

Multilink PPP for DDR – Basic Configuration and Verification

Document ID: 10239

Contents

Introduction

Before You Begin

- Conventions

- Prerequisites

- Components Used

What Multilink PPP Does

Configuring Multilink PPP

- Commands

- Legacy DDR

Dialer Profiles

Verify MPPP Operation

Related Information

Introduction

Multilink PPP (also referred to as MP, MPPP, MLP, or Multilink) provides a method for spreading traffic across multiple physical WAN links while providing packet fragmentation and reassembly, proper sequencing, multivendor interoperability, and load balancing on inbound and outbound traffic.

MPPP allows packets to be fragmented. These fragments are sent simultaneously over multiple point-to-point links to the same remote address. The multiple physical links come up in response to a user-defined load threshold. This load can be measured on just inbound traffic, on just outbound traffic, or on either; however, it cannot be measured on the combined load of both inbound and outbound traffic.

For dial connections, MPPP can be configured for ISDN Basic Rate Interfaces (BRIs) and Primary Rate Interfaces (PRIs), as well as for asynchronous serial interfaces. It can also be configured for non-dial serial interfaces, though this functionality is not specifically addressed in this document. This document will address configuration of basic MPPP for Dial-on-Demand Routing (DDR). Multichassis Multilink PPP will not be covered in this document; see the Multichassis Multilink PPP (MMP) documentation for more information.

Before You Begin

Conventions

For more information on document conventions, see the Cisco Technical Tips Conventions.

Prerequisites

There are no specific prerequisites for this document.

Components Used

The information in this document is based on the software and hardware versions below.

- Multilink PPP was first introduced in Cisco IOS® Software Release 11.0(3)
- Cisco IOS Software Release 11.3 was used in this example.

The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you are working in a live network, ensure that you understand the potential impact of any command before using it.

What Multilink PPP Does

MPPP is a method for splitting, recombining, and sequencing datagrams across multiple logical data links. See RFC 1990 [↗](#) RFC 1990 [↗](#) for a good description of MPPP. It was originally motivated by the desire to exploit multiple bearer channels in ISDN, but it is equally applicable to any situation in which multiple PPP links connect two systems, including async links.

Traffic routed across an MPPP link via its controlling interface (a Virtual Access interface) will be fragmented, with the fragments being sent across the different physical links. At the remote end of the link, the fragments are reassembled and forwarded to the next hop toward their ultimate destination.

Configuring Multilink PPP

This section addresses the commands and the different methods of configuring MPPP on a router.

Commands

| Required Command | Description |
|---|---|
| ppp multilink | Configure the PPP multilink command (on both routers) under the physical interface and the dialer interface (if using dialer profiles). Note: If you add this command, you must disconnect any existing connections and then reconnect for the new multilink parameters to be applied. Because multilink is negotiated during the call setup, any changes to multilink are not implemented on connections that have completed the link control protocol (LCP) negotiation. |
| dialer load-threshold 5 outbound | Interface load (from 1 to 255) beyond which the dialer will initiate another call to the destination. The bandwidth is defined as a ratio of 255, where 255 would be 100 percent of the available bandwidth. In this example, the additional channel will be brought up when the outbound load on the link is 5/255 or 2 percent. Vary this value depending on your needs. The outbound argument sets the load calculation to be made only on outbound traffic. The inbound argument does the same, but for |

