authentication** parameter specifies the authentication protocol to be used on the physical interface or channel. If **chap** is specified, the Challenge-Handshake Authentication Protocol (CHAP) is used. If **pap** is specified, the Password Authentication Protocol (PAP) is used. If **either** is specified, the router uses the option negotiation process to negotiate the authentication protocol to be used with the device at the remote end of the link, specifying CHAP as the first choice. If **none** is specified, no authentication protocol is used. The default is **none**.

The **bap** parameter specifies whether the Bandwidth Allocation Protocol is used for negotiating the activation of demand PPP links. The default is **on**.

The **bapmode** parameter specifies which peer originates another link to add to the multilink bundle. For **callback** mode, the number to call must be configured on the call at the lower layer (ISDN, ACC, L2TP). The default is **call**.

The **cbdelay** parameter specifies the delay in tenths of a second between bringing down a call for callback and actually making the call back to the peer. This parameter handles different timing requirements, for example of various ISDN switches, and is valid when the callback mode is **request**. The default is **1**.

The **cbmode** parameter specifies whether a callback request is made or accepted during the LCP negotiation. If **request** is specified, a request is made for callback. If **accept** is specified, requests for callback received from the peer are accepted and processed; however, **authentication** must be set to PAP or CHAP. If **off** is specified, no callback requests are made and callback requests are not accepted. The default is **off**.

The **cbnumber** parameter specifies the number to include when requesting a callback with the **cboperation** parameter set to **e164number**. The number specified should be a phone number as specified in the E.164 standard.

The **cboperation** parameter specifies the callback operation to be included in the callback request to specify to the peer how to determine the callback number. If **userauth** is specified, the peer uses the username and password supplied during authentication to look up the callback number. If **e164number** is specified, the callback number specified by the **cbnumber** parameter is included in the callback request. The default is **userauth**.

The **compalgorithm** parameter specifies the compression algorithm to use when compressing and decompressing PPP packets. If **predictor** is specified, the Predictor algorithm is used with type 1 encapsulation as specified in RFC 1978. If **staclzs** is specified, the Stac LZS algorithm is used as specified in RFC 1974. The default is **staclzs**. This parameter does not apply to L2TP calls.

The **compression** parameter enables or disables the use of compression for the interface. When used with multilink, setting **compression** to **on** compresses packets before they are sent to the individual links. Setting **compression** to **link** enables compression for the link specified by the **over** parameter. The default is **off**. The **link** option should be used when compression is required on some physical interfaces and not on others. For example, if a PPP multilink uses a compressing modem link and a normal dedicated leased line, **compression** should be set to **off** on the physical interface to which the modem is connected, and LINK for the physical interface to which the dedicated leased line is connected. If compression is required on all physical interfaces of a PPP interface, the **compression** parameter should be set to **on**. This parameter does not apply to L2TP calls.

The **debugmaxbytes** parameter specifies the maximum number of bytes that are displayed for each packet when the **ctrlpkt, datapkt, packet**, and **pkt** debug options are enabled. The default is **60**.

The **description** parameter specifies a user-defined description for the interface, to make it easier to distinguish between a number of PPP interfaces.

The **echo** parameter specifies whether LCP Echo Request and Echo Reply messages are used to determine link quality. If three consecutive Echo Request messages are transmitted without receiving an Echo Reply response, the link is deemed to be down. If **off** is specified, Echo Request messages are not transmitted. If **on** is specified, these messages are transmitted every 10 seconds. If a period in seconds is specified, Echo Request messages are transmitted at the specified interval. The LQR function has precedence over the echo function. Both cannot be enabled at the same time. However, both **lqr** and **echo** can be **on** at the same time if **echo** is turned on first.

The **encryption** parameter enables or disables the use of encryption for the interface. The default is **off**.

The **fragment** parameter applies to a multilink bundle interface and determines whether packets are fragmented. The default is **off**. Fragmentation cannot be enabled if compression or encryption are configured.

The **fragoverhead** parameter specifies the maximum allowable overhead, as a percentage, for fragmenting packets using the variable fragmentation scheme for multilink PPP. When a packet exceeds this limit, it is fragmented using the fixed fragmentation scheme. The default is **5**. The variable fragmentation scheme spreads the packet over all the links in the multilink bundle by splitting the packet into variable sized fragments to match the speed of individual links. Larger fragments are transmitted over faster links, thereby providing an inherent load balancing scheme. The fixed fragmentation scheme spreads the packet over all the links in the multilink bundle by splitting the packet into equal fixed sized fragments. If the number of links is large and the packet is relatively small a fragment is not transmitted over every link.

The **idle** parameter controls the dial-on-demand feature. If **on** is specified, dial-on-demand is enabled with a default timer of 60 seconds. If a time is specified, dial-on-demand is enabled with the timer set to the specified time. If **off** is specified, dial-on-demand is disabled. When the dial-on-demand feature is activated, PPP brings up the link when there is traffic to be sent, and takes down the link when there has been no traffic for the specified timer period. The effect on a PPP interface using an on-demand call is to connect the call when traffic is to be sent and disconnect the call when no traffic has been sent or received for the specified timer period. For leased lines, this parameter has no effect, as the links are always connected. The default is **off**.

Note that you can configure the interface so that OSPF keeps on-demand links active regardless of whether the interface is Up or Down. To do this, set the parameter **notifyospfdown=off** in the **add ip interface** or **set ip interface** commands. See "add ip interface" on page 22-80 of Chapter 22, Internet Protocol (IP) for further information.

The **indatalimit** parameter specifies the input data threshold, in megabytes, for the PPP interface. When the interface's cumulative input data counter exceeds this limit, the link is closed and any further attempts to open the link fail. If the input data counter for the interface is cleared with the **reset ppp** command on page 15-82, the link can be reopened. The default is **none**.

The **ippool** parameter specifies the IP pool to use to allocate IP addresses for dynamic dial-in PPP connections. If **none** is specified, IP addresses are not allocated from an IP pool. The default is **none**. See "IP Address Pools" on page 22-52 of Chapter 22, Internet Protocol (IP) for more information about creating IP address pools.

The **iprequest** parameter specifies whether to make a request for an IP address to be allocated by the peer during the IPCP negotiation. If **on** is specified, a request is made. If **off** is specified, a request is not made. The default is **off**.

The **login** parameter specifies which login procedure the call creating this dynamic interface must use when it is activated. If **radius** is specified, the router sends requests to the configured RADIUS server(s) to authenticate the call. If **tacacs** is specified, the router sends requests to the configured TACACS server(s) to authenticate the call. If **user** is specified, the router checks the User Authentication Database to authenticate the call. If **all** is specified, the router tries all methods to authenticate the call.

The **lqr** parameter sets the LQR timer for link quality management. If **on** is specified, LQR is enabled with a default timer of 60 seconds. If a time is specified, LQR is enabled with the timer set to the specified time. If **off** is specified, LQR is disabled. The LQR function has precedence over the echo function. Both cannot be enabled at the same time. However, both **lqr** and **echo** can be **on** at the same time if **echo** is turned on first. The default is **on**.

The **magic** parameter enables or disables negotiation of the magic number option. The default is **on**. The magic number determines if a interface is looped back. The interface does not reach an open state when there is a loopback.

The **maxlinks** parameter specifies the maximum number of links allowed in a multilink PPP interface created using this template. The default is **2**.

When using the **maxlinks** parameter without BAP, a link is brought up fully before being checked. If the remote station is configured to bring up more links than the local station accepts, links may be brought up and torn down repeatedly. Therefore, to stop this situation occurring, also implement BAP when **maxlinks** is implemented.

The **mru** parameter specifies whether the MRU (Maximum Receive Unit) option is included, and what value is sent in LCP configuration requests while bringing up this interface. If **off** is specified, the MRU option is omitted. If a decimal value is specified, the transmitted MRU option is set to this value. If **on** is specified, the MRU is calculated normally. The default is **on**.

The **mssheader** parameter specifies the number of bytes within an MTU that are reserved for packet headers. This amount is subtracted from the MTU of the interface to define its Maximum Segment Size (MSS). The MSS specifies the maximum amount of data that can be contained within each TCP packet. The default is 120 bytes.

The **mssheader** parameter may only be used with an Ethernet or VLAN physical interface (PPPoE).

The **mtu** parameter specifies the value of the maximum transmission unit. The allowable MTU values for the PPP lower layers are shown in Table 15-12.

Table 15-12: Allowable MTU values

| Interface | Minimum MTU | Maximum MTU | Default MTU |
|---|---|---|---|
| PPP (except over Ethernet) | 256 | 1500 | 1500 |
| PPP over Ethernet | 256 | 1492 | 1492 |

We recommend that you leave the default MTU values unchanged for normal operations.

The **multilink** parameter specifies whether incoming dynamic PPP calls using this template are multilinked together. If **on** is specified, incoming dynamic PPP calls from the same peer and with the same authentication information are multilinked together. If **off** is specified, incoming dynamic PPP calls are not multilinked under any circumstances. The default is **on**.

The **nullfragtimer** parameter specifies the maximum time, in seconds, a link in a multilink bundle may be idle before a NULL fragment is transmitted over the link. NULL fragments are used to keep the last sequence number transmitted over the link up to date. The default is **3**.

The **onlinelimit** parameter specifies the up-time threshold, in hours, for the PPP interface. When the interface's cumulative up-time counter exceeds this limit, the link is closed and any further attempts to open the link fail. If the up-time counter for the interface is cleared using the **reset ppp** command on page 15-82, the link can be reopened. The default is **none**.

The **outdatalimit** parameter specifies the output data threshold, in megabytes, for the PPP interface. When the interface's cumulative output data counter exceeds this limit, any further attempts to open the link fail. If the output data counter for the interface is cleared using the **reset ppp** command on page 15-82, the link can be reopened. The default is **none**.

