GRE IPv6 Tunnels

The GRE IPv6 Tunnels feature enables the delivery of packets from other protocols through an IPv6 network and allows the routing of IPv6 packets between private networks across public networks with globally routed IPv6 addresses. Generic routing encapsulation (GRE) is a unicast protocol that offers the advantages of encapsulating broadcast and multicast traffic (multicast streaming or routing protocols) or other non-IP protocols and of being protected by IPsec.

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 GRE IPv6 Tunnels

- GRE tunnel keepalive packets are not supported.
- Multipoint GRE (mGRE) IPv6 tunneling is not supported.

- There is limited support for tunnel transport in virtual routing and forwarding (VRF). The limited support in VRF is applicable to IPv6 point-to-point GRE without tunnel protection.

Information About GRE IPv6 Tunnels

Overview of GRE IPv6 Tunnels

The GRE IPv6 Tunnels feature enables the delivery of packets from other protocols through an IPv6 network and allows the routing of IPv6 packets between private networks across public networks with globally routed IPv6 addresses.

For point-to-point GRE tunnels, each tunnel interface requires a tunnel source IPv6 address and a tunnel destination IPv6 address when being configured. All packets are encapsulated with an outer IPv6 header and a GRE header.

GRE IPv6 Tunnel Protection

GRE IPv6 tunnel protection allows devices to work as security gateways, establish IPsec tunnels between other security gateway devices, and provide crypto IPsec protection for traffic from internal networks when the traffic is sent across the public IPv6 Internet. The GRE IPv6 tunnel protection functionality is similar to the security gateway model that uses GRE IPv4 tunnel protection.

How to Configure GRE IPv6 Tunnels

Configuring GRE IPv6 Tunnels

Perform this task to configure a GRE tunnel on an IPv6 network. GRE tunnels can be configured to run over an IPv6 network layer and transport IPv6 and IPv4 packets through IPv6 tunnels.

Note You must enable IPv6 or configure IPv6 MTU size more than 1500 on a tunnel's exit interface to avoid receiving warning messages.

Before You Begin

When GRE IPv6 tunnels are configured, IPv6 addresses are assigned to the tunnel source and the tunnel destination. The tunnel interface can have either IPv4 or IPv6 addresses (this is not shown in the task below). The host or device at each end of the configured tunnel must support both IPv4 and IPv6 protocol stacks.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **tunnel source** {*ipv6-address* | *interface-type interface-number*}
5. **tunnel destination** *ipv6-address*
6. **tunnel mode gre ipv6**
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 tunnel** *tunnel-number* **Example:** `Device(config)# interface tunnel 0`	Specifies a tunnel interface and number and enters interface configuration mode.
Step 4	**tunnel source** {*ipv6-address* | *interface-type interface-number*} **Example:** `Device(config-if)# tunnel source ethernet 0`	Specifies the source IPv6 address or the source interface type and number for the tunnel interface. • If an interface type and number are specified, the interface must be configured with an IPv6 address. **Note** Only the syntax used in this context is displayed. For more details, see the IPv6 Command Reference.
Step 5	**tunnel destination** *ipv6-address* **Example:** `Device(config-if)# tunnel destination 2001:0DB8:0C18:2::300`	Specifies the destination IPv6 address for the tunnel interface. **Note** Only the syntax used in this context is displayed. For more details, see the IPv6 Command Reference.
Step 6	**tunnel mode gre ipv6** **Example:** `Device(config-if)# tunnel mode gre ipv6`	Specifies a GRE IPv6 tunnel. **Note** The **tunnel mode gre ipv6** command specifies GRE as the encapsulation protocol for the tunnel interface. Only the syntax used in this context is displayed. For more details, see the IPv6 Command Reference.

	Command or Action	Purpose
Step 7	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring GRE IPv6 Tunnel Protection

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **tunnel source** {*ipv6-address* | *interface-type interface-number*}
5. **tunnel destination** *ipv6-address*
6. **tunnel mode gre ipv6**
7. **tunnel protection ipsec profile** *profile-name*
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 tunnel** *tunnel-number* **Example:** `Device(config)# interface tunnel 0`	Specifies a tunnel interface and number and enters interface configuration mode.
Step 4	**tunnel source** {*ipv6-address* | *interface-type interface-number*}	Specifies the source IPv6 address or the source interface type and number for the tunnel interface.

