Configuring IP Tunnels

This chapter describes how to configure IP tunnels using Generic Route Encapsulation (GRE) on Cisco NX-OS devices.

This chapter includes the following sections:

- Information About IP Tunnels, page 8-1
- Licensing Requirements for IP Tunnels, page 8-3
- Prerequisites for IP Tunnels, page 8-4
- Guidelines and Limitations, page 8-4
- Default Settings, page 8-4
- Configuring IP Tunnels, page 8-4
- Verifying the IP Tunnel Configuration, page 8-10
- Configuration Examples for IP Tunneling, page 8-10
- Additional References, page 8-11
- Feature History for Configuring IP Tunnels, page 8-12

Information About IP Tunnels

IP tunnels can encapsulate a same-layer or higher layer protocol and transport the result over IP through a tunnel created between two devices.

This section includes the following topics:

- IP Tunnel Overview, page 8-1
- GRE Tunnels, page 8-2
- Path MTU Discovery, page 8-3
- Virtualization Support, page 8-3
- High Availability, page 8-3

IP Tunnel Overview

IP tunnels consists of the following three main components:

- Passenger protocol—The protocol that needs to be encapsulated. IPv4 is an example of a passenger protocol.
- Carrier protocol—The protocol that is used to encapsulate the passenger protocol. Cisco NX-OS supports GRE as a carrier protocol.
- Transport protocol—The protocol that is used to carry the encapsulated protocol. IPv4 is an example of a transport protocol.

An IP tunnel takes a passenger protocol, such as IPv4, and encapsulates that protocol within a carrier protocol, such as GRE. The device then transmits this carrier protocol over a transport protocol, such as IPv4.

You configure a tunnel interface with matching characteristics on each end of the tunnel.

For more information, see the "Configuring IP Tunnels" section on page 8-4.

You must enable the tunnel feature before you can see configure it. Beginning in Cisco NX-OS Release 4.2, the system automatically takes a checkpoint prior to disabling the feature, and you can roll back to this checkpoint. See the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 5.x*, for information on roll backs and checkpoints.

Beginning with Cisco NX-OS Release 4.2, a tunnel configured in one VDC is isolated from a tunnel with the same number configured in another VDC. For example, Tunnel 0 in VDC 1 is independent of tunnel 0 in VDC 2.

Beginning with Cisco NX-OS Release 4.2, your tunnel source IP address and destination IP address should be in the same VRF.

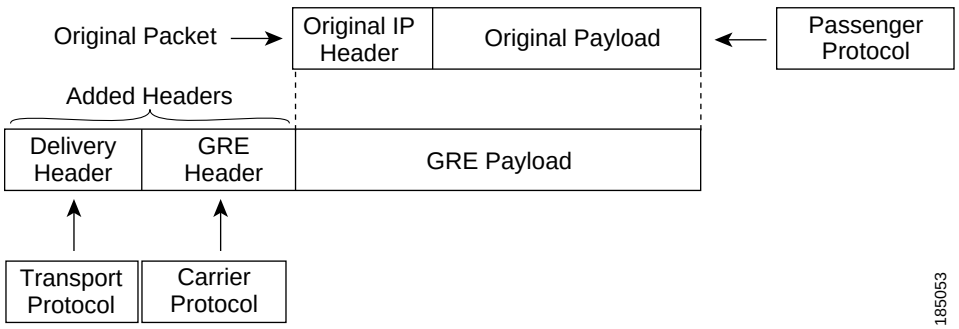
GRE Tunnels

Note Beginning with Cisco NX-OS Release 5.1(1), the software supports multicasting over GRE tunnels.

You can use generic routing encapsulation (GRE) as the carrier protocol for a variety of passenger protocols.

Figure 8-1 shows the IP tunnel components for a GRE tunnel. The original passenger protocol packet becomes the GRE payload and the device adds a GRE header to the packet. The device then adds the transport protocol header to the packet and transmits it.

Figure 8-1 GRE PDU



Path MTU Discovery

Path maximum transmission unit (MTU) discovery (PMTUD) prevents fragmentation in the path between two endpoints by dynamically determining the lowest MTU along the path from the packet's source to its destination. PMTUD reduces the send MTU value for the connection if the interface receives information that the packet would require fragmentation.

