Configuring Multilink PPP Connections for Broadband and Serial Topologies

This module describes how to configure Multilink PPP over broadband and serial interfaces. Configuring Multilink PPP over broadband includes configuring Multilink PPP over ATM (MLPoA), Multilink PPP over Ethernet (MLPoE), Multilink PPP over Ethernet over ATM (MLPoEoA), Multilink PPP over Queue-in-Queue (MLPoQinQ), and Multilink PPP over VLAN (MLPoVLAN).

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for Multilink PPP Connections for Broadband and Serial Topologies

Multilink PPP over Broadband Restrictions

This section lists the common limitations and caveats for all broadband Multilink PPP models supported by Cisco software. For information about basic broadband restrictions, limitations, and caveats, see the *Broadband Access Aggregation and DSL Configuration Guide*.

- Some Cisco broadband Multilink PPP implementations support the creation of a Multilink PPP bundle for some broadband configurations either through a virtual template interface or through an interface multilink. Cisco software supports virtual template interfaces only for broadband Multilink PPP bundle types. Interface multilink and multilink groups are supported on Multilink PPP over serial interfaces (that is, nonbroadband Multilink PPP).

Note   A virtual access interface (VAI) will be created for a session that is based on a virtual template configuration.

- Cisco software supports a maximum of 4096 virtual template interfaces. In many cases, a single common virtual template configuration may be used to create multiple bundle VAI instances, especially for single-link bundle sessions.

- Depending on your release, there may be a limit of one link per Multilink PPP bundle. The **ppp multilink links maximum** command should be configured on the virtual template to ensure that requests for additional links in a bundle are rejected.

- Cisco In-Service Software Upgrade (ISSU) and stateful switchover (SSO) for broadband Multilink PPP sessions are not supported.

QoS Restrictions

- Quality of service (QoS) is supported only on a Multilink PPP bundle. QoS is not supported on PPP sessions that make up the Multilink PPP bundle. Multilink PPP does not function properly if queuing policies are attached to PPP sessions that make up the Multilink PPP bundle.

- QoS Bandwidth Remaining Ratio (BRR) is not supported with Multilink PPP. BRR will require QoS policies to be applied to PPP sessions that make up the Multilink PPP bundle, which will cause Multilink PPP to not function properly.

Information About Multilink PPP Connections for Broadband and Serial Topologies

Multilink PPP

The Multilink PPP feature provides load balancing functionality over multiple WAN links while providing multivendor interoperability and support for packet fragmentation, proper sequencing, and load calculation on both inbound and outbound traffic. The Multilink PPP feature supports the fragmentation and packet sequencing specifications described in RFC 1990.

Multilink PPP allows packets to be fragmented and fragments to be sent at the same time over multiple point-to-point links to the same remote address. Multiple links come up in response to a defined dialer load threshold. The load can be calculated on inbound or outbound traffic, as required, for the traffic between specific sites. Multilink PPP provides bandwidth on demand and reduces transmission latency across WAN links.

Multilink PPP can work over synchronous and asynchronous serial type of single or multiple interfaces that have been configured to support both dial-on-demand rotary groups and PPP encapsulation.

Multilink PPP Bundles

Multilink PPP combines multiple physical links into a logical bundle called a Multilink PPP bundle. A Multilink PPP bundle is a single, virtual interface that connects to the peer system. Having a single interface (Multilink PPP bundle interface) provides a single point to apply hierarchical queueing, shaping, and policing to traffic flows. Individual links in a bundle do not perform any hierarchical queueing. None of the links have any knowledge about the traffic on parallel links. Hierarchical queueing and quality of service (QoS) cannot be applied uniformly to the entire aggregate traffic between a system and its peer system. A single, virtual interface also simplifies the task of monitoring traffic to the peer system (for example, all traffic statistics run on one interface).

Figure 1: Multilink PPP Bundle
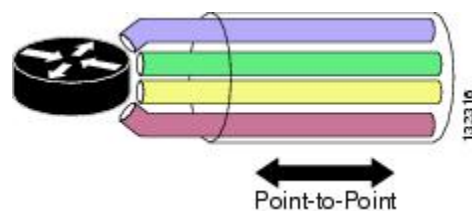
Multilink PPP Bundles and PPP Links

Multilink PPP works with fully functional PPP interfaces. A Multilink PPP bundle can have multiple links connecting peer devices. These links can be serial links or broadband links (Ethernet or ATM). As long as each link behaves like a standard serial interface, mixed links work properly in a bundle.

To designate a link to a specified bundle, use the **ppp multilink group** command for configuring the link. This command restricts the link to join only the specified bundle. When a link negotiates to join a Multilink PPP bundle, the link must provide proper identification that is associated with the Multilink PPP bundle. If the negotiation is successful, the link is assigned to the requested Multilink PPP bundle. If the link provides identification that coincides with the identification associated with a different Multilink PPP bundle in the system or if the link fails to match the identity of a Multilink PPP bundle that is already active on the multilink group interface, the connection terminates.

A link joins a Multilink PPP bundle only if it negotiates to use the bundle when a connection is established and the identification information exchanged matches that of an existing bundle.

When you configure the **ppp multilink group** command on a link, the command applies the following restrictions on the link:

- The link is not allowed to join any bundle other than the indicated group interface.

- The PPP session must be terminated if the peer device attempts to join a different bundle.

A link joins a bundle only when the identification keys for that link match the identification keys for an existing bundle. (See the "Factors that Govern a Link Joining a Bundle" section.) Merely configuring the **ppp multilink group** command on a link does not allow the link to join the corresponding bundle; the link must have matching identification keys to join the corresponding bundle. Identification keys are always used as the determining factor for matching links with bundles.

Because the **ppp multilink group** command merely places a restriction on a link, any Multilink-PPP-enabled link that is not assigned to a particular multilink group can join the dedicated bundle interface if the Multilink-PPP-enabled link provides correct identification keys for that dedicated bundle. Removing the **ppp multilink group** command from an active link that is currently a member of a multilink group does not make that link leave the bundle because the link is still a valid member of the multilink group. However, the link is no longer restricted to this one bundle.

The table below lists the different configurations of Multilink PPP and the number of links supported by each one of them.

Feature Name	Number of Links Supported Per Bundle
Multilink PPP over Serial	10
Multilink PPP over Ethernet	1 or 8 (depending on your release)
Multilink PPP over Ethernet over VLAN	1 or 8 (depending on your release)*
Multilink PPP over Ethernet over QinQ	1 or 8 (depending on your release)*
Multilink PPP over Ethernet over ATM	1 or 8 (depending on your release)*
Multilink PPP over ATM (AAL5MUX, AL5SNAP)	1 or 8 (depending on your release)*
Multilink PPP over LNS	1 or 8 (depending on your release)*

* See the Release Notes specific to your platform and release.

Link Fragmentation and Interleaving

The main benefit of Multilink PPP is the support for link fragmentation and interleaving (LFI). LFI minimizes packet latency on delay-sensitive voice, video, and interactive applications when data is sent over low-speed interfaces.

With LFI, the latency of delay-sensitive traffic is minimized because Multilink PPP breaks the nonpriority or nonlatency-sensitive traffic into smaller fragments. The delay-sensitive traffic is then PPP encapsulated and interleaved with nonpriority Multilink PPP fragments or packets. At the receiver, Multilink PPP fragments or packets are reordered and reassembled while the PPP-encapsulated packets are received and immediately forwarded. (Multilink PPP is bypassed; no reordering or reassembly is performed.)

Types of Multilink PPP Bundle Interfaces

Multilink PPP bundle interfaces can be one of the following types:

- Multilink group interfaces
- Virtual access interfaces (VAIs)

Both these types of interfaces provide the same level of PPP and multilink functionality after a bundle is established. All PPP and multilink-related features run identically on a bundle.

Multilink Group Interface

A multilink group interface is a static interface that exists whether or not it is being used at a particular point in time. A static interface is defined in the startup configuration file on a device. This type of interface is created automatically when a device boots up. Multilink group interfaces are dedicated to specific remote users. These interfaces are used in leased-line environments, where you have information about all physical links connections and where the number of users is defined by the number of physical connections in the system.

Multilink group interfaces allow you to track a specific user's activity. By examining a user's associated interface, you can easily see whether a user is connected and how much traffic the user has sent or received. You can monitor the state of a multilink group interface for issues such as network outages.

Virtual Access Interface

A virtual access interface (VAI) is a type of interface that is used for Multilink PPP bundles, specifically for Multilink PPP over broadband. A VAI is created dynamically for a multilink connection and released as soon as a multilink connection is torn down. A bundle interface of this type exists only as long as a user is connected. As soon as a user disconnects, a virtual access interface no longer exists.

A VAI is the default type of bundle interface when Multilink PPP is running in a broadband topology. For broadband topologies, a VAI replaces multilink group interfaces, which are used in serial topologies.

The use of a VAI has the following advantages:

- The number of bundle interfaces depends only on the number of currently active multilink users and not on the size of the user database.

• Because a local configuration source does not exist for per-user information, this information is derived from another source, such as an authentication, authorization, and accounting (AAA) server.

The disadvantage of using a VAI lies in its monitoring capability. A multilink group interface, used in serial deployments, allows the monitoring of each user's traffic through a serial link. With a VAI, there is no dedicated interface defined in a device's configuration file. Therefore, you must track a user's activity by using other means, such as the accounting mechanism of an AAA server.

Factors that Govern a Link Joining a Bundle

A link joins a bundle when identification keys for that link match identification keys for an existing bundle.

The following two keys define the identity of a remote system: the PPP username and Multilink PPP endpoint discriminator. PPP authentication mechanisms (for example, password authentication protocol [PAP] or Challenge-Handshake Authentication Protocol [CHAP]) learn the PPP username. The endpoint discriminator is an option negotiated by the Link Control Protocol (LCP). Therefore, a bundle consists of links that have the same PPP usernames and endpoint discriminators.

A link that does not provide a PPP username or an endpoint discriminator is an anonymous link. Multilink PPP collects all anonymous links into a single bundle that is referred to as the anonymous bundle or the default bundle. Typically, there can be only one anonymous bundle. Any anonymous links that negotiate Multilink PPP join (or create) the anonymous bundle.

When you use multilink group interfaces, more than one anonymous bundle is allowed. When you preassign a link to a Multilink PPP bundle by using the **ppp multilink group** command and the link is anonymous, the link joins the bundle interface to which it is assigned if the interface is not already active and is associated with an anonymous user.

