



CHAPTER 12

Understanding and Configuring IP Multiplexing

This chapter discusses IP multiplexing for satellite topologies in the following major sections:

- [Understanding IP Multiplexing, page 12-1](#)
- [Configuring IP Multiplexing, page 12-2](#)
- [Verifying the IP Multiplexing Configuration, page 12-12](#)

Understanding IP Multiplexing

You can use IP multiplexing to optimize IPv4 and IPv6 traffic in environments where packet-per-second transmission limitations cause inefficient bandwidth utilization, such as a satellite network. IP multiplexing addresses this constraint by bundling smaller packets into one larger UDP packet, known as a superframe. The router then sends the superframe to the destination router which demultiplexes the individual packets out of the superframe and routes them to their final destination.

IP multiplexing uses Cisco IOS access control lists (ACLs) to identify outbound packets. You can configure standard, extended, or named ACLs to use with IP multiplexing. IP multiplexing maintains a cache of recent ACL lookup results to optimize traffic classification.

The following interface types support IP multiplexing:

- Ethernet
- Fast Ethernet
- Gigabit Ethernet
- IPv4 GRE tunnel
- IPv6 GRE tunnel
- Ethernet, Fast Ethernet, and Gigabit Ethernet VLAN
- VMI over Ethernet, Fast Ethernet, and Gigabit Ethernet
- Virtual-Template on VMI

Both endpoints of the multiplex connection must be configured for multiplexing with corresponding source and destination addresses. If a superframe arrives at an interface with IP multiplexing not configured or not configured to receive superframes from the destination router, the superframe is not demultiplexed, and the superframe is routed normally. If IP multiplexing is not configured, then outbound packets are routed normally.

Configuring IP Multiplexing

When configuring IP multiplexing, you must configure each device before enabling the configuration. Failure to do so will result in lost packets at the end that is not yet configured.

Configuring IP multiplexing requires the following procedures:

- [Configuring ACLs to Identify Traffic, page 12-2](#)
- [Configuring an IP Multiplex Profile, page 12-2](#)
- [Configuring IP Multiplexing on an Interface, page 12-6](#)

The following procedures are optional and can be used to optimized IP multiplexing:

- [Configuring the Multiplex Lookup Cache Size, page 12-8](#)
- [Configuring IP Multiplexing on an Interface, page 12-6](#)

Configuring ACLs to Identify Traffic

IP multiplexing uses ACL definitions to identify traffic selected for multiplexing treatment. You can configure standard, extended or named ACLs to define traffic you want to multiplex. Packets that are not identified by an ACL used for multiplexing are routed normally.

Refer to the following URL on Access Control Lists for more information on how to configure an ACL: http://www.cisco.com/en/US/docs/ios/sec_data_plane/configuration/guide/sec_acc_list_ov_ps10591_TSD_Products_Configuration_Guide-Chapter.html.

In general, an ACL statement for IP multiplexing should have the following format:

```
permit udp any host destination_IP_address UDP_port_number
```

IP Multiplexing makes caching decisions based on destination IP address, destination port, and protocol type. Although ACLs can be defined to filter packets based on other attributes, using other attributes in an IP Multiplexing ACL may have unexpected and/or unwanted results.

Configuring an IP Multiplex Profile

The attributes associated with an IP multiplexing connection between two routers are configured in an IP multiplex profile.



Tip

You must configure an IP multiplex profile for each endpoint of an IP multiplex connection in the network.

You must define the following information for an IP multiplex profile:

- Profile name
- Access control list (ACL) used to classify outbound IP packets as IP multiplex traffic
- Source and destination IP addresses to be included in the superframe header
- Maximum amount of time the router waits to fill a superframe before sending a partial superframe

You can define the following optional information for an IP multiplex profile:

- Maximum size of an outbound IP packet to be considered for multiplexing
- Maximum MTU size of a superframe
- TTL value to be included in the superframe IP header

