Implementing Tunneling for IPv6

Last Updated: July 31, 2012

This module describes how to configure overlay tunneling techniques used by the Cisco IOS software to support the transition from IPv4-only networks to integrated IPv4- and IPv6-based networks. Tunneling encapsulates IPv6 packets in IPv4 packets and uses the IPv4 network as a link-layer mechanism.

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 at the end of this module.

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 Implementing Tunneling for IPv6

- In Cisco IOS Release 12.0(21)ST and Cisco IOS Release 12.0(22)S and earlier releases, the Cisco 12000 series Gigabit Switch Router (GSR) gives a very low priority to the processing of IPv6 tunneled packets. Therefore, we strongly recommend that you limit the use of IPv6 tunnels on the GSR using these releases to topologies that sustain a low level of network traffic and require a minimal amount of process-switching resources.
- IPv6 manually configured tunnel traffic in Cisco IOS Release 12.0(23)S is processed in software on the CPU of the line card, instead of in the Route Processor (RP) in the GSR, resulting in enhanced performance.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

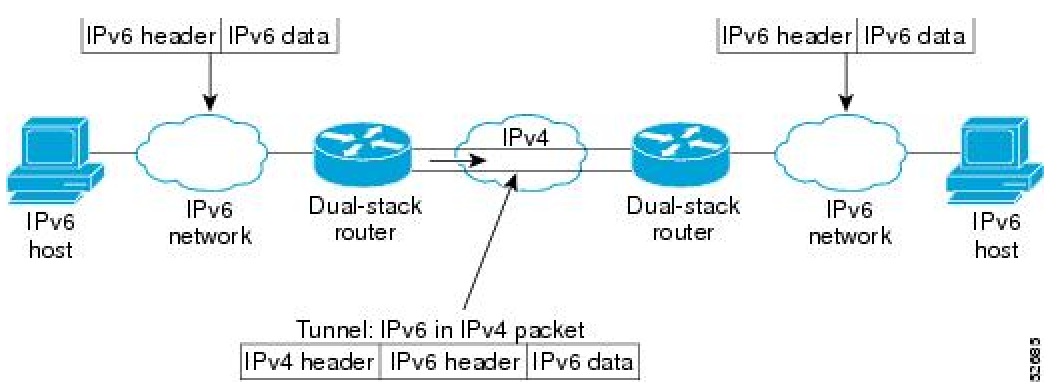
Information About Implementing Tunneling for IPv6

Overlay Tunnels for IPv6

Overlay tunneling encapsulates IPv6 packets in IPv4 packets for delivery across an IPv4 infrastructure (a core network or the Internet (see the figure below). By using overlay tunnels, you can communicate with isolated IPv6 networks without upgrading the IPv4 infrastructure between them. Overlay tunnels can be configured between border routers or between a border router and a host; however, both tunnel endpoints must support both the IPv4 and IPv6 protocol stacks. IPv6 supports the following types of overlay tunneling mechanisms:

- Manual
- Generic routing encapsulation (GRE)
- IPv4-compatible
- 6to4
- Intrasite Automatic Tunnel Addressing Protocol (ISATAP)

Figure 1 Overlay Tunnels



Note Overlay tunnels reduce the maximum transmission unit (MTU) of an interface by 20 octets (assuming the basic IPv4 packet header does not contain optional fields). A network using overlay tunnels is difficult to troubleshoot. Therefore, overlay tunnels connecting isolated IPv6 networks should not be considered as a final IPv6 network architecture. The use of overlay tunnels should be considered as a transition technique toward a network that supports both the IPv4 and IPv6 protocol stacks or just the IPv6 protocol stack.

Use the table below to help you determine which type of tunnel you want to configure to carry IPv6 packets over an IPv4 network.

Table 1 Suggested Usage of Tunnel Types to Carry IPv6 Packets over an IPv4 Network

Tunneling Type	Suggested Usage	Usage Notes
Manual	Simple point-to-point tunnels that can be used within a site or between sites	Can carry IPv6 packets only.
GRE- and IPv4- compatible	Simple point-to-point tunnels that can be used within a site or between sites	Can carry IPv6, Connectionless Network Service (CLNS), and many other types of packets.
IPv4- compatible	Point-to-multipoint tunnels	Uses the ::/96 prefix. We do not now recommend using this tunnel type.
6to4	Point-to-multipoint tunnels that can be used to connect isolated IPv6 sites	Sites use addresses from the 2002::/16 prefix.
6RD	IPv6 service is provided to customers over an IPv4 network by using encapsulation of IPv6 in IPv4.	Prefixes can be from the SP’s own address block.
ISATAP	Point-to-multipoint tunnels that can be used to connect systems within a site	Sites can use any IPv6 unicast addresses.

Individual tunnel types are discussed in detail in this document. We recommend that you review and understand the information about the specific tunnel type that you want to implement. When you are familiar with the type of tunnel you need, see the table below for a summary of the tunnel configuration parameters that you may find useful.

Table 2 Tunnel Configuration Parameters by Tunneling Type

Tunneling Type	Tunnel Configuration Parameter		
Tunnel Mode	Tunnel Source	Tunnel Destination	Interface Prefix or Address

Tunneling Type	Tunnel Configuration Parameter			
Manual	ipv6ip	An IPv4 address, or a reference to an interface on which IPv4 is configured.	An IPv4 address.	An IPv6 address.
GRE/IPv4	gre ip		An IPv4 address.	An IPv6 address.
IPv4- compatible	ipv6ip auto-tunnel		Not required. These are all point-to-multipoint tunneling types. The IPv4 destination address is calculated, on a per-packet basis, from the IPv6 destination.	Not required. The interface address is generated as ::*tunnel-source*/96.
6to4	ipv6ip 6to4			An IPv6 address. The prefix must embed the tunnel source IPv4 address
6RD	ipv6ip 6rd			An IPv6 address.
ISATAP	ipv6ip isatap			An IPv6 prefix in modified eui-64 format. The IPv6 address is generated from the prefix and the tunnel source IPv4 address.