Multilink PPP determines the bundle a link joins by following these steps:

1 When a PPP session is initiated, Multilink PPP creates a bundle name identifier for the link.

2 Multilink PPP then searches for a bundle with the same bundle name identifier. The following scenarios are possible:

   • If a bundle with the same identifier exists, the link joins that bundle.

   • If a bundle with the same identifier does not exist, Multilink PPP creates a new bundle with the same identifier as the link, and the link becomes the first link to join the bundle.

The table below describes commands and associated algorithms that are used to generate a bundle name. In the table, "username" typically means an authenticated username; however, an alternate name can be used instead. The alternate name is usually an expanded version of the username (for example, virtual private dialup network [VPDN] tunnels might include the network access server name) or a name derived from other sources.

Table 1: Bundle Name Generation Commands

Command	Bundle Name Generation Algorithm
multilink bundle-name authenticated	The bundle name is derived from the username that is defined on a peer device. The username is passed from the peer device during PPP negotiation. If a peer does not provide a username, the algorithm uses the peer's endpoint discriminator. **Note** The **authenticated** keyword specifies that the bundle name should be derived by Cisco software. Because multiple bundles may exist on a device concurrently, Cisco software must ensure a unique name for each bundle. The endpoint discriminator is ignored entirely, unless it is the only name that can be found. The **multilink bundle-name authenticated** command is the default naming policy.
multilink bundle-name both	The bundle name is a concatenation of the username and the endpoint discriminator.
multilink bundle-name endpoint	The bundle name is a peer's endpoint discriminator. If there is no endpoint discriminator, the algorithm uses the peer's username.

Rate of Session Establishment for Multilink PPP Bundles

When devices running Cisco software begin negotiating a large number of broadband sessions, a peer device may be constrained by its processing capabilities. This limitation may cause an excessive number of timeouts (because the peer device may be renegotiating hundreds of sessions due to timeouts) while trying to negotiate PPP parameters. Cisco software provides an internal mechanism to control the rate of session establishment (to prevent excessive timeouts) for broadband sessions by using the Call Admission Control (CAC) functionality. CAC can be configured to control the number of sessions that can be negotiated in a given period of time. Controlling the rate of session establishment is also known as throttling. The mechanism of throttling helps to prevent unnecessary negotiation timeouts with slower devices. The following commands show how to configure the CAC functionality:

```
Device(config)# call admission new-model
Device(config)# call admission limit 500
Device(config)# call admission cpu-limit 80
Device(config)# call admission pppoe 10 1
Device(config)# call admission pppoa 10 1
Device(config)# call admission vpdn 10 1
```

- The **call admission new-model** command enables the new-model-based CAC, which regulates session establishment based on both CPU utilization and incoming session requests.

- The **call admission limit** command (also referred to as charge limit) sets the maximum total concurrent charge threshold. If this threshold is exceeded, any additional session requests are rejected.

- The **call admission cpu-limit** command specifies the CPU utilization threshold, as a percentage. If this threshold is exceeded, new sessions are rejected. CAC uses the 5-second CPU utilization of IOS daemon (IOSd) for this calculation.

- Session requests are rejected if either the **cpu-limit** (80% in the example above) or the charge **limit** (1000 in the example above) is exceeded.

- The **call admission pppoe 10 1** command specifies the charge values for a single PPP over Ethernet (PPPoE) session. In the above example, the session charge is 10, and the lifetime is 1 second. The charge values are set per call type (PPP over ATM [PPPoA], PPPoE, or virtual private dialup network [VPDN]). The extended lifetime is calculated as the sum of two lifetime values. For the above example, the extended lifetime is 1 + 1 = 2.

- You can calculate calls per second (CPS) as follows:
  `CPS = Charge Limit / (Session Charge * Extended Lifetime)`
  For the above example, CPS = 500 / (10 * 2) = 25

For a more detailed explanation, refer to the "Broadband Scalability and Performance" module in the *Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide*.

Multilink PPP Packet Overhead

Multilink PPP encapsulation adds six extra bytes (four header and two checksum) to each outbound packet. These overhead bytes reduce the effective bandwidth of a connection; therefore, the throughput for a Multilink PPP bundle is slightly less than that for an equivalent bandwidth connection that is not using Multilink PPP. If the average packet size is large, the extra Multilink PPP overhead is not readily apparent. However, if the average packet size is small, the extra overhead becomes more noticeable.

Multilink PPP fragmentation adds additional overhead to a packet. Each fragment contains six bytes of Multilink PPP header plus a link encapsulation header. The size of the link encapsulation header varies based on the topology of a network. The Layer 2 headers for Ethernet, ATM, and serial interfaces add different number of bytes to a packet.

Multilink PPP over Serial Interfaces

The Multilink PPP over Serial Interfaces feature enables you to bundle T1 interfaces into a single, logical connection called a Multilink PPP bundle. (See the "Multilink PPP Bundles" section.) The Multilink PPP over Serial Interfaces feature also provides the following functionalities:

- Load balancing—Multilink PPP provides bandwidth on demand and uses load balancing across all member links (up to ten) to transmit packets and packet fragments. Multilink PPP mechanisms calculate the load on inbound or outbound traffic between specific sites. Because Multilink PPP splits packets and fragments across all member links during transmission, Multilink PPP reduces transmission latency across WAN links. Ideally, all member links in a bundle would be of the same bandwidth (for example, T1s). Load balancing and fragmentation and interleaving also allow for a mix of unequal cost member links for situations where a small increment in the bundle bandwidth is required.

- Increased redundancy—Multilink PPP allows traffic to flow over remaining member links when a port fails. When you configure a Multilink PPP bundle that consists of T1 lines from more than one line card and if one line card stops operating, part of the bundle on other line cards continues to operate.

- Link fragmentation and interleaving (LFI)—The Multilink PPP fragmenting mechanism fragments large, nonreal-time packets and sends fragments at the same time over multiple point-to-point links to the same

remote address. Smaller, real-time packets remain intact. The Multilink PPP interleaving mechanism sends real-time packets between fragments of nonreal-time packets, thus reducing real-time packet delay.

The figure below shows a Multilink PPP bundle that consists of T1 interfaces from three T3 interfaces.

Figure 2: Multilink PPP Bundle for Multilink PPP over Serial Interfaces
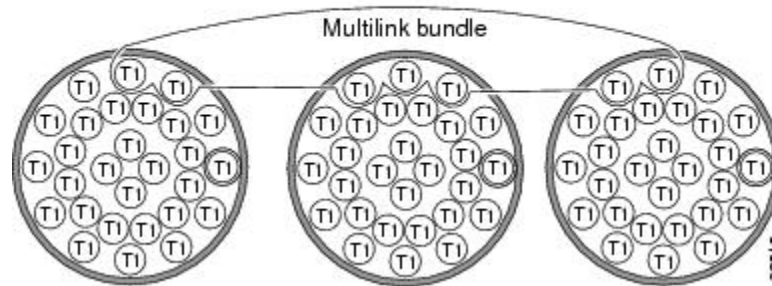
Multilink PPP over Broadband

The Multilink PPP over Broadband feature allows you to combine Multilink PPP over Ethernet (MLPoE) and Multilink PPP over ATM (MLPoA) links into a multilink bundle. Functionally, Multilink PPP over broadband is the same as Multilink PPP over serial interfaces, with the exception of interface management. In Multilink PPP over serial interfaces, interfaces are statically defined in the configuration database, that is, in the startup configuration. In Multilink PPP over broadband, link interfaces are created dynamically by Cisco software while negotiating a PPP session.

The Multilink PPP feature operates in the following two deployment schemes: "PTA mode" and "LNS mode." In the PPP termination and aggregation (PTA) mode, a device acts as the PTA device and terminates Multilink PPP sessions coming from the customer premises equipment (CPE). In the Layer 2 Tunneling Protocol (L2TP) Network Server (LNS) mode, a device terminates Multilink PPP sessions (carried in a Layer 2 tunnel that originates on a L2TP Access Concentrator [LAC] device) coming from the CPE device.

Note Cisco software allows a device to function as a LAC switch. Therefore, in the LNS mode, Cisco software can run on both LAC and LNS devices. A Cisco device cannot act as a CPE device.

Cisco software can terminate Multilink PPP over ATM (MLPoA) ATM adaptation layer 5 (AAL5) multiplexer (MUX) or AAL5 Subnetwork Access Protocol (SNAP), Multilink PPP over Ethernet (MLPoE), Multilink PPP over Ethernet over ATM (MLPoEoA) AAL5 SNAP, Multilink PPP over Ethernet over Queue-in-Queue (MLPoEoQinQ), or Multilink PPP over Ethernet over VLAN (MLPoEoVLAN) sessions acting as the PTA node. In the LNS mode, Cisco software can terminate Multilink PPP over LNS (MLPoLNS) sessions by using Ethernet (Gigabit Ethernet or 10 Gigabit Ethernet) and ATM (AAL5 MUX and AAL5 SNAP) as the LAC-to-LNS connection to the LAC device. Cisco software also provides support to act as a LAC device to switch broadband Multilink PPP sessions between a CPE or Digital Subscriber Line Access Multiplexer (DSLAM) and an LNS device.

PTA Mode

In the PPP termination and aggregation (PTA) mode, Multilink PPP bundles are terminated at the customer premises equipment (CPE) and at the PTA device. A PTA device terminates the PPP or Multilink PPP session and assigns the network layer address to a client. The client data, present as payload in the Multilink PPP packet, is then forwarded to the backbone network. Depending on the access network, PPP or Multilink PPP frames can be transported to the PTA device by using PPP over ATM (PPPoA) or PPP over Ethernet (PPPoE).

The figure below shows the PTA mode.