Perform the following task to configure an IP multiplex profile.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **{ip | ipv6} mux profile** *profile_name*
4. **access-list** *access-list name or number*
5. **source** *{ip_address | interface name}*
6. **destination** *ip_address*
7. **(Optional) holdtime** *milliseconds*
8. **(Optional) maxlength** *bytes*
9. **(Optional) mtu** *bytes*
10. **(Optional) ttl** *hops*
11. **(Optional) no singlepacket**
12. **no shutdown**
13. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<pre>enable</pre> <p>Example: Router> enable Router#</p>	<p>Enables privileged EXEC mode.</p> <p>Enter your password if prompted.</p>
Step 2	<pre>configure terminal</pre> <p>Example: Router# configure terminal Enter configuration commands, one per line. End with CNTL/Z. Router(config)#</p>	<p>Enters global configuration mode.</p>
Step 3	<pre>{ip ipv6} mux profile profile_name</pre> <p>Example: Router(config)#ip mux profile routeRTP-SJ Router(config-ipmux-profile)#</p>	<p>Creates an IP multiplex profile with the specified name and enters IP multiplexing mode profile mode.</p> <p>Use the ip keyword to create an IPv4 profile. Use the ipv6 keyword create an IPv6 profile.</p>
Step 4	<pre>access-list access-list name or number</pre> <p>Example: Router(config-ipmux-profile)# access-list routeRTP-SJ Router(config-ipmux-profile)#</p>	<p>Applies the specified access list to the profile and uses the statements in the access list to identify outbound traffic for multiplexing.</p>
Step 5	<pre>source {ip_address interface interface-type}</pre> <p>Example: Router(config-ipmux-profile)#source 172.16.1.1 Router(config-ipmux-profile)#</p>	<p>Designates the source IP address for the profile. The source address is the IP address assigned to the outbound interface. If you created an IPv4 profile, then use an IPv4 address. If you created an IPv6 profile, then use an IPv6 address.</p> <p>If you use the interface keyword, IP multiplexing will use the IP address configured for that interface. Beware if you are using the interface keyword for an IPv6 interface with multiple IP addresses assigned to it. IP multiplexing may not use the IP address you want for multiplexing.</p> <p>The profile must be shutdown in order to change the source address.</p> <p> Note This source address must be configured as the destination address in the corresponding profile at the other end of the IP multiplexing connection.</p>

	Command or Action	Purpose
Step 6	<p><code>destination ip_address</code></p> <p>Example: Router(config-ipmux-profile)# destination 172.172.16.2.1 Router(config-ipmux-profile)#</p>	<p>Designates the IP address to which superframes will be sent from the particular profile. The destination address must match the source address of the corresponding profile on the destination router. If you created an IPv4 profile, then use an IPv4 address. If you created an IPv6 profile, then use an IPv6 address.</p> <p>The profile must be shutdown in order to change the destination address.</p> <p> Note This destination address must be configured as the source address in the corresponding profile at the other end of the IP multiplexing connection.</p>
Step 7	<p><code>holdtime milliseconds</code></p> <p>Example: Router(config-ipmux-profile)# holdtime 150 Router(config-ipmux-profile)#</p>	<p>(Optional) Configures the amount of time in milliseconds that a multiplex profile waits to fill the superframe before sending a partial superframe.</p> <p>Valid values range from 20 to 250 milliseconds</p> <p>If you do not set a hold time, the profile uses 20 milliseconds as a default</p>
Step 8	<p><code>maxlength bytes</code></p> <p>Example: Router(config-ipmux-profile)# maxlength 128 Router(config-ipmux-profile)#</p>	<p>(Optional) Configures the largest packet size that the multiplex profile can hold for multiplexing. A larger packet size will not be multiplexed even if it correctly matches the ACL attached to the profile.</p> <p>Valid values range from 64 to 1472 bytes.</p> <p>If you do not configure a maximum packet length, then any packet that fits into the superframe is multiplexed.</p>
Step 9	<p><code>mtu bytes</code></p> <p>Example: Router(config-ipmux-profile)# maxlength 512 Router(config-ipmux-profile)#</p>	<p>(Optional) Configures the maximum size for the outbound superframe. Valid values range from 256 to 1500 bytes.</p> <p>If you do not configure a MTU values, the profile uses 1500 bytes as a default.</p> <p>The superframe size specified in the mtu command includes the IP and UDP headers for the superframe of 48 bytes for IPv6 and 28 bytes for IPv4 packets. Therefore an IPv6 mtu configured to 1400 bytes will accept 1352 bytes of data before sending a full superframe. An IPv4 mtu configured to 1400 bytes will accept 1372 bytes of data before sending a full superframe.</p>
Step 10	<p><code>ttl hops</code></p> <p>Example: Router(config-ipmux-profile)#ttl 128 Router(config-ipmux-profile)#</p>	<p>(Optional) Configures the superframe time-to-live (ttl) for the IP header of the superframe.</p> <p>Valid values range from 1 to 255 hops.</p> <p>By default, the ttl value is set to 64 hops.</p>