IPv6 Manually Configured Tunnels

A manually configured tunnel is equivalent to a permanent link between two IPv6 domains over an IPv4 backbone. The primary use is for stable connections that require regular secure communication between two edge routers or between an end system and an edge router, or for connection to remote IPv6 networks.

An IPv6 address is manually configured on a tunnel interface, and manually configured IPv4 addresses are assigned to the tunnel source and the tunnel destination. The host or router at each end of a configured tunnel must support both the IPv4 and IPv6 protocol stacks. Manually configured tunnels can be configured between border routers or between a border router and a host. Cisco Express Forwarding switching can be used for IPv6 manually configured tunnels, or Cisco Express Forwarding switching can be disabled if process switching is needed.

GRE IPv4 Tunnel Support for IPv6 Traffic

IPv6 traffic can be carried over IPv4 GRE tunnels using the standard GRE tunneling technique that is designed to provide the services necessary to implement any standard point-to-point encapsulation scheme. As in IPv6 manually configured tunnels, GRE tunnels are links between two points, with a separate tunnel for each link. The tunnels are not tied to a specific passenger or transport protocol, but in this case, carry IPv6 as the passenger protocol with the GRE as the carrier protocol and IPv4 or IPv6 as the transport protocol.

The primary use of GRE tunnels is for stable connections that require regular secure communication between two edge routers or between an edge router and an end system. The edge routers and the end systems must be dual-stack implementations.

GRE Support over IPv6 Transport

GRE has a protocol field that identifies the passenger protocol. GRE tunnels allow Intermediate System-to-Intermediate System (IS-IS) or IPv6 to be specified as a passenger protocol, which allows both IS-IS and IPv6 traffic to run over the same tunnel. If GRE did not have a protocol field, it would be impossible to distinguish whether the tunnel was carrying IS-IS or IPv6 packets. The GRE protocol field is why it is desirable that you tunnel IS-IS and IPv6 inside GRE.

mGRE Tunnels Support over IPv6

To enable service providers deploy IPv6 in their core infrastructure, multipoint generic routing encapsulation (mGRE) tunnels over IPv6 are supported. Dynamic Multipoint Virtual Private Network (DMVPN) customers may run either IPv4 or IPv6 in their local networks, so the overlay endpoints can be either IPv4 or IPv6. For an IPv6 transport endpoint, the overlay endpoint can either be an IPv4 or IPv6 private network address.

GRE has a protocol field that identifies the passenger protocol. GRE tunnels allow Intermediate System-to-Intermediate System (IS-IS) or IPv6 to be specified as a passenger protocol, which allows both IS-IS and IPv6 traffic to run over the same tunnel. If GRE did not have a protocol field, it would be impossible to distinguish whether the tunnel was carrying IS-IS or IPv6 packets. The GRE protocol field is why it is desirable that you tunnel IS-IS and IPv6 inside GRE.

GRE CLNS Tunnel Support for IPv4 and IPv6 Packets

GRE tunneling of IPv4 and IPv6 packets through CLNS networks enables Cisco CLNS Tunnels (CTunnels) to interoperate with networking equipment from other vendors. This feature provides compliance with RFC 3147.

The optional GRE services defined in header fields, such as checksums, keys, and sequencing, are not supported. Any packet received requesting such services will be dropped.

Automatic 6to4 Tunnels

An automatic 6to4 tunnel allows isolated IPv6 domains to be connected over an IPv4 network to remote IPv6 networks. The key difference between automatic 6to4 tunnels and manually configured tunnels is that the tunnel is not point-to-point; it is point-to-multipoint. In automatic 6to4 tunnels, routers are not configured in pairs because they treat the IPv4 infrastructure as a virtual nonbroadcast multiaccess (NBMA) link. The IPv4 address embedded in the IPv6 address is used to find the other end of the automatic tunnel.

An automatic 6to4 tunnel may be configured on a border router in an isolated IPv6 network, which creates a tunnel on a per-packet basis to a border router in another IPv6 network over an IPv4 infrastructure. The tunnel destination is determined by the IPv4 address of the border router extracted from the IPv6 address that starts with the prefix 2002::/16, where the format is 2002:*border-router-IPv4-address* ::/48. Following the embedded IPv4 address are 16 bits that can be used to number networks within the site. The border router at each end of a 6to4 tunnel must support both the IPv4 and IPv6 protocol stacks. 6to4 tunnels are configured between border routers or between a border router and a host.

The simplest deployment scenario for 6to4 tunnels is to interconnect multiple IPv6 sites, each of which has at least one connection to a shared IPv4 network. This IPv4 network could be the global Internet or a corporate backbone. The key requirement is that each site have a globally unique IPv4 address; the Cisco software uses this address to construct a globally unique 6to4/48 IPv6 prefix. As with other tunnel mechanisms, appropriate entries in a Domain Name System (DNS) that map between hostnames and IP addresses for both IPv4 and IPv6 allow the applications to choose the required address.