Figure 3: PTA Mode
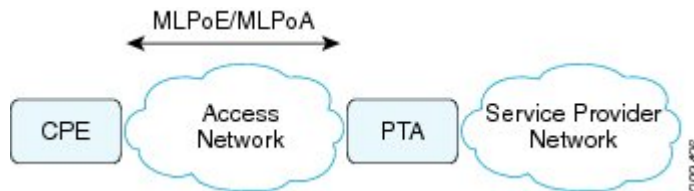
LNS Mode

A customer premises equipment (CPE) provides access to a network through various types of access network topologies, including ISDN, asymmetric digital subscriber line (ADSL), and Fiber to the Home (FTTH). If the network protocol used by the CPE device is IPv4, the IP payload will be carried over PPP or Multilink PPP, and the L2TP Access Concentrator (LAC) device will carry the client PPP or Multilink PPP session data to the Layer 2 Tunneling Protocol (L2TP) Network Server (LNS) device by using Layer 2 Tunneling Protocol Version 2 (L2TPv2). The LNS device terminates the PPP or Multilink PPP session and assigns the client its network layer address. The client data, present as the payload in the IPv4 packet, is then forwarded to the backbone network. Depending on the access network, PPP or Multilink PPP frames can be transported to the LAC device by using PPP over ATM (PPPoA) or PPP over Ethernet (PPPoE).

Note   Because the LAC device performs Layer 2 Forwarding (L2F) between a CPE and an LNS device, the LAC device does not physically terminate the Multilink PPP session; the LAC device forwards the Multilink PPP session traffic by using L2F.

CPE and LAC devices exchange PPP or Multilink PPP packets that are typically encapsulated using the PPPoE protocol and carried over an Ethernet network. The requirements of an LNS device are independent of protocols that are in use between clients and the LAC device. The LAC device creates L2TPv2 tunnels to all LNS devices to which clients need termination. Multiple tunnels can exist between a given LAC-to-LNS pair of devices for load sharing and redundancy considerations.

The figure below shows the LNS mode.

Figure 4: LNS Mode
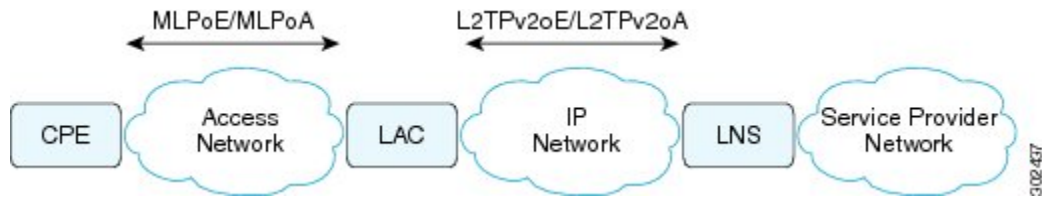
For each PPP or Multilink PPP session that is initiated by the CPE device, the LAC device signals the LNS device to add another session to the tunnel. After a session is set up, all traffic from the client, including PPP control packets, is forwarded to the LNS device. After the decision to negotiate the initial Link Control Protocol (LCP) and forward packets to an LNS device has been made, the LAC device becomes Layer 2 transparent to subsequent packets that are received from the client and the LNS device, including further PPP control messages. PPP or Multilink PPP sessions are terminated at the LNS device, and IP packets from the client are routed on the attached ISP or corporate network towards their final destination. The LNS device performs authentication, authorization, and accounting (AAA) actions on PPP or Multilink PPP sessions.

Note In the case of PPP control packet exchanges at PPP or Multilink PPP session initiation time, the Multilink PPP session is terminated at the CPE device and the LNS device, not at the LAC device. The LAC device does not have the knowledge that this is a Multilink PPP session. As a result, when the client CPE first initiates a PPP-LCP connection, the Multilink (Maximum Receive Reconstructed Unit [MRRU]) negotiation included as part of the LCP negotiation may fail if the LAC device has not yet established the connection with the LNS device (which is typically the case). When a connection between the LAC device and the LNS device is established, the LNS device can renegotiate Multilink (MRRU) LCP options with the CPE device, and the two peer devices can then establish an end-to-end Multilink PPP connection.

Performance- and Scalability-Related Commands

When working in large-scale network configurations with many Multilink PPP (or PPP) subscribers, the traffic load when bringing up a scaled session can overwhelm the network and packets may be dropped. For example, such a scenario may occur after a full device reload or after an outage on a network interface that supports large numbers of users.

To minimize recovery issues due to dropped session establishment packets, we recommend that you configure the following commands on the multilink interface (that is, interface multilink, virtual template, or serial member link interfaces):

```
Device(config)# interface GigabitEthernet 1/0/0
Device(config-if)# ppp max-configure 30
Device(config-if)# ppp max-failure 30
Device(config-if)# ppp timeout retry 5
Device(config-if)# keepalive 30
```
The above commands are the recommended starting points. These values can be configured as required depending on the scale of the network.

Some other issues related to network packet drops, observed when bringing up large-scale networks, include PPP or Multilink PPP sessions coming up but missing adjacencies in IP forwarding tables (that is, IP addresses are not assigned to sessions).

The **keepalive** command listed above affects how quickly a link is flagged as disabled if it is no longer functional. If a Multilink PPP bundle is used with multiple member link sessions and if one or more links are removed, the higher the keepalive interval the longer Multilink PPP will require to detect the disabled link and remove from the list of member links. This delay can cause a period of packet loss and delays on the bundle until the disabled link has been detected. The default keepalive interval, if not specified by the user, is 10 seconds, and the default number of keepalive retries is 4.

Note   When a link is disabled due to a loss of signal, shut down, or due to a major alarm type of condition, the disabled link may be detected outside of the keepalive mechanism and reported as disabled before the keepalive timeout. The keepalive mechanism applies wherever the link is disabled but appears functional at the physical layer.

When configuring Broadband Aggregation (BBA) groups over an Ethernet interface, there are limits that must be adjusted to match the scale of the configuration. The number of broadband sessions can be limited per MAC address, per VLAN, and per virtual circuit (VC). The following example shows how to use the **bba-group pppoe** command to configure a BBA group over Ethernet:

```
Device(config)# bba-group pppoe global
Device(config-bba-group)# sessions per-mac limit 2000
Device(config-bba-group)# sessions per-vlan limit 2000
Device(config-bba-group)# sessions per-vc limit 2000 threshold 2000
```
If these per-session numbers are too low, there will be fewer established sessions between devices.

Multilink PPP over ATM on the PTA Device

Cisco software supports Multilink PPP over ATM (MLPoA) by using single-link and multilink bundles. Cisco software supports MLPoA over ATM adaptation layer 5 (AAL5) multiplexer (MUX) or AAL5 Subnetwork Access Protocol (SNAP).

The network topology and functional support for MLPoA is similar to the Multilink PPP over Ethernet (MLPoE) topology on the PTA device. The difference is that the connection between the PTA device and downstream devices is ATM instead of Ethernet.

Multilink PPP over Ethernet over ATM on the PTA Device

Cisco software supports Multilink PPP over Ethernet over ATM (MLPoEoA) by using single-link and multilink bundles. Cisco software supports MLPoEoA over ATM adaptation layer 5 (AAL5) Subnetwork Access Protocol (SNAP).

The network topology and functional support for MLPoEoA is similar to the Multilink PPP over Ethernet (MLPoE) topology on the PTA device. The difference is that the connection between the PTA device and downstream devices is ATM instead of Ethernet.

Multilink PPP over LNS

Cisco software supports Multilink PPP over LNS (MLPoLNS) by using single-link and multilink bundles. The data link layer for MLPoLNS is Layer 2 Tunneling Protocol Version 2 (L2TPv2). For MLPoLNS, Ethernet is used for transport between L2TP Network Server (LNS) and L2TP Access Concentrator (LAC) devices.

The Multilink PPP over LNS feature bundles one or more virtual private dialup network (VPDN) sessions in a single logical connection, which forms a Multilink PPP bundle on the LNS device. From a Multilink PPP perspective, Multilink PPP over LNS is similar to Multilink PPP over ATM or over Ethernet or over Ethernet

over ATM, except that in this case, Multilink PPP or PPP packets are L2TPv2 encapsulated. The bandwidth of a member link session can be determined on a LAC device by using the *connect speed* attribute-value pair.

The L2TP tunnel between a LAC device and an LNS device carries both PPP and Multilink PPP traffic. The LAC device acts as a switch and forwards both PPP and Multilink PPP session packets between the CPE device and the LNS device. The figure below shows a LAC-to-LNS dialup network.

Figure 5: Dialup LAC to LNS
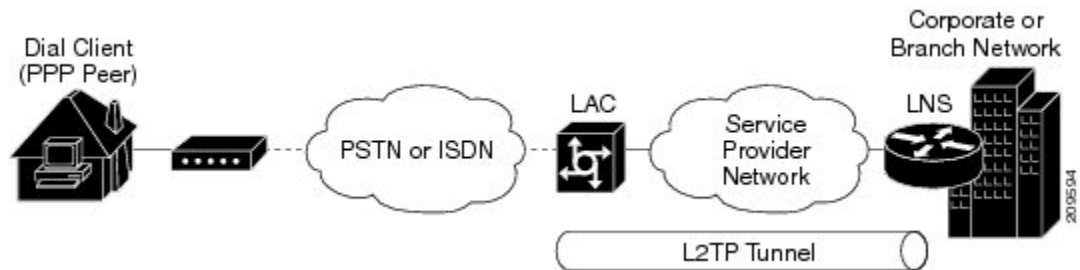
L2TP-client-initiated tunneling allows a client device to initiate L2TP tunnels. A client device can initiate an L2TP tunnel to the LNS device without the intermediate Network Access Server (NAS) participating in tunnel negotiation or establishment. The benefit of L2TP-client-initiated tunneling is that client devices can initiate L2TP tunnels.

The figure below shows an example of a client-initiated tunneling scenario. The client connects to the NAS through a medium, such as a dialup modem, Digital Subscriber Line (DSL), ISDN, or a cable modem, that supports PPP. The client can initiate an L2TP tunnel to the LNS device.

Figure 6: L2TPv2 Tunnel to an LNS Device
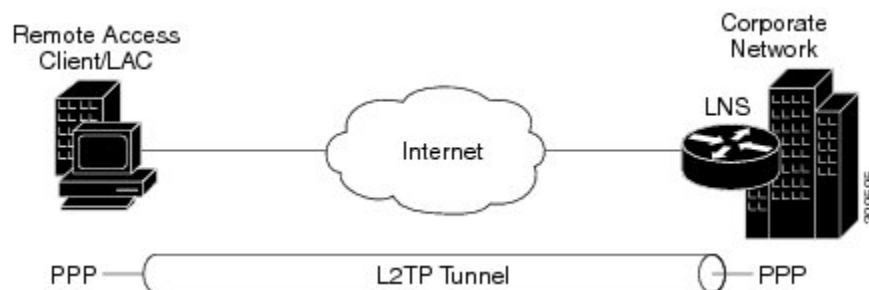
The L2TP Client-Initiated Tunneling feature uses a virtual PPP interface, which adds Layer 2 encapsulation to Layer 3 packets, allowing these packets to be sent to the LNS device over an L2TPv2 tunnel.