The **password** parameter specifies the password to use when the peer requests authentication using either CHAP or PAP. This is normally required for network lines between routers, for which an authentication protocol has been selected with the **authentication** parameter.

The **predcheck** parameter specifies the type of CRC to be used for Predictor compression. The Predictor RFC specifies using CRC-16, however some router manufacturers have implemented Predictor with CRC-CCITT, which is the CRC specified in RFC 1662, "PPP in HDLC-link Framing". This value is not negotiated so the same value needs to be configured at both ends of the link for Predictor compression to work correctly. This parameter does not apply to L2TP calls.

The **rechallenge** parameter specifies if the CHAP rechallenge function is on, off or the maximum rechallenge period. PPP calculates a random period for the CHAP rechallenge between the maximum rechallenge period and a minimum of 5 minutes. If **on** is specified, the CHAP rechallenges take place with a maximum rechallenge period of 15 minutes. If **off** is specified, then CHAP rechallenges do not take place. If a time is specified, then the time, which is between 360 and 3600 seconds, is used as the upper limit on the rechallenge time calculation. The default is **on**.

The **restart** parameter specifies the time between successive retransmissions of unacknowledged configure requests or terminate requests. The default is 3 seconds.

The **staccheck** parameter specifies the check mode to used for the Stac LZS compression algorithm. If **sequence** is specified an incrementing sequence number is used to determine whether a packet has been lost and therefore whether the compression history needs to be reset. If **lcb** is specified an LCB value is used to determine if an error has occurred in a packet. The default is **sequence**. This parameter does not apply to L2TP calls.

The **starentity** parameter specifies the star entity and the encryption algorithm to be used by the encryption channel configured by the PPP interface. This parameter must be specified if PPP encryption is enabled.

The **totaldatalimit** parameter sets the total data throughput threshold, in megabytes, for the PPP interface. When the interface's cumulative total data counter exceeds this limit, the link is closed and any further attempts to open the link fail. If the total data counter for the interface is cleared using the **reset ppp** command on page 15-82, the link can be reopened. The default is **none**.

The **username** parameter specifies the username to be used when generating PAP authentication requests and when responding to CHAP authentication challenges. If the **username** is not set, the router's system name is used by default.

The **vjc** parameter specifies whether Van Jacobson Header Compression will be negotiated on dynamic PPP interfaces created from the template. The default is **off**.

## Examples

To modify template 1 to use LCP Echo for link quality management, use the command:

```
set ppp template=1 echo=on
```

To modify template 0 to the MTU value 1400, use the command:

```
set ppp template=0 mtu=1400
```

To set a PPPoE template that has an MSSH value of 200, use the command:

```
set ppp temp=1 mssheader=200
```

## Related Commands

**create ppp template**
**destroy ppp template**
**disable ppp template debug**
**enable ppp template debug**
**reset ppp**
**show ppp pppoe**
**show ppp limits**
**show ppp template**

# show ppp

**Syntax** `SHOW PPP[=`*`ppp-interface`*`]`

where *ppp-interface* is the PPP interface number from 0 to 1023

**Description** This command displays a list of each PPP interface, users of the interface, physical interfaces that the interface is running over, and the current state of the interface (Figure 15-15, Table 15-13).

This command requires a user with security officer privilege when the router is in security mode.

Figure 15-15: Example output from the **show ppp** command for a PPP link

```
Name             Enabled  ifIndex  Over                    CP           State
-------------------------------------------------------------------------------
ppp0               YES    4                                IPCP         OPENED
                                   syn0                    LCP          OPENED
                                   isdn-demand             LCP          OPENED
-------------------------------------------------------------------------------
```

Table 15-13: Parameters in output of the **show ppp** command

| Parameter | Meaning |
|---|---|
| Name | The name of the PPP interface. If the interface is a dynamic PPP interface an asterisk ("*") is displayed in front of the interface name. |
| Enabled | YES if the PPP interface is enabled; NO if it is disabled. |
| Ifindex | The value of ifIndex for the PPP interface. |
| Over | The lower layer(s) used by the PPP interface; SYN*n*, ISDN-*callname*, ACC-*callname*, MIOX*n-circuitname*, TDM-*groupname,* ETH*n-servicename*, VLAN*n-servicename,* TNL-*callname*. |
| CP | A list of the network and link control protocols running over the PPP interface; one or more of "IPCP", "IPV6CP", "IPXCP", "BCP", "DNCP", "ATCP", "LCP", "CCP", "ILCCP", "ECP", "BACP" or "MULTI". |
| State | The state of the PPP links; one of "INITIAL", "STARTING", "CLOSED", "STOPPED", "CLOSING", "STOPPING", "REQ SENT", "ACK RCVD", "ACK SENT" or "OPENED". |

**Examples** To display information about PPP interface 2, use the command:

```
show ppp=2
```

**Related Commands** **show ppp config**
**show ppp count**

# show ppp config

**Syntax** `SHOW PPP[=`*`ppp-interface`*`] CONFIG`

where *ppp-interface* is the PPP interface number from 0 to 1023

**Description** This command displays the configuration of a PPP interface (Figure 15-16, Table 15-14 on page 15-100).

Figure 15-16: Example output from the **show ppp config** command

```
Interface - description
  Parameter                          Configured                 Negotiated
-------------------------------------------------------------------------------
ppp0 - Link to Southern Regional Office
                                                          Local            Peer
  Bandwidth Allocation Protocol             ON
  Bandwidth Allocation Call Mode          CALL
  Multilink Fragmentation                  OFF
  Acceptable Fragment Overhead (%)           5
  Null Fragment Timer (seconds)              3
  Session Timer (seconds)                  OFF
  Idle Timer (seconds)                     OFF
  Maximum Receive Unit (bytes)            0FF             NONE            NONE
  Compression                              ON               ON              ON
  Encryption                              OFF              OFF             OFF
  Username                            NOT SET
  Password                            NOT SET
  Bundle Endpoint Discr Class               0
  Bundle Endpoint Discr Value              []
  Bundle Username                     NOT SET
  Bundle Maximum Links                       2
  acc-btb
  Type                                primary
  Restart Timer (seconds)                    3
  Max-Configure                     continuous
  Max-Terminate                              2
  Echo Request Timer (seconds)             OFF
  Callback Mode                            OFF
  Link Compression                          ON               ON              ON
  LQR Timer (seconds)                       60              OFF             OFF
  Magic Number                              ON              OFF             OFF
  Link Discriminator                      0000              OFF             OFF
  Link Endpoint Discr Class                  0
  Link Endpoint Discr Value
  Authentication                           CHAP             NONE            NONE
  Authentication Mode                     INOUT
  CHAP Rechallenge (max. period seconds) 900
  Utilisation (%)                            0
Compression
  Algorithm                           STACLZS          STACLZS         STACLZS
  Stac LZS Checkmode                 SEQUENCE         SEQUENCE        SEQUENCE
PPPoE
  Session ID                                                 4B5C            4B5C
  MAC Address of Peer                                              00-00-cd-00-5e-65
  Service Name                        aardvark
```

Figure 15-16: Example output from the **show ppp config** command (cont.)

```
IP
  IP Compression Protocol               NONE            NONE            NONE
  IP Pool                               Test
  IP Address Request                     OFF
  IP Address                    192.168.1.1     192.168.1.1     192.168.1.2
  Primary DNS Address           192.168.2.3            NONE            NONE
  Secondary DNS Address         192.168.5.1            NONE            NONE
  Primary WinS Address          192.168.5.5                            NONE
  Secondary WinS Address            NOT SET                            NONE
 Debug
  Maximum packet bytes to display         22
------------------------------------------------------------------------------
```

Table 15-14: Parameters in output of the **show ppp config** command

| Parameter | Meaning |
|---|---|
| Configured | This column specifies the value that has been configured for a parameter. The value may be modified by the negotiation process between the local and remote ends of the PPP link. |
| Negotiated/Local | For a link in the open state, this column contains the value that the local end of the link uses for a parameter as a result of the negotiation process. For a link that is not open, this column contains the initial value for a parameter. |
| Negotiated/Peer | For a link in the open state, this column contains the value that the remote end of the link uses for a parameter as a result of the negotiation process. For a link that is not open, this column contains the initial value for a parameter. |
| ppp<*n*> - <*description*> | The name and description of the interface. Following fields display information about the interface as a whole. |
| Bandwidth Allocation Protocol | Whether the Bandwidth Allocation Protocol is enabled on the interface. |
| Bandwidth Allocation Call Mode | The call mode for the Bandwidth Allocation Protocol, if the Bandwidth Allocation Protocol is enabled on the interface; either "CALL" or "CALLBACK". |
| Multilink fragmentation | Whether multilink packets may be fragmented. |
| Acceptable Fragment Overhead (%) | The maximum amount of overhead allowed to be added to each packet due to variable fragmentation. If this level is exceeded when fragmentation of a packet is done using the variable fragmentation scheme, then the fixed fragmentation scheme is used instead. |
| Null Fragment Timer (seconds) | Seconds that the link must be idle for before a Null fragment is sent on a link in a multilink bundle. |
| Session Timer (seconds) | Seconds before a link is disconnected, or whether the session timer is disabled. |
| Idle Timer (seconds) | Seconds that a link must be idle before it is disconnected, or whether the idle timer is disabled. |
| Maximum Receive Unit (bytes) | The maximum allowable length for packets received at the PPP layer. The MRU of the peer is used as the MTU of the upper layers so that they do not transmit anything that is too long for the peer to handle. |
| Compression | Whether compression is enabled for the entire PPP interface. |

Table 15-14: Parameters in output of the **show ppp config** command (cont.)