Automatic IPv4-Compatible IPv6 Tunnels

Automatic IPv4-compatible tunnels use IPv4-compatible IPv6 addresses. IPv4-compatible IPv6 addresses are IPv6 unicast addresses that have zeros in the high-order 96 bits of the address, and an IPv4 address in the low-order 32 bits. They can be written as 0:0:0:0:0:0:A.B.C.D or ::A.B.C.D, where "A.B.C.D" represents the embedded IPv4 address.

The tunnel destination is automatically determined by the IPv4 address in the low-order 32 bits of IPv4-compatible IPv6 addresses. The host or router at each end of an IPv4-compatible tunnel must support both the IPv4 and IPv6 protocol stacks. IPv4-compatible tunnels can be configured between border-routers or between a border-router and a host. Using IPv4-compatible tunnels is an easy method to create tunnels for IPv6 over IPv4, but the technique does not scale for large networks.

IPv6 Rapid Deployment Tunnels

The IPv6 Rapid Deployment (6RD) feature is an extension of the 6to4 feature. The 6RD feature allows a service provider to provide a unicast IPv6 service to customers over its IPv4 network by using encapsulation of IPv6 in IPv4.

The main differences between 6RD and 6to4 tunneling are as follows:

- 6RD does not require addresses to have a 2002::/16 prefix; therefore, the prefix can be from theservice provider’s own address block. This function allows the 6RD operational domain to be within the SP network. From the perspective of customer sites and the general IPv6 Internet connected to a 6RD-enabled service provider network, the IPv6 service provided is equivalent to the native IPv6.
- All 32 bits of the IPv4 destination need not be carried in the IPv6 payload header. The IPv4 destination is obtained from a combination of bits in the payload header and information on the router. Furthermore, the IPv4 address is not at a fixed location in the IPv6 header as it is in 6to4.

ISATAP Tunnels

ISATAP is an automatic overlay tunneling mechanism that uses the underlying IPv4 network as an NBMA link layer for IPv6. ISATAP is designed for transporting IPv6 packets within a site where a native IPv6 infrastructure is not yet available; for example, when sparse IPv6 hosts are deployed for testing. ISATAP tunnels allow individual IPv4 or IPv6 dual-stack hosts within a site to communicate with other such hosts on the same virtual link, basically creating an IPv6 network using the IPv4 infrastructure.

The ISATAP router provides standard router advertisement network configuration support for the ISATAP site. This feature allows clients to automatically configure themselves as they would do if they were connected to an Ethernet. It can also be configured to provide connectivity out of the site. ISATAP uses a well-defined IPv6 address format composed of any unicast IPv6 prefix (/64), which can be link local, or global (including 6to4 prefixes), enabling IPv6 routing locally or on the Internet. The IPv4 address is encoded in the last 32 bits of the IPv6 address, enabling automatic IPv6-in-IPv4 tunneling.

Although the ISATAP tunneling mechanism is similar to other automatic tunneling mechanisms, such as IPv6 6to4 tunneling, ISATAP is designed for transporting IPv6 packets within a site, but not between sites.

Although the ISATAP tunneling mechanism is similar to other automatic tunneling mechanisms, such as IPv6 6to4 tunneling, ISATAP is designed for transporting IPv6 packets within a site, not between sites.

ISATAP uses unicast addresses that include a 64-bit IPv6 prefix and a 64-bit interface identifier. The interface identifier is created in modified EUI-64 format in which the first 32 bits contain the value 000:5EFE to indicate that the address is an IPv6 ISATAP address. The table below describes an ISATAP address format.

Table 3 IPv6 ISATAP Address Format

64 Bits	32 Bits	32 Bits
Link local or global IPv6 unicast prefix	0000:5EFE	IPv4 address of the ISATAP link

As shown in the table above, an ISATAP address consists of an IPv6 prefix and the ISATAP interface identifier. This interface identifier includes the IPv4 address of the underlying IPv4 link. The following example shows what an actual ISATAP address would look like if the prefix is 2001:DB8:1234:5678::/64 and the embedded IPv4 address is 10.173.129.8. In the ISATAP address, the IPv4 address is expressed in hexadecimal as 0AAD:8108 (for example, 2001:DB8:1234:5678:0000:5EFE:0AAD:8108).

IPv6 IPsec Site-to-Site Protection Using Virtual Tunnel Interface

The IPv6 IPsec feature provides IPv6 crypto site-to-site protection of all types of IPv6 unicast and multicast traffic using native IPsec IPv6 encapsulation. The IPsec virtual tunnel interface (VTI) feature provides this function, using IKE as the management protocol.

An IPsec VTI supports native IPsec tunneling and includes most of the properties of a physical interface. The IPsec VTI alleviates the need to apply crypto maps to multiple interfaces and provides a routable interface.

The IPsec VTI allows IPv6 routers to work as security gateways, establish IPsec tunnels between other security gateway routers, and provide crypto IPsec protection for traffic from internal network when being transmitting across the public IPv6 Internet.

For further information on VTIs, see Implementing IPsec in IPv6 Security.

How to Implement Tunneling for IPv6

Configuring Manual IPv6 Tunnels

Perform this task to configure manual IPv6 tunnels.