	Command or Action	Purpose
Step 11	singlepacket Example: Router(config-ipmux-profile)# singlepacket Router(config-ipmux-profile)#	Configures the router to send the original packet unmodified if there is only one packet to multiplex when the hold timer expires. By default, single packets are multiplexed into superframes when the hold timer expires.
Step 12	no shutdown Example: Router(config-ipmux-profile)# no shutdown Router(config-ipmux-profile)#	Activates the multiplex profile If you want to change the ACL associated with the profile or the contents of the ACL, you must enter the shutdown command for the profile, make the changes and then enter the no shutdown command.
Step 13	exit Example: Router(config-ipmux-profile)# exit Router(config)#	Exits the configuration mode and returns to global configuration mode.

Configuring IP Multiplexing on an Interface

IP multiplexing must be configured on an interface and the interface enabled to activate IP multiplexing. Once IP multiplexing is configured on an interface, all multiplex profiles are used to classify IP packets routed for transmission on the interface. The following Cisco IOS interfaces support IP Multiplexing:

- Ethernet
- Fast Ethernet
- Gigabit Ethernet
- IPv4 GRE tunnel
- IPv6 GRE tunnel
- Ethernet, Fast Ethernet, and Gigabit Ethernet VLAN
- VMI over Ethernet, Fast Ethernet, and Gigabit Ethernet
- Virtual-Template on VMI

Perform the following procedure to enable IP multiplexing on an interface:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type/slot*
4. **{ip | ipv6} mux**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable Router#	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal Enter configuration commands, one per line. End with CNTL/Z. Router(config)#	Enters global configuration mode.
Step 3	interface type/slot Example: Router(config)# interface fastethernet0/1 Router(config-if)#	Enters interface configuration mode for the specified interface.
Step 4	{ip ipv6} mux Example: Router(config-if)# ipv6 mux Router(config-if)#	Enables IP multiplexing on the interface. Use ip mux for an IPv4 interface and ipv6 mux for an IPv6 interface.  Note You can use the show interface command to verify that the interface is administratively up and whether the interface has an IPv4 or IPv6 address configured for the interface.
Step 5	exit Example: Router(config-ipmux-policy)# exit Router(config)#	Exits IP multiplex policy mode.

Configuring UDP Port for Superframe Traffic

The receiving router identifies incoming superframes by destination IP address, protocol type (UDP), and a UDP port number. A single UDP port number is used for all IP multiplexing traffic in the network.


Note

If you do not configure a UDP port for IP multiplexing traffic, the system uses the default value of 6682. This value is inserted in the UDP header of the outbound superframe. If you use the default UDP port value, make sure that all routers sending or receiving IP multiplexing traffic use the same value.

Perform this task to configure the UDP port for IP multiplexing traffic.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **{ip | ipv6} mux udpport *port_number***

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable Router#	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal Enter configuration commands, one per line. End with CNTL/Z. Router(config)#	Enters global configuration mode.
Step 3	{ip ipv6} mux udpport <i>port_number</i> Example: Router(config)# ip mux udpport 5000 Router(config)#	Configures the UDP port for IP multiplexing. Valid Values range from 1024 to 49151.

Configuring the Multiplex Lookup Cache Size

The lookup cache maps the destination address, protocol type, and port number to a multiplex profile to reduce performance overhead related to ACL lookups. You can configure the maximum size of the cache to manage memory utilization on the router.

The maximum size of the IPv6 cache can range from 1,000,000 to 4,294,967,295 bytes which corresponds to 10,419 to 44,739,242 entries. The maximum size of the IPv4 cache can range from 1,000,000 to 4,294,967,295 bytes which corresponds to 11,363 to 49,367,440 entries.



Note

If you do not configure the cache size, the cache size defaults to 1,000,000 bytes, which will hold 11,363 entries for IPv4 multiplex and 10,419 for IPv6 multiplex.

Perform this task to configure the size of the lookup cache.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. `ip mux cache size`

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable Router#	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Router# <code>configure terminal</code> Enter configuration commands, one per line. End with CNTL/Z. Router(config)#	Enters global configuration mode.
Step 3	ip mux cache size Example: Router(config)# <code>ip mux cache 5000000</code> Router(config)#	Configures the size of the IP multiplexing look cache. Valid Values range from 1000000 to 4294967295 bytes.

Configuring the IP Multiplex Policy

An IP multiplex policy is used to retain DSCP priorities of the underlying data traffic. An IP multiplex policy approximates QoS. If you configure an IP multiplex policy, then you can configure DSCP values for the superframe header and you can specify that only the packets with a specified DSCP value be placed into the superframe. Note that a policy can match more than one DSCP value.