	Command or Action	Purpose
	Example: `Device(config-if)# tunnel source ethernet 0`	• If an interface type and number are specified, the interface must be configured with an IPv6 address. **Note** Only the syntax used in this context is displayed. For more details, see the IPv6 Command Reference.
Step 5	**tunnel destination** *ipv6-address* **Example:** `Device(config-if)# tunnel destination 2001:0DB8:0C18:2::300`	Specifies the destination IPv6 address for the tunnel interface. **Note** Only the syntax used in this context is displayed. For more details, see the IPv6 Command Reference.
Step 6	**tunnel mode gre ipv6** **Example:** `Device(config-if)# tunnel mode gre ipv6`	Specifies a GRE IPv6 tunnel. **Note** The **tunnel mode gre ipv6** command specifies GRE as the encapsulation protocol for the tunnel interface. Only the syntax used in this context is displayed. For more details, see the IPv6 Command Reference.
Step 7	**tunnel protection ipsec profile** *profile-name* **Example:** `Device(config-if)# tunnel protection ipsec profile ipsec-profile`	Associates the tunnel interface with an IPsec profile. **Note** For the *profile-name* argument, specify the IPsec profile configured in global configuration mode.
Step 8	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for GRE IPv6 Tunnels

Example: Configuring GRE IPv6 Tunnels

The following example shows how to configure a GRE tunnel over an IPv6 transport. In this example, Ethernet0/0 has an IPv6 address, and this is the source address used by the tunnel interface. The destination IPv6 address of the tunnel is specified directly. In this example, the tunnel carries both IPv4 and IS-IS traffic.

```
interface Tunnel0
 ip address 10.1.1.1 255.255.255.0
 ip router isis
 tunnel source Ethernet0/0
 tunnel destination 2001:DB8:1111:2222::1
 tunnel mode gre ipv6
!
interface Ethernet0/0
 no ip address
 ipv6 address 2001:DB8:1111:1111::1/64
!
```

```
router isis
 net 49.0001.0000.0000.000a.00
```

Example: Configuring GRE IPv6 Tunnel Protection

The following example shows how to associate the IPsec profile “ipsec-profile” with a GRE IPv6 tunnel interface. The IPsec profile is configured using the **crypto ipsec profile** command.

```
crypto ipsec profile ipsec-profile
 set transform-set ipsec-profile
!
interface Tunnel1
 ip address 192.168.1.1 255.255.255.252
 tunnel source FastEthernet2/0
 tunnel destination 10.13.7.67
 tunnel protection ipsec profile ipsec-profile
```

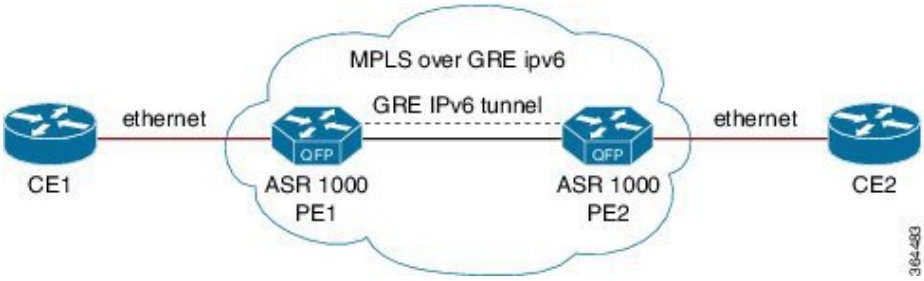
Information About EoMPLS over IPv6 GRE Tunnel

Ethernet over MPLS (EoMPLS) is a tunneling mechanism that allows you to tunnel Layer 2 traffic through a Layer 3 MPLS network. EoMPLS is also known as Layer 2 tunneling.

The EoMPLS over IPv6 GRE Tunnel feature supports tunneling of EoMPLS traffic via an IPv6 network by using GRE tunnels. Effective from Cisco IOS XE Release 3.15s, EoMPLS is supported over IPv6 GRE tunnel.

The following figure shows a deployment model of the EoMPLS over IPv6 GRE Tunnel on a Cisco ASR 1000 Series Aggregation Services Router.