When you enable PMTUD, the interface sets the Don't Fragment (DF) bit on all packets that traverse the tunnel. If a packet that enters the tunnel encounters a link with a smaller MTU than the MTU value for the packet, the remote link drops the packet and sends an ICMP message back to the sender of the packet. This message indicates that fragmentation was required (but not permitted) and provides the MTU of the link that dropped the packet.

Note PMTUD on a tunnel interface requires that the tunnel endpoint can receive ICMP messages generated by devices in the path of the tunnel. Check that ICMP messages can be received before using PMTUD over firewall connections.

Virtualization Support

You can configure IP tunnels only in the default virtual device context (VDC) and the default Virtual Routing and Forwarding (VRF) instance.

Beginning with Cisco NX-OS Release 4.2, you can configure a tunnel interface as a member of a Virtual Routing and Forwarding (VRF) instance and as a member of any VDC. By default, Cisco NX-OS places you in the default VDC and default VRF unless you specifically configure another VDC and VRF. A tunnel configured in one VDC is isolated from a tunnel with the same number configured in another VDC. For example, Tunnel 0 in VDC 1 is independent of tunnel 0 in VDC 2.

Your tunnel source IP address and destination IP address should be in the same VRF. You can also configure what VRF to use to look up the tunnel destination. This VRF should match the VRF of the tunnel source IP address.

See the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide, Release 5.x*, for information about VDCs and see the *Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 5.x,* for information about VRFs.

High Availability

IP tunnels support stateful restarts. A stateful restart occurs on a supervisor switchover. After the switchover, Cisco NX-OS applies the runtime configuration after the switchover.

Licensing Requirements for IP Tunnels

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	IP tunnels require an Enterprise Services license. For a complete explanation of the Cisco NX-OS licensing scheme and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide.*

Prerequisites for IP Tunnels

IP tunnels have the following prerequisites:

- You must be familiar with TCP/IP fundamentals to configure IP tunnels.
- You are logged on to the switch.
- You have installed the Enterprise Services license for Cisco NX-OS.
- You must enable the tunneling feature in a device before you can configure and enable any IP tunnels.

Guidelines and Limitations

IP tunnels have the following configuration guidelines and limitations:

- Cisco NX-OS supports the GRE Header defined in IETF RFC 2784. Cisco NX-OS does not support tunnel keys and other options from IETF RFC 1701.
- Beginning with Cisco NX-OS Release 5.2(5) and for later 5.2(x) releases, you can configure the tunnel interface and the tunnel transport in different Virtual Routing and Forward instances (VRF). The tunnels are supported only on M1 Series cards on Cisco Nexus 7000 Series platforms.
- Cisco NX-OS does not support WCCP on tunnel interfaces.
- Cisco NX-OS does not support GRE tunnel keepalives.

Default Settings

Table 8-1 lists the default settings for IP tunnel parameters.

Table 8-1 Default IP Tunnel Parameters

Parameters	Default
Path MTU discovery age timer	10 minutes
Path MTU discovery minimum MTU	64
Tunnel feature	Disabled

Configuring IP Tunnels

This section includes the following topics:

- Enabling Tunneling, page 8-5
- Creating a Tunnel Interface, page 8-5
- Configuring a GRE Tunnel, page 8-7
- Enabling Path MTU Discovery, page 8-8
- Assigning VRF Membership to a Tunnel Interface, page 8-9

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Enabling Tunneling

You must enable the tunneling feature before you can configure any IP tunnels.

SUMMARY STEPS

1. **configure terminal**
2. **feature tunnel**
3. **exit**
4. **show feature**
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	`configure terminal`
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	`feature tunnel`
	
	Example:
	`switch(config)# feature tunnel`
	`switch(config-if)#`
		Allows the creation of a new tunnel interface.
		To disable the tunnel interface feature, use the **no** form of this command.

Step 3	`exit`
	
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface mode and returns to the configuration mode.

Step 4	`show feature`
	
	Example:
	`switch(config-if)# show feature`
		(Optional) Displays information about the features enabled on the device.

Step 5	`copy running-config startup-config`
	
	Example:
	`switch(config-if)# copy running-config startup-config`
		(Optional) Saves this configuration change.

Creating a Tunnel Interface

You can create a tunnel interface and then configure this logical interface for your IP tunnel.