Switching Multilink PPP Traffic Through a LAC Device

Cisco software supports the switching of broadband Multilink PPP sessions between customer premises equipment (CPE) and LNS devices through a LAC device.

In the LAC switching mode, the LAC device provides the Multilink PPP over ATM (MLPoATM), Multilink PPP over Ethernet (MLPoE), or Multilink PPP over Ethernet over ATM (MLPoEoA) connection to the CPE device and the virtual private dialup network (VPDN) connection to the LNS device. The LAC-to-CPE connection may be any of the variations supported through MLPoA, MLPoE, or MLPoEoA in the PPP termination and aggregation (PTA) mode.

When switching Multilink PPP traffic through a LAC device, Multilink PPP sessions are not terminated. Multilink PPP sessions are terminated by CPE and LNS devices. The LAC device manages the establishment of the member link VPDN session between the CPE and LNS devices and the switching of the session data. Data from the CPE device is L2TPv2 encapsulated and passed to the LNS device. Data from the LNS device is stripped off of the L2TPv2 data and passed on to the CPE device.

QoS Traffic and Shaping

Quality of service (QoS) is applied to a Multilink PPP bundle by using a service policy. This policy is then applied to the Multilink PPP bundle interface, which appears as a PPP interface.

Note   For Multilink PPP over serial interfaces, the service policy is applied to the multilink interface configuration. For broadband topologies, the service policy is applied to the virtual template configuration.

QoS allows you to manually account for the extra overhead imposed by the external interface and other Layer 2 encapsulations by using the **account** and **user-defined** keywords in the **shape** command. One such example includes ATM cell overhead (extra bits or bytes are added to packets when running traffic on an ATM network). This additional overhead is not accounted for because the extra bytes are removed by the external interface before the QoS policy is applied to the packets. For more information, see the *QoS: Policing and Shaping Configuration Guide*.

How to Configure Multilink PPP Connections for Broadband and Serial Topologies

Configuring Multilink PPP

Before configuring Multilink PPP connections between CPE and PTA devices, you must configure Multilink PPP by performing the following tasks:

Creating a Multilink Bundle

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface multilink** *group-number*
4. **ip address** *address mask*
5. **ppp multilink**
6. **ppp multilink group** *group-number*
7. **end**
8. **show ppp multilink**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface multilink** *group-number* **Example:** `Device(config)# interface multilink 10`	Assigns a multilink bundle group number and enters interface configuration mode.

Step 4	**ip address** *address mask*
	
	Example:
	`Device(config-if)# ip address 192.0.2.9`
	`255.255.255.224`
		Assigns an IP address to the multilink interface.

Step 5	**ppp multilink** **Example:** `Device(config-if)# ppp multilink`	Enables Multilink PPP.
Step 6	**ppp multilink group** *group-number* **Example:** `Device(config-if)# ppp multilink group 12`	Restricts a physical link to join only the designated multilink-group interface.
Step 7	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.
Step 8	**show ppp multilink** **Example:** `Device# show ppp multilink`	Displays Multilink PPP bundle information.

Assigning an Interface to a Multilink Bundle

⚠️

Caution Do not install a route to the peer address while configuring a Multilink PPP lease line. The route can be disabled using the **no ppp peer-neighbor-route** command on the Multilink PPP bundle interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial** *slot*/*subslot*/*port*
4. **no ip address**
5. **encapsulation ppp**
6. **keepalive** *seconds*
7. **ppp multilink**
8. **ppp multilink group** *group-number*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface serial** *slot*/*subslot*/*port* **Example:** `Device(config)# interface serial 1/3/0`	Assigns a multilink bundle group number and enters interface configuration mode.
Step 4	**no ip address** **Example:** `Device(config-if)# no ip address`	Removes any specified IP address.
Step 5	**encapsulation ppp** **Example:** `Device(config-if)# encapsulation ppp`	Enables PPP encapsulation.

	Command or Action	Purpose
Step 6	**keepalive** *seconds* **Example:** `Device(config-if)# keepalive 50`	Sets the frequency of keepalive packets.
Step 7	**ppp multilink** **Example:** `Device(config-if)# ppp multilink`	Enables Multilink PPP.
Step 8	**ppp multilink group** *group-number* **Example:** `Device(config-if)# ppp multilink group 12`	Restricts a physical link to join only the designated multilink-group interface.
Step 9	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Minimum Multilink PPP Links

Perform this task to configure the minimum number of links in a Multilink PPP bundle, which are required to keep that bundle active.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface multilink** *group-number*
4. **ppp multilink**
5. **ppp multilink min-links** *links* **mandatory**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** Device# configure terminal	Enters global configuration mode.
Step 3	**interface multilink** *group-number* **Example:** Device(config)# interface multilink 10	Assigns a multilink bundle group number and enters interface configuration mode.
Step 4	**ppp multilink** **Example:** Device(config-if)# ppp multilink	Enables Multilink PPP.
Step 5	**ppp multilink min-links** *links* **mandatory** **Example:** Device(config-if)# ppp multilink min-links 5 mandatory	Specifies the required minimum number of links in a Multilink PPP bundle. • If the minimum number of links in the Multilink PPP bundle falls below the number specified by the *links* argument, the Multilink PPP bundle is disabled.
Step 6	**end** **Example:** Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Changing the Default Endpoint Discriminator

By default, when the system negotiates the use of Multilink PPP with a peer device, the value that is supplied for the endpoint discriminator is the same as the username used for authentication. The username is configured for the interface by using the **ppp chap hostname** or **ppp pap sent-username** command. If not configured, the username defaults to the globally configured hostname (or stack group name if this interface is a Stack Group Bidding Protocol [SGBP] group member).

Perform this task to override or change the default endpoint discriminator.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface virtual-template** *number*
4. **ppp multilink endpoint** {**hostname** | **ip** *ip-address* | **mac** *lan-interface* | **none** | **phone** *telephone-number* | **string** *char-string*}
5. **end**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Device# configure terminal	Enters global configuration mode.
Step 3	**interface virtual-template** *number* **Example:** Device(config)# interface virtual-template 1	Creates a virtual template interface that can be configured and applied dynamically for creating VAIs and enters interface configuration mode.
Step 4	**ppp multilink endpoint** {**hostname** | **ip** *ip-address* | **mac** *lan-interface* | **none** | **phone** *telephone-number* | **string** *char-string*} **Example:** Device(config-if)# ppp multilink endpoint ip 209.165.201.20	Overrides or changes the default endpoint discriminator that the system uses when negotiating the use of Multilink PPP with a peer.
Step 5	**end** **Example:** Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Multilink PPP Interleaving and Queueing

Multilink PPP support for interleaving can be configured on virtual templates. To configure interleaving, first configure a virtual template and then configure Multilink PPP and interleaving on the interface or template. These tasks are described in the "Configuring Multilink PPP Interleaving" section.

Note   Fair queueing, which is enabled by default, must remain enabled on the interface.

This section covers the following tasks:

Configuring Multilink PPP Interleaving

Interleaving statistics can be displayed by using the **show interfaces** command, specifying the particular interface on which interleaving is enabled. Interleaving data is displayed only if there are interleaves. For example, the following line shows interleaves:

```
Output queue: 315/64/164974/31191 (size/threshold/drops/interleaves)
```

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface virtual-template** *number*
4. **ppp multilink**
5. **ppp multilink interleave**
6. **ppp multilink fragment delay** *milliseconds*
7. **end**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface virtual-template** *number* **Example:** `Device(config)# interface virtual-template 1`	Creates a virtual template interface that can be configured and applied dynamically for creating VAIs and enters interface configuration mode.
Step 4	**ppp multilink** **Example:** `Device(config-if)# ppp multilink`	Enables Multilink PPP.
Step 5	**ppp multilink interleave** **Example:** `Device(config-if)# ppp multilink interleave`	Enables the interleaving of packets among fragments of larger packets on a Multilink PPP bundle.

	Command or Action	Purpose
Step 6	**ppp multilink fragment delay** *milliseconds* **Example:** Device(config-if)# ppp multilink fragment delay 50	Specifies the maximum size, in units of time, for packet fragments on a Multilink PPP bundle.
Step 7	**end** **Example:** Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Disabling PPP Multilink Fragmentation

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface multilink** *group-number*
4. **ppp multilink fragment disable**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Device# configure terminal	Enters global configuration mode.
Step 3	**interface multilink** *group-number* **Example:** Device(config)# interface multilink 10	Assigns a multilink group number and enters interface configuration mode.

	Command or Action	Purpose

Step 4	**ppp multilink fragment disable**
	
	Example:
	`Device(config-if)# ppp multilink fragment`
	`disable`
		Disables PPP multilink fragmentation.

Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Multilink PPP over Broadband

The following sections provide information about configuring Multilink PPP connections between CPE and PTA devices for MLPoA, MLPoE, MLPoEoA, and MLPoLNS:

Creating a Class Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** *class-map-name* [**match-all** | **match-any**]
4. **match ip precedence** *precedence-criteria*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	**Purpose**
Step 3	**class-map** *class-map-name* [**match-all** | **match-any**] **Example:** `Device(config)# class-map ip-prec-1 match-all`	Creates a class map to be used for matching traffic with the specified class and enters QoS class-map configuration mode.
Step 4	**match ip precedence** *precedence-criteria* **Example:** `Device(config-cmap)# match ip precedence 1`	(Optional) Specifies the match criteria in a class map.
Step 5	**end** **Example:** `Device(config-cmap)# end`	Exits QoS class-map configuration mode and returns to privileged EXEC mode.

Creating a Policy Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** {*class-name* \| **class-default**}
5. **priority percent** *percentage*
6. **shape** {**average** \| **peak**} *mean-rate*
7. **service-policy** *policy-map*
8. **end**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	**Purpose**

Step 3	**policy-map** *policy-map-name*
	
	Example:
	`Device(config)# policy-map`
	`mlp-parent-policy-10m`
		Specifies the name of the policy map to be created and enters QoS policy-map configuration mode.