Figure 1: EoMPLS over IPv6 GRE Tunnel Deployment on a Cisco ASR 1000 Series Aggregation Services Router



Configuring EoMPLS over IPv6 GRE Tunnel

EoMPLS over IPv6 GRE Tunnel can be configured in the following two methods:

Using Legacy Commands, on page 6

Using Protocol-based Commands, on page 9

Using Legacy Commands

This section describes how to configure EoMPLS over IPv6 GRE Tunnel using legacy commands. The following are relevant configurations from both Provider Edge 1 Router and Provider Edge 2 Router:

SUMMARY STEPS

1. configure terminal
2. ipv6 unicast-routing
3. mpls label protocol ldp
4. mpls ldp router-id Loopback0 [force]
5. interface *type number*
6. ip address *ip-address mask*
7. interface gigabitethernet slot/port
8. encapsulation dot1 *vlan-id*
9. xconnect *peer-ipaddress vc-id* encapsulation mpls
10. interface tunnel *interface number*
11. ip address *ip-address mask*
12. tunnel source {*ip-address* | *interface-type interface-number*}
13. tunnel mode gre ipv6
14. tunnel destination *ipv6-address*
15. mpls ip
16. interface gigabitethernet slot/port
17. ipv6 address { *ipv6-prefix*/*prefix-length* | *prefix-name sub-bits*/*prefix-length* }

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure terminal **Example:** `Router#configure terminal`	Enters global configuration mode.
Step 2	ipv6 unicast-routing **Example:** `Router(config)#ipv6 unicast-routing`	Enables the forwarding of IPv6 unicast datagrams globally on the router.
Step 3	mpls label protocol ldp **Example:** `Router(config)#mpls label protocol ldp`	Enables Label Distribution Protocol (LDP).
Step 4	mpls ldp router-id Loopback0 [force] **Example:** `Router(config)#mpls ldp router-id Loopback0 [force]`	Configures the LDP Router ID. **Note** The optional force keyword ensures that the IP address on interface loopback 0, and not the IP address of any other interface, becomes the LDP router ID.
Step 5	interface *type number*	Enters configuration mode for the loopback interface.

	Command or Action	Purpose
	Example: `Router(config)#interface Loopback 0`	
Step 6	ip address *ip-address mask* **Example:** `Router(config-if)#ip address 10.1.1.2 255.255.255.255`	Sets the IP address and subnet mask for the loopback interface.
Step 7	interface gigabitethernet slot/port **Example:** `Router(config-if)#interface GigabitEthernet0/0/1.2`	Enters the configuration mode for a Gigabit Ethernet interface on the router.
Step 8	encapsulation dot1 *vlan-id* **Example:** `Router(config-subif)#encapsulation dot1q 200`	Enables 802.1Q trunking on a router.
Step 9	xconnect *peer-ipaddress vc-id* encapsulation mpls **Example:** `Router(config-subif)#xconnect 10.1.1.1 100 encapsulation mpls`	Enables the attachment circuit and specifies the IP address of the peer, a VC ID, and the data encapsulation method.
Step 10	interface tunnel *interface number* **Example:** `Router(config)#interface tunnel 10`	Designates a tunnel interface and enters interface configuration mode.
Step 11	ip address *ip-address mask* **Example:** `Router(config-if)#ip address 41.0.0.1 255.255.255.0`	Sets the IP address and subnet mask for the loopback interface.
Step 12	tunnel source {*ip-address* | *interface-type interface-number*} **Example:** `Router(config-if)#tunnel source GigabitEthernet 0/0/0`	Specifies the source IPv4 address or the source interface type and number for the tunnel interface.
Step 13	tunnel mode gre ipv6 **Example:** `Router (config-if)#tunnel mode gre ipv6`	Specifies that the GRE over IPv6 encapsulation protocol is used in the tunnel.
Step 14	tunnel destination *ipv6-address* **Example:** `Router(config-if)#tunnel destination 2002::2`	Specifies the destination IPv6 address for the tunnel interface.
Step 15	mpls ip	Enables mpls processing on the tunnel interface.

	Command or Action	Purpose
	Example: `Router(config-if)#mpls ip`	
Step 16	interface gigabitethernet slot/port **Example:** `Router(config-if)#interface GigabitEthernet0/0/0`	Enters the configuration mode for a Gigabit Ethernet interface on the router.
Step 17	ipv6 address { *ipv6-prefix*/*prefix-length* | *prefix-name sub-bits*/*prefix-length* } **Example:** `Router(config-if)#ipv6 address 2002::1/112`	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.

Using Protocol-based Commands

This section describes how to configure EoMPLS over IPv6 GRE Tunnel using Protocol-based commands.