BEFORE YOU BEGIN

Beginning with Cisco NX-OS Release 5.2(5) and for later 5.2(x) releases, you can configure the tunnel source and the tunnel destination in different VRFs. Ensure that you have enabled the tunneling feature.

SUMMARY STEPS

1. **configure terminal**
2. **interface tunnel** *number*
3. **tunnel source** {*ip-address* | *interface-name*}
4. **tunnel destination** {*ip-address* | *host-name*}
5. **tunnel use-vrf** *vrf-name*
6. **show interfaces tunnel** *number*
7. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters configuration mode.

Step 2	**interface tunnel** *number*
	
	Example:
	`switch(config)# interface tunnel 1`
	`switch(config-if)#`
		Creates a new tunnel interface.

Step 3	**tunnel source** {*ip-address* | *interface-name*} **Example:** `switch(config-if)# tunnel source ethernet 1/2`	Configures the source address for this IP tunnel.
Step 4	**tunnel destination** {*ip-address* | *host-name*} **Example:** `switch(config-if)# tunnel destination 192.0.2.1`	Configures the destination address for this IP tunnel.
Step 5	**tunnel use-vrf** *vrf-name* **Example:** `switch(config-if)# tunnel use-vrf blue`	(Optional) Uses the configured VRF to look up the tunnel IP destination address.

	Command	Purpose
Step 6	**show interfaces tunnel** *number* **Example:** `switch(config-if)# show interfaces tunnel 1`	(Optional) Displays the tunnel interface statistics.
Step 7	**copy running-config startup-config** **Example:** `switch(config-if)# copy running-config startup-config`	(Optional) Saves this configuration change.

Use the **no interface tunnel** command to remove the tunnel interface and all associated configuration.

Command	Purpose
no interface tunnel *number* **Example:** `switch(config)# no interface tunnel 1`	Deletes the tunnel interface and the associated configuration.

You can configure the following optional parameters to tune the tunnel in interface configuration mode:

Command	Purpose
description *string* **Example:** `switch(config-if)# description GRE tunnel`	Configures a description for the tunnel.
mtu *value* **Example:** `switch(config-if)# mtu 1400`	Sets the MTU of IP packets sent on an interface.
tunnel ttl *value* **Example:** `switch(config-if)# tunnel ttl 100`	Sets the tunnel time-to-live value. The range is from 1 to 255.

This example shows how to create a tunnel interface:

```
switch# configure terminal
switch(config)# interface tunnel 1
switch(config-if)# tunnel source ethernet 1/2
switch(config-if)# tunnel destination 192.0.2.1
switch(config-if)# copy running-config startup-config
```

Configuring a GRE Tunnel

You can set a tunnel interface to GRE tunnel mode.

BEFORE YOU BEGIN

Ensure that you have enabled the tunneling feature.

SUMMARY STEPS

1. **configure terminal**
2. **interface tunnel** *number*
3. **tunnel mode gre ip**
4. **show interfaces tunnel** *number*
5. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal** **Example:** switch# configure terminal switch(config)#	Enters configuration mode.
Step 2	**interface tunnel** *number* **Example:** switch(config)# interface tunnel 1 switch(config-if)#	Enters a tunnel interface configuration mode.
Step 3	**tunnel mode gre ip** **Example:** switch(config-if)# tunnel mode gre ip	Sets this tunnel mode to GRE
Step 4	**show interfaces tunnel** *number* **Example:** switch(config-if)# show interfaces tunnel 1	(Optional) Displays the tunnel interface statistics.
Step 5	**copy running-config startup-config** **Example:** switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure the tunnel interface to GRE and set the GRE tunnel keepalives:

```
switch# configure terminal
switch(config)# interface tunnel 1
switch(config-if)# tunnel mode gre ip
switch(config-if)# copy running-config startup-config
```

Enabling Path MTU Discovery

Use the **tunnel path-mtu discovery** command to enable path MTU discovery on a tunnel.

Command	Purpose
tunnel path-mtu-discovery [**age-timer** *min*] [**min-mtu** *bytes*] **Example:** switch(config-if)# tunnel path-mtu-discovery 25 1500	Enables Path MTU Discovery (PMTUD) on a tunnel interface. The parameters are as follows: • *mins*—Number of minutes. The range is from 10 to 30. The default is 10. • *mtu-bytes*—Minimum MTU recognized. The range is from 92 to 65535. The default is 92.