| | <p>inbound traffic only. Using the either argument sets the load as the larger of the outbound and inbound loads.</p> <p>Tip: Often, customers will configure the command dialer load-threshold 1 because they want all of their B-channels to be used immediately for every call. The theory behind this is that if all the B-channels go up at once and the entire ISDN pipe is used for each call, the call should be shorter in duration because it will take less time to transfer the user data.</p> <p>While this theory is sound, in practice it is a good idea never to set your dialer load-threshold value to anything less than $\frac{1}{2}$. Setting this value to something less than $\frac{1}{2}$ can cause multiple ISDN channels to go up at once which can lead to contention between both channels and a failure to connect with any of them.</p> |
|---|---|
| Optional Commands | Description |
| <p>ppp timeout multilink link remove <i>seconds</i></p> | <p>This command may be used to prevent the multilink connections from flapping when the load varies. For example, when the load threshold is set to 15 (that is, $15/255 = 6$ percent) and the traffic exceeds the threshold, additional lines are brought up. When the traffic falls below the threshold, the additional lines are dropped. In situations where data rates are highly variable, it is advantageous for the multiple channels to stay up for a specified period of time even if the load-threshold falls below the specified value. Assign this multilink timeout to be less than that specified for dialer idle-timeout which controls the timeout for all links.</p> |
| <p>ppp timeout multilink link add <i>seconds</i></p> | <p>This command can be used to prevent multiple links from being added to the MP bundle until high traffic is received for a specified interval. This can prevent bursts of traffic from unnecessarily bringing up additional lines.</p> |
| <p>ppp multilink max-link or ppp multilink links maximum (IOS 12.2 or higher)</p> | <p>The value set in the ppp multilink links maximum command specifies the maximum number of links allowed in a bundle. When more links than the number assigned with the ppp multilink links maximum command try to enter the bundle,</p> |

| | |
|--|---|
| | MLP hangs up its dialer channels to reduce the number of links. This can be used to prevent a multilink connection from bringing up too many connections. |
| ppp multilink min-link or ppp multilink links minimum (IOS 12.2 or higher) | The value set in the ppp multilink links minimum command specifies the minimum number of links that MLP will try to keep in a bundle. MLP attempts to dial up additional links to obtain the number specified by the links argument, even if the load does not exceed the load threshold. This can be used to force a certain number of channels up |
| multilink bundle-name | This command can be used to change the criteria with which a multilink bundle is identified. |

Legacy DDR

This section addresses how to configure Multilink PPP using Legacy DDR (rotary-group and dialer maps).

Method 1: Only One Physical Interface – ex. ISDN

Because ISDN interfaces are considered to be "Dialer" interfaces, few commands are required to make an ISDN interface capable of making MPPP connections. For example, it is not necessary to configure a dialer rotary group unless you are using more than one BRI or PRI.



Following is an example of a BRI configured to make a simple dial-on-demand PPP connection:

```
!
interface BRI0
 ip address 192.168.12.3 255.255.255.240
 encapsulation ppp
 dialer map IP 192.168.12.1 name ROUTER1 5554321
 dialer-group 1
 ppp authentication chap
 isdn spid1 40855512120000 5551212
 isdn spid2 40855512340000 5551234
!
```

Only two commands must be added to this interface's configuration to make MPPP possible. The router at the other end of the call must be similarly configured. These two commands are:

```
ppp multilink
dialer load-threshold load [outbound | inbound | either]
```

Method 2: Multiple Physical Interfaces – ISDN, Async, and Serial

In circumstances where two or more physical interfaces need to be bundled together (for example, when using async or serial interfaces, or more than one ISDN interface) a different method must be used. In these cases, a dialer rotary group must be configured and a Dialer interface must be added to the configuration of the router in order to control the MPPP connection. In brief, a "logical" interface must control the "physical" interfaces.

In order to accomplish this, you must:

1. Place the physical interfaces into a rotary group.
2. Create a logical ("Dialer") interface as the lead for the rotary group.
3. Configure the Dialer interface to do MPPP.

Follow these steps to configure MPPP on multiple interfaces:

1. Put the physical interfaces into a rotary group by using the **dialer rotary-group** *number* command. In this example, the asynchronous interface is put into rotary group 1:

```
router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
router(config)#interface async 1
router(config-if)#dialer rotary-group 1
router(config-if)#^Z
router#
```

Note: Be sure to use the **no shutdown** interface configuration command if the router has never been configured or if the router has been set back to its default configuration.

2. To create a Dialer interface, use the **interface dialer** *number* global configuration command. In this example, interface Dialer 1 is created:

```
router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
router(config)#interface dialer 1
router(config-if)#end
router#
```

Note: The number argument of the **interface dialer** command must be the same as the number of the rotary group configured in Step 1.

