Chapter 30

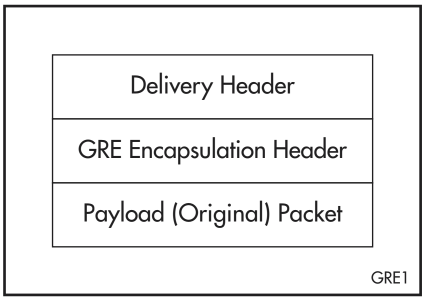
Generic Routing Encapsulation (GRE)

Introduction

Generic Routing Encapsulation (GRE) is a mechanism for encapsulating any network layer protocol over any other network layer protocol. The general specification is described in RFC 1701, and the encapsulation of IP packets over IP is defined in RFC 1702 as a specific implementation of GRE.

In the general case, a network layer packet, called the *payload packet*, is encapsulated in a GRE packet, which may also include source route information. The resulting GRE packet is then encapsulated in some other network layer protocol, called the *delivery protocol*, and then forwarded (see figure below).

Figure 30-1: Format of a packet with GRE encapsulation



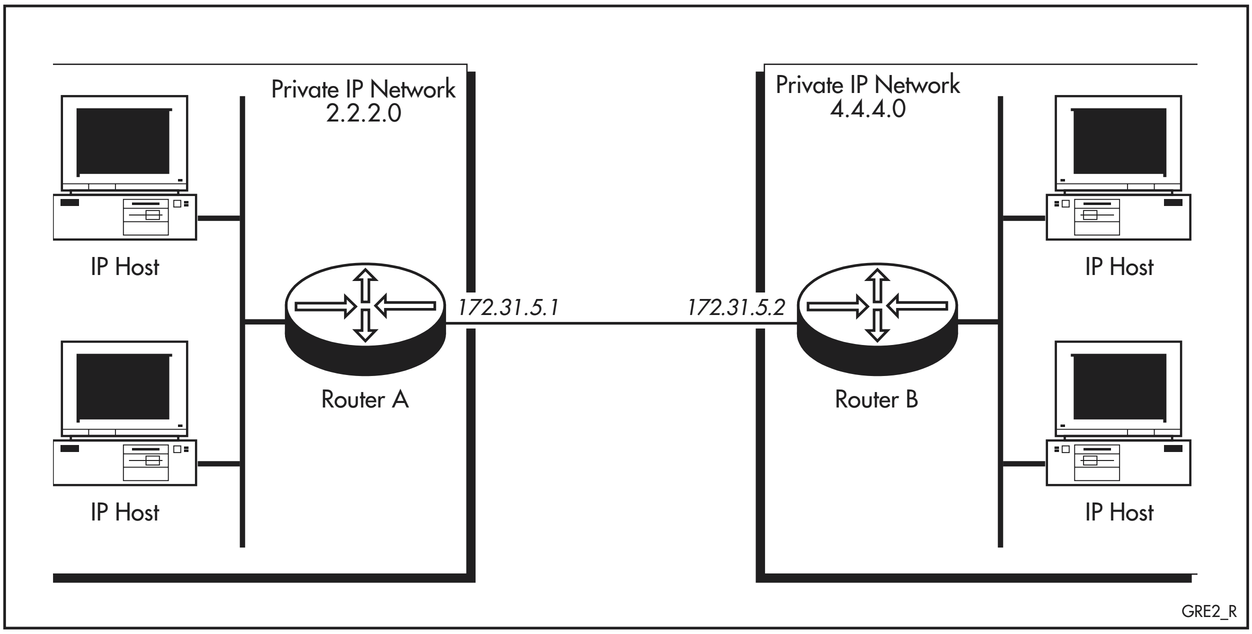
The specific standard for GRE encapsulation is IP over IP (RFC 1702) and this is the standard supported by the router. The main use of the RFC 1702 standard is to route IP packets between private IP networks across an internet that uses globally assigned IP addresses. Private IP networks may either use IP addresses from the ranges of IP addresses reserved for private networks in RFC 1597 (see figure below), or worse, any randomly selected range of IP addresses.

Network Class	**Reserved IP Address Range**
A	10.0.0.0 – 10.255.255.255
B	172.16.0.0 – 172.31.255.255
C	192.168.0.0 – 192.168.255.255

In either case, the administration of a private IP network must ensure that packets using such IP addresses are not transmitted to external networks, to prevent routing conflicts. GRE allows hosts in one private IP network to communicate with hosts in another private IP network by effectively providing a tunnel between two routers across an internet (Figure 30-2 on page 30-3).

In Figure 30-2 on page 30-3 IP packets from the private IP network 2.2.2.0 destined for a host in the private IP network 4.4.4.0 are encapsulated by Router A and forwarded to Router B. Intermediate routers route the packets using addresses in the delivery protocol header. Router B extracts the original payload packet and routes it to the appropriate destination within network 4.4.4.0.

Figure 30-2: Typical scenario for GRE encapsulation of IP over IP



Some interface and port types mentioned in this chapter may not be supported on your router. The interface and port types that are available vary depending on your product's model, and whether an expansion unit (PIC, NSM) is installed. For more information, see the Hardware Reference.

GRE on the Router

The router supports RFC 1702, which defines the encapsulation of IP packets over IP.