With manually configured IPv6 tunnels, an IPv6 address is configured on a tunnel interface, and manually configured IPv4 addresses are assigned to the tunnel source and the tunnel destination. The host or router at each end of a configured tunnel must support both the IPv4 and IPv6 protocol stacks.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **ipv6 address** *ipv6-prefix* / *prefix-length* [**eui-64**]
5. **tunnel source** {*ip-address*| *interface-t ype interface-number*}
6. **tunnel destination** *ip-address*
7. **tunnel mode ipv6ip**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *tunnel-number* **Example:** `Router(config)# interface tunnel 0`	Specifies a tunnel interface and number, and enters interface configuration mode.
Step 4	**ipv6 address** *ipv6-prefix* / *prefix-length* [**eui-64**] **Example:** `Router(config-if)# ipv6 address 3ffe:b00:c18:1::3/127`	Specifies the IPv6 network assigned to the interface and enables IPv6 processing on the interface.
Step 5	**tunnel source** {*ip-address*| *interface-t ype interface-number*} **Example:** `Router(config-if)# tunnel source ethernet 0`	Specifies the source IPv4 address or the source interface type and number for the tunnel interface. • If an interface is specified, the interface must be configured with an IPv4 address.

	Command or Action	Purpose
Step 6	**tunnel destination** *ip-address* **Example:** `Router(config-if)# tunnel destination 192.168.30.1`	Specifies the destination IPv4 address or hostname for the tunnel interface.
Step 7	**tunnel mode ipv6ip** **Example:** `Router(config-if)# tunnel mode ipv6ip`	Specifies a manual IPv6 tunnel. **Note** The **tunnel mode ipv6ip** command specifies IPv6 as the passenger protocol and IPv4 as both the encapsulation and transport protocol for the manual IPv6 tunnel.

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 to transport IPv6 packets in IPv6 tunnels and IPv4 packets in IPv6 tunnels.

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 assigned (this is not shown in the task). The host or router at each end of a configured tunnel must support both the IPv4 and IPv6 protocol stacks.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **ipv6 address** *ipv6-prefix* / *prefix-length* [**eui-64**]
5. **tunnel source** {*ip-address* | *ipv6-address* | *interface-type interface-number*}
6. **tunnel destination** {*host-name* | *ip-address* | *ipv6-address*}
7. **tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre**| **gre multipoint** | **gre ipv6** | **ipip** [**decapsulate-any**] | **iptalk** | **ipv6** | **mpls** | **nos**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *tunnel-number* **Example:** `Router(config)# interface tunnel 0`	Specifies a tunnel interface and number, and enters interface configuration mode.
Step 4	**ipv6 address** *ipv6-prefix* / *prefix-length* [**eui-64**] **Example:** `Router(config-if)# ipv6 address 3ffe:b00:c18:1::3/127`	Specifies the IPv6 network assigned to the interface and enables IPv6 processing on the interface.
Step 5	**tunnel source** {*ip-address* | *ipv6-address* | *interface-type interface-number*} **Example:** `Router(config-if)# tunnel source ethernet 0`	Specifies the source IPv4 address or the source interface type and number for the tunnel interface. • If an interface is specified, the interface must be configured with an IPv4 address.
Step 6	**tunnel destination** {*host-name* | *ip-address* | *ipv6-address*} **Example:** `Router(config-if)# tunnel destination 2001:DB8:1111:2222::1/64`	Specifies the destination IPv6 address or hostname for the tunnel interface.
Step 7	**tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre**| **gre multipoint** | **gre ipv6** | **ipip** [**decapsulate-any**] | **iptalk** | **ipv6** | **mpls** | **nos** **Example:** `Router(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.

Configuring Automatic 6to4 Tunnels

Perform this task to configure automatic 6to4 tunnels.

With 6to4 tunnels, the tunnel destination is determined by the border router IPv4 address, which is concatenated to the prefix 2002::/16 in the format 2002:*border-router-IPv4-address*::/48. The border router at each end of a 6to4 tunnel must support both the IPv4 and IPv6 protocol stacks.

Note

The configuration of only one IPv4-compatible tunnel and one 6to4 IPv6 tunnel is supported on a router. If you choose to configure both of those tunnel types on the same router, we strongly recommend that they do not share the same tunnel source.

The reason that a 6to4 tunnel and an IPv4-compatible tunnel cannot share an interface is that both of them are NBMA "point-to-multipoint" access links and only the tunnel source can be used to reorder the packets from a multiplexed packet stream into a single packet stream for an incoming interface. So when a packet with an IPv4 protocol type of 41 arrives on an interface, that packet is mapped to an IPv6 tunnel interface based on the IPv4 address. However, if both the 6to4 tunnel and the IPv4-compatible tunnel share the same source interface, the router is not able to determine the IPv6 tunnel interface to which it should assign the incoming packet.

IPv6 manually configured tunnels can share the same source interface because a manual tunnel is a "point-to-point" link, and both the IPv4 source and IPv4 destination of the tunnel are defined.

>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **ipv6 address** *ipv6-prefix* / *prefix-length* [**eui-64**]
5. **tunnel source** {*ip-address*| *interface-t ype interface-number*}
6. **tunnel mode ipv6ip 6to4**
7. **exit**
8. **ipv6 route** *ipv6-prefix* / *prefix-length* **tunnel** *tunnel-number*