A router may have up to three multiplex policies for IPv6 and three multiplex policies for IPv4 defined on it. Multiplex policies are global and apply to all multiplex profiles on a router.

If the DSCP value assigned to a packet does not match any multiplex policy, then the router uses the default multiplex policy for superframe multiplexing. Superframes for the default policy have a DSCP value set to 0.

If you do not configure an IP multiplex policy, then all IP multiplex packets are sent using the default IP multiplex policy with a DSCP value equal to 0.

The DSCP values in each packet header remains intact as the packet goes through the multiplexing and demultiplexing processes.

Configuring DSCP Value for Outbound Superframes

Perform this task to create a multiplex policy, specify the matching DSCP values for a superframe, and specify the outbound DSCP value for the header of the superframe.

If you do not configure a DSCP value for an outbound superframe, superframes are sent with DSCP equal to 0.

If the DSCP value for packets selected for multiplexing does not match any multiplex policy **matchdscp** values, then these packets are sent using the default multiplex policy which has a DSCP set to 0.

A packet found to match the **matchdscp** value is put in the superframe with the corresponding multiplex policy.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **{ip | ipv6} mux policy *policy_name***
4. **outdscp *DSCP_value***
5. **matchdscp *DSCP_value***
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable Router#	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal Enter configuration commands, one per line. End with CNTL/Z. Router(config)#	Enters global configuration mode.
Step 3	{ip ipv6} mux policy <i>policy-name</i> Example: Router(config)# ip mux policy RouterTP-SJ Router(config-ipmux-policy)#	Configures an IP policy with the specified name and enters IP multiplex policy configuration mode.

	Command or Action	Purpose
Step 4	<pre> outdscp <i>DSCP_value</i> </pre> <p>Example: Router(config-ipmux-policy)# outdscp 10 Router(config-ipmux-policy)#</p>	<p>Configures the DSCP value for the outbound superframe.</p> <p>Valid values range from 0 to 63. The following DSCP values are also valid:</p> <ul style="list-style-type: none"> af11 Match packets with AF11 dscp (001010) af12 Match packets with AF12 dscp (001100) af13 Match packets with AF13 dscp (001110) af21 Match packets with AF21 dscp (010010) af22 Match packets with AF22 dscp (010100) af23 Match packets with AF23 dscp (010110) af31 Match packets with AF31 dscp (011010) af32 Match packets with AF32 dscp (011100) af33 Match packets with AF33 dscp (011110) af41 Match packets with AF41 dscp (100010) af42 Match packets with AF42 dscp (100100) af43 Match packets with AF43 dscp (100110) cs1 Match packets with CS1(precedence 1) dscp (001000) cs2 Match packets with CS2(precedence 2) dscp (010000) cs3 Match packets with CS3(precedence 3) dscp (011000) cs4 Match packets with CS4(precedence 4) dscp (100000) cs5 Match packets with CS5(precedence 5) dscp (101000) cs6 Match packets with CS6(precedence 6) dscp (110000) cs7 Match packets with CS7(precedence 7) dscp (111000) default Match packets with default dscp (000000) ef Match packets with EF dscp (101110)

	Command or Action	Purpose
Step 5	<pre>matchdscp DSCP_value</pre> <p>Example: Router(config-ipmux-policy)# matchdscp 45 Router(config-ipmux-policy)#</p>	<p>Configures the DSCP value that IP multiplexing uses to compare against the DSCP value in packets bound for multiplexing. A match puts the packet in the superframe that corresponds to the IP multiplex policy.</p> <p>You can enter more than one value.</p> <p>Valid values range from 0 to 63. The following DSCP values are also valid:</p> <ul style="list-style-type: none"> af11 Match packets with AF11 dscp (001010) af12 Match packets with AF12 dscp (001100) af13 Match packets with AF13 dscp (001110) af21 Match packets with AF21 dscp (010010) af22 Match packets with AF22 dscp (010100) af23 Match packets with AF23 dscp (010110) af31 Match packets with AF31 dscp (011010) af32 Match packets with AF32 dscp (011100) af33 Match packets with AF33 dscp (011110) af41 Match packets with AF41 dscp (100010) af42 Match packets with AF42 dscp (100100) af43 Match packets with AF43 dscp (100110) cs1 Match packets with CS1(precedence 1) dscp (001000) cs2 Match packets with CS2(precedence 2) dscp (010000) cs3 Match packets with CS3(precedence 3) dscp (011000) cs4 Match packets with CS4(precedence 4) dscp (100000) cs5 Match packets with CS5(precedence 5) dscp (101000) cs6 Match packets with CS6(precedence 6) dscp (110000) cs7 Match packets with CS7(precedence 7) dscp (111000) default Match packets with default dscp (000000) ef Match packets with EF dscp (101110)
Step 6	<pre>exit</pre> <p>Example: Router(config-ipmux-policy)#exit Router(config)#</p>	<p>Exits IP multiplex policy mode.</p>