Each router that provides GRE encapsulation defines one or more GRE entities. An entity contains a list of one or more patterns specifying a source interface or a source IP address/mask (and optionally a destination IP address/mask), and a target IP address. An entity is created by using the command:

```
add gre=gre-number target=ipadd {interface=interface|
    source=ipadd [smask=ipadd] [destination=ipadd]
    [dmask=ipadd]} [entry=entry-number]
```

The remote end of the tunnel must be defined with the **add gre tunnel** command before you can specify the tunnel's target IP address with the **add gre** command. You have to execute the **add gre tunnel** and **add gre** commands at each end of the tunnel.

You can optionally assign a *tunnel key* to a GRE entity. A tunnel key is a 32-bit number is assigned to both ends of the tunnel. A key is added with the **add gre tunnel** command, and can be modified or deleted with the **set gre tunnel** command. The tunnel key provides a weak form of security because packets injected into the tunnel by an external party are rejected unless they contain the correct tunnel key value. The key also allows packets to travel through specific tunnels in multi-point networks because the key identifies each end of one tunnel.

The router supports both point-to-point and multi-point configurations. In a point-to-point configuration, two private networks are connected across the Internet via a single link, and the GRE entity can be defined using the **interface** or **source** and **smask** parameters. An IP packet received from the source interface or from the source IP address (and mask) is encapsulated using GRE and forwarded to the target IP address.

In a multi-point configuration, multiple private networks are connected across the Internet. Each private network has multiple links to other private networks, and the GRE entity must be defined using the **source**, **smask**, **destination**, and **dmask** parameters. An IP packet received from the source IP address and mask and destined for the destination IP address and mask is encapsulated using GRE and forwarded to the target IP address.

The router specified by the target address decapsulates the GRE packet and forwards the IP packet to its destination. IP packets that do not match patterns in the GRE entity are treated as normal IP packets and processed accordingly.

To modify a GRE entity, use the command:

```
set gre=gre-number entry=entry-number [{interface=interface|
   source=ipadd [smask=ipadd] [destination=ipadd]
   [dmask=ipadd]}] [target=ipadd]
```

To delete a GRE entity, use the command:

```
delete gre=gre-number entry={entry-number|all}
```

To associate a GRE entity with an IP interface, use one of these commands before the entity becomes active:

```
add ip interface=interface ipaddress={ipadd|dhcp}
   [broadcast={0|1}] [directedbroadcast={yes|no|on|off}]
   [filter={0..99|none}] [fragment={yes|no}] [gre={0..100|
   none}] [mask=ipadd] [metric=1..16] [multicast={off|send|
   receive|both|on}] [ospfmetric=1..65534]
   [policyfilter={100..199|none}] [priorityfilter={200..299|
   none}] [proxyarp={on|off}] [ripmetric=1..16]
   [samode={block|passthrough}] [vjc={on|off}]
set ip interface=interface [broadcast={0|1}]
   [directedbroadcast={yes|no|on|off}] [filter={0..99|none}]
   [fragment={yes|no}] [gre={0..100|none}] [ipaddress=ipadd|
   dhcp] [mask=ipadd] [metric=1..16] [multicast={off|send|
   receive|both|on}] [ospfmetric=1..65534]
   [policyfilter={100..199|none}] [priorityfilter={200..299|
   none}] [proxyarp={on|off}] [ripmetric=1..16]
   [samode={block|passthrough}] [vjc={on|off}]
```

A GRE entity may be associated with one or more IP interfaces. However, each IP interface may be associated with only one GRE entity.

IP packets received via the IP interface are checked against the patterns in the GRE entity and if a match is found the IP packet is encapsulated using GRE, and forwarded to the specified target address. At the target address, the router's GRE module extracts the payload packet and forwards it to the IP module for normal processing. If the packet does not match a pattern in the entity, then the packet is treated as an ordinary IP packet and is processed as usual. The order in which the patterns are listed in a GRE entity is an important factor in the efficiency of the encapsulation process.

To enable, disable, or modify GRE encapsulation on a specific IP interface, use the command:

```
set ip interface=interface gre={0..100|none}
```

To enable or disable GRE encapsulation for all IP interfaces on the router, use the commands:

```
enable gre
disable gre
```

Configuration Examples

This sections shows the following configurations:

- Basic Configuration
- Multi-Point Configuration

Basic Configuration

This example shows the steps to configure point-to-point GRE based on the configuration in Figure 30-2 on page 30-3. It assumes that IP has already been configured correctly and is operational on both routers. The following table lists the parameter values in the example.

Parameter	Router A	Router B
Private network IP address	2.2.2.0	4.4.4.0
Private network mask	255.255.255.0	255.255.255.0
Ethernet interface IP address	2.2.2.1	4.4.4.1
Local hosts gateway IP address	2.2.2.1	4.4.4.1
WAN interface IP address	172.31.5.1	172.31.5.2
Target IP address	172.31.5.2	172.31.5.1

1. **Enable the GRE module.**

   The GRE module must be enabled on Router A and Router B by using the following command on each router:

   ```
   enable gre
   ```

2. **Create the GRE tunnel.**

   The tunnel must be defined at both ends, and a key can optionally be assigned to the tunnel.