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**interface tunnel** *tunnel-number* **Example:** `Router(config)# interface tunnel 0`	Specifies a tunnel interface and number, and enters interface configuration mode.
Step 4	**ipv6 address** *ipv6-prefix* / *prefix-length* [**eui-64**] **Example:** `Router(config-if)# ipv6 address 2002:c0a8:6301:1::1/64`	Specifies the IPv6 address assigned to the interface and enables IPv6 processing on the interface. • The 32 bits following the initial 2002::/16 prefix correspond to an IPv4 address assigned to the tunnel source.
Step 5	**tunnel source** {*ip-address*| *interface-t ype interface-number*} **Example:** `Router(config-if)# tunnel source ethernet 0`	Specifies the source interface type and number for the tunnel interface. **Note** The interface type and number specified in the **tunnel source** command must be configured with an IPv4 address.
Step 6	**tunnel mode ipv6ip 6to4** **Example:** `Router(config-if)# tunnel mode ipv6ip 6to4`	Specifies an IPv6 overlay tunnel using a 6to4 address.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode, and returns the router to global configuration mode.
Step 8	**ipv6 route** *ipv6-prefix* / *prefix-length* **tunnel** *tunnel-number* **Example:** **Example:** `Router(config)# ipv6 route 2002::/16 tunnel 0`	Configures a static route for the IPv6 6to4 prefix 2002::/16 to the specified tunnel interface. **Note** When configuring a 6to4 overlay tunnel, you must configure a static route for the IPv6 6to4 prefix 2002::/16 to the 6to4 tunnel interface. • The tunnel number specified in the **ipv6 route** command must be the same tunnel number specified in the **interface tunnel**command.

Configuring IPv4-Compatible IPv6 Tunnels

Perform this task to configure IPv4-compatible IPv6 tunnels.

With an IPv4-compatible tunnel, the tunnel destination is automatically determined by the IPv4 address in the low-order 32 bits of IPv4-compatible IPv6 addresses. The host or router at each end of an IPv4-compatible tunnel must support both the IPv4 and IPv6 protocol stacks.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **tunnel source** {*ip-address*| *interface-t ype interface-number*}
5. **tunnel mode ipv6ip auto-tunnel**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *tunnel-number* **Example:** `Router(config)# interface tunnel 0`	Specifies a tunnel interface and number, and enters interface configuration mode.
Step 4	**tunnel source** {*ip-address*| *interface-t ype interface-number*} **Example:** `Router(config-if)# tunnel source ethernet 0`	Specifies the source interface type and number for the tunnel interface. **Note** The interface type and number specified in the **tunnel source** command is configured with an IPv4 address only.
Step 5	**tunnel mode ipv6ip auto-tunnel** **Example:** `Router(config-if)# tunnel mode ipv6ip auto-tunnel`	Specifies an IPv4-compatible tunnel using an IPv4-compatible IPv6 address.

Configuring 6RD Tunnels

Perform this task to configure 6RD tunnels.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **tunnel source** {*ip-address*| *interface-t ype interface-number*}
5. **tunnel mode ipv6ip** [**6rd** | **6to4** | **auto-tunnel** | **isatap**]
6. **tunnel 6rd prefix** *ipv6-prefix* / *prefix-length*
7. **tunnel 6rd ipv4** {**prefix-length** *length*} {**suffix-length** *length*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *tunnel-number* **Example:** `Router(config)# interface tunnel 1`	Specifies a tunnel interface and number, and enters interface configuration mode.
Step 4	**tunnel source** {*ip-address*| *interface-t ype interface-number*} **Example:** `Router(config-if)# tunnel source Ethernet2/0`	Specifies the source interface type and number for the tunnel interface.
Step 5	**tunnel mode ipv6ip** [**6rd** | **6to4** | **auto-tunnel** | **isatap**] **Example:** `Router(config-if)# tunnel mode ipv6ip 6rd`	Configures a static IPv6 tunnel interface.

	Command or Action	Purpose
Step 6	**tunnel 6rd prefix** *ipv6-prefix* / *prefix-length* **Example:** `Router(config-if)# tunnel 6rd prefix 2001:B000::/32`	Specifies the common IPv6 prefix on IPv6 rapid 6RD tunnels.
Step 7	**tunnel 6rd ipv4** {**prefix-length** *length*} {**suffix-length** *length* **Example:** `Router(config-if)# tunnel 6rd ipv4 prefix-length 16 suffix 8`	Specifies the prefix length and suffix length of the IPv4 transport address common to all the 6RD routers in a domain.

Configuring ISATAP Tunnels

The **tunnel source** command used in the configuration of an ISATAP tunnel must point to an interface with an IPv4 address configured. The ISATAP IPv6 address and prefix (or prefixes) advertised are configured as for a native IPv6 interface. The IPv6 tunnel interface must be configured with a modified EUI-64 address because the last 32 bits in the interface identifier are constructed using the IPv4 tunnel source address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **ipv6 address** *ipv6-prefix* / *prefix-length* [**eui-64**]
5. **no ipv6 nd ra suppress**
6. **tunnel source** {*ip-address*| *interface-type interface-number*}
7. **tunnel mode ipv6ip isatap**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *tunnel-number* **Example:** `Router(config)# interface tunnel 1`	Specifies a tunnel interface and number, and enters interface configuration mode.
Step 4	**ipv6 address** *ipv6-prefix* / *prefix-length* [**eui-64**] **Example:** `Router(config-if)# ipv6 address 2001:DB8:6301::/64 eui-64`	Specifies the IPv6 address assigned to the interface and enables IPv6 processing on the interface. **Note** Refer to the *Configuring Basic Connectivity for IPv6* module for more information on configuring IPv6 addresses.
Step 5	**no ipv6 nd ra suppress** **Example:** `Router(config-if)# no ipv6 nd ra suppress`	Sending of IPv6 router advertisements is disabled by default on tunnel interfaces. This command reenables the sending of IPv6 router advertisements to allow client autoconfiguration.
Step 6	**tunnel source** {*ip-address*| *interface-type interface-number*} **Example:** `Router(config-if)# tunnel source ethernet 1/0`	Specifies the source interface type and number for the tunnel interface. **Note** The interface type and number specified in the **tunnel source** command must be configured with an IPv4 address.
Step 7	**tunnel mode ipv6ip isatap** **Example:** `Router(config-if)# tunnel mode ipv6ip isatap`	Specifies an IPv6 overlay tunnel using a ISATAP address.