| Parameter | Meaning |
|---|---|
| Encryption | Whether encryption is enabled for the entire PPP interface. |
| Username | The username used by the PPP interface for both PAP and CHAP authentication or whether a username has not been set. |
| Password | Whether a password has been set for the entire PPP interface. |
| Up Rate (%utilisation) | The utilisation level on the link where an additional channel is opened when the interface has on-demand links. |
| Up Time (seconds) | Seconds that the utilisation level on the link must exceed *Up Rate* before an additional channel is opened when the interface has on-demand links. |
| Down Rate (%utilisation) | The utilisation level on the link below which a channel is closed when the interface has on-demand links. |
| Down Time (seconds) | Seconds that the utilisation level on the link must be below *Down Rate* before a channel is closed when the interface has on-demand links. |
| Bundle Endpoint Discr Class | The class of endpoint discriminator used to uniquely identify this link's endpoint. |
| Bundle Endpoint Discr Value | The hexadecimal value of the endpoint discriminator used to uniquely identify this link's endpoint. |
| Bundle Username | The username assigned to the multilink bundle or whether a username has not been set. |
| Bundle Maximum Links | The maximum number of links allowed in a multilink bundle. If the Maximum Links value is `0xFFFFFFFF`, this value is displayed as 'N/A' rather than as a huge number. If the Maximum Receive Unit is set to OFF, OFF is displayed. |
| **LCP Information** | **This section is repeated once for each LCP (physical interface) operating over the PPP interface.** |
| *<lcp-name>* | The name of an LCP operating over this PPP interface. Following fields display information about this LCP (link). |
| Number of primary channels | The number of channels with a **type** of **primary** carried over the ISDN call, if this physical interface is an ISDN call. |
| Number of secondary channels | The number of channels with a **type** of **secondary** carried over the ISDN call, if this physical interface is an ISDN call. |
| Number of demand channels | The number of channels with a **type** of **demand** carried over the ISDN call, if this physical interface is an ISDN call. |
| Type | The role of this physical interface for bandwidth on demand and leased line backup; either "demand", "primary" or "secondary". |
| Modem Control | Whether modem control is enabled (valid on synchronous interfaces). |
| Restart Timer | Seconds between configure requests for this physical interface. |
| Max-Configure | The maximum number of configure requests sent before PPP gives up trying to open this link, or "continuous". |
| Max-Terminate | The maximum number of Terminate requests sent before PPP gives up trying to open this link and declares this link down, or "continuous". |

Table 15-14: Parameters in output of the **show ppp config** command (cont.)

| Parameter | Meaning |
|---|---|
| Echo Request Timer (seconds) | The time, in seconds, between transmissions of LCP *Echo Request* messages when LCP *Echo Request/Echo Reply* messages are used to monitor link state, or "OFF" if LQR is used to determine link status. |
| Callback Mode | Whether this link requests callback, accepts callback, or does neither. |
| Callback Operation | The callback operation to include in the callback request when the callback mode is **request**; either "USERAUTH" or "E164NUMBER". This field is displayed if *Callback Mode* is set to "REQUEST". |
| Callback Number | The callback number included in callback requests when the callback mode is **request** and the callback operation is E164NUMBER. This field is displayed if *Callback Mode* is set to "REQUEST" and *Callback Operation* is set to "E164NUMBER". |
| Callback Delay (tenths of a second) | The delay, in tenths of a second, between deactivating a call for callback and making the return call. This field is displayed if *Callback Mode* is set to "ACCEPT". |
| Link Compression | Whether compression is enabled for this link rather than the entire PPP interface. |
| LQR Timer (seconds) | Seconds between LQR packets transmitted over this physical interface. |
| Magic Number | Whether the magic number option is enabled for this physical interface. |
| Link Discriminator | The link discriminator value for this physical interface, or "OFF" if the link discriminator LCP option is not enabled. |
| Link Endpoint Discr Class | The class of link endpoint discriminator assigned to this end of the physical interface. |
| Link Endpoint Discr Value | The value the of link endpoint discriminator assigned to this end of the physical interface, expressed in hexadecimal. |
| Authentication | The authentication protocol in use on this physical interface; either NONE, PAP, CHAP, or EITHER. |
| Authentication Mode | Whether authentication is requested on incoming ISDN calls, outgoing ISDN calls, or both; either IN, OUT, or INOUT. |
| CHAP Rechallenge | Seconds for the maximum interval between CHAP rechallenges, or "OFF" if the CHAP rechallenge is disabled. This parameter is displayed when CHAP is configured. |
| Utilisation (%) | The bandwidth utilisation, as a percentage of time the interface is transmitting data, for this physical interface. |
| Link Compression | Information about link compression on this physical interface if link compression is enabled on this physical interface. |
| Algorithm | The compression algorithm used to compress packets on this physical interface; either "PREDICTOR" or "STAC_LZS". |
| Stac LZS Checkmode | The check mode used by the Stac LZS compression algorithm to determine if a decompression history is unsynchronised on this physical interface; either "NONE", "LCB", "CRC", "SEQUENCE" or "EXTENDED". |

Table 15-14: Parameters in output of the **show ppp config** command (cont.)

| Parameter | Meaning |
|---|---|
| Predictor LZS Checkmode | The check mode used by the Predictor compression algorithm to determine if a decompression history is unsynchronised on this physical interface; either "CRC-16" or "CRC-CCITT". |
| Channel Information | This section is displayed when the LCP (physical interface) is an ISDN interface, and is repeated once for each channel in the physical interface. Basic Rate ISDN interfaces have 2 channels. Primary Rate ISDN interfaces have 30 channels. |
| bri<*n*> - channel <*n*><br>pri<*n*> - channel <*n*> | The interface and channel number of physical interfaces that are ISDN calls. Following fields display information specific to this channel. |
| Type | The role of this channel for bandwidth on demand and leased line backup; either "demand", "primary" or "secondary". |
| Utilisation (%) | The bandwidth utilisation, as a percentage of time the interface is transmitting data, for the physical interface. |
| Link Compression | Whether compression is enabled for the link rather than the entire PPP interface. |
| LQR Timer (seconds) | Seconds between LQR packets transmitted over the physical interface. |
| Magic Number | Whether the magic number option is enabled for the physical interface. |
| Link Discriminator | The link discriminator value for the physical interface, or whether the link discriminator LCP option is not enabled. |
| Link Endpoint Discr Class | The class of link endpoint discriminator assigned to this end of the physical interface. |
| Link Endpoint Discr Value | The value the of link endpoint discriminator assigned to this end of the physical interface, expressed in hexadecimal. |
| Authentication | The authentication protocol in use on the physical interface; either NONE, PAP, CHAP, or EITHER. |
| **NCP Information** | **This section is repeated once for each NCP configured on the PPP interface.** |
| Encryption | Information about encryption on the PPP interface, if encryption is enabled on the interface. |
| Star Entity Identifier | The star entity and encryption algorithm used on the PPP interface. |
| Link Compression | Information about link compression on the PPP interface if link compression is enabled on the PPP interface. |
| Algorithm | The compression algorithm to use for compressing packets on the PPP interface; either "PREDICTOR" or "STAC_LZS". |
| Stac LZS Checkmode | The check mode used by the Stac LZS compression algorithm to determine if a decompression history is unsynchronised on the PPP interface; either "NONE", "LCB", "CRC", "SEQUENCE" or "EXTENDED". |
| Predictor LZS Checkmode | The check mode used by the Predictor compression algorithm to determine if a decompression history is unsynchronised on the PPP interface; either "CRC-16" or "CRC-CCITT". |

Table 15-14: Parameters in output of the **show ppp config** command (cont.)

| Parameter | Meaning |
|---|---|
| IP | Information about the IP NCP on the PPP interface, if IP is enabled on this PPP interface. |
| IP Compression Protocol | The IP compression protocol enabled on the PPP interface; either "VJC" or "NONE". |
| IP Pool | The name of the IP address pool used to assign IP addresses for this PPP interface, or "NOT SET" if an IP address pool has not been assigned. |
| IP Address Request | Whether an IP address is requested from the peer during IPCP negotiation; either "ON" or "OFF". |
| IP Address | The IP address configured at each end of the link, "0.0.0.0" if the PPP interface is an unnumbered interface, or "NONE" if an IP address has not been assigned. |
| Primary DNS Address | The IP address of the primary DNS server, passed to a peer in response to an IPCP primary DNS request. |
| Secondary DNS Address | The IP address of the secondary DNS server, passed to a peer in response to an IPCP secondary DNS request. |
| Primary WinS Address | The IP address of the primary WINS server, passed to a peer in response to an IPCP primary WINS server request. |
| Secondary WinS Address | The IP address of the secondary WINS server, passed to a peer in response to an IPCP secondary WINS server request. |
| Debug | Information about debugging on the PPP interface. |
| Maximum packet bytes to display | The maximum number of bytes of each PPP packet displayed by the PACKET debugging option. |
| **PPPoE Information** | **This section is displayed when the LCP (physical interface) is an ethernet interface.** |
| Session ID | The hexadecimal value of the session ID number for the current PPP over Ethernet session. The number is allocated by the Access Concentrator, which is the local router when the router acts as an access concentrator or the remote peer when the router acts as a client. |
| MAC Address of Peer | The MAC address of the peer where the router is currently connected via the PPP over Ethernet session. |
| Service Name | The name of the Ethernet service that the PPP interface is using, of "ANY" if the special service name ANY was specified when configuring the PPP interface. |

## Examples

To display the configuration for PPP interface 2, use the command:

```
show ppp=2 config
```

## Related Commands

**show ppp**
**show ppp count**
**show ppp utilisation**

# show ppp count

**Syntax** `SHOW PPP[=`*`ppp-interface`*`] COUNT[={INTERFACE|LCP|MULTILINK|NCP|PPPOE}]`

where *ppp-interface* is a number from 0 to 1023

**Description** This command displays counters for the interface.

If **count=interface** is specified, counters from the Interfaces MIB are displayed (Figure 15-17, Table 15-15 on page 15-106).

If **count=lcp** is specified, counters for LCP, LQR, CCP, ECP and authentication protocols are displayed (Figure 15-18 on page 15-107, Table 15-16 on page 15-108).