Use the **show running-config** command to see the default configuration of a dialer interface:

```
!
interface Dialer1
  no ip address
  no cdp enable
!
```

3. Next, configure the Dialer interface in order to place or receive calls. The essential commands for MPPP are the same as in Step 1:

```
!
interface Dialer1
  ip address 192.168.10.1 255.255.255.0
  encapsulation ppp
  dialer in-band
  dialer idle-timeout 300
  dialer map ip 192.168.10.11 name RemoteRouter broadcast 5551234
  dialer load-threshold 100
  dialer-group 1
  no fair-queue
```

```

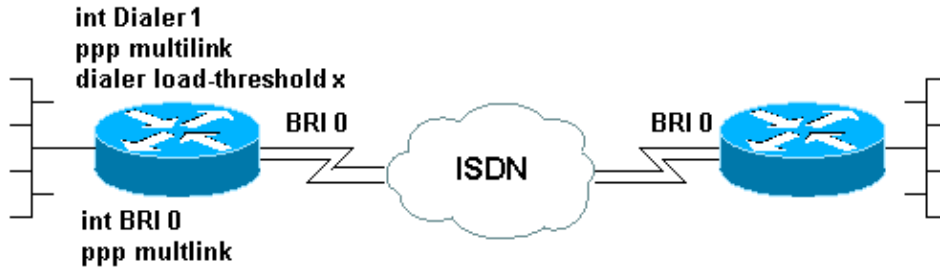
ppp multilink
ppp authentication chap
!

```

For examples of complete DDR configurations with MPPP, see the PPP Support Page

Dialer Profiles

Configuring Multilink PPP on Dialer Profiles is similar to that for Legacy DDR. The **ppp multilink** command must be configured on **both** the physical interface and dialer interface. The **dialer load-threshold** command should be configured on the Dialer interface. For example,



```

interface BRI0
  no ip address
  encapsulation ppp
  dialer pool-member 1
  isdn switch-type basic-5ess
  ppp authentication chap
  ppp multilink

! -- Configure multilink on both physical and dialer interfaces

!
interface Dialer1
  ip address 172.22.85.1 255.255.255.0
  encapsulation ppp
  dialer pool 1

! -- Defines the pool of physical resources from which the Dialer

! -- interface may draw B channels as needed.

  dialer remote-name R1
  dialer string 6661000
  dialer load-threshold 128 outbound
  dialer-group 5
  ppp authentication chap
  ppp multilink

! -- Configure multilink on both physical and dialer interfaces

```

For more information on dialer profiles refer to the document [Configuring and Troubleshooting Dialer Profiles](#)

Verify MPPP Operation

In order to verify the proper operation of an MPPP connection, use the **debug ppp negotiation** command. The critical elements that must be negotiated in the LCP phase are the Maximum Receive Reconstructed Unit (MRRU) and the Endpoint Discriminator (EndpointDisc):

```
As1 LCP: O CONFREQ [Listen] id 1 len 26
As1 LCP:   AuthProto CHAP (0x0305C22305)
As1 LCP:   MagicNumber 0x10963BD1 (0x050610963BD1)
As1 LCP:   MRRU 1524 (0x110405F4)
As1 LCP:   EndpointDisc 1 Local (0x13070174657374)
As1 LCP: I CONFREQ [REQsent] id 3 Len 27
As1 LCP:   MRU 1500 (0x010405DC)
As1 LCP:   MagicNumber 0x2CBF9DAE (0x05062CBF9DAE)
As1 LCP:   MRRU 1500 (0x110405DC)
As1 LCP:   EndpointDisc 1 Local (0x1306011AC16D)
As1 LCP: I CONFACK [REQsent] id 1 Len 26
As1 LCP:   AuthProto CHAP (0x0305C22305)
As1 LCP:   MagicNumber 0x10963BD1 (0x050610963BD1)
As1 LCP:   MRRU 1524 (0x110405F4)
As1 LCP:   EndpointDisc 1 Local (0x13070174657374)
As1 LCP: O CONFACK [ACKrcvd] id 3 Len 24
As1 LCP:   MRU 1500 (0x010405DC)
As1 LCP:   MagicNumber 0x2CBF9DAE (0x05062CBF9DAE)
As1 LCP:   MRRU 1500 (0x110405DC)
As1 LCP:   EndpointDisc 1 Local (0x1306011AC16D)
As1 LCP: State is Open
```