Verifying IPv6 Tunnel Configuration and Operation

Perform this task to verify IPv6 tunnel configuration and operation.

SUMMARY STEPS

1. **enable**
2. **show interfaces tunnel** *number* [**accounting**]
3. **ping** [*protocol*] *destination*
4. **show ip route** [*address*[*mask*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show interfaces tunnel** *number* [**accounting**] **Example:** `Router# show interfaces tunnel 0`	(Optional) Displays tunnel interface information. • Use the *number* argument to display information for a specified tunnel.
Step 3	**ping** [*protocol*] *destination* **Example:** `Router# ping 10.0.0.1`	(Optional) Diagnoses basic network connectivity.
Step 4	**show ip route** [*address*[*mask*]] **Example:** `Router# show ip route 10.0.0.2`	(Optional) Displays the current state of the routing table. **Note** Only the syntax relevant for this task is shown.

• Examples, page 17

Examples

Sample Output to check remote endpoint address from the ping Command

This example is a generic example suitable for both IPv6 manually configured tunnels and IPv6 over IPv4 GRE tunnels. In the example, two routers are configured to be endpoints of a tunnel. Router A has Ethernet interface 0/0 configured as tunnel interface 0 with an IPv4 address of 10.0.0.1 and an IPv6 prefix of 2001:DB8:1111:2222::1/64. Router B has Ethernet interface 0/0 configured as tunnel interface 1 with an IPv4 address of 10.0.0.2 and an IPv6 prefix of 2001:DB8:1111:2222::2/64. To verify that the tunnel source and destination addresses are configured, use the **show interfaces tunnel** command on Router A.

```
RouterA# show interfaces tunnel 0
```

```
Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 10.0.0.1 (Ethernet0/0), destination 10.0.0.2, fastswitch TTL 255
  Tunnel protocol/transport GRE/IP, key disabled, sequencing disabled
  Tunnel TTL 255
  Checksumming of packets disabled,  fast tunneling enabled
  Last input 00:00:14, output 00:00:04, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue :0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     4 packets input, 352 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
     8 packets output, 704 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Sample Output from the ping Command

To check that the local endpoint is configured and working, use the **ping** command on Router A:

```
RouterA# ping 2001:DB8:1111:2222::2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:1111:2222::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/20/20 ms
```

Sample Output from the show ip route Command

To check that a route exists to the remote endpoint address, use the **show ip route** command:

```
RouterA# show ip route 10.0.0.2
Routing entry for 10.0.0.0/24
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Routing Descriptor Blocks:
  * directly connected, via Ethernet0/0
      Route metric is 0, traffic share count is 1
```

Sample Output from the ping Command

To check that the remote endpoint address is reachable, use the **ping** command on Router A.

Note The remote endpoint address may not be reachable using the **ping** command because of filtering, but the tunnel traffic may still reach its destination.

```
RouterA# ping 10.0.0.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/21/28 ms
```

To check that the remote IPv6 tunnel endpoint is reachable, use the **ping** command again on Router A. The same note on filtering also applies to this example.

```
RouterA# ping 1::2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1::2, timeout is 2 seconds:
```

```
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/20/20 ms
```

These steps may be repeated at the other endpoint of the tunnel.

Configuration Examples for Implementing Tunneling for IPv6

Example: Configuring Manual IPv6 Tunnels

The following example configures a manual IPv6 tunnel between Router A and Router B. In the example, tunnel interface 0 for both Router A and Router B is manually configured with a global IPv6 address. The tunnel source and destination addresses are also manually configured.

Router A Configuration

```
interface ethernet 0
 ip address 192.168.99.1 255.255.255.0
interface tunnel 0
 ipv6 address 3ffe:b00:c18:1::3/127
 tunnel source ethernet 0
 tunnel destination 192.168.30.1
 tunnel mode ipv6ip
```

Router B Configuration

```
interface ethernet 0
 ip address 192.168.30.1 255.255.255.0
interface tunnel 0
 ipv6 address 3ffe:b00:c18:1::2/127
 tunnel source ethernet 0
 tunnel destination 192.168.99.1
 tunnel mode ipv6ip
```

Example Configuring GRE Tunnels

Example: GRE Tunnel Running IS-IS and IPv6 Traffic

The following example configures a GRE tunnel running both IS-IS and IPv6 traffic between Router A and Router B:

Router A Configuration

```
ipv6 unicast-routing
clns routing
!
interface tunnel 0
 no ip address
 ipv6 address 3ffe:b00:c18:1::3/127
 ipv6 router isis
 tunnel source Ethernet 0/0
 tunnel destination 2001:DB8:1111:2222::1/64
 tunnel mode gre ipv6
!
interface Ethernet0/0
 ip address 10.0.0.1 255.255.255.0
!
router isis
 net 49.0000.0000.000a.00
```

Router B Configuration

```
ipv6 unicast-routing
clns routing
!
interface tunnel 0
 no ip address
 ipv6 address 3ffe:b00:c18:1::2/127
 ipv6 router isis
 tunnel source Ethernet 0/0
 tunnel destination 2001:DB8:1111:2222::2/64
 tunnel mode gre ipv6
!
interface Ethernet0/0
 ip address 10.0.0.2 255.255.255.0
!
router isis
 net 49.0000.0000.000b.00
 address-family ipv6
 redistribute static
 exit-address-family
```

Example: Tunnel Destination Address for IPv6 Tunnel