   On Router A, set the remote end IP address of the tunnel to 172.31.5.2, and create the key "0000ABCD" by using the command:

   ```
   add gre tunnel remote=172.31.5.2 key=0000abcd
   ```

   On Router B, set the remote end IP address of the tunnel to 172.31.5.1, and create the key "0000ABCD" by using the command:

   ```
   add gre tunnel remote=172.31.5.1 key=0000abcd
   ```

3. **Create the GRE entity.**

   A GRE entity must be created on each router to specify the IP packets to be encapsulated and where to forward them.

   On Router A, create a GRE entity to match all IP packets received from the private IP network 2.2.2.0 and forward them to Router B by using the command:

   ```
   add gre=1 source=2.2.2.0 smask=255.255.255.0
      target=172.31.5.2
   ```

   On Router B, create a GRE entity to match all IP packets received from the private IP network 4.4.4.0 and forward them to Router A by using the command:

   ```
   add gre=1 source=4.4.4.0 smask=255.255.255.0
      target=172.31.5.1
   ```

   The IP addresses specified for the TARGET parameter must be valid, globally assigned public IP addresses.

4. **Associate the GRE entity with an IP interface.**

   A GRE entity must be associated with an IP interface before it becomes active. In this example the GRE entity is associated with the router's Ethernet interface that is attached to the private IP network.

   On Router A, associate the GRE entity with IP interface vlan1 by using the command:

   ```
   add ip interface=vlan1 IP=2.2.2.1 gre=1
   ```

   On Router B, associate the GRE entity with IP interface vlan1 by using the command:

   ```
   add ip interface=vlan1 IP=4.4.4.1 gre=1
   ```

5. **Assign the gateway for local hosts.**

   Local hosts in each private IP network must be configured to use the appropriate router as their gateway to the internet and the other private IP network. Hosts in private network 2.2.2.0 need to be configured to use Router A as their gateway, and specifically the router's vlan1 interface that has the IP address 2.2.2.1. Hosts in private network 4.4.4.0 need to be configured to use Router B as their gateway, and specifically the router's vlan1 interface that has the IP address 4.4.4.1.

   The exact method depends on the particular TCP/IP software used on the hosts, but typically for TCP/IP software running on PCs, there is a configuration file on the PC containing a line like:

   ```
   gateway=2.2.2.1; for hosts in private network 2.2.2.0

   gateway=4.4.4.1; for hosts in private network 4.4.4.0
   ```

   Consult the documentation for the TCP/IP software used on the local hosts.

Multi-Point Configuration

This example shows the steps to configure multi-point GRE based on the configuration in Figure 30-3 on page 30-7. It assumes that IP has not already been configured. IP is enabled on each router and IP interfaces are added for the local private IP network and the WAN link to the Internet. Table 30-1 on page 30-8 lists the parameter values used in the example.

Some interface and port types mentioned in this example may not be supported on your router. The interface and port types that are available vary depending on your product's model, and whether an expansion unit (PIC, NSM) is installed. For more information, see the Hardware Reference.

Figure 30-3: Example configuration for multi-point GRE encapsulation of IP over IP