If **count=multilink** is specified, counters for the multilink protocol are displayed (Figure 15-19 on page 15-113, Table 15-17 on page 15-113).

If **count=ncp** is specified, counters for the NCPs are displayed (Figure 15-20 on page 15-114, Table 15-18 on page 15-114).

If **count=pppoe** is specified, counters for PPPoE active discovery packets that have been sent and received are displayed (Figure 15-21 on page 15-117, Table 15-20 on page 15-117).

If a category is not specified, all counters are displayed, including those for BAP and BACP (Table 15-19 on page 15-115).

Figure 15-17: Example output from the **show ppp count=interface** command

```
ppp0                  1519 seconds     Last change at:        974 seconds
Interface Counters
  ifInOctets                116554     ifOutOctets                 91792
  ifInUcastPkts                  0     ifOutUcastPkts                  0
  ifInNUcastPkts              2098     ifOutNUcastPkts              1538
  ifInDiscards                   0     ifOutDiscards                   0
  ifInErrors                     3     ifOutErrors                     0
  ifInUnknownProtos              0     ifOutQLen                       0
```

Table 15-15: Parameters in output of the **show ppp count=interface** command

| Parameter | Meaning |
|---|---|
| ppp0 | The interface name. |
| seconds | Seconds since the interface was last re-initialised. |
| Last change at | Seconds since the interface entered its current operational state. |
| ifInOctets | The total number of octets received over the interface, including two octets per frame for PPP address and control information, two octets per frame for the FCS, one octet per frame for a flag and two octets per frame for the PPP header (six for multilink), and the number of octets in the user data packets and PPP control packets. |
| ifInUcastPkts | The total number of subnetwork-unicast packets delivered to a higher-layer protocol. |
| ifInNUcastPkts | The total number of non-unicast packets delivered to a higher-layer protocol. |
| ifInDiscards | The total number of inbound packets chosen to be discarded to prevent their being delivered to a higher-layer protocol even though no errors had been detected. One reason for discarding such packets would be to free up buffer space. |
| ifInErrors | The total number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. |
| ifInUnknownProtos | The total number of discarded packets received via the interface because of an unknown or unsupported protocol. |
| ifOutOctets | The total number of octets transmitted over the interface, including two octets per frame for PPP address and control information, two octets per frame for the FCS, one octet per frame for a flag and two octets per frame for the PPP header (six for multilink), and the number of octets in the user data packets and PPP control packets. |
| ifOutUcastPkts | The total number of packets that higher-layer protocols requested be transmitted to a subnetwork-unicast address, including those that were discarded or not sent. |
| ifOutNUcastPkts | The total number of packets that higher-layer protocols requested be transmitted to a non-unicast address, including those that were discarded or not sent. |
| ifOutDiscards | The total number of outbound packets chosen to be discarded to prevent their being transmitted even though no errors had been detected. One possible reason for discarding such a packet could be to free up buffer space. |
| ifOutErrors | The total number of outbound packets with errors preventing them from being transmitted. |
| ifOutQLen | The length of the output packet queue. |

Figure 15-18: Example output from the **show ppp count=lcp** command

```
CCP
  inOctets                    52456     outOctets                   38959
  inUserPkts                   2101     outUserPkts                  1538
  inConfigureRequest              3     outConfigureRequest             3
  inConfigureAcknowledge          3     outConfigureAcknowledge         3
  inConfigureNAK                  0     outConfigureNAK                 0
  inConfigureReject               0     outConfigureReject              0
  inTerminateRequest              0     outTerminateRequest             0
  inTerminateAcknowledge          0     outTerminateAcknowledge         0
  inCodeReject                    0     outCodeReject                   0
  decodeSsuccesses             2098     encodeSuccesses              1538
  decodeFailures                  3     encodeFailures                  0
  decodeDiscards                  0     encodeDiscards                  0
  inResetRequests                 8     outResetRequests                3
  inResetAcks                     3     outResetAcks                    2
  encoEventsWithLcpDown           0

LQM    OVER: syn1
  lqrFailures                     0     loopbacksDetected               0

  inLQRs                         16     outLQRs                        16
  inPktLost                       0     outPktLost                      0
  inOctetLost                     0     outOctetLost                    0
                                        outLQRsLost                     0
                                        outLQRsTransit                  0

PAP    OVER: syn1
  inRequest                       1     outRequest                      0
  inAck                           0     outAck                          1
  inNak                           0     outNak                          0

LCP    OVER: syn1
  inOctets                    25316     outOctets                   19158
  inUserPkts                    929     outUserPkts                   749
  inConfigureRequest              3     outConfigureRequest             7
  inConfigureAcknowledge          3     outConfigureAcknowledge         3
  inConfigureNAK                  0     outConfigureNAK                 0
  inConfigureReject               3     outConfigureReject              0
  inTerminateRequest              0     outTerminateRequest             1
  inTerminateAcknowledge          1     outTerminateAcknowledge         0
  inCodeReject                    0     outCodeReject                   0
  inProtocolReject                3     outProtocolReject               0
  inEchoRequest                   0     outEchoRequest                  0
  inEchoReply                     0     outEchoReply                    0
  inDiscardRequest                0     outDiscardRequest               0
  echoFailures                    0     badEchoReplies                  0
```

Table 15-16: Parameters in output of the **show ppp count=lcp** command

| Parameter | Meaning |
|---|---|
| ECP | Information about the encryption control protocol (ECP). |
| inOctets | The number of octets received by the encryption protocol. This includes two octets per frame for the PPP encryption header, the number of octets of encrypted data received, and the number of octets in control protocol packets (ECP). For multilinks an extra six octets per frame are included for the multilink header. |
| inUserPkts | The number of packets received for the encryption control protocol. |
| inConfigureRequest | The number of *Configure-Request* packets received for the encryption control protocol. |
| inConfigureAcknowledge | The number of *Configure-Acknowledge* packets received for the encryption control protocol. |
| inConfigureNAK | The number of *Configure-NAK* packets received for the encryption control protocol. |
| inConfigureReject | The number of *Configure-Reject* packets received for encryption control protocol. |
| inTerminateRequest | The number of *Terminate-Request* packets received for encryption control protocol. |
| inTerminateAcknowledge | The number of *Terminate-Acknowledge* packets received for the encryption control protocol. |
| inCodeReject | The number of *Code-Reject* packets received for the encryption control protocol. |
| inResetRequests | The number of *Reset-Request* packets received to reset the encryption history. |
| inResetACKs | The number of *Reset-Acknowledge* packets received to reset the encryption history. |
| decodeSuccesses | The number of encrypted packets successfully decoded. |
| decodeFailures | The number of encrypted packets that failed to be decoded. |
| decodeDiscards | The number of packets to be decoded that were discarded. |
| getSessKeySuccesses | The number of times a session key has been retrieved from the STAR module. |
| getMktSuccesses | The number of times a master key table has been retrieved from the STAR module. |
| starEventsNotAttached | The number of times the PPP interface received an event from the STAR module when it was not attached. |
| abortedNegotiations | The number of times an ECP negotiation was aborted |
| outOctets | The number of octets transmitted by the encryption protocol. This includes two octets per frame for the PPP encryption header, the number of octets of encrypted data transmitted, and the number of octets in control protocol packets (ECP). For multilinks an extra six octets per frame are included for the multilink header. |
| outUserPkts | The number of packets transmitted by the encryption control protocol. |
| outConfigureRequest | The number of *Configure-Request* packets transmitted by the encryption control protocol. |

Table 15-16: Parameters in output of the **show ppp count=lcp** command (cont.)

| Parameter | Meaning |
|---|---|
| outConfigureAcknowledge | The number of *Configure-Acknowledge* packets transmitted by the encryption control protocol. |
| outConfigureNAK | The number of *Configure-NAK* packets transmitted by the encryption control protocol. |
| outConfigureReject | The number of *Configure-Reject* packets transmitted by the encryption control protocol. |
| outTerminateRequest | The number of *Terminate-Request* packets transmitted by the encryption control protocol. |
| outTerminateAcknowledge | The number of *Terminate-Acknowledge* packets transmitted by the encryption control protocol. |
| outCodeReject | The number of *Code-Reject* packets transmitted by the encryption control protocol. |
| outResetRequests | The number of *Reset-Request* packets transmitted to reset the compression history. |
| outResetACKs | The number of *Reset-Acknowledge* packets transmitted to reset the compression history. |
| encodeSuccesses | The number of packets successfully encoded. |
| encodeFailures | The number of packets that failed to be encoded correctly. |
| encodeDiscards | The number of packets to be encoded that were discarded. |
| getSessKeyFailures | The number of times the PPP interface failed to retrieve a session key from the STAR module. |
| getMktFailures | The number of times the PPP interface failed to retrieve a master key from the STAR module. |
| starEventsWithLcpDown | The number of times the PPP interface received an event from the STAR module when the interface's LCP was not in the opened state. |
| CCP<br>ILCCP OVER: *<interface>* | Information about the compression control protocol (CCP) or ILCCP and the physical interface over which ILCCP is running. |
| inOctets | The number of octets received by the compression protocol. This includes two octets per frame for the PPP compression header, the number of octets of compressed data received, and the number of octets in control protocol packets (CCP). For multilinks an extra six octets per frame are included for the multilink header. |
| inUserPkts | The number of packets received by the compression control protocol. |
| inConfigureRequest | The number of *Configure-Request* packets received by the compression control protocol. |
| inConfigureAcknowledge | The number of *Configure-Acknowledge* packets received by the compression control protocol. |
| inConfigureNAK | The number of *Configure-NAK* packets received by the compression control protocol. |
| inConfigureReject | The number of *Configure-Reject* packets received by the compression control protocol. |
| inTerminateRequest | The number of *Terminate-Request* packets received by the compression control protocol. |

Table 15-16: Parameters in output of the **show ppp count=lcp** command (cont.)