Step 4	**class** {*class-name* | **class-default**} **Example:** `Device(config-pmap)# class ip-prec-1`	Specifies the name of the class and enters QoS policy-map class configuration mode. • This class is associated with the class map created in the "Creating a Class Map" section.
Step 5	**priority percent** *percentage* **Example:** `Device(config-pmap-c)# priority percent 10`	(Optional) Specifies the percentage of the total available bandwidth to be set aside for the priority class.
Step 6	**shape** {**average** | **peak**} *mean-rate* **Example:** `Device(config-pmap-c)# shape average 250000`	Shapes traffic to the indicated bit rate according to the algorithm specified.

Step 7	**service-policy** *policy-map*
	
	Example:
	`Device(config-pmap-c)# service-policy`
	`mlp-parent-policy-10m`
		Attaches a policy map to a class.

Step 8	**end** **Example:** `Device(config-pmap-c)# end`	Exits QoS policy-map class configuration mode and returns to privileged EXEC mode.

Defining a PPP over Ethernet Profile

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bba-group pppoe** {*group-name* \| **global**}
4. **virtual-template** *template-number*
5. **end**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**bba-group pppoe** {*group-name* | **global**} **Example:** `Device(config)# bba-group pppoe mlpppoe-bba-group-10m`	Defines a PPPoE profile and enters BBA group configuration mode.
Step 4	**virtual-template** *template-number* **Example:** `Device(config-bba-group)# virtual-template 18`	Specifies the virtual template to be used to clone VAIs for all PPPoE ports that use this PPPoE profile.
Step 5	**end** **Example:** `Device(config-bba-group)# end`	Exits BBA group configuration mode and returns to privileged EXEC mode.

Configuring a Virtual Template Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface virtual-template** *number*
4. **peer default ip address pool** [*pool-name*]
5. **ppp multilink**
6. **ppp multilink interleave**
7. **ppp multilink endpoint string** *char-string*
8. **ppp multilink retry** *seconds*
9. **service-policy output** *policy-map-name*
10. **end**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Device# configure terminal	Enters global configuration mode.
Step 3	**interface virtual-template** *number* **Example:** Device(config)# interface virtual-template 1	Creates a virtual template interface that can be configured and applied dynamically for creating VAIs and enters interface configuration mode.
Step 4	**peer default ip address pool** [*pool-name*] **Example:** Device(config-if)# peer default ip address pool MLP-IPv4-Pool	Specifies an IP address from an IP address pool to be returned to a remote peer that is connected to this interface.
Step 5	**ppp multilink** **Example:** Device(config-if)# ppp multilink	Enables Multilink PPP.
Step 6	**ppp multilink interleave** **Example:** Device(config-if)# ppp multilink interleave	Enables interleaving of packets among fragments of larger packets on a multilink bundle.
Step 7	**ppp multilink endpoint string** *char-string* **Example:** Device(config-if)# ppp multilink endpoint string Dialer2-CPE-MLPoE	Overrides or changes the default endpoint discriminator that the system uses when negotiating the use of Multilink PPP with a peer.
Step 8	**ppp multilink retry** *seconds* **Example:** Device(config-if)# ppp multilink retry 4	Sets the maximum waiting period for a response during PPP negotiation.
Step 9	**service-policy output** *policy-map-name* **Example:** Device(config-if)# service-policy output mlp-parent-10m	Attaches the previously created traffic policy (policy map). • The policy map evaluates and applies QoS features for traffic leaving the interface.

	Command or Action	Purpose
Step 10	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Multilink PPP over ATM on the CPE Device

Before configuring Multilink PPP over ATM, you must complete the following tasks:

- Creating a Class Map
- Creating a Policy Map
- Creating a Dialer Interface

This section covers the following tasks:

Configuring Multilink PPP over ATM Using AAL5 MUX Encapsulation

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *interface-number*[**.***subinterface-number* {**mpls** | **multipoint** | **point-to-point**}]
4. **pvc** *vpi/vci*
5. **vbr-nrt** *output-pcr output-scr output-maxburstsize*
6. **encapsulation aal5mux** *protocol*
7. **dialer pool-number** *number*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	**Purpose**
Step 3	**interface atm** *interface-number*[*.subinterface-number* {**mpls** | **multipoint** | **point-to-point**}] **Example:** Device(config)# interface atm 6/0.20 point-to-point	Specifies the ATM interface for which Multilink PPP must be configured and enters interface configuration mode.
Step 4	**pvc** *vpi/vci* **Example:** Device(config-if)# pvc 20/10	Specifies the encapsulation type on an ATM permanent virtual circuit (PVC) and enters ATM virtual circuit configuration mode.
Step 5	**vbr-nrt** *output-pcr output-scr output-maxburstsize* **Example:** Device(config-if-atm-vc)# vbr-nrt 512 256 20	Configures the variable bit rate-nonreal time (VBR-NRT) and specifies output peak cell rate (PCR), output sustainable cell rate (SCR), and output maximum burst cell size for an ATM PVC.
Step 6	**encapsulation aal5mux** *protocol* **Example:** Device(config-if-atm-vc)# encapsulation aal5mux ppp dialer	Configures the ATM adaptation layer (AAL) and encapsulation type for an ATM virtual circuit (VC).
Step 7	**dialer pool-number** *number* **Example:** Device(config-if-atm-vc)# dialer pool-number 4	Configures a physical interface as a member of a dialer profile dialing pool.
Step 8	**end** **Example:** Device(config-if-atm-vc)# end	Exits ATM virtual circuit configuration mode and returns to privileged EXEC mode.

Configuring Multilink PPP over ATM Using AAL5 SNAP Encapsulation

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *interface-number*[*.subinterface-number* {**mpls** | **multipoint** | **point-to-point**}]
4. **pvc** *vpi*/*vci*
5. **vbr-nrt** *output-pcr output-scr output-maxburstsize*
6. **encapsulation aal5snap**
7. **protocol ppp dialer**
8. **dialer pool-number** *number*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Device# configure terminal	Enters global configuration mode.
Step 3	**interface atm** *interface-number*[*.subinterface-number* {**mpls**	**multipoint**	**point-to-point**}] **Example:** Device(config)# interface atm 6/0.20 point-to-point	Specifies the ATM interface for which Multilink PPP must be configured and enters interface configuration mode.
Step 4	**pvc** *vpi*/*vci* **Example:** Device(config-if)# pvc 20/10	Specifies the encapsulation type on an ATM permanent virtual circuit (PVC) and enters ATM virtual circuit configuration mode.
Step 5	**vbr-nrt** *output-pcr output-scr output-maxburstsize* **Example:** Device(config-if-atm-vc)# vbr-nrt 512 256 20	Configures the variable bit rate-nonreal time (VBR-NRT) and specifies output peak cell rate (PCR), output sustainable cell rate (SCR), and output maximum burst cell size for an ATM PVC.

	Command or Action	**Purpose**
Step 6	**encapsulation aal5snap** **Example:** Device(config-if-atm-vc)# encapsulation aal5snap	Configures the ATM adaptation layer (AAL) and encapsulation type for an ATM virtual circuit (VC).
Step 7	**protocol ppp dialer** **Example:** Device(config-if-atm-vc)# protocol ppp dialer	Configures a static map for an ATM PVC.
Step 8	**dialer pool-number** *number* **Example:** Device(config-if-atm-vc)# dialer pool-number 4	Configures a physical interface as a member of a dialer profile dialing pool.
Step 9	**end** **Example:** Device(config-if-atm-vc)# end	Exits ATM virtual circuit configuration mode and returns to privileged EXEC mode.

Configuring Multilink PPP over Ethernet over ATM at the CPE

Before You Begin

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *interface-number*[.*subinterface-number* {**mpls** | **multipoint** | **point-to-point**}]
4. **pvc** *vpi/vci*
5. **vbr-nrt** *output-pcr output-scr output-maxburstsize*
6. **pppoe-client dial-pool-number** *number*
7. **end**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	**Purpose**
Step 2	**configure terminal** **Example:** Device# configure terminal	Enters global configuration mode.
Step 3	**interface atm** *interface-number*[.*subinterface-number* {**mpls** | **multipoint** | **point-to-point**}] **Example:** Device(config)# interface atm 6/0.20 point-to-point	Specifies the ATM interface for which Multilink PPP must be configured and enters interface configuration mode.
Step 4	**pvc** *vpi*/*vci* **Example:** Device(config-if)# pvc 20/10	Specifies the encapsulation type on an ATM permanent virtual circuit (PVC) and enters ATM virtual circuit configuration mode.
Step 5	**vbr-nrt** *output-pcr* *output-scr* *output-maxburstsize* **Example:** Device(config-if-atm-vc)# vbr-nrt 512 256 20	Configures the variable bit rate-nonreal time (VBR-NRT) and specifies output peak cell rate (PCR), output sustainable cell rate (SCR), and output maximum burst cell size for an ATM PVC.
Step 6	**pppoe-client dial-pool-number** *number* **Example:** Device(config-if-atm-vc)# pppoe-client dial-pool-number 6	Configures a PPPoE client.
Step 7	**end** **Example:** Device(config-if-atm-vc)# end	Exits ATM virtual circuit configuration mode and returns to privileged EXEC mode.

Configuring Multilink PPP over ATM on the PTA Device

Before configuring Multilink PPP over ATM, you must complete the following tasks:

- Creating a Class Map
- Creating a Policy Map
- Defining a PPP over Ethernet Profile
- Configuring a Virtual Template Interface

This section covers the following tasks:

Configuring Multilink PPP over ATM Using AAL5 MUX Encapsulation

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *interface-number*[*.subinterface-number* {**mpls** | **multipoint** | **point-to-point**}]
4. **pvc** *vpi/vci*
5. **vbr-nrt** *output-pcr output-scr output-maxburstsize*
6. **encapsulation aal5mux** *protocol* **virtual-template** *virtual-template*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface atm** *interface-number*[*.subinterface-number* {**mpls**	**multipoint**	**point-to-point**}] **Example:** `Device(config)# interface atm 2/2/1.20 point-to-point`	Specifies the ATM interface for which Multilink PPP must be configured and enters interface configuration mode.
Step 4	**pvc** *vpi/vci* **Example:** `Device(config-if)# pvc 20/10`	Specifies the encapsulation type on an ATM PVC and enters ATM virtual circuit configuration mode.
Step 5	**vbr-nrt** *output-pcr output-scr output-maxburstsize* **Example:** `Device(config-if-atm-vc)# vbr-nrt 512 256 20`	Configures the variable bit rate-nonreal time (VBR-NRT) and specifies output peak cell rate (PCR), output sustainable cell rate (SCR), and output maximum burst cell size for an ATM PVC.
Step 6	**encapsulation aal5mux** *protocol* **virtual-template** *virtual-template* **Example:** `Device(config-if-atm-vc)# encapsulation aal5mux ppp virtual-template 18`	Configures the ATM adaptation layer (AAL) and encapsulation type for an ATM virtual circuit (VC) and assigns the VC to the previously specified virtual template.