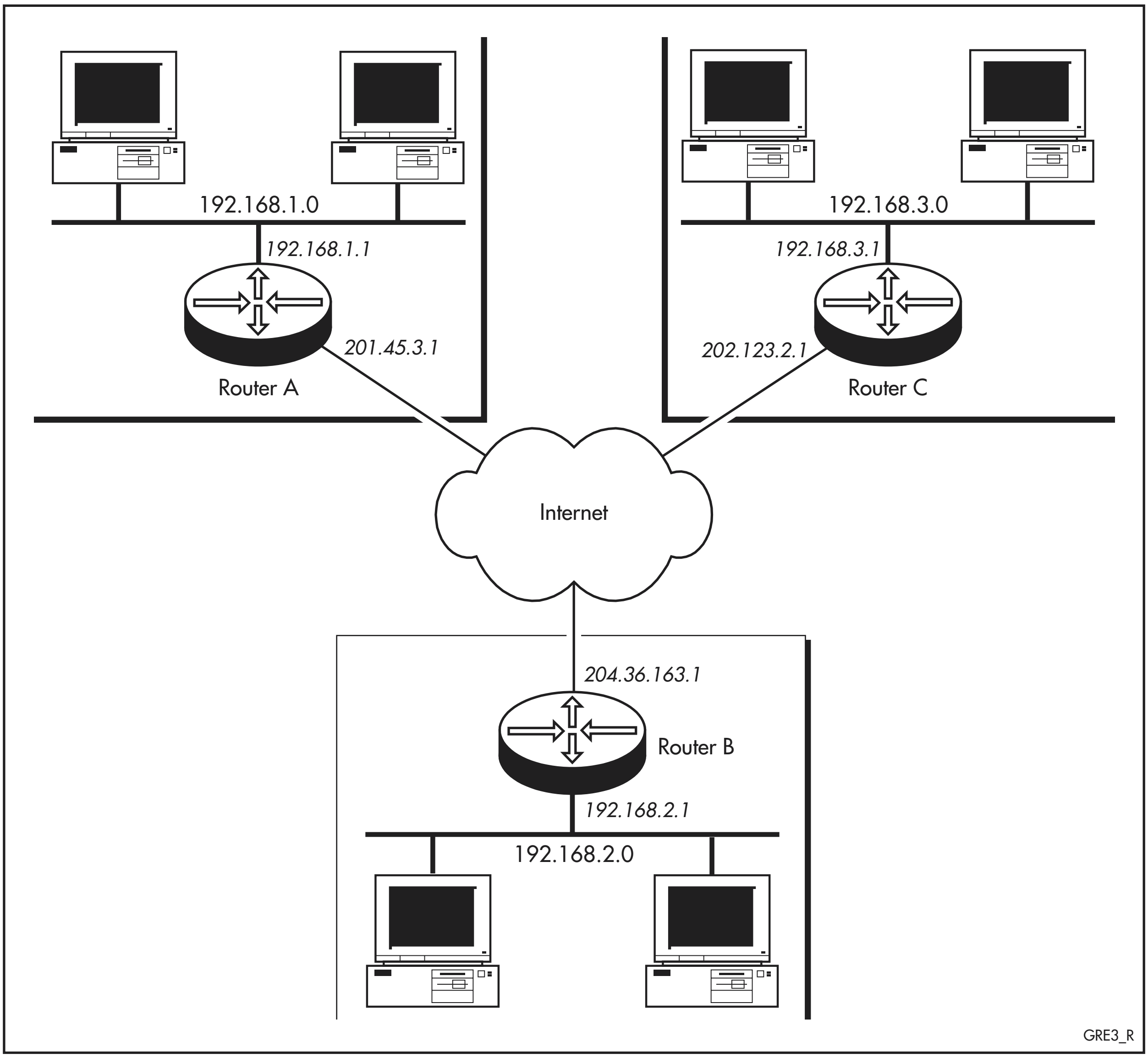


Table 30-1: Example configuration parameters for multi-point GRE

Parameter	**Router A**	**Router B**	**Router C**
Private network IP address	192.168.1.0	192.168.2.0	192.168.3.0
Private network mask	255.255.255.0	255.255.255.0	255.255.255.0
Ethernet interface	`vlan1`	`vlan1`	`vlan1`
Ethernet interface IP address	192.168.1.1	192.168.2.1	192.168.3.1
Local hosts gateway IP address	192.168.1.1	192.168.2.1	192.168.3.1
PPP interface	ppp0	ppp0	ppp0
PPP interface IP address	201.45.3.1	204.36.163.1	202.123.2.1
Target IP addresses	204.36.163.1, 202.123.2.1	201.45.3.1, 202.123.2.1	201.45.3.1, 204.36.163.1

To configure multi-point GRE

1. **Enable the GRE module.**

   The GRE module must be enabled on routers A, B and C by using the following command on each router:

   ```
   enable gre
   ```

2. **Create the GRE tunnels.**

   All of the GRE tunnels within the multi-point configuration must be defined. An optional tunnel key can also be configured if desired.

   On Router A, two tunnels exist. One is between Router A and Router B, and the other is between Router A and Router C. Define the tunnel between Router A and Router B by using the command:

   ```
   add gre tunnel remote=204.36.163.1
   ```

   Define the tunnel between Router A and Router C by using the command:

   ```
   add gre tunnel remote=204.123.2.1
   ```

   On Router B, two tunnels exist. One is between Router B and Router A, and the other is between Router B and Router C. Define the tunnel between Router B and Router A by using the command:

   ```
   add gre tunnel remote=201.45.3.1
   ```

   Define the tunnel between Router B and Router C by using the command:

   ```
   add gre tunnel remote=202.123.2.1
   ```

   On Router C, two tunnels exist. One is between Router C and Router A, and the other is between Router C and Router B. Define the tunnel between Router C and Router A by using the command:

   ```
   add gre tunnel remote=201.45.3.1
   ```

   Define the tunnel between Router C and Router B by using the command:

   ```
   add gre tunnel remote=204.36.163.1
   ```

3. **Create the GRE entity.**

   A GRE entity must be created on each router to specify which IP packets are to be encapsulated, and where to forward the encapsulated packets.

   On Router A, create a GRE entity to match all IP packets from the local private IP network 192.168.1.0 and destined for the remote private IP network 192.168.2.0, and forward them to Router B. Add a second entry to the GRE entity to match all IP packets from the local private IP network 192.168.1.0 and destined for the remote private IP network 192.168.3.0, and forward them to Router C by using the commands:

```
add gre=1 source=192.168.1.0 smask=255.255.255.0
   dest=192.168.2.0 dmask=255.255.255.0
   target=204.36.163.1

add gre=1 source=192.168.1.0 smask=255.255.255.0
   dest=192.168.3.0 dmask=255.255.255.0
   target=202.123.2.1
```

On Router B, create a GRE entity to match all IP packets from the local private IP network 192.168.2.0 and destined for the remote private IP network 192.168.1.0, and forward them to Router A. Add a second entry to the GRE entity to match all IP packets from the local private IP network 192.168.2.0 and destined for the remote private IP network 192.168.3.0, and forward them to Router C by using the commands:

```
add gre=1 source=192.168.2.0 smask=255.255.255.0
   dest=192.168.1.0 dmask=255.255.255.0 target=201.45.3.1

add gre=1 source=192.168.2.0 smask=255.255.255.0
   dest=192.168.3.0 dmask=255.255.255.0
   target=202.123.2.1
```