| Parameter | Meaning |
|---|---|
| inTerminateAcknowledge | The number of *Terminate-Acknowledge* packets received by the compression control protocol. |
| inCodeReject | The number of *Code-Reject* packets received by the compression control protocol. |
| decodeSuccesses | The number of packets successfully decoded by the compression or encryption control protocol. |
| decodeFailures | The number of packets that failed to be decoded correctly by the compression or encryption control protocol. |
| decodeDiscards | The number of packets that were discarded by the compression or encryption control protocol. |
| inResetRequests | The number of *Reset-Request* packets received to reset the compression history. |
| inResetACKs | The number of *Reset-Acknowledge* packets received to reset the compression history. |
| encoEventsWithLcpDown | The number of times the PPP interface received an event from the ENCO module when the interface's LCP was not in the opened state. |
| outOctets | The number of octets transmitted by the compression protocol. This includes two octets per frame for the PPP compression header, the number of octets of compressed data transmitted, and the number of octets in control protocol packets (CCP). For multilinks an extra six octets per frame are included for the multilink header. |
| outUserPkts | The number of packets transmitted by the compression control protocol. |
| outConfigureRequest | The number of *Configure-Request* packets transmitted by the compression control protocol. |
| outConfigureAcknowledge | The number of *Configure-Acknowledge* packets transmitted by the compression control protocol. |
| outConfigureNAK | The number of *Configure-NAK* packets transmitted by the compression control protocol. |
| outConfigureReject | The number of *Configure-Reject* packets transmitted by the compression control protocol. |
| outTerminateRequest | The number of *Terminate-Request* packets transmitted by the compression control protocol. |
| outTerminateAcknowledge | The number of *Terminate-Acknowledge* packets transmitted by the compression control protocol. |
| outCodeReject | The number of *Code-Reject* packets transmitted by the compression control protocol. |
| encodeSuccesses | The number of packets successfully encoded. |
| encodeFailures | The number of packets that failed to be encoded correctly. |
| encodeDiscards | The number of packets to be encoded that were discarded. |
| outResetRequests | The number of *Reset-Request* packets transmitted to reset the compression history. |
| outResetACKs | The number of *Reset-Acknowledge* packets transmitted to reset the compression history. |

Table 15-16: Parameters in output of the **show ppp count=lcp** command (cont.)

| Parameter | Meaning |
|---|---|
| LQM OVER: *<interface>* | Information about LQR and the physical interface over which LQR is running. |
| lqrFailures | The number of times the LQR timer has timed out. |
| loopbacksDetected | The number of times the link entered loopback mode. |
| inLQRs | The number of LQR packets received. |
| inPktLost | The number of inbound LQR packets lost. |
| inOctetLost | The number of inbound LQR octets lost. |
| outLQRs | The number of LQR packets transmitted. |
| outPktLost | The number of outbound LQR packets lost. |
| outOctetLost | The number of outbound LQR octets lost. |
| outLQRsLost | The number of outbound LQR packets lost. |
| outLQRsTransit | The number of outbound LQR packets in transit. |
| PAP OVER: *<interface>* | Information about PAP and the physical interface over which PAP is running. |
| inRequest | The number of PAP *Authenticate-Request* packets received. |
| inAck | The number of PAP *Authenticate-Acknowledgement* packets received. |
| inNak | The number of PAP *Authenticate-Negative-Acknowledgement* packets received. |
| outRequest | The number of PAP *Authenticate-Request* packets transmitted. |
| outAck | The number of PAP *Authenticate-Acknowledgement* packets transmitted. |
| outNak | The number of PAP *Authenticate-Negative-Acknowledgement* packets transmitted. |
| CHAP OVER: *<interface>* | Information about CHAP and the physical interface over which CHAP is running. |
| inChallenge | The number of CHAP *Challenge* packets received for. |
| inResponse | The number of CHAP *Response* packets received. |
| inSuccess | The number of CHAP *Success* packets received. |
| inFailure | The number of CHAP *Failure* packets received. |
| outChallenge | The number of CHAP *Challenge* packets transmitted. |
| outResponse | The number of CHAP *Response* packets transmitted. |
| outSuccess | The number of CHAP *Success* packets transmitted. |
| outFailure | The number of CHAP *Failure* packets transmitted. |
| LCP OVER: *<interface>* | Information about the LCP and the physical interface over which LCP is running. |

Table 15-16: Parameters in output of the **show ppp count=lcp** command (cont.)

| Parameter | Meaning |
|---|---|
| inOctets | The number of octets received by the link control protocol. This includes the number of octets in control protocol packets (e.g. LCP, LQR, PAP, CHAP), plus the number of octets of data received. The number of octets of data equals the number of inOctets recorded for compression or encryption when enabled. If compression or encryption is not enabled, the number of inOctets of data equals the sum of the inOctets recorded for all the user protocols on this link. For multilinks an extra six octets per frame are included for the multilink header. |
| inUserPkts | The number of packets received for the LCP. |
| inConfigureRequest | The number of *Configure-Request* packets received for the LCP. |
| inConfigureAcknowledge | The number of *Configure-Acknowledge* packets received for the LCP. |
| inConfigureNAK | The number of *Configure-NAK* packets received for the LCP. |
| inConfigureReject | The number of *Configure-Reject* packets received for the LCP. |
| inTerminateRequest | The number of *Terminate-Request* packets received for the LCP. |
| inTerminateAcknowledge | The number of *Terminate-Acknowledge* packets received for the LCP. |
| inCodeReject | The number of *Code-Reject* packets received for the LCP. |
| inProtocolReject | The number of Protocol Reject packets received for the LCP. |
| inEchoRequest | The number of Echo Request packets received for the LCP. |
| inEchoReply | The number of Echo Reply packets received for the LCP. |
| inDiscardRequest | The number of Discard Request packets received for the LCP. |
| echoFailures | The number of times the ECHO timer has timed out. |
| outOctets | The number of octets transmitted by the link control protocol. This includes the number of octets in control protocol packets (e.g. LCP, LQR, PAP, CHAP), plus the number of octets of data received. The number of octets of data equals the number of inOctets recorded for compression or encryption when enabled. If compression or encryption is not enabled, the number of outOctets of data equals the sum of the outOctets recorded for all the user protocols on this link. For multilinks an extra six octets per frame are included for the multilink header. |
| outUserPkts | The number of packets sent for the LCP. |
| outConfigureRequest | The number of *Configure-Request* packets sent for the LCP. |
| outConfigureAcknowledge | The number of *Configure-Acknowledge* packets sent for the LCP. |
| outConfigureNAK | The number of *Configure-NAK* packets sent for the LCP. |
| outConfigureReject | The number of *Configure-Reject* packets sent for the LCP. |
| outTerminateRequest | The number of *Terminate-Request* packets sent for the LCP. |
| outTerminateAcknowledge | The number of *Terminate-Acknowledge* packets sent for the LCP. |

Table 15-16: Parameters in output of the **show ppp count=lcp** command (cont.)

| Parameter | Meaning |
|---|---|
| outCodeReject | The number of *Code-Reject* packets sent for the LCP. |
| outProtocolReject | The number of Protocol Reject packets sent for the LCP. |
| outEchoRequest | The number of Echo Request packets sent for the LCP. |
| outEchoReply | The number of Echo Reply packets sent for the LCP. |
| outDiscardRequest | The number of Discard Request packets sent for the LCP. |
| badEchoReplies | The number of *Echo Reply* packets received with a different ID than the original *Echo Request* packet. |

Figure 15-19: Example output from the **show ppp count=multilink** command

```
Multilink Counters
  inWholeFragments            1538     outWholeFragments           1538
  inStartFragments               0     outStartFragments              0
  inMiddleFragments              0     outMiddleFragments             0
  inEndFragments                 0     outEndFragments                0
  inNullFragments               54     outNullFragments              54
```

Table 15-17: Parameters in output of the **show ppp count=multilink** command

| Parameter | Meaning |
|---|---|
| inWholeFragments | The number of multilink encapsulated fragments received that contain a whole packet. |
| inStartFragments | The number of multilink encapsulated fragments received that contain the start of a packet. |
| inMiddleFragments | The number of multilink encapsulated fragments received that contain part of a packet that is not the start or the end. |
| inEndFragments | The number of multilink encapsulated fragments received that contain the end of a packet. |
| inNullFragments | The number of NULL multilink encapsulated fragments that have been received. |
| outWholeFragments | The number of multilink encapsulated fragments transmitted that contain a whole packet. |
| outStartFragments | The number of multilink encapsulated fragments transmitted that contain the start of a packet. |
| outMiddleFragments | The number of multilink encapsulated fragments transmitted that contain part of a packet that is not the start or the end. |
| outEndFragments | The number of multilink encapsulated fragments transmitted that contain the end of a packet. |
| outNullFragments | The number of NULL multilink encapsulated fragments that have been transmitted. |

Figure 15-20: Example output from the **show ppp count=ncp** command

```
IPCP
  inOctets                    63611     outOctets                   91768
  inUserPkts                   2098     outUserPkts                  1538
  inConfigureRequest              1     outConfigureRequest             1
  inConfigureAcknowledge          1     outConfigureAcknowledge         1
  inConfigureNAK                  0     outConfigureNAK                 0
  inConfigureReject               0     outConfigureReject              0
  inTerminateRequest              0     outTerminateRequest             0
  inTerminateAcknowledge          0     outTerminateAcknowledge         0
  inCodeReject                    0     outCodeReject                   0
```