	Command or Action	Purpose
Step 7	**end** **Example:** Device(config-if-atm-vc)# end	Exits ATM virtual circuit configuration mode returns to privileged EXEC mode.

Configuring Multilink PPP over ATM Using AAL5 SNAP Encapsulation

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *interface-number*[*.subinterface-number* {**mpls** | **multipoint** | **point-to-point**}]
4. **pvc** *vpi*/*vci*
5. **vbr-nrt** *output-pcr output-scr output-maxburstsize*
6. **encapsulation aal5snap**
7. **protocol** *protocol* **virtual-template** *virtual-template*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Device# configure terminal	Enters global configuration mode.
Step 3	**interface atm** *interface-number*[*.subinterface-number* {**mpls**	**multipoint**	**point-to-point**}] **Example:** Device(config)# interface atm 2/2/1.22 point-to-point	Specifies the ATM interface for which Multilink PPP must be configured and enters interface configuration mode.
Step 4	**pvc** *vpi*/*vci* **Example:** Device(config-if)# pvc 20/10	Specifies the encapsulation type on an ATM PVC and enters ATM virtual circuit configuration mode.

	Command or Action	Purpose
Step 5	**vbr-nrt** *output-pcr output-scr output-maxburstsize* **Example:** `Device(config-if-atm-vc)# vbr-nrt 512 256 20`	Configures the variable bit rate-nonreal time (VBR-NRT) and specifies output peak cell rate (PCR), output sustainable cell rate (SCR), and output maximum burst cell size for an ATM PVC.
Step 6	**encapsulation aal5snap** **Example:** `Device(config-if-atm-vc)# encapsulation aal5snap`	Configures the ATM adaptation layer (AAL) and encapsulation type for an ATM virtual circuit (VC).
Step 7	**protocol** *protocol* **virtual-template** *virtual-template* **Example:** `Device(config-if-atm-vc)# protocol ppp virtual-template 18`	Specifies parameters that PPPoA sessions use.
Step 8	**end** **Example:** `Device(config-if-atm-vc)# end`	Exits ATM virtual circuit configuration mode and returns to privileged EXEC mode.

Configuring Multilink PPP over Ethernet over ATM on the PTA Device

Before configuring Multilink PPP over Ethernet over ATM, you must complete the following tasks:

- Creating a Class Map
- Creating a Policy Map
- Defining a PPP over Ethernet Profile
- Configuring a Virtual Template Interface

This section covers the following tasks:

Configuring a Virtual Circuit Class

Before configuring MLPoEoA, you must create and configure a virtual circuit (VC) class.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vc-class atm** *name*
4. **protocol pppoe** [**group** *group-name*]
5. **vbr-nrt** *output-pcr output-scr output-maxburstsize*
6. **encapsulation aal5snap**
7. **create-on demand**
8. **idle-timeout** *minutes*
9. **end**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vc-class atm** *name* **Example:** `Device(config)# vc-class atm mlpoeoa-vc-class`	Creates a VC class and enters VC-class configuration mode.

Step 4	**protocol pppoe** [**group** *group-name*]
	
	Example:
	`Device(config-vc-class)# protocol pppoe group`
	`mlpoeoa-bba-group`
		Enables PPP over Ethernet (PPPoE) sessions to be established on permanent virtual circuits (PVCs).

Step 5	**vbr-nrt** *output-pcr output-scr output-maxburstsize* **Example:** `Device(config-vc-class)# vbr-nrt 512 256 20`	Configures the variable bit rate-nonreal time (VBR-NRT) and specifies the output peak cell rate (PCR), output sustainable cell rate (SCR), and output maximum burst cell size for an ATM PVC.

Step 6	**encapsulation aal5snap**
	
	Example:
	`Device(config-vc-class)# encapsulation`
	`aal5snap`
		Configures the ATM adaptation layer (AAL) and encapsulation type for an ATM VC.

	Command or Action	Purpose
Step 7	**create-on demand** **Example:** `Device (config-vc-class)# create-on demand`	Configures ATM PVC autoprovisioning, which enables a PVC or range of PVCs to be created automatically on demand.
Step 8	**idle-timeout** *minutes* **Example:** `Device(config-vc-class)# idle-timeout 10`	Sets a time to keep the session alive in the absence of any data traffic.
Step 9	**end** **Example:** `Device(config-vc-class)# end`	Exits VC-class configuration mode and returns to privileged EXEC mode.

Configuring Multilink PPP over Ethernet over ATM Using AAL5 SNAP Encapsulation

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm** *interface-number*[*.subinterface-number* {**mpls** | **multipoint** | **point-to-point**}]
4. **pvc** *vpi/vci*
5. **class-vc** *name*
6. **vbr-nrt** *output-pcr output-scr output-maxburstsize*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface atm** *interface-number*[**.***subinterface-number* {**mpls** | **multipoint** | **point-to-point**}] **Example:** `Router (config)# interface atm 2/2/1.24 point-to-point`	Specifies the ATM interface for which Multilink PPP must be configured and enters interface configuration mode.
Step 4	**pvc** *vpi*/*vci* **Example:** `Device(config-if)# pvc 20/10`	Specifies the encapsulation type on an ATM permanent virtual circuit (PVC) and enters ATM virtual circuit configuration mode.
Step 5	**class-vc** *name* **Example:** `Device(config-if-atm-vc)# class-vc mlpppoeoa-vc-class`	Assigns a virtual circuit (VC) class to an ATM PVC.
Step 6	**vbr-nrt** *output-pcr* *output-scr* *output-maxburstsize* **Example:** `Device(config-if-atm-vc)# vbr-nrt 512 256 20`	Configures the variable bit rate-nonreal time (VBR-NRT) and specifies the output peak cell rate (PCR), output sustainable cell rate (SCR), and output maximum burst cell size for an ATM PVC.
Step 7	**end** **Example:** `Device(config-if-atm-vc)# end`	Exits ATM virtual circuit configuration mode and returns to privileged EXEC mode.

Configuring Multilink PPP over LNS

Configuring an LNS Device to Initiate and Receive L2TP Traffic

Before performing this task, you must configure the virtual template interface as described in the "Configuring a Virtual Template Interface" section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vpdn enable**
4. **vpdn-group** *group-name*
5. **accept-dialin**
6. **protocol l2tp**
7. **virtual-template** *template-number*
8. **exit**
9. **terminate-from hostname** *hostname*
10. **lcp renegotiation** {**always** | **on-mismatch**}
11. **no l2tp authentication**
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vpdn enable** **Example:** `Device(config)# vpdn enable`	Enables virtual private dialup network (VPDN) on the device. • The device looks for tunnel definitions in a local database and on a remote authorization server (home gateway) if one is present.
Step 4	**vpdn-group** *group-name* **Example:** `Device(config)# vpdn-group group1`	Defines a local group name for which you can assign other VPDN variables and enters VPDN group configuration mode.
Step 5	**accept-dialin** **Example:** `Device(config-vpdn)# accept-dialin`	Configures the LNS device to accept tunneled PPP connections from the LAC device, creates an accept-dialin VPDN subgroup, and enters VPDN accept-dialin group configuration mode.

	Command or Action	**Purpose**
Step 6	**protocol l2tp** **Example:** `Device(config-vpdn-acc-in)# protocol l2tp`	Specifies the Layer 2 Tunnel Protocol (L2TP).
Step 7	**virtual-template** *template-number* **Example:** `Device(config-vpdn-acc-in)# virtual-template 18`	Specifies the virtual template to be used to clone virtual access interfaces (VAIs).
Step 8	**exit** **Example:** `Device(config-vpdn-acc-in)# exit`	Returns to VPDN group configuration mode.
Step 9	**terminate-from hostname** *hostname* **Example:** `Device(config-vpdn)# terminate-from hostname LAC3`	Specifies the hostname of the remote LAC device that will be required when accepting a VPDN tunnel.
Step 10	**lcp renegotiation** {**always** | **on-mismatch**} **Example:** `Device(config-vpdn)# lcp renegotiation always`	Allows the LNS device to renegotiate the PPP Link Control Protocol (LCP).
Step 11	**no l2tp authentication** **Example:** `Device(config-vpdn)# no l2tp authentication`	Disables L2TP tunnel authentication.
Step 12	**end** **Example:** `Device(config-vpdn)# end`	Returns to privileged EXEC mode.

Configuring a LAC Device to Initiate and Receive L2TP Traffic

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **vpdn enable**
4. **vpdn-group** *group-name*
5. **request-dialin**
6. **protocol l2tp**
7. **exit**
8. **initiate-to ip** *ip-address* [**priority** *priority-number*]
9. **local name** *hostname*
10. **no l2tp authentication**
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**vpdn enable** **Example:** `Router(config)# vpdn enable`	Enables virtual private dialup network (VPDN) on the device.
Step 4	**vpdn-group** *group-name* **Example:** `Device(config)# vpdn-group group1`	Defines a local group name for which you can assign other VPDN variables and enters VPDN group configuration mode.
Step 5	**request-dialin** **Example:** `Device(config-vpdn)# request-dialin`	Enables the LAC device to request Layer 2 Tunneling Protocol (L2TP) tunnels and enters VPDN request-dialin group configuration mode.

	Command or Action	**Purpose**
Step 6	**protocol l2tp** **Example:** Device(config-vpdn-req-in)# protocol l2tp	Specifies the L2TP protocol.
Step 7	**exit** **Example:** Device(config-vpdn-req-in)# exit	Returns to VPDN group configuration mode.
Step 8	**initiate-to ip** *ip-address* [**priority** *priority-number*] **Example:** Device(config-vpdn)# initiate-to ip 10.0.0.4	Specifies the LNS IP address and optionally, the priority of the IP address (1 is the highest).
Step 9	**local name** *hostname* **Example:** Device(config-vpdn)# local name LAC3	Specifies a local hostname that the tunnel will use to identify itself.
Step 10	**no l2tp authentication** **Example:** Device(config-vpdn)# no l2tp authentication	Disables L2TP tunnel authentication.
Step 11	**end** **Example:** Device(config-vpdn)# end	Exits VPDN group configuration mode and returns to privileged EXEC mode.