On Router C, create a GRE entity to match all IP packets from the local private IP network 192.168.3.0 and destined for the remote private IP network 192.168.1.0, and forward them to Router A. Add a second entry to the GRE entity to match all IP packets from the local private IP network 192.168.3.0 and destined for the remote private IP network 192.168.2.0, and forward them to Router B by using the commands:

```
add gre=1 source=192.168.3.0 smask=255.255.255.0
   dest=192.168.1.0 dmask=255.255.255.0 target=201.45.3.1

add gre=1 source=192.168.3.0 smask=255.255.255.0
   dest=192.168.2.0 dmask=255.255.255.0
   target=204.36.163.1
```

The IP addresses specified for the TARGET parameter must be valid, globally assigned public IP addresses.

4. **Associate the GRE entity with an IP interface.**

A GRE entity must be associated with an IP interface before it becomes active. Enable IP routing and add two IP interfaces, one for the Ethernet interface to which the local private IP network is attached, and one for the PPP connection to the Internet. Associate the GRE entity with the Ethernet interface.

On Router A, use the commands:

```
enable ip

add ip interface=vlan1 ip=192.168.1.1 mask=255.255.255.0
   gre=1

create ppp=0 over=syn0

add ip interface=ppp0 ip=201.45.3.1
```

On Router B, use the commands:

```
enable ip

add ip interface=vlan1 ip=192.168.2.1 mask=255.255.255.0
   gre=1

create ppp=0 over=syn0

add ip interface=ppp0 ip=204.36.163.1
```

On Router C, use the commands:

```
enable ip

add ip interface=vlan1 ip=192.168.3.1 mask=255.255.255.0
   gre=1

create ppp=0 over=syn0

add ip interface=ppp0 ip=202.123.2.1
```

5. **Assign the gateway for local hosts.**

   Local hosts in each private IP network must be configured to use the appropriate router as their gateway to the internet and the other private IP network. Hosts in private network 192.168.1.0 need to be configured to use Router A as their gateway, and specifically the router's vlan1 interface that has the IP address 192.168.1.1. Hosts in private network 192.168.2.0 need to be configured to use Router B as their gateway, and specifically the router's vlan1 interface that has the IP address 192.168.2.1.

   The exact method depends on the particular TCP/IP software used on the hosts, but typically for TCP/IP software running on PCs, there is a configuration file on the PC containing a line like:

```
gateway=192.168.1.1; for hosts in network 192.168.1.0

gateway=192.168.2.1; for hosts in network 192.168.2.0

gateway=192.168.3.1; for hosts in network 192.168.3.0
```

   Consult the documentation for the TCP/IP software used on the local hosts.

Command Reference

This section describes the commands available on the router to configure and manage Generic Routing Encapsulation (GRE).

GRE requires the IP module to be enabled and configured correctly. See Chapter 22, Internet Protocol (IP) for detailed descriptions of the commands required to enable and configure IP.

Some interface and port types mentioned in this chapter may not be supported on your router. The interface and port types that are available vary depending on your product's model, and whether an expansion unit (PIC, NSM) is installed. For more information, see the Hardware Reference.

The shortest valid command is denoted by capital letters in the Syntax section. See “Conventions” on page lxv of About this Software Reference in the front of this manual for details of the conventions used to describe command syntax. See Appendix A, Messages for a complete list of error messages and their meanings.

add gre

Syntax

```
ADD GRE=gre-number TARget=ipadd {INTerface=interface|
   SOurce=ipadd [SMask=ipadd] [DEStination=ipadd]
   [DMask=ipadd]} [ENTry=entry-number]
```

where:

- *gre-number* is a decimal number from 1 to 100.
- *interface* is a valid interface.
- *ipadd* is an IP address in dotted decimal notation.
- *entry-number* is a non-zero decimal number.

Description

This command is used to add a pattern to a GRE entity in the encapsulation table. The target specified must correspond to a tunnel that has previously been defined using the **add gre tunnel** command on page 30-14. The exact pattern should not already exist in the GRE entity. The pattern comprises either an interface name or an IP source address and mask to match IP packets against, and a destination IP address for matching packets.

The **gre** parameter specifies the number of the GRE entity to which the pattern is to be added.

The **target** parameter specifies the IP address of an interface on a destination router. The encapsulated IP packets are sent to this address. The IP address must correspond to a previously defined remote end tunnel address, specified with the **remote** parameter of the **add gre tunnel** command on page 30-14. The IP address must also be a valid global (non-private) IP address.

The **interface** parameter specifies the name of an interface used by the IP module. IP packets received via the interface are encapsulated and forwarded using GRE. If **interface** is specified, **source**, **smask**, **destination**, and **dmask** cannot be specified. Valid interfaces are:

- eth (such as eth0)
- PPP (such as ppp0)
- VLAN (such as vlan1)
- FR (such as fr0)
- X.25 DTE (such as x25t0)

The interface must already exist. To see list of all currently available interfaces, use the **show interface** command on page 9-72 of Chapter 9, Interfaces.