Table 15-18: Parameters in output of the **show ppp count=ncp** command

| Parameter | Meaning |
|---|---|
| inOctets | The number of octets received by the network protocol. This includes two octets per frame for the PPP protocol header, the number of octets of user data to be passed up to the user protocol, and the number of octets in control protocol packets (e.g. IPCP, ATCP). |
| inUserPkts | The number of packets received for the network control protocol. |
| inConfigureRequest | The number of Configure-Request packets received for the network control protocol. |
| inConfigureAcknowledge | The number of Configure-Acknowledge packets received for the network control protocol. |
| inConfigureNAK | The number of Configure-NAK packets received for the network control protocol. |
| inConfigureReject | The number of Configure-Reject packets received for the network control protocol. |
| inTerminateRequest | The number of Terminate-RequestTerminate-Acknowledge packets received for the network control protocol. |
| inTerminateAcknowledge | The number of Terminate-Acknowledge packets received for the network control protocol. |
| inCodeReject | The number of Code-Reject packets received for the network control protocol. |
| outOctets | The number of octets transmitted by the network protocol. This includes two octets per frame for the PPP protocol header, the number of octets of user data passed down from the user protocol, and the number of octets in control protocol packets (e.g. IPCP, ATCP). |
| outUserPkts | The number of packets sent for the network control protocol. |
| outConfigureRequest | The number of Configure-Request packets sent for the network control protocol. |
| outConfigureAcknowledge | The number of Configure-Acknowledge packets sent for the network control protocol. |
| outConfigureNAK | The number of Configure-NAK packets sent for the network control protocol. |
| outConfigureReject | The number of Configure-Reject packets sent for the network control protocol. |

Table 15-18: Parameters in output of the **show ppp count=ncp** command (cont.)

| Parameter | Meaning |
|---|---|
| outTerminateRequest | The number of Terminate-Request packets sent for the network control protocol. |
| outTerminateAcknowledge | The number of Terminate-Acknowledge packets sent for the network control protocol. |
| outCodeReject | The number of Code-Reject packets sent for the network control protocol. |

Table 15-19: Parameters in output of the **show ppp count** command for **bap** and **bacp**

| Parameter | Meaning |
|---|---|
| BAP | Information about the operation of BAP. |
| inCallReq | The number of *Call-Request* packets received by the BAP protocol. |
| inCallResp | The number of *Call-Response* packets received by the BAP protocol. |
| inCallbackReq | The number of *Callback-Request* packets received by the BAP protocol. |
| inCallbackResp | The number of *Callback-Response* packets received by the BAP protocol. |
| inLinkDropQueryReq | The number of *Link-Drop-Query-Request* packets received by the BAP protocol. |
| inLinkDropQueryResp | The number of *Link-Drop-Query-Response* packets received by the BAP protocol. |
| inCallStatusInd | The number of *Call-Status-Indication* packets received by the BAP protocol. |
| inCallStatusResp | The number of *Call-Status-Response* packets received by the BAP protocol. |
| inErrors | The number of packets received by the BAP protocol that contained errors. |
| inDiscards | The number of packets received by the BAP protocol that were discarded. |
| outCallReq | The number of *Call-Request* packets transmitted by the BAP protocol. |
| outCallResp | The number of *Call-Response* packets transmitted by the BAP protocol. |
| outCallbackReq | The number of *Callback-Request* packets transmitted by the BAP protocol. |
| outCallbackResp | The number of *Callback-Response* packets transmitted by the BAP protocol. |
| outLinkDropQueryReq | The number of *Link-Drop-Query-Request* packets transmitted by the BAP protocol. |
| outLinkDropQueryResp | The number of *Link-Drop-Query-Response* packets transmitted by the BAP protocol. |
| outCallStatusInd | The number of *Call-Status-Indication* packets transmitted by the BAP protocol. |
| outCallStatusResp | The number of *Call-Status-Response* packets transmitted by the BAP protocol. |

Table 15-19: Parameters in output of the **show ppp count** command for **bap** and **bacp** (cont.)

| Parameter | Meaning |
|---|---|
| BACP | Information about the operation of BACP. |
| inOctets | The number of octets received by the BACP protocol. |
| inUserPkts | The number of packets received by the BACP protocol. |
| inConfigureRequest | The number of *Configure-Request* packets received by the BACP protocol. |
| inConfigureAcknowledge | The number of *Configure-Acknowledge* packets received by the BACP protocol. |
| inConfigureNAK | The number of *Configure-NAK* packets received by the BACP protocol. |
| inConfigureReject | The number of *Configure-Reject* packets received by the BACP protocol. |
| inTerminateRequest | The number of *Terminate-Request* packets received by the BACP protocol. |
| inTerminateAcknowledge | The number of *Terminate-Acknowledge* packets received by the BACP protocol. |
| inCodeReject | The number of *Code-Reject* packets received by the BACP protocol. |
| outOctets | The number of octets transmitted by the BACP protocol. |
| outUserPkts | The number of packets transmitted by the BACP protocol. |
| outConfigureRequest | The number of *Configure-Request* packets transmitted by the BACP protocol. |
| outConfigureAcknowledge | The number of *Configure-Acknowledge* packets transmitted by the BACP protocol. |
| outConfigureNAK | The number of *Configure-NAK* packets transmitted by the BACP protocol. |
| outConfigureReject | The number of *Configure-Reject* packets transmitted by the BACP protocol. |
| outTerminateRequest | The number of *Terminate-Request* packets transmitted by the BACP protocol. |
| outTerminateAcknowledge | The number of *Terminate-Acknowledge* packets transmitted by the BACP protocol. |
| outCodeReject | The number of *Code-Reject* packets transmitted by the BACP protocol. |

Figure 15-21: Example output from the **show ppp count=pppoe** command

```
ppp0                  1519 seconds     Last change at:        974 seconds
PPPoE Counters
  PADIs Rx                       1     PADIs Tx                       0
  PADOs Rx                       0     PADOs Tx                       1
  PADRs Rx                       1     PADRs Tx                       0
  PADSs Rx                       0     PADSs Tx                       1
  PADTs Rx                       0     PADTs Tx                       0
```

Table 15-20: Parameters in output of the **show ppp count=pppoe** command

| Parameter | Meaning |
|---|---|
| PADIs Rx | The number of PADI packets received. |
| PADIs Tx | The number of PADI packets transmitted. |
| PADOs Rx | The number of PADO packets received. |
| PADOs Tx | The number of PADO packets transmitted. |
| PADRs Rx | The number of PADR packets received. |
| PADRs Tx | The number of PADR packets transmitted. |
| PADSs Rx | The number of PADS packets received. |
| PADSs Tx | The number of PADS packets transmitted. |
| PADT Rx | The number of PADT packets received. |
| PADT Tx | The number of PADT packets transmitted. |

**Examples** To display the interface counters for PPP interface 1, use the command:

```
show ppp=1 count=interface
```

**Related Commands**

**show ppp**
**show ppp config**
**show ppp idletimer**
**show ppp multilink**
**show ppp utilisation**

# show ppp debug

**Syntax** `SHOW PPP[=`*`ppp-interface`*`] DEBUG`

where *ppp-interface* is a number from 0 to 1023

**Description** This command displays the debugging options that are currently enabled for the specified or all PPP interfaces (Figure 15-22, Table 15-21).

Figure 15-22: Example output from the **show ppp debug** command

```
Interface        Enabled Debug Modes
-------------------------------------------------------------
ppp0             AUTH, DECODE, LCP, PKT, UTILISATION
-------------------------------------------------------------
```

Table 15-21: Parameters in output of the **show ppp debug** command

| Parameter | Meaning |
|---|---|
| Interface | The interface name. |
| Enabled Debug Modes | The list of currently enabled debug modes for the interface; one or more of "AUTH", "BAPSTATE", "CALLBACK", "CTRLPKT", "DATAPKT", "DECODE", "DEMAND", "ENCO", "LCP", "LQR", "NCP", "PKT", "PPPOE" or "UTILISATION". |

**Examples** To display the debugging options set for all PPP interfaces, use the command:

```
show ppp debug
```

**Related Commands**

**disable ppp debug**
**disable ppp template debug**
**enable ppp debug**
**enable ppp template debug**

# show ppp idletimer

**Syntax** `SHOW PPP[=`*`ppp-interface`*`] IDLETIMER`

where *ppp-interface* is a number, from 0 to 1023

**Description** This command displays the configured and current values of the PPP idle timer for the specified or all PPP interfaces (Figure 15-23, Table 15-22).

Figure 15-23: Example output from the **show ppp idletimer** command

```
Interface
  Configured Idle Time       Idle Timer Value
-----------------------------------------------
ppp0
                     60                EXPIRED
-----------------------------------------------
```

Table 15-22: Parameters in output of the **show ppp idletimer** command

| Parameter | Meaning |
|---|---|
| ppp0 | The interface name. |
| Configured Idle Time | Seconds that a link must be idle before it is disconnected, or whether the idle timer is disabled. |
| Idle Timer Value | Seconds remaining until the link is disconnected or whether the timer has expired. |

**Examples** To display the idle timers for all PPP interfaces, use the command:

```
show ppp idletimer
```

**Related Commands**

**show ppp**
**show ppp config**
**show ppp count**
**show ppp multilink**

# show ppp limits

**Syntax** `SHOW PPP[=`*`ppp-interface`*`] LIMITS`

where *ppp-interface* is a number from 0 to 1023

**Description** This command displays information about the accumulated up-time and input/output data counters for all PPP interfaces or a specific one. It shows the current values of the counters, and any defined threshold limits (See Figure 15-24, Table 15-23).