Configuring Multilink PPP over Serial Interfaces

To configure Multilink PPP over serial interfaces, configure serial interfaces to support PPP encapsulation and Multilink PPP. Repeat the steps below for as many serial interfaces as required.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial** *number*
4. **no ip address**
5. **encapsulation ppp**
6. **ppp multilink**
7. **end**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** Device# configure terminal	Enters global configuration mode.
Step 3	**interface serial** *number* **Example:** Device(config)# interface serial 1	Specifies an asynchronous interface and enters interface configuration mode.
Step 4	**no ip address** **Example:** Device(config-if)# no ip address	Removes any specified IP address.
Step 5	**encapsulation ppp** **Example:** Device(config-if)# encapsulation ppp	Enables PPP encapsulation.
Step 6	**ppp multilink** **Example:** Device(config-if)# ppp multilink	Enables Multilink PPP.
Step 7	**end** **Example:** Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for Multilink PPP Connections for Broadband and Serial Topologies

Example: Configuring Multilink PPP

Multilink PPP provides characteristics most similar to hardware inverse multiplexers with good manageability and Layer 3 services support. The following example shows the configuration of Multilink PPP with traffic shaping and QoS. In this example, two bundles with four links each are configured between two devices. The **ppp chap hostname** command entries are required for originating and terminating multiple bundles on a single pair of devices.

```
controller T3 0/3/1
 framing c-bit
 cablelength 224
 t1 1 channel-group 0 timeslots 1-24
 t1 2 channel-group 0 timeslots 1-24
 t1 3 channel-group 0 timeslots 1-24
 t1 4 channel-group 0 timeslots 1-24
 t1 5 channel-group 0 timeslots 1-24
 t1 6 channel-group 0 timeslots 1-24
 t1 7 channel-group 0 timeslots 1-24
 t1 8 channel-group 0 timeslots 1-24
!
class-map match-all DETERMINISTICOUT
  match ip precedence 3
class-map match-all VOICEVIDEOCONTROLOUT
  match ip precedence 2
class-map match-all VOICEOUT
  match ip precedence 1
class-map match-all ROUTINGPROTOCOLS
  match ip precedence 5
class-map match-all CONTROLLEDLOADOUT
  match ip precedence 4
!
policy-map QOS304QCHILD
 class VOICEOUT
    priority level 1
    police cir percent 30
 class VOICEVIDEOCONTROLOUT
    priority level 2
    police cir percent 5
 class DETERMINISTICOUT
    bandwidth remaining ratio 20
 class CONTROLLEDLOADOUT
    bandwidth remaining ratio 18
 class ROUTINGPROTOCOLS
    bandwidth remaining ratio 4
 class class-default
    bandwidth remaining ratio 22
policy-map ASRMLP6MBPARENT
 class class-default
    shape average percent 98
    service-policy QOS304QCHILD
!
interface multilink 1
 ip address 192.168.1.1 255.255.255.0
 ppp chap hostname multilink_name-1
 ppp multilink
 ppp multilink group 1
 service-policy output ASRMLP6MBPARENT
!
interface multilink 2
 ip address 192.168.2.1 255.255.255.0
```

```
 ppp chap hostname multilink_name-2
 ppp multilink
 ppp multilink group 2
 service-policy output ASRMLP6MBPARENT
!
interface serial 0/3/1/1:0
 no ip address
 encapsulation ppp
 no keepalive
 ppp chap hostname multilink_name-1
 ppp multilink
 ppp multilink group 1
!
interface serial 0/3/1/2:0
 no ip address
 encapsulation ppp
 no keepalive
 ppp chap hostname multilink_name-1
 ppp multilink
 ppp multilink group 1
!
interface serial 0/3/1/3:0
 no ip address
 encapsulation ppp
 no keepalive
 ppp chap hostname multilink_name-1
 ppp multilink
 ppp multilink group 1
!
interface serial 0/3/1/4:0
 no ip address
 encapsulation ppp
 no keepalive
 ppp chap hostname multilink_name-1
 ppp multilink
 ppp multilink group 1
!
interface serial 0/3/1/5:0
 no ip address
 encapsulation ppp
 no keepalive
 ppp chap hostname multilink_name-2
 ppp multilink
 ppp multilink group 2
!
interface serial 0/3/1/6:0
 no ip address
 encapsulation ppp
 no keepalive
 ppp chap hostname multilink_name-2
 ppp multilink
 ppp multilink group 2
!
interface serial 0/3/1/7:0
 no ip address
 encapsulation ppp
 no keepalive
 ppp chap hostname multilink_name-2
 ppp multilink
 ppp multilink group 2
!
interface serial 0/3/1/8:0
 no ip address
 encapsulation ppp
 no keepalive
 ppp chap hostname multilink_name-2
 ppp multilink
 ppp multilink group 2
!
```

Example: Configuring Multilink PPP over ATM on the PTA Device

Example: Configuring Multilink PPP over ATM Using AAL5 MUX Encapsulation

The following example shows how to configure Multilink PPP over ATM (MLPoA) with AAL5 MUX encapsulation on the PTA device:

```
class-map match-all ip-prec-1
 match ip precedence 1
policy-map mlp-child-lfi-policy
 class ip-prec-1
  priority percent 10
policy-map mlp-parent-250K
 class class-default
  shape average 250000
  service-policy mlp-child-lfi-policy
policy-map mlp-parent-10M
 class class-default
  shape average 10000000
  service-policy mlp-child-lfi-policy
interface virtual-template 18
 description MLPoEoA/MLPoA aal5mux/aal5snap (single-link bundle) Virtual Template
 ip address negotiated
 peer default ip address pool MLP-IPv4-Pool
 ppp max-failure 30
 ppp chap password 0 password1
 ppp multilink
 ppp multilink interleave
 ppp multilink endpoint magic-number
 ppp timeout retry 4
 service-policy output mlp-parent-250K
bba-group pppoe mlpoeoa-bba-group-250K
 virtual-template 18
ip local pool MLP-IPv4-Pool 209.165.201.2 209.165.201.10
ip forward-protocol nd
no ip http server
no ip http secure-server
ip route 10.0.0.0 255.0.0.0 172.16.0.1
ip route 192.168.0.1 255.255.255.255 198.51.100.1
ip route 192.168.0.2 255.255.255.255 198.51.100.2
ip route 192.168.0.3 255.255.255.255 198.51.100.3
ip route 192.168.0.4 255.255.255.255 198.51.100.4
ip route 192.168.0.5 255.255.255.255 198.51.100.5
ip route 192.168.0.6 255.255.255.255 198.51.100.6
ip route vrf Mgmt-intf 10.0.0.0 255.0.0.0 192.168.0.1
ip route vrf Mgmt-intf 172.16.0.0 255.0.0.0 192.168.0.1
ip route vrf Mgmt-intf 209.165.202.128 255.0.0.0 192.168.0.1
ip route vrf Mgmt-intf 192.168.0.0 255.255.240.0 192.168.0.1
interface atm 2/2/1.20 point-to-point
 description MLPoA aal5mux (Single-Link Bundles) Session (PTA Mode)
 no atm enable-ilmi-trap
 pvc 20/10
  vbr-nrt 512 256 1
  encapsulation aal5mux ppp Virtual-Template18
```

Example: Configuring Multilink PPP over ATM Using AAL5 SNAP Encapsulation

The following example shows how to configure Multilink PPP over ATM (MLPoA) with AAL5 SNAP encapsulation on the PTA device:

```
class-map match-all ip-prec-1
 match ip precedence 1
policy-map mlp-child-lfi-policy
 class ip-prec-1
  priority percent 10
```

```
policy-map mlp-parent-250K
 class class-default
  shape average 250000
  service-policy mlp-child-lfi-policy
policy-map mlp-parent-10M
 class class-default
  shape average 10000000
  service-policy mlp-child-lfi-policy
interface virtual-template 18
 description MLPoEoA/MLPoA aal5mux/aal5snap (single-link bundle) Virtual Template
 ip address negotiated
 peer default ip address pool MLP-IPv4-Pool
 ppp max-failure 30
 ppp chap password 0 password1
 ppp multilink
 ppp multilink interleave
 ppp multilink endpoint magic-number
 ppp timeout retry 4
 service-policy output mlp-parent-250K
bba-group pppoe mlpoeoa-bba-group-250K
 virtual-template 18
ip local pool MLP-IPv4-Pool 209.165.201.2 209.165.201.10
ip forward-protocol nd
no ip http server
no ip http secure-server
ip route 10.0.0.0 255.0.0.0 172.16.0.1
ip route 192.168.0.1 255.255.255.255 198.51.100.1
ip route 192.168.0.2 255.255.255.255 198.51.100.2
ip route 192.168.0.3 255.255.255.255 198.51.100.3
ip route 192.168.0.4 255.255.255.255 198.51.100.4
ip route 192.168.0.5 255.255.255.255 198.51.100.5
ip route 192.168.0.6 255.255.255.255 198.51.100.6
ip route vrf Mgmt-intf 10.0.0.0 255.0.0.0 192.168.0.1
ip route vrf Mgmt-intf 172.16.0.0 255.0.0.0 192.168.0.1
ip route vrf Mgmt-intf 209.165.202.128 255.0.0.0 192.168.0.1
ip route vrf Mgmt-intf 192.168.0.0 255.255.240.0 192.168.0.1
interface atm 2/2/1.22 point-to-point
 description MLPoA aal5snap (Single-Link Bundles) Session (PTA Mode)
 no atm enable-ilmi-trap
 pvc 20/14
  vbr-nrt 512 256 20
  encapsulation aal5snap
  protocol ppp Virtual-Template18
```