The **source** parameter specifies a source IP address in dotted decimal notation. IP packets originating from this IP address are treated as IP packets from a private IP address and are encapsulated. The **source** parameter can be used with the **smask** parameter to specify a range of IP addresses. The **source** and **mask** parameters must be compatible. For each bit in **source** that is zero, the equivalent bit in **mask** must also be zero. If **interface** is specified, **source**, **smask**, **destination** and **dmask** cannot be specified.

The **smask** parameter specifies the network mask, in dotted decimal notation, to be used with the source IP address specified by the **source** parameter. The **source** parameter can be used with the **smask** parameter to specify a range of IP addresses. The **source** and **smask** parameters must be compatible. For each bit in **source** that is zero, the equivalent bit in **mask** must also be zero. If **interface** is specified, **source**, **smask**, **destination** and **dmask** cannot be specified.

The **destination** parameter specifies the destination IP address of the packet to be encapsulated, in dotted decimal notation. IP packets originating from the IP address specified by the **source** parameter *and* destined for this IP address are treated as IP packets from a private IP address and are encapsulated. The **destination** parameter can be used with the **dmask** parameter to specify a range of IP addresses. The **destination** and **dmask** parameters must be compatible. For each bit in **destination** that is zero, the equivalent bit in **dmask** must also be zero. If **interface** is specified, **source**, **smask**, **destination** and **dmask** cannot be specified.

The **dmask** parameter specifies the network mask, in dotted decimal notation, to be used with the destination IP address specified by the **destination** parameter to define a range of IP addresses. The **destination** and **dmask** parameters must be compatible. For each bit in **destination** that is zero, the equivalent bit in **dmask** must also be zero. If **interface** is specified, **source**, **smask**, **destination** and **dmask** cannot be specified.

The **entry** parameter specifies the entry number in the GRE entity that this pattern occupies. Existing patterns with the same or higher entry numbers are pushed down the list. If the number is greater than the total number of patterns in the list, then this pattern is added to the end of the list. The default is to add the pattern to the end of the list.

Examples

To add a pattern to GRE entity 1 to encapsulate IP packets from hosts with IP addresses from 192.168.23.0 to 192.168.23.255 and forward them to the router with IP address 202.50.100.23, use the command:

```
add gre=1 tar=202.50.100.23 so=192.168.23.0 sm=255.255.255.0
```

Related Commands

delete gre
set gre
show gre

add gre tunnel

Syntax `ADD GRE TUNnel REMote=`*`ipadd`*` [KEY={`*`keyval`*`|NONE}]`

where:

- *ipadd* is a valid global (non-private) IP address in dotted decimal notation.
- *keyval* is a hexadecimal number from 1 to ffffffff.

Description This command adds a new tunnel definition, and optionally a tunnel key.

The **remote** parameter specifies the IP address of an interface on a destination router that is to be the remote end of the tunnel. The IP address of the remote end of the tunnel should not match the remote end IP address of any previously defined tunnels.

The **key** parameter assigns a tunnel key number to the tunnel. Once assigned, only packets containing the tunnel key in the GRE header are allowed through the tunnel. The same key must be assigned to both ends of the tunnel. The **none** parameter disables GRE tunnel keys over this tunnel. The default is **none**.

Examples To create a new GRE tunnel with the remote end having the IP address 202.50.100.23 and a tunnel key value of ABCD, use the command:

```
add gre tun rem=202.50.100.23 key=abcd
```

Related Commands

add gre
delete gre
delete gre tunnel
set gre
set gre tunnel
show gre
show gre tunnel

delete gre

Syntax

```
DELete GRE=gre-number ENTry={entry-number|ALL}
```

where:

- *gre-number* is a decimal number from 1 to 100.
- *entry-number* is a non-zero decimal number.

Description

This command is used to delete a pattern or patterns from a GRE entity in the encapsulation table. The pattern must exist in the GRE entity.

The **gre** parameter specifies the number of the GRE entity from which the pattern is to be deleted.

The **entry** parameter specifies the entry number in the GRE entity to be deleted. If **all** is specified, all patterns in the GRE entity are deleted. Existing patterns with the same or higher entry numbers are pushed up the list to fill the vacant entry.

Examples

To delete GRE entry number 3 from the GRE entity 1 pattern list, use:

```
del gre=1 ent=3
```

Related Commands

add gre
set gre
show gre

delete gre tunnel

Syntax

```
DELete GRE TUNnel REMote=ipadd
```

where *ipadd* is a valid global (non-private) IP address in dotted decimal notation

Description

This command is used to delete a previously defined GRE tunnel.

The **remote** parameter specifies the IP address of an interface on a destination router that is the remote end of the tunnel. Currently defined GRE entry patterns cannot have a target IP address that matches the **remote** parameter.

Examples

To delete a GRE tunnel with a remote end IP address of 202.50.100.23, use the command:

```
del gre tun rem=202.50.100.23
```

Related Commands

add gre tunnel
delete gre
set gre
show gre
show gre tunnel

disable gre

Syntax `DISable GRE`

Description This command disables GRE encapsulation on the router. GRE encapsulation must currently be enabled.

Examples To disable GRE module, use the command:

```
dis gre
```

Related Commands enable gre

disable gre debug

Syntax `DISable GRE DEBug`

Description This command disables GRE debugging. GRE debugging must be enabled prior to executing this command. The **purge gre** command also halts GRE debugging. Debugging is disabled by default.