```
Router(config
)
# interface Tunnel0
Router(config
-if)
# no ip address
Router(config
-if)
# ipv6 router isis
Router(config
-if)
# tunnel source Ethernet 0/0
Router(config
-if)
# tunnel destination 2001:DB8:1111:2222::1/64
Router(config
-if)
# tunnel mode gre ipv6
Router(config
-if)
# exit
!
Router(config
)
# interface Ethernet0/0
```

```
Router(config
-if)
# ip address 10.0.0.1 255.255.255.0
Router(config
-if)
# exit
!
Router(config
)
# ipv6 unicast-routing
Router(config
)
# router isis

Router(config
)
# net 49.0000.0000.000a.00
```

Example: Configuring CTunnels in GRE Mode to Carry IPv6 Packets in CLNS

The following example configures a GRE CTunnel running both IS-IS and IPv6 traffic between Router A and Router B in a CLNS network. The **ctunnel mode gre** command allows tunneling between Cisco and third-party networking devices and carries both IPv4 and IPv6 traffic.

The **ctunnel mode gre** command provides a method of tunneling that is compliant with RFC 3147 and allows tunneling between Cisco equipment and third-party networking devices.

Router A

```
ipv6 unicast-routing
clns routing
interface ctunnel 102
 ipv6 address 2001:DB8:1111:2222::1/64
 ctunnel destination 49.0001.2222.2222.2222.00
 ctunnel mode gre
interface Ethernet0/1
 clns router isis
router isis
 net 49.0001.1111.1111.1111.00
```

Router B

```
ipv6 unicast-routing
clns routing
interface ctunnel 201
 ipv6 address 2001:DB8:1111:2222::2/64
 ctunnel destination 49.0001.1111.1111.1111.00
 ctunnel mode gre
interface Ethernet0/1
 clns router isis
router isis
 net 49.0001.2222.2222.2222.00
```

To turn off GRE mode and restore the CTunnel to the default Cisco encapsulation routing only between endpoints on Cisco equipment, use either the **no ctunnel mode** command or the **ctunnel mode cisco** command. The following example shows the same configuration modified to transport only IPv4 traffic.

Example: Configuring 6to4 Tunnels

The following example configures a 6to4 tunnel on a border router in an isolated IPv6 network. The IPv4 address is 192.168.99.1, which translates to the IPv6 prefix of 2002:c0a8:6301::/48. The IPv6 prefix is subnetted into 2002:c0a8:6301::/64 for the tunnel interface: 2002:c0a8:6301:1::/64 for the first IPv6 network, and 2002:c0a8:6301:2::/64 for the second IPv6 network. The static route ensures that any other traffic for the IPv6 prefix 2002::/16 is directed to tunnel interface 0 for automatic tunneling.

```
interface Ethernet0
 description IPv4 uplink
 ip address 192.168.99.1 255.255.255.0
!
interface Ethernet1
 description IPv6 local network 1
 ipv6 address 2002:c0a8:6301:1::1/64
!
interface Ethernet2
 description IPv6 local network 2
 ipv6 address 2002:c0a8:6301:2::1/64
!
interface Tunnel0
 description IPv6 uplink
 no ip address
 ipv6 address 2002:c0a8:6301::1/64
 tunnel source Ethernet 0
 tunnel mode ipv6ip 6to4
!
ipv6 route 2002::/16 tunnel 0
```

Example: Configuring IPv4-Compatible IPv6 Tunnels

The following example configures an IPv4-compatible IPv6 tunnel that allows Border Gateway Protocol (BGP) to run between a number of routers without having to configure a mesh of manual tunnels. Each router has a single IPv4-compatible tunnel, and multiple BGP sessions can run over each tunnel, one to each neighbor. Ethernet interface 0 is used as the tunnel source. The tunnel destination is automatically determined by the IPv4 address in the low-order 32 bits of an IPv4-compatible IPv6 address. Specifically, the IPv6 prefix 0:0:0:0:0:0 is concatenated to an IPv4 address (in the format 0:0:0:0:0:0:A.B.C.D or ::A.B.C.D) to create the IPv4-compatible IPv6 address. Ethernet interface 0 is configured with a global IPv6 address and an IPv4 address (the interface supports both the IPv6 and IPv4 protocol stacks).

Multiprotocol BGP is used in the example to exchange IPv6 reachability information with the peer 10.67.0.2. The IPv4 address of Ethernet interface 0 is used in the low-order 32 bits of an IPv4-compatible IPv6 address and is also used as the next-hop attribute. Using an IPv4-compatible IPv6 address for the BGP neighbor allows the IPv6 BGP session to be automatically transported over an IPv4-compatible tunnel.

```
interface tunnel 0
 tunnel source Ethernet 0
 tunnel mode ipv6ip auto-tunnel
interface ethernet 0
 ip address 10.27.0.1 255.255.255.0
 ipv6 address 3000:2222::1/64
router bgp 65000
 no synchronization
 no bgp default ipv4-unicast
neighbor ::10.67.0.2 remote-as 65002
address-family ipv6
 neighbor ::10.67.0.2 activate
 neighbor ::10.67.0.2 next-hop-self
 network 2001:2222:d00d:b10b::/64
```

Example: Configuring 6RD Tunnels

The following example shows the running configuration of a 6RD tunnel and the corresponding output of the **show tunnel 6rd** command:

```
interface Tunnel1
 ipv6 address 2001:B000:100::1/32
```

```
 tunnel source Ethernet2/1
 tunnel mode ipv6ip 6rd
 tunnel 6rd prefix 2001:B000::/32
 tunnel 6rd ipv4 prefix-len 16 suffix-len 8
end
Router# show tunnel 6rd tunnel 1
Interface Tunnel1:
  Tunnel Source: 10.1.1.1
  6RD: Operational, V6 Prefix: 2001:B000::/32
       V4 Common Prefix Length: 16, Value: 10.1.0.0
       V4 Common Suffix Length: 8, Value: 0.0.0.1
```

Example: Configuring ISATAP Tunnels

The following example shows the tunnel source defined on Ethernet 0 and the **tunnel mode** command used to configure the ISATAP tunnel. Router advertisements are enabled to allow client autoconfiguration.