Figure 15-24: Example output from the **show ppp** Limits command

```
Name                          Current      Limit           Remaining
------------------------------------------------------------------------------
ppp0            Up Time       16:12        25 hrs           08:47
                In Data       EXCEEDED     50 MB          0.0 MB
                Out Data      21.5 MB      Unlimited        --
                Total Data    71.5 MB      Unlimited        --
------------------------------------------------------------------------------
```

Table 15-23: Parameters in output of the **show ppp limits** command

| Parameter | Meaning |
|---|---|
| Name | The name of the PPP interface. |
| Over | The lower layer(s) used by the PPP interface; SYN*n*, ISDN-*callname*, ACC-*callname*, MIOX*n-circuitname*, TDM-*groupname,* ETH*n-servicename*, VLAN*n-servicename,* TNL-*callname*. |
| Up-Time | The cumulative up-time, in hours, for the interface. |
| In Data | The cumulative input data throughput, in megabytes, for the interface. |
| Out Data | The cumulative output data throughput, in megabytes, for the interface. |
| Total Data | The cumulative total data throughput, in megabytes, for the interface. |
| Current | The current value of the cumulative counter, or "EXCEEDED" if the interface counter has exceeded the corresponding threshold limit. |
| Limit | The threshold limits for the interface, or "Unlimited" if no value has been specified. |
| Remaining | The remaining time/data throughput allowed before the limit is exceeded and the link closed. |

**Example** To display the up-time and input/output data counters for PPP interface 0, use the command:

```
show ppp=0 limits
```

**Related Commands**

**add ppp acservice**
**create ppp template**
**reset ppp**
**set ppp**
**set ppp acservice**

# show ppp multilink

**Syntax** `SHOW PPP[=`*`ppp-interface`*`] MULTILINK`

where *ppp-interface* is a number from 0 to 1023

**Description** This command displays information about a multilink bundle for all PPP interfaces or a specific one (Figure 15-25, Table 15-24).

Figure 15-25: Example output from the **show ppp multilink** command

```
Interface
  Parameter                                                                Value
-------------------------------------------------------------------------------
ppp0
  Multilink Enabled                                                          Yes
  Fragmentation Enabled                                                       No
  Acceptable fragmentation overhead for VF scheme (%)                          5
  Minimum packet size for fragmentation using VF scheme (bytes)              120
  Null fragment timer (seconds)                                                3
  Number of links in bundle                                                    4
  Total bandwidth of bundle (bps)                                         256000
  Number of packets fragmented using VF scheme                                 0
  Number of packets fragmented using FF scheme                                 0
  Number of packets not fragmented                                           971
  Next output sequence number                                                972
  Minimum sequence number received on bundle                                 863
  Next expected sequence number                                              866
  Length of receive queue                                                      0
  Discards from receive queue                                                  0
-------------------------------------------------------------------------------
```

Table 15-24: Parameters in output of the **show ppp multilink** command

| Parameter | Meaning |
|---|---|
| ppp0 | The interface name. |
| Multilink Enabled | Whether multilink is enabled on this PPP interface. |
| Fragmentation Enabled | Whether fragmentation is enabled on this PPP interface. |
| Acceptable fragmentation overhead for VF scheme (%) | The maximum acceptable overhead for fragmentation using the variable fragmentation scheme, as a percentage of packet size. |
| Minimum packet size for fragmentation using VF scheme (bytes) | The minimum size packet that may be fragmented using the variable fragmentation scheme. |
| Null fragment timer (seconds) | Seconds that the link must be idle before a null fragment is transmitted. |
| Number of links in bundle | The number of links in the multilink bundle. |
| Total bandwidth of bundle (bps) | The total bandwidth in bits per second of the multilink bundle. |
| Number of packets fragmented using VF scheme | The number of packets that have been fragmented using the variable fragmentation scheme. |
| Number of packets fragmented using FF scheme | The number of packets that have been fragmented using the fixed fragmentation scheme. |

Table 15-24: Parameters in output of the **show ppp multilink** command (cont.)

| Parameter | Meaning |
|---|---|
| Number of packets not fragmented | The number of packets that have not been fragmented. |
| Next output sequence number | The sequence number to use in the next transmission over the multilink bundle. |
| Minimum sequence number received on bundle | The lowest sequence number received via the multilink bundle. |
| Next expected sequence number | The next sequence number expected via the multilink bundle. |
| Length of receive queue | The current length of the receive queue. |
| Discards from receive queue | The number of packets discarded from the receive queue due to lost packets causing sequence number synchronisation to be lost. |

## Examples

To display multilink information for all PPP interfaces, use the command:

```
show ppp multilink
```

## Related Commands

**show ppp**
**show ppp config**
**show ppp count**
**show ppp idletimer**

# show ppp nameserver

**Syntax** `SHOW PPP NAMESERVER`

**Description** This command displays information about the currently configured global DNS and WINS servers (Figure 15-26, Table 15-25).

Figure 15-26: Example output from the **show ppp nameserver** command

```
Name Server                        Address
-------------------------------------------------
Primary DNS ..................... 192.168.2.3
Secondary DNS ................... 192.168.5.1
Primary WinS .................... 192.168.5.5
Secondary WinS .................. Not Set
-------------------------------------------------
```

Table 15-25: Parameters in output of the **show ppp nameserver** command

| Parameter | Meaning |
|---|---|
| Primary DNS Address | The IP address of the primary DNS server, passed to a peer in response to an IPCP primary DNS request. |
| Secondary DNS Address | The IP address of the secondary DNS server, passed to a peer in response to an IPCP secondary DNS request. |
| Primary WinS Address | The IP address of the primary WINS server, passed to a peer in response to an IPCP primary WINS server request. |
| Secondary WinS Address | The IP address of the secondary WINS server, passed to a peer in response to an IPCP secondary WINS server request. |

**Examples** To display the currently configure DNS and WINS servers, use the command:

```
show ppp nameserver
```

**Related Commands** **set ppp**
**show ppp**

# show ppp pppoe

**Syntax** `SHow PPP PPPOE`

**Description** This command displays information about PPPoE interfaces and services that are currently configured (Figure 15-27, Table 15-26).

Figure 15-27: Example output from the **show ppp pppoe** command

```
PPPOE
------------------------------------------------------------
PPP1:
  Service Name ................. bob
  Peer Mac Address ............. 00-00-cd-00-ab-a3
  Interface .................... eth0
  Session ID ................... a1a3
  Maximum Segment Size ......... 1292

  Access Concentrator Mode ..... Enabled

Services:
  bob
    Max sessions ................ 2
    Current Sessions ............ 1
    Template .................... 1
    Interface ................... eth1
    MAC RADIUS Authentication ... YES
  carol
    Max sessions ................ 5
    Current Sessions ............ 0
    Template .................... 1
    Interface ................... vlan1
    MAC RADIUS Authentication ... YES

PPPOE Counters:
  Rejected PADI packets ...... 0
  Rejected PADO packets ...... 0
  Rejected PADR packets ...... 0
  Rejected PADS packets ...... 0
  Rejected PADT packets ...... 0
------------------------------------------------------------
```

Table 15-26: Parameters in output of the **show ppp pppoe** command

| Parameter | Meaning |
|---|---|
| Service Name | The name of the service either that the PPPoE interface is offering (if the interface is in access concentrator mode) or that the PPPoE interface is requesting/using (if the interface is not in access concentrator mode). |
| Peer Mac Address | The MAC address of the PPPoE peer. |
| Interface | The interface that the PPPoE Access Concentrator or PPPoE Client is using. If all Ethernet interfaces are being used, "ethernet" will be displayed. |
| Session ID | The ID of the current session. |

Table 15-26: Parameters in output of the **show ppp pppoe** command (cont.)

| | |
|---|---|
| Maximum Segment Size | The maximum number of bytes that the data payload may occupy in a TCP packet. This figure is derived by subtracting the clamped MSS header size from the MTU of the interface |
| Access Concentrator Mode | Enabled indicates that the router is acting as a PPPoE Access Concentrator. Disabled indicates that it is not acting as a PPPoE Access Concentrator. A router can act as both a PPPoE Access Concentrator and PPPoE client simultaneously. |
| Services | The list of PPPoE services that the router can offer. This list is displayed when the router is in Access Concentrator mode. |
| Max Session | The maximum number of simultaneous instances of the service that are permitted. |
| Current Sessions | The number of instances of the service currently in use. |
| Template | The number of the PPP template used to make PPP sessions that run over the service. |
| MAC RADIUS Authentication | On indicates that when RADIUS User Authentication is in use, the MAC address of the PPPoE peer is sent with the username and password. Off indicates that the peer MAC address is not sent. |
| Rejected PADI packets | The number of PPPoE PADI packets rejected because the requested service name was not available. |
| Rejected PADO packets | The number of PPPoE PADO packets rejected because no PPPoE interface was expecting a PADO or the PADO was not matched with a PPPoE interface. |
| Rejected PADR packets | The number of PPPoE PADR packets rejected because the PADR packet was not matched with a PPPoE interface. |
| Rejected PADS packets | The number of PPPoE PADS packets rejected because the PADS packet was not matched with a PPPoE interface. |
| Rejected PADT packets | The number of PPPoE PADT packets rejected because the PADT packet was not matched with a PPPoE interface. |

## Examples

To display information about all the currently configured PPPoE interfaces and general PPPoE counters, use the command:

```
sh ppp pppoe
```

## Related Commands

**add ppp**
**create ppp**
**enable ppp debug**
**show ppp**
**show ppp config**

# show ppp template

**Syntax** `SHow PPP TEMPLATE[=`*`template`*`] [DEBUG]`

where *template* is a number from 0 to 31

**Description** This command displays information about PPP templates.

The **template** parameter specifies the number of the template to display. If a template is not specified, information is displayed about all templates, including the default template. If a template is specified, information about the specified template is displayed (Figure 15-28, Table 15-27 on page 15-127). If no templates have been defined, the default template is displayed.

If **debug** is specified, the debugging modes enabled for the template or all templates are displayed (Figure 15-29 on page 15-128, Table 15-28 on page 15-128).