SUMMARY STEPS

1. template type pseudowire [*pseudowire-name*]
2. encapsulation mpls
3. end
4. interface pseudowire *number*
5. source template type pseudowire
6. encapsulation mpls
7. neighbor *peer-address vcid-value*
8. end
9. l2vpn xconnect context *context-name*
10. member pseudowire *interface-number*
11. member gigabit ethernet *interface-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	template type pseudowire [*pseudowire-name*] **Example:** `Router(config)# template type pseudowire eompls`	Specifies the name of a Layer 2 pseudowire class and enters pseudowire class configuration mode.
Step 2	encapsulation mpls	Specifies the tunneling encapsulation.

	Command or Action	Purpose
	Example: `Router(config-pw-class)# encapsulation mpls`	
Step 3	end **Example:** `Router(config-pw-class)# end`	Exits to privileged EXEC mode.
Step 4	interface pseudowire *number* **Example:** `Router(config)# interface pseudowire 100`	Specifies the pseudowire interface and enters interface configuration mode.
Step 5	source template type pseudowire **Example:** `Router(config-if)# source template type pseudowire eompls`	Configures the source template of type pseudowire named EoMPLS.
Step 6	encapsulation mpls **Example:** `Router(config-pw-class)# encapsulation mpls`	Specifies the tunneling encapsulation.
Step 7	neighbor *peer-address vcid-value* **Example:** `Router(config-if)# neighbor 154.154.154.154 100`	Specifies the peer IP address and virtual circuit (VC) ID value of a Layer 2 VPN (L2VPN) pseudowire.
Step 8	end **Example:** `Router(config-if)# end`	Exits to privileged EXEC mode.
Step 9	l2vpn xconnect context *context-name* **Example:** `Router(config)# l2vpn xconnect context eompls_100`	Creates a Layer 2 VPN (L2VPN) cross connect context and enters xconnect configuration mode.
Step 10	member pseudowire *interface-number* **Example:** `Router(config-xconnect)# member pseudowire 100`	Specifies a member pseudowire to form a Layer 2 VPN (L2VPN) cross connect.
Step 11	member gigabit ethernet *interface-number* **Example:** `Router(config-xconnect)# member GigabitEthernet0/0/1`	Specifies the location of the Gigabit Ethernet member interface.

Verifying the EoMPLS over IPv6 GRE Tunnel Configuration

Use the following commands to verify that the EoMPLS over IPv6 GRE Tunnel feature is correctly configured.

SUMMARY STEPS

1. show inter tunnel [*tunnel-id*]
2. show xconnect all [detail]
3. show mpls l2transport vc id detail

DETAILED STEPS

	Command or Action	Purpose

Step 1	show inter tunnel [*tunnel-id*]	`Router# show inter tunnel10`
		`Tunnel10 is up, line protocol is up`
		`  Hardware is Tunnel`
		`  Internet address is 41.0.0.2/24`
		`  MTU 1456 bytes, BW 100 Kbit/sec, DLY 50000 usec,`
		`     reliability 255/255, txload 1/255, rxload 1/255`
		`  Encapsulation TUNNEL, loopback not set`
		`  Keepalive not set`
		`  Tunnel linestate evaluation up`
		`  Tunnel source 2002::2 (GigabitEthernet0/0/0), destination 2002::1`
		`   Tunnel Subblocks:`
		`      src-track:`
		`         Tunnel10 source tracking subblock associated with GigabitEthernet0/0/0`
		`          Set of tunnels with source GigabitEthernet0/0/0, 1 member (includes iterators),`
		`on interface <OK>`
		`  Tunnel protocol/transport GRE/IPv6`
		`    Key disabled, sequencing disabled`
		`    Checksumming of packets disabled`
		`  Tunnel TTL 255`
		`  Path MTU Discovery, ager 10 mins, min MTU 1280`
		`  Tunnel transport MTU 1456 bytes`
		`  Tunnel transmit bandwidth 8000 (kbps)`
		`  Tunnel receive bandwidth 8000 (kbps)`
		`  Last input never, output never, output hang never`
		`  Last clearing of "show interface" counters 04:41:12`
		`  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0`
		`  Queueing strategy: fifo`
		`  Output queue: 0/0 (size/max)`
		`  30 second input rate 0 bits/sec, 0 packets/sec`
		`  30 second output rate 0 bits/sec, 0 packets/sec`
		`     8363 packets input, 1074130 bytes, 0 no buffer`
		`     Received 0 broadcasts (0 IP multicasts)`
		`     0 runts, 0 giants, 0 throttles`
		`     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort`
		`     8384 packets output, 1076628 bytes, 0 underruns`
		`     0 output errors, 0 collisions, 0 interface resets`
		`     0 unknown protocol drops`
		`     0 output buffer failures, 0 output buffers swapped out`