```
ipv6 unicast-routing
interface tunnel 1
 tunnel source ethernet 0
 tunnel mode ipv6ip isatap
 ipv6 address 2001:DB8::/64 eui-64
 no ipv6 nd ra suppress
 exit
```

Additional References

Related Documents

Related Topic	Document Title
IPsec VTIs	Implementing IPsec in IPv6 Security
IPv6 supported feature list	"Start Here: Cisco IOS Software Release Specifics for IPv6 Features," *Cisco IOS IPv6 Configuration Guide*
CLNS tunnels	*Cisco IOS ISO CLNS Configuration Guide*
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS IPv6 Command Reference*

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

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

RFCs

RFC	Title
RFC 2473	*Generic Packet Tunneling in IPv6 Specification*
RFC 2893	*Transition Mechanisms for IPv6 Hosts and Routers*
RFC 3056	*Connection of IPv6 Domains via IPv4 Clouds*
RFC 4214	*Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)*

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 Implementing Tunneling for IPv6

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 4 Feature Information for Implementing Tunneling for IPv6

Feature Name	Releases	Feature Information
CEFv6 Switching for 6to4 Tunnels	12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(18)SXE 12.2(12)T 12.4 15.0(1)S 15.1(1)SG	Cisco Express Forwarding switching can be used for IPv6 manually configured tunnels.
IPv6 Tunneling--6RD IPv6 Rapid Deployment	15.1(3)T	The 6RD feature allows a service provider to provide a unicast IPv6 service to customers over its IPv4 network by using encapsulation of IPv6 in IPv4.
IPv6 Tunneling--Automatic 6to4 Tunnels	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(33)SRA12.2(18)SXE 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T 15.0(1)S	An automatic 6to4 tunnel allows isolated IPv6 domains to be connected over an IPv4 network to remote IPv6 networks.
IPv6 Tunneling--Automatic IPv4-Compatible Tunnels	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(33)SRA 12.2(18)SXE 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T 15.0(1)S	Automatic IPv4-compatible tunnels use IPv4-compatible IPv6 addresses.
IPv6 Tunneling--IPv6 GRE Tunnels in CLNS Networks	12.2(25)S 12.2(28)SB 12.2(33)SRA 12.3(7)T 12.4 12.4(2)T	GRE tunneling of IPv4 and IPv6 packets through CLNS networks enables Cisco CTunnels to interoperate with networking equipment from other vendors.
IPv6 Tunneling--IP over IPv6 GRE Tunnels	12.2(30)S 12.3(7)T 12.4 12.4(2)T	GRE tunnels are links between two points, with a separate tunnel for each link.
IPv6 Tunneling--IPv4 over IPv6 Tunnels	12.2(30)S 12.2(33)SRA 12.3(7)T 12.4 12.4(2)T 15.0(1)S	IPv6 supports this feature

Feature Name	Releases	Feature Information
IPv6 Tunneling--IPv6 over IPv4 GRE Tunnels	12.0(22)S[1] 12.2(14)S 12.2(28)SB 12.2(33)SRA 12.2(17a)SX1 12.2(4)T 12.3 12.3(2)T 12.4 12.4(2)T 15.0(1)S	GRE tunnels are links between two points, with a separate tunnel for each link. The tunnels are not tied to a specific passenger or transport protocol, but in this case carry IPv6 as the passenger protocol with the GRE as the carrier protocol and IPv4 or IPv6 as the transport protocol.
IPv6 Tunneling--IPv6 over IPv6 Tunnels	12.2(30)S 12.3(7)T 12.4 12.4(2)T	IPv6 supports this feature
IPv6 Tunneling--IPv6 over UTI Using a Tunnel Line Card[2]	12.0(23)S	IPv6 supports this feature.
IPv6 Tunneling--ISATAP Tunnel Support	12.2(14)S 12.2(28)SB 12.2(33)SRA 12.2(17a)SX1 12.2(15)T 12.3 12.3(2)T 12.4 12.4(2)T 15.0(1)S 15.1(1)SG	ISATAP is an automatic overlay tunneling mechanism that uses the underlying IPv4 network as a NBMA link layer for IPv6.
IPv6 Tunneling--Manually Configured IPv6 over IPv4 Tunnels	12.0(23)S[3] 12.2(14)S 12.2(28)SB 12.2(33)SRA 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T 15.0(1)S	A manually configured tunnel is equivalent to a permanent link between two IPv6 domains over an IPv4 backbone.
mGRE Tunnels over IPv6	15.2(1)T	mGRE tunnels are configured to enable service providers deploy IPv6 in their core infrastructure.

1 IPv6 over IPv4 GRE tunnels are not supported on the GSR.

2 Feature is supported on the GSR only.

3 In Cisco IOS Release 12.0(23)S, the GSR provides enhanced performance for IPv6 manually configured tunnels by processing traffic on the line card.