Figure 15-28: Example output from the **show ppp template** command

```
Template - Description
  Parameter                                          Value
-------------------------------------------------------------------------------
pppt0 - Template for calls from Head Office
  Multilink                                            ON
  Maximum links                                         4
  Bandwidth Allocation Protocol                        ON
  Bandwidth Allocation Call Mode                     CALL
  Multilink fragmentation                             OFF
  Acceptable Fragment Overhead (%)                      5
  Null Fragment Timer (seconds)                         3
  Idle Timer (seconds)                                OFF
  Compression                                          ON
  Compression Algorithm                           STACLZS
  Compression Checkmode                               LCB
  Encryption                                          OFF
  Username                                        NOT SET
  Password                                        NOT SET
  Login Servers                     RADIUS,TACACS,USER
  IP Pool                                         NOT SET
  Request IP Address                                   NO
  VJC                                                 OFF
  Clamped MSS Header Size (bytes)                     200
Link
  Authentication                                     NONE
  CHAP Rechallenge (max. period seconds)              900
  Callback Mode                                       OFF
  Callback Operation                                 USER
  Callback Number                                       -
  Callback Delay (seconds)                              5
  Echo Timer (seconds)                                 10
  LQR Timer (seconds)                                  60
  Magic Number                                         ON
  Maximum Receive Unit                                OFF
  Restart Timer (seconds)                               3
Debug
  Maximum packet bytes to display                      32
-------------------------------------------------------------------------------
```

Table 15-27: Parameters in output of the **show ppp template** command

| Parameter | Meaning |
|---|---|
| pppT<*template*> - <*description*> | The number and description of the template. |
| Multilink | Whether dynamic PPP calls can be multilinked together; either "ON" or "OFF". |
| Maximum links | The maximum number of links allowed in a multilink bundle created with this template. |
| Bandwidth Allocation Protocol | Whether the Bandwidth Allocation Protocol is enabled; either "ON" or "OFF". |
| Bandwidth Allocation Call Mode | The call mode for the Bandwidth Allocation Protocol, if the Bandwidth Allocation Protocol is enabled; either "CALL" or "CALLBACK". |
| Multilink fragmentation | Whether multilink packets may be fragmented. |
| Acceptable Fragment Overhead(%) | The maximum amount of overhead allowed to be added to each packet due to variable fragmentation. If this level is exceeded when fragmentation of a packet is done using the variable fragmentation scheme, then the fixed fragmentation scheme is used instead. |
| Null Fragment Timer | Seconds that the link must be idle for before a Null fragment is sent on a link in a multilink bundle. |
| Idle Timer (seconds) | Seconds that a link must be idle before it is disconnected or whether the idle timer is disabled. |
| Compression | Whether compression is enabled. |
| Compression Algorithm | The compression algorithm to use for compressing packets; either "PREDICTOR" or "STACLZS". |
| Compression Checkmode | The check mode used by the compression algorithm to determine when a decompression history becomes unsynchronised; either SEQUENCE, LCB, CRC16, or CRCCCITT. |
| Encryption | Whether encryption is enabled. |
| Username | The username used by the PPP interface for both PAP and CHAP authentication, or "NOT SET" if a username has not been set. |
| Password | Whether a password has been set for the entire PPP interface. |
| Login Servers | The authentication servers to use; "USER", "RADIUS", "TACACS", or "NOT SET" if a login server has not been set. |
| IP Pool | The name of the IP address pool used to assign IP addresses for this PPP interface, or "NOT SET" if an IP address pool has not been assigned. |
| Request IP Address | Whether an IP address is requested from the peer during IPCP negotiation. |
| VJC | Whether Van Jacobson Header Compression is configured on the PPP template. Either ON or OFF. |
| Clamped MSS Header Size | The amount of space, in bytes, within an MTU that is reserved for packet headers. This amount is subtracted from the MTU of the interface to define the Maximum Segment Size (MSS). |

Table 15-27: Parameters in output of the **show ppp template** command (cont.)

| Parameter | Meaning |
|---|---|
| Authentication | The authentication protocol in use; either NONE, PAP, CHAP, or EITHER. |
| CHAP Rechallenge | Seconds for the maximum interval between CHAP rechallenges or whether CHAP rechallenge is disabled. |
| Callback Mode | Whether the link requests callback, accepts callback or does neither. |
| Callback Operation | The callback operation included in callback requests; either "USERAUTH" or "E164NUMBER". |
| Callback Number | The callback number to include in callback requests when the callback operation is E164NUMBER. |
| Callback Delay (seconds) | Seconds for the delay between deactivating a call for callback and making the call back to the peer. |
| Echo Timer (seconds) | Seconds for the interval between transmissions of LCP *Echo Request* messages. |
| LQR Timer (seconds) | Seconds between LQR packets transmitted over the physical interface. |
| Maximum Receive Unit | The MRU value to include in LCP requests, DEFAULT, or OFF. |
| Maximum Transmission Unit | The value for the maximum transmission unit; the default is 1500. |
| Magic Number | Whether the magic number option is enabled. |
| Restart Timer | Seconds between configure requests for the physical interface. |
| Maximum packet bytes to display | The maximum number of bytes of each PPP packet displayed by the PACKET debugging option. |

Figure 15-29: Example output from the **show ppp template debug** command

```
Template      Call                      Enabled Debug Modes
------------------------------------------------------------------------
pppT0                                   PKT,LCP,NCP
------------------------------------------------------------------------
```

Table 15-28: Parameters in output of the **show ppp template debug** command

| Parameter | Meaning |
|---|---|
| Template | The name of a PPP template. |
| Call | The lower layer call using this template, if any. |
| Enabled Debug Modes | The debugging modes enabled for the template (and call); one or more of "AUTH", "BAPSTATE", "CALLBACK", "DEMAND", "ENCO", "LCP", "NCP", "PKT" or "UTILISATION". |

**Examples** To display the configuration for all templates, use the command:

```
show ppp template
```

To display the configuration for template 1, use the command:

```
show ppp template=1
```

To display the debugging modes enabled for template 3, use the command:

```
show ppp template=3 debug
```

**Related Commands**

**create ppp template**
**destroy ppp template**
**disable ppp template debug**
**enable ppp template debug**
**set ppp acservice**
**set ppp template**

# show ppp txstatus

**Syntax** `SHOW PPP[=`*`ppp-interface`*`] TXSTATUS`

where *ppp-interface* is the PPP interface number, from 0 to 1023

**Description** This command displays information about the status of a PPP transmission queue for the specified interface of all interfaces (Figure 15-30, Table 15-29).

Figure 15-30: Example output from the **show ppp txstatus** command

```
Interface
  Parameter                                          Value
-------------------------------------------------------------
ppp0
  Interface transmission queue length .......... 0

eth0
  Packets started transmission ................. 198
  Packets being transmitted .................... 0
  Packets lost during transmission ............. 3
  Packets finished transmission ................ 195
  Packets discarded in pipe .................... 0
  Link transmission queue length ............... 0
  Driver bandwidth (bps) ....................... 48000
  Driver transmission delay (ms) ............... 0
  Driver transmission status ................... Ready
-------------------------------------------------------------
```

Table 15-29: Parameters in output of the **show ppp txstatus** command

| Parameter | Meaning |
|---|---|
| ppp<*n*> | Name of a PPP interface. |
| Interface transmission queue length | The length of the output queue for this PPP interface. |
| <*physical-interface*> | The name of a physical interface or channel forming part of this PPP interface. |
| Packets started transmission | The total number of packets that higher-layer protocols requested be transmitted to a non-unicast address, including those that were discarded or not sent. |
| Packets being transmitted | The number of packets currently being transmitted on this physical interface or channel. |
| Packets lost during transmission | The number of packets lost during transmission on this physical interface or channel. |
| Packets finished transmission | The number of packets that have been transmitted and acknowledged on this physical interface or channel. |
| Packets discarded in pipe | The number of packets discarded on this physical interface or channel. |

Table 15-29: Parameters in output of the **show ppp txstatus** command (cont.)

| Parameter | Meaning |
|---|---|
| Link transmission queue length | The length of the output queue for this physical interface or channel. |
| Driver bandwidth (bps) | The bandwidth of the layer 1 device driver for this physical interface or channel. |
| Driver transmission delay (ms) | Milliseconds of the delay in the layer 1 device driver for this physical interface or channel. |
| Driver transmission status | Whether the layer 1 device driver for this physical interface or channel is busy or ready. |

**Examples** To show the status of the PPP transmission queue, use the command:

```
show ppp txstatus
```

**Related Commands** **show ppp**

# show ppp utilisation

**Syntax** `SHOW PPP[=`*`ppp-interface`*`] UTILISATION`

where *ppp-interface* is the PPP interface number, from 0 to 1023

**Description** This command shows PPP utilisation measurements for each lower layer interface and the overall utilisation for all PPP interfaces, or a particular interface if specified (Figure 15-31, Table 15-30). The data is sent to the port, or telnet session, where the command was entered.

Figure 15-31: Example output from the **show ppp utilisation** command

```
Interface: ppp0              Time: 15:45:39
  Over: tdm-0                  Utilisation(%): 93       Bandwidth(bps): 2048000
  Over: tdm-1                  Utilisation(%): 2        Bandwidth(bps): 2048000
  Utilisation Overall(%): 47   Up Timer(seconds): 0     Down Timer(seconds): 0
```

Table 15-30: Parameters in output of the **show ppp utilisation** command

| Parameter | Meaning |
|---|---|
| Interface | The PPP interface being queried. |
| Time | The time the command was processed. |
| Over | The name of each lower layer interface of the PPP interface in question. |
| Utilisation(%) | The current utilisation rate of each lower layer interface. |
| Bandwidth(bps) | The bandwidth of each lower layer interface. |
| Utilisation Overall(%) | The overall utilisation rate of the PPP interface. |
| Up Timer(seconds) | The time remaining after which an additional channel opens, providing that the overall utilisation rate remains above the threshold specified by the **uprate** parameter when the **create/set ppp** command was issued. |
| Down Timer(seconds) | The time remaining after which a channel closes, providing that the overall utilisation rate remains below the threshold specified by the **downrate** parameter when the **create/set ppp** command was issued. |

**Examples** To show utilisation statistics on PPP interface 2, use the command:

```
show ppp=2 utilisation
```

**Related Commands**

**enable ppp debug**
**disable ppp debug**
**enable ppp template debug**
**disable ppp template debug**
**show ppp config**
**show ppp debug**