Examples To disable GRE debugging, use the command:

```
dis gre deb
```

Related Commands enable gre debug
purge gre

enable gre

Syntax `ENAble GRE`

Description This command enables GRE encapsulation on the router. GRE encapsulation must currently be disabled.

Examples To enable GRE module, use the command:

```
ena gre
```

Related Commands disable gre

enable gre debug

Syntax

```
ENAble GRE DEBug
```

Description

This command enables GRE debugging when it is disabled. Debugging is disabled by default.

When debugging is enabled, information from GRE headers is displayed in real time whenever GRE packets are sent or received by the router. For GRE packets that are sent or received, the information displayed consists of the GRE version, the payload packet protocol type and the GRE tunnel key. Received packets also show whether the packet was rejected, and if so, the reason why. For detailed debugging of IP over IP GRE packets, use this command in conjunction with the **enable ip debug=packet** command.

Examples

To enable GRE debugging, use the command:

```
ena gre deb
```

Related Commands

disable gre debug
disable ip debug in Chapter 22, Internet Protocol (IP)
enable ip debug in Chapter 22, Internet Protocol (IP)
purge gre

purge gre

Syntax

```
PURge GRE
```

Description

This command resets GRE encapsulation on the router, and purges all GRE configurations and patterns in non-volatile storage.

Examples

To purge GRE module, use the command:

```
pur gre
```

Related Commands

reset gre

reset gre

Syntax RESET GRE

Description This command resets GRE encapsulation on the router. GRE encapsulation must currently be enabled. This has the same effect as entering the command sequence:

```
disable gre
enable gre
```

Examples To reset GRE module, use the command:

```
reset gre
```

Related Commands **disable gre**
enable gre

set gre

Syntax

```
SET GRE=gre-number ENTry=entry-number
   [{INTerface=interface|SOurec=ipadd [SMask=ipadd]
   [DEStination=ipadd] [DMask=ipadd]}] [TARget=ipadd]
```

where:

- *gre-number* is a decimal number from 1 to 100.
- *interface* is a valid interface name.
- *ipadd* is an IP address in dotted decimal notation.
- *entry-number* is a non-zero decimal number.

Description

This command is used to modify a pattern in a GRE entity in the encapsulation table. The pattern must exist in the GRE entity. The pattern comprises either an interface name or an IP source address and mask to match IP packets against, and a destination IP address for matching packets.

The **gre** parameter specifies the number of the GRE entity in which the pattern is to be changed.

The **entry** parameter specifies the entry number in the GRE entity to be changed.

The **interface** parameter specifies the name of an interface used by the IP module that is already assigned and configured. IP packets received via the interface are encapsulated and forwarded using GRE. If **interface** is specified, **source**, **smask**, **destination**, and **dmask** may not be specified. Valid interfaces are:

- eth (such as eth0)
- PPP (such as ppp0)
- VLAN (such as vlan1)
- FR (such as fr0)
- X.25 DTE (such as x25t0)

To see list of all currently available interfaces, use the **show interface** command on page 9-72 of Chapter 9, Interfaces.

The **source** parameter specifies a source IP address in dotted decimal notation. IP packets originating from this IP address are treated as IP packets from a private IP address and are encapsulated. The **source** parameter can be used with the **smask** parameter to specify a range of IP addresses. The **source** and **mask** parameters must be compatible. For each bit in **source** that is zero, the equivalent bit in **mask** must also be zero. If **interface** is specified, **source**, **smask**, **destination**, and **dmask** cannot be specified.

The **smask** parameter specifies the network mask, in dotted decimal notation, to be used with the source IP address specified by the **source** parameter to define a range of IP addresses. The **source** and **smask** parameters must be compatible. For each bit in **source** that is zero, the equivalent bit in **mask** must also be zero. If **interface** is specified, **source**, **smask**, **destination**, and **dmask** cannot be specified.

The **destination** parameter specifies the destination IP address of the packet to be encapsulated, in dotted decimal notation. IP packets originating from the IP address specified by the **source** parameter *and* destined for this IP address are treated as IP packets from a private IP address and are encapsulated. The **destination** parameter can be used with the **dmask** parameter to specify a range of IP addresses. The **destination** and **dmask** parameters must be compatible. For each bit in **destination** that is zero, the equivalent bit in **dmask** must also be zero. If **interface** is specified, **source**, **smask**, **destination,** and **dmask** cannot be specified.

The **dmask** parameter specifies the network mask, in dotted decimal notation, to be used with the destination IP address specified by the **destination** parameter to define a range of IP addresses. The **destination** and **dmask** parameters must be compatible. For each bit in **destination** that is zero, the equivalent bit in **dmask** must also be zero. If **interface** is specified, **source**, **smask**, **destination,** and **dmask** cannot be specified.

The **target** parameter specifies the IP address of an interface on a destination router. The encapsulated IP packets are sent to this address. The IP address must correspond to a previously defined remote end tunnel address, specified with the **remote** parameter in the **add gre tunnel** command on page 30-14. The IP address must also be a valid global (non-private) IP address.