Assigning VRF Membership to a Tunnel Interface

You can add a tunnel interface to a VRF.

BEFORE YOU BEGIN

Ensure that you have enabled the tunneling feature.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Assign the IP address for a tunnel interface after you have configured the interface for a VRF.

SUMMARY STEPS

1. **configure terminal**
2. **interface tunnel** *number*
3. **vrf member** *vrf-name*
4. **ip-address** *ip-prefix/length*
5. **show vrf** [*vrf-name*] **interface** *interface-type number*
6. **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	**configure terminal** **Example:** switch# configure terminal switch(config)#	Enters configuration mode.
Step 2	**interface tunnel** *number* **Example:** switch(config)# interface tunnel 0 switch(config-if)#	Enters interface configuration mode.
Step 3	**vrf member** *vrf-name* **Example:** switch(config-if)# vrf member RemoteOfficeVRF	Adds this interface to a VRF.

	Command	Purpose
Step 4	**ip address** *ip-prefix/length* **Example:** switch(config-if)# ip address 192.0.2.1/16	Configures an IP address for this interface. You must do this step after you assign this interface to a VRF.
Step 5	**show vrf** [*vrf-name*] **interface** *interface-type number* **Example:** switch(config-vrf)# show vrf Enterprise interface tunnel 0	(Optional) Displays VRF information.
Step 6	**copy running-config startup-config** **Example:** switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

The following example shows how to add a tunnel interface to the VRF:

```
switch# configure terminal
switch(config)# interface tunnel 0
switch(config-if)# vrf member RemoteOfficeVRF
switch(config-if)# ip address 209.0.2.1/16
switch(config-if)# copy running-config startup-config
```

Verifying the IP Tunnel Configuration

To verify IP tunnel configuration information, perform one of the following tasks:

Command	Purpose
show interface tunnel *number*	Displays the configuration for the tunnel interface (MTU, protocol, transport, and VRF). Displays input and output packets, bytes, and packet rates.
show interface tunnel *number* **brief**	Displays the operational status, IP address, encapsulation type, and MTU of the tunnel interface.
show interface tunnel *number* **description**	Displays the configured description of the tunnel interface.
show interface tunnel *number* **status**	Displays the operational status of the tunnel interface.
show interface tunnel *number* **status err-disabled**	Displays the error disabled status of the tunnel interface.

Configuration Examples for IP Tunneling

The following example shows a simple GRE tunnel. Ethernet 1/2 is the tunnel source for router A and the tunnel destination for router B. Ethernet interface 2/1 is the tunnel source for router B and the tunnel destination for router A.

router A:

```
feature tunnel
interface tunnel 0
  ip address 209.165.20.2/8
  tunnel source ethernet 1/2
  tunnel destination 192.0.2.2
  tunnel mode gre ip
  tunnel path-mtu-discovery 25 1500
interface ethernet1/2
  ip address 192.0.2.55/8
```

router B:

```
feature tunnel
interface tunnel 0
  ip address 209.165.20.1/8
  tunnel source ethernet2/1
  tunnel destination 192.0.2.55
  tunnel mode gre ip
interface ethernet 2/1
ip address 192.0.2.2/8
```

Additional References

For additional information related to implementing IP tunnels, see the following sections:

- Related Documents, page 8-12
- Standards, page 8-12

Related Documents

Related Topic	Document Title
IP Tunnel commands	*Cisco Nexus 7000 Series NX-OS Interfaces Command Reference, Release 5.x*
IP Fragmentation and Path MTU discovery	*Resolve IP Fragmentation, MTU, MSS, and PMTUD Issues with GRE and IPSEC*

Standards

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

Feature History for Configuring IP Tunnels

Table 8-2 lists the release history for this feature.

Table 8-2 Feature History for Configuring IP Tunnels

Feature Name	Releases	Feature Information
Support for tunnel and its transport in different VRFs for M1 series I/O modules only	5.2(5)	This enhancement was supported for NX-OS Release 5.2(5) and for later 5.2(x) releases.
IP tunnels in VDC and VRF other than default	4.2(1)	This feature was introduced.
IP tunnels	4.0(1)	This features was introduced.