Verifying the IP Multiplexing Configuration

The following procedures can be used for verifying the IP Multiplexing configuration on the router:

[Displaying IP Multiplex Statistics, page 12-13](#)

[Displaying IP Multiplexing Cache Statistics, page 12-13](#)

[Displaying IP Multiplex Profiles, page 12-13](#)

[Displaying IP Multiplexing Statistics for an Interface, page 12-14](#)

Displaying IP Multiplex Statistics

You can use the **show {ip | ipv6} mux** command to display IP multiplexing statistics.

The following example shows how to display IPv4 multiplex statistics:

```
router#show ip mux
Superframe UDP Port: 6682

Multiplex Policies
muxpol          Outbound DSCP:      19
                  Match DSCP values:  af21 19
muxpol2         Outbound DSCP:      af11
                  Match DSCP values:  11
muxpol3         Outbound DSCP:      2
                  Match DSCP values:  1

IPv4 Multiplex Cache Statistics
Current Entries:          3
Maximum Number of Entries: 56818
Cache High Water Mark:   3
Total Stale Entries:     0
Total Do-Not-Multiplex Entries: 0

router#
```

Displaying IP Multiplexing Cache Statistics

You can use the **show {ip | ipv6} mux cache** command to display IP multiplexing cache statistics.

The following example shows how to display the cache statistics:

```
router#show ip mux cache

IPv4 Multiplex Cache Statistics
Current Entries:          3
Maximum Number of Entries: 56818
Cache High Water Mark:   3
Total Stale Entries:     0
Total Do-Not-Multiplex Entries: 0

IPv4 Multiplex Cache Contents
Destination Address  Port      Protocol  Profile
-----
20.20.20.24         0         UDP       r1
20.20.20.20         1000      UDP       r1
20.20.20.21         1000      UDP       r1

router#
```

Displaying IP Multiplex Profiles

You can use the **show {ip | ipv6} mux profile** command to display IP multiplex profile statistics. If you do not enter a profile name, this command displays statistics for all multiplex profiles.

The following example shows how to display the profile statistics for the IPv6 profile r1v6:

```
router#show ipv6 mux profile
Profile r1v6
Shutdown:          No
Destination:      2000:0:1:2:A8BB:CFF:FE01:5610
```

```

Source:                2000:0:1:2:A8BB:CCFF:FE01:5510
Access-list:           muxv6acl
TTL:                   64
  Max mux length:      1452
MTU:                   1500
  Hold time(ms):       20
  Single packet superframes: Enabled

Inbound (demux) Statistics
  Superframes received: 0
  Packets demultiplexed: 0
  Avg. Inbound Multiplex ratio: N/A

Outbound (mux) Statistics
Default Policy
  Packets: 40825  Full Superframes: 0      Partial Superframes: 20293
  Avg. Outbound Multiplex ratio: 2.1:1  Mux length exceeded: 0

Policy policy1
  Packets: 1273  Full Superframes: 0      Partial Superframes: 532
  Avg. Outbound Multiplex ratio: 2.39:1  Mux length exceeded: 0

router#

```

Displaying IP Multiplexing Statistics for an Interface

You can use the **show {ip | ipv6} mux interface** command to display IP multiplexing statistics for a specific interface.

If you do not specify a specific interface, this command displays statistics for all interfaces with IP multiplexing configured.

The following example shows how to display IP multiplex statistics for Ethernet 0/1:

```

router#show ip mux interface Ethernet0/1
IPv4 Multiplexing statistics for Ethernet0/1
Transmit
  IPv4 superframes transmitted: 20430
  IPv4 packets multiplexed: 30555
  Average TX mux ratio: 1.49:1
Receive
  IPv4 super frames received: 22009
  IPv4 packets demuxed: 32634
  IPv4 superframes rejected: 0
  IPv4 format errors: 0
  Average RX mux ratio: 1.48:1
router#

```