As with the other elements of LCP negotiation, the MRRU and EndpointDisc must be agreed to by both ends of the connection during the exchange of CONFREQs and CONFACKs. Both ends of the connection must send CONFACKs for the protocol to be established. For more information on how to read **debug ppp negotiation** output refer to the document Understanding **debug ppp negotiation** Output.

After MPPP has been successfully negotiated during the LCP phase of PPP negotiation and Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP) have completed successfully, a Virtual Access interface will be created by the Cisco IOS Software to represent the MPPP bundle. For more information on the uses and theory behind Virtual Access interfaces, please see the Virtual Access PPP Features in Cisco IOS documentation.

The creation of the Virtual Access interface is signaled in the **debug ppp negotiation** output by the following:

```
As1 PPP: Phase is VIRTUALIZED
```

From this point forward, PPP negotiation of the Network Control Protocols is handled by the Virtual Access interface. For example:

```
Vi1 PPP: Treating connection as a dedicated line
Vi1 PPP: Phase is ESTABLISHING, Active Open
Vi1 LCP: O CONFREQ [Closed] id 1 Len 37
...
Vi1 PPP: Phase is UP
Vi1 IPCP: O CONFREQ [Closed] id 1 len 10
Vi1 IPCP:   Address 192.168.10.1 (0x0306C0A80A01)
...
```

Once the MPPP connection has been established, information on the connection can be found in the output of the **show ppp multilink** command:

```
router#show ppp multilink
```

```

Virtual-Access1, bundle name is RemoteRouter
  0 lost fragments, 0 reordered, 0 unassigned, sequence 0x29/0x17 rcvd/sent
  0 discarded, 0 lost received, 1/255 load
  Member links: 1 (max not set, min not set)
  Async1

```

The *bundle name* is the authenticated username of the connected client device. The *member links* are a list of the physical interfaces that are active members of the bundle. In the example above, only one link is currently active, however the router can add more links to the bundle at some point. To disconnect a specific link (rather than the whole bundle) using the command **clear interface interface** . For example, **clear interface Async1**.

The order of which naming convention will be tried first (as seen in bundle name) can be changed using the command **multilink bundle-name** .

In addition, the **show interface** command is valid for the Virtual Access interface as it is for any other physical or logical interface. The same type of information will be presented as would appear in any other **show interface** output.

```

router#show interface virtual-access 1
Virtual-Access1 is up, line protocol is up
Hardware is Virtual Access interface
Description: Multilink PPP to RemoteRouter

!-- This VAccess interface is conencted to "RemoteRouter"

Internet address is 192.168.10.1/24
MTU 1500 bytes, BW 7720 Kbit, DLY 100000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation PPP, loopback not set
Keepalive set (10 sec)
DTR is pulsed for 5 seconds on reset
LCP Open, multilink Open

! -- multilink state should be Open for a successful connection

Open: IPCP
Last input 00:00:01, output never, output hang never
Last clearing of "show interface" counters 04:25:13
Queueing strategy: fifo
Output queue 0/40, 0 drops; input queue 0/75, 0 drops
5 minute input rate 12000 bits/sec, 2 packets/sec
5 minute output rate 12000 bits/sec, 2 packets/sec
2959 packets input, 2075644 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
2980 packets output, 2068142 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions

```

Related Information

- [Understanding debug ppp negotiation Output](#)
 - [Troubleshooting Second B-channel Call Failures on ISDN BRI Links](#)
 - [Configuring BRI-to-BRI Dialup with DDR Dialer Maps](#)
 - [Virtual Access PPP Features in Cisco IOS](#)
 - [PPP Design and Debugging](#) 
 - [PPP Support Page](#)
 - [Technical Support – Cisco Systems](#)
-