Example: Configuring Multilink PPP over Ethernet over ATM on the PTA Device

The following example shows how to configure Multilink PPP over Ethernet over ATM (MLPoEoA) with AAL5 SNAP encapsulation on the PTA device:

```
class-map match-all ip-prec-1
 match ip precedence 1
policy-map mlp-child-lfi-policy
 class ip-prec-1
  priority percent 10
policy-map mlp-parent-250K
 class class-default
  shape average 250000
  service-policy mlp-child-lfi-policy
policy-map mlp-parent-10M
 class class-default
  shape average 10000000
  service-policy mlp-child-lfi-policy
interface virtual-template 18
 description MLPoEoA/MLPoA aal5mux/aal5snap (single-link bundle) Virtual Template
 ip address negotiated
 peer default ip address pool MLP-IPv4-Pool
 ppp max-failure 30
 ppp chap password 0 password1
 ppp multilink
 ppp multilink interleave
```

```
 ppp multilink endpoint magic-number
 ppp timeout retry 4
 service-policy output mlp-parent-250K
bba-group pppoe mlpoeoa-bba-group-250K
 virtual-template 18
vc-class atm mlpoeoa-vc-class-250K
  protocol pppoe group mlpoeoa-bba-group-250K
  vbr-nrt 512 256 20
  encapsulation aal5snap
  create on-demand
  idle-timeout 30
ip local pool MLP-IPv4-Pool 209.165.201.2 209.165.201.10
ip forward-protocol nd
no ip http server
no ip http secure-server
ip route 10.0.0.0 255.0.0.0 172.16.0.1
ip route 192.168.0.1 255.255.255.255 198.51.100.1
ip route 192.168.0.2 255.255.255.255 198.51.100.2
ip route 192.168.0.3 255.255.255.255 198.51.100.3
ip route 192.168.0.4 255.255.255.255 198.51.100.4
ip route 192.168.0.5 255.255.255.255 198.51.100.5
ip route 192.168.0.6 255.255.255.255 198.51.100.6
ip route vrf Mgmt-intf 10.0.0.0 255.0.0.0 192.168.0.1
ip route vrf Mgmt-intf 172.16.0.0 255.0.0.0 192.168.0.1
ip route vrf Mgmt-intf 209.165.202.128 255.0.0.0 192.168.0.1
ip route vrf Mgmt-intf 192.168.0.0 255.255.240.0 192.168.0.1
interface atm 2/2/1.24 point-to-point
 description MLPoEoA aal5snap (Single-Link Bundles) Session (PTA Mode)
 no atm enable-ilmi-trap
 pvc 23/32
  class-vc mlpoeoa-vc-class-250K
  vbr-nrt 512 256 20
```

Example: Configuring Multilink PPP over LNS

Example: Configuring an LNS Device to Initiate and Receive L2TP Traffic

The following example shows how to set up a tunnel on the Gigabit Ethernet interface on which virtual private dialup network (VPDN) member links are negotiated and added to the Multilink PPP bundle that is cloned from virtual template 500:

```
aaa new-model
!
!
aaa authentication ppp default local
aaa authentication ppp TESTME group radius
aaa authorization network default local
aaa authorization network TESTME group radius
!
aaa session-id common
buffers small perm 15000
buffers mid perm 12000
buffers big perm 8000
!
vpdn enable
!
vpdn-group LNS_1
 accept-dialin
  protocol l2tp
  virtual-template 500
 terminate-from hostname LAC1-1
 local name LNS1-1
 lcp renegotiation always
 l2tp tunnel receive-window 100
 L2tp tunnel password 0 password1
 l2tp tunnel nosession-timeout 30
 l2tp tunnel retransmit retries 7
```

```
       l2tp tunnel retransmit timeout min 2
       l2tp tunnel retransmit timeout max 8
      !
      !
      interface GigabitEthernet 2/0/0
       ip address 209.165.202.140 255.255.255.0
       negotiation auto
       hold-queue 4096 in
      !
      !
      interface virtual-template 500
       ip unnumbered Loopback1
       peer default ip address pool pool-1
       ppp mtu adaptive
       ppp timeout authentication 100
       ppp max-configure 110
       ppp max-failure 100
       ppp timeout retry 5
      keepalive 30
       ppp authentication pap TESTME
       ppp authorization TESTME
       ppp multilink
      !
      ip local pool pool-1 209.165.201.3 209.165.201.30
      radius-server host 10.0.0.10 auth-port 1645 acct-port 1646 key password1
      radius-server retransmit 0
```

Example: Configuring a LAC Device to Initiate and Receive L2TP Traffic

The following example shows how to set up a tunnel on a LAC device to initiate and receive traffic:

```
vpdn enable
vpdn search-order domain
!
vpdn-group cisco-1.com
 request-dialin
  protocol l2tp
  domain cisco-1.com
 initiate-to ip 10.0.0.4
 local name LAC1
 no l2tp tunnel authentication
!
vpdn-group cisco-2.com
 request-dialin
  protocol l2tp
  domain cisco-2.com
 initiate-to ip 10.0.0.4
 local name LAC2
 no l2tp tunnel authentication
!
vpdn-group cisco-3.com
 request-dialin
  protocol l2tp
  domain cisco-3.com
 initiate-to ip 10.0.0.4
 local name LAC3
 no l2tp tunnel authentication
!
vpdn-group cisco-4.com
 request-dialin
  protocol l2tp
  domain cisco-4.com
 initiate-to ip 10.0.0.4
 local name LAC4
 no l2tp tunnel authentication
!
bba-group pppoe cpe-lac-lns-group
 virtual-template 99
!
interface GigabitEthernet 0/0/0
 description GE connection to cpp-rtp-7200-41 0/1 (PTA-to-CPE)
```

```
 no ip address
 negotiation auto
!
interface GigabitEthernet 0/0/0.23
 description PPPoEoVLAN-to-PPPoLNS (non-MLP) Sessions CPE-LAC-LNS
 encapsulation dot1Q 23
 pppoe enable group cpe-lac-lns-group
!
interface GigabitEthernet 0/0/0.24
 description PPPoEoQinQ-to-PPPoLNS (non-MLP) Sessions CPE-LAC-LNS
 encapsulation dot1Q 24 second-dot1q 240
 pppoe enable group cpe-lac-lns-group
!
interface GigabitEthernet 0/2/0
 description GE connection to cpp-rtp-7200-41 0/2  (PTA-to-CPE)
 no ip address
 negotiation auto
!
interface GigabitEthernet 0/2/0.23
 description MLPPPoEoVLAN-to-MLPPPoLNS (Single Link Bundles) Sessions CPE-LAC-LNS
 encapsulation dot1Q 23
 pppoe enable group cpe-lac-lns-group
!
interface GigabitEthernet 0/2/0.24
 description MLPPPoEoQinQ-to-MLPPPoLNS (Single Link Bundles) Sessions CPE-LAC-LNS
 encapsulation dot1Q 24 second-dot1q 240
 pppoe enable group cpe-lac-lns-group
!
interface TenGigabitEthernet 1/0/0
 description TenGig connection to cpp-rtp-mcp6ru-01 1/0/0 (LAC-to-LNS)
 ip address 10.0.0.3 255.255.0.0
!
interface virtual-template 99
 description VT for PPPoE/MLPPPoE sessions from CPE being forwarded to LNS CPE-LAC-LNS
 no ip address
 no peer default ip address
 ppp mtu adaptive
 ppp authentication chap pap
```

Example: Configuring Multilink PPP over Serial Interfaces

The figure below shows a typical inverse multiplexing application using two Cisco routers and Multilink PPP over four T1 lines.

Figure 7: Inverse Multiplexing Application Using Multilink PPP
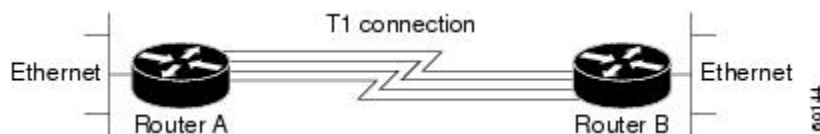
The example below shows the configuration commands that are used to create the inverse multiplexing application.

Router A Configuration

```
hostname RouterA
!
!
username RouterB password passwordA
ip subnet-zero
multilink virtual-template 1
!
interface virtual-template 1
```

```
 ip unnumbered Ethernet0
 ppp authentication chap
 ppp multilink
!
interface serial 0
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
!
interface serial 1
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
!
interface serial 2
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
!
interface serial 3
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
!
interface GigabitEthernet 0/0/0
 ip address 10.17.1.254 255.255.255.0
!
router rip
network 10.0.0.0
!
end
```

Router B Configuration

```
hostname RouterB
!
!
username RouterB password passwordA
ip subnet-zero
multilink virtual-template 1
!
interface virtual-template 1
 ip unnumbered Ethernet0
 ppp authentication chap
 ppp multilink
!
interface serial 0
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
!
interface serial 1
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
!
interface serial 2
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
!
interface serial 3
 no ip address
 encapsulation ppp
 no fair-queue
```

```
 ppp multilink
!
interface GigabitEthernet 0/0/0
 ip address 10.17.2.254 255.255.255.0
!
router rip
network 10.0.0.0
!
end
```

Additional References for Multilink PPP Connections for Broadband and Serial Topologies

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
PPP commands	Dial Technologies Command Reference
Broadband configuration tasks	*Broadband and DSL Configuration Guide*
Multilink PPP	*Multilink PPP Feature Functionality on the ASR 1000 Series Aggregation Services Router*

Standards and RFCs

Standard/RFC	Title
RFC 1990	*The PPP Multilink Protocol (MP)*
RFC 2686	*The Multi-Class Extension to Multi-Link PPP*

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Multilink PPP Connections for Broadband and Serial Topologies

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 2: Feature Information for Multilink PPP Connections for Broadband and Serial Topologies

Feature Name	Releases	Feature Information
MLPoA and MLPoEoA—PTA	Cisco IOS XE Release 3.4S	Configuring Multilink PPP over broadband includes configuring Multilink PPP over ATM (MLPoA), Multilink PPP over Ethernet (MLPoE), Multilink PPP over Ethernet over ATM (MLPoEoA), Multilink PPP over Queue-in-Queue (MLPoQinQ), and Multilink PPP over VLAN (MLPoVLAN). **Note** For Multilink PPP over Serial, Multilink PPP over Ethernet, and Multilink PPP over LNS, see the Release Notes specific to your platform and release. This feature also provides support for Link Fragmentation and Interleaving (LFI) to meet additional needs of service providers to manage the latency of their delay-sensitive voice, video, and interactive application traffic on slower broadband links.