Examples To modify entry 3 in GRE entity 1 to use a target of 192.168.163.45, use the command:

```
set gre=1 ent=3 tar=192.168.163.45
```

Related Commands

delete gre
set gre
show gre

set gre tunnel

Syntax

```
SET GRE TUNnel REMote=ipadd KEY={keyval|NONE}
```

where:

- *ipadd* is a valid global (non-private) IP address in dotted decimal notation.
- *keyval* is a hexadecimal number from 1 to ffffffff.

Description

This command changes the GRE tunnel key details of a previously defined GRE tunnel key.

The **remote** parameter specifies the IP address of an interface on a destination router that is to be the remote end of the tunnel.

The **key** parameter assigns a tunnel key number to the tunnel. Once assigned, only packets containing the tunnel key in the GRE header are allowed through the tunnel. The same key must be assigned to both ends of the tunnel. The **none** parameter disables GRE tunnel keys over this tunnel.

Examples

To set the GRE tunnel key of an existing GRE tunnel that has a remote IP address 202.50.100.23to the value “DCBA”, use the command:

```
add gre tun rem=202.50.100.23 key=dcba
```

Related Commands

add gre
add gre tunnel
delete gre
delete gre tunnel
set gre
show gre
show gre tunnel

show gre

Syntax `SHow GRE[=`*`gre-number`*`]`

Description This command displays information about GRE entities as a pattern list of a specific entity or all entities.

The **gre** parameter specifies the number of the GRE entity to be displayed. If one is not specified, all GRE entities are displayed (Figure 30-4, Table 30-2).

Figure 30-4: Example output from the **show gre** command

```
-----------------------------------------------------------------------------
GRE   Entry   Interface   Source            Destination       Target
                          Source Mask       Dest. Mask        Match
-----------------------------------------------------------------------------
  1     1     -           192.168.1.0       192.168.10.0      202.36.163.5
                          255.255.255.0     255.255.255.0              0
        2     -           192.168.2.0       192.168.8.0       202.36.137.21
                          255.255.255.0     255.255.255.0              0

          Requests:          0            Translations:          0
-----------------------------------------------------------------------------
```

Table 30-2: Parameters in output of the **show gre** command

Parameter	Meaning
GRE	GRE entity number.
Entry	Entry number for this pattern in the GRE entity.
Interface	Interface name for this pattern or a dash.
Source	Source IP address of the packet for this pattern or a dash.
Source Mask	Source network mask for this pattern or a dash.
Destination	Destination IP address of the packet for this pattern or dash.
Dest. Mask	Destination network mask for this pattern or a dash.
Target	Target (router interface) IP address for this pattern.
Match	Number of IP packets processed that matched this pattern.
Requests	Number of IP packets checked against this GRE entity.
Translations	Number of IP packets processed that matched a pattern in this entity and were encapsulated by GRE.

Examples To show the pattern list for GRE entity 1, use the command:

```
sh gre=1
```

Related Commands

add gre
delete gre
show gre general

show gre general

Syntax `SHow GRE GENeral`

Description This command displays general information about received GRE packets (Figure 30-5, Table 30-3).

Figure 30-5: Example output from the **show gre tunnel** command

```
GRE General Information
------------------------------------------------------------
Status .................................. Enabled
Total GRE Pkts In ....................... 0

Total Discarded Packets ................. 0
    Invalid GRE Version Rcvd ............ 0
    Invalid GRE Protocol Type Rcvd ...... 0
    Invalid GRE Key Value Rcvd .......... 0
    GRE Not Enabled Discard Pkts ........ 0
    Tunnel Unconfigured Discard Pkts .... 0
```

Table 30-3: Parameters in output of the **show gre general** command

Parameter	Meaning
Status	Whether GRE is enabled.
Total GRE Pkts In	Total number of packets received containing a GRE header.
Total Discarded Packets	Total number of received GRE packets that have been discarded.
Invalid GRE Version Rcvd	Number of received GRE packets that were discarded due to an invalid GRE version in the header (any version other than zero is invalid).
Invalid GRE Protocol Type Rcvd	Number of received GRE packets discarded due to the payload packet being an unsupported protocol type (any protocol type other than Internet Protocol is invalid).
Invalid GRE Key Value Rcvd	Number of received GRE packets discarded due to an invalid tunnel key in the header.
GRE Not Enabled Discard Pkts	Number of received GRE packets discarded because GRE was not enabled.
Tunnel Unconfigured Discard Pkts	Number of received GRE packets discarded because the local end of the tunnel has not been configured.

Examples To show general information about received GRE packets, use the command:

```
sh gre gen
```

Related Commands

add gre
delete gre
show gre

show gre tunnel

Syntax `SHow GRE TUNnel`

Description This command displays all currently defined GRE tunnels and their associated GRE tunnel keys (Figure 30-6, Table 30-4).

Figure 30-6: Example output from the **show gre tunnel** command

```
---------------------------------------------
Tunnel Remote End      GRE Key
---------------------------------------------
10.1.2.214             0000ABCD
10.1.2.215             None
10.1.2.216             12345678
---------------------------------------------
```

Table 30-4: Parameters displayed in output of the **show gre tunnel** command

Parameter	Meaning
Tunnel Remote End	IP address of the remote end of the tunnel.
GRE Key	Name of the GRE tunnel key associated with the tunnel.

Examples To show all defined GRE tunnels, use the command:

```
sh gre tun
```

Related Commands

purge gre
reset gre
show gre
show gre general