Step 2	show xconnect all [detail]	`Router# show xconnect all`
		`Legend:    XC ST=Xconnect State  S1=Segment1 State  S2=Segment2 State`
		`  UP=Up       DN=Down            AD=Admin Down      IA=Inactive`
		`  SB=Standby  HS=Hot Standby     RV=Recovering      NH=No Hardware`
		
		`XC ST  Segment 1                         S1 Segment 2                          S2`
		`------+---------------------------------+--+---------------------------------+--`
		`UP pri   ac Gi0/0/0.2:200(Eth VLAN)      UP mpls 10.1.1.2:100                  UP`
		
		`asr1001#show xconnect all detail`

	Command or Action	Purpose

		`Legend: XC ST=Xconnect State S1=Segment1 State S2=Segment2 State`
		`UP=Up DN=Down AD=Admin Down IA=Inactive`
		`SB=Standby HS=Hot Standby RV=Recovering NH=No Hardware`
		
		`XC ST Segment 1 S1 Segment 2 S2`
		`------+---------------------------------+--+---------------------------------+--`
		`UP pri ac Gi0/0/0.2:200(Eth VLAN) UP mpls 10.1.1.2:100 UP`
		`Interworking: ethernet Local VC label 17`
		`Remote VC label 17`

Step 3	show mpls l2transport vc id detail	`Router# show mpls l2transport vc 100 detail`
		`Local interface: Gi0/0/0.2 up, line protocol up, Eth VLAN 200 up`
		`Interworking type is Ethernet`
		`Destination address: 10.1.1.2, VC ID: 100, VC status: up`
		`Output interface: Tu10, imposed label stack {17}`
		`Preferred path: not configured`
		`Default path: active`
		`Next hop: point2point`
		`Create time: 05:52:23, last status change time: 05:52:07`
		`Last label FSM state change time: 05:52:07`
		`Signaling protocol: LDP, peer 10.1.1.2:0 up`
		`Targeted Hello: 10.1.1.1(LDP Id) -> 10.1.1.2, LDP is UP`
		`Graceful restart: configured and not enabled`
		`Non stop routing: not configured and not enabled`
		`Status TLV support (local/remote) : enabled/supported`
		`LDP route watch : enabled`
		`Label/status state machine : established, LruRru`
		`Last local dataplane status rcvd: No fault`
		`Last BFD dataplane status rcvd: Not sent`
		`Last BFD peer monitor status rcvd: No fault`
		`Last local AC circuit status rcvd: No fault`
		`Last local AC circuit status sent: No fault`
		`Last local PW i/f circ status rcvd: No fault`
		`Last local LDP TLV status sent: No fault`
		`Last remote LDP TLV status rcvd: No fault`
		`Last remote LDP ADJ status rcvd: No fault`
		`MPLS VC labels: local 17, remote 17`
		`Group ID: local 0, remote 0`
		`MTU: local 1500, remote 1500`
		`Remote interface description:`
		`Sequencing: receive disabled, send disabled`
		`Control Word: On (configured: autosense)`
		`SSO Descriptor: 10.1.1.2/100, local label: 17`
		`Dataplane:`
		`SSM segment/switch IDs: 4098/4097 (used), PWID: 1`
		`VC statistics:`
		`transit packet totals: receive 0, send 0`
		`transit byte totals: receive 0, send 0`
		`transit packet drops: receive 0, seq error 0, send 0`

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Commands List, All Releases
Tunnel commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	Interface and Hardware Component Command Reference
IPv6 commands: complete command syntax, command mode, defaults, command history, usage guidelines, and examples	IPv6 Command Reference

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 GRE IPv6 Tunnels

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 1: Feature Information for GRE IPv6 Tunnels

Feature Name	Releases	Feature Information
GRE IPv6 Tunnels	Cisco IOS XE Release 3.7S	The GRE IPv6 Tunnels feature enables the delivery of packets from other protocols through an IPv6 network and allows the routing of IPv6 packets between private networks across public networks with globally routed IPv6 addresses.
EoMPLS over IPv6 GRE Tunnel	Cisco IOS XE Release 3.15S	The EoMPLS over IPv6 GRE Tunnel feature supports tunneling of EoMPLS traffic via an IPv6 network by using GRE tunnels.