Configuration Guide

5991-3731
December 2005

ProCurve Networking
HP Innovation

Generic Routing Encapsulation (GRE)

GRE Tunneling to Provide Traditional Point-to-Point Functionality over Non-Traditional Network Infrastructures

This configuration guide explains the concepts behind configuring your Secure Router Operating System (SROS) product to use GRE tunneling, defines related commands, and provides sample configurations. For detailed information regarding specific command syntax, refer to the *SROS Command Line Interface Reference Guide* on your *ProCurve SROS Documentation* CD.

This guide consists of the following sections:

- *Understanding the Need for GRE* on page 2
- *Overview of GRE* on page 3
- *Configuring GRE* on page 5
- *Configuration Example* on page 11

Understanding the Need for GRE

True point-to-point networks are not always possible in today's corporate networking environment. Many networks deploy non-traditional methods of connection (for example, DSL or broadband) at remote sites or branch offices. The branch office, telecommuter, or business traveler then becomes separated from the corporate network. Some method of tunneling becomes imperative to connect all the network sites together.

Virtual Private Networking (VPN) is often deployed to create private tunnels through the public network system for passing data to remote sites. While VPN is sufficient for the average business traveler, it is not a good solution for branch site connectivity. VPN configurations must include statically maintained access lists to identify traffic through the tunnel. These access lists are often tedious to configure for larger networks and are prone to errors.

VPNs do not permit multicast traffic to pass; therefore routing protocols such as Routing Information Protocol (RIP) and Open Shortest Path First (OSPF) are no longer options for dynamic routing updates. All new additions to the network topology must be manually added to the various configured access lists. Without dynamic routing from one site to another, network management is severely hampered. Network managers need their non-heterogeneous networks (see Figure 1) to function like traditional point-to-point networks (see Figure 2) so that traditional management methods (once available only on point-to-point circuits) can apply to the entire network.

How do you accomplish this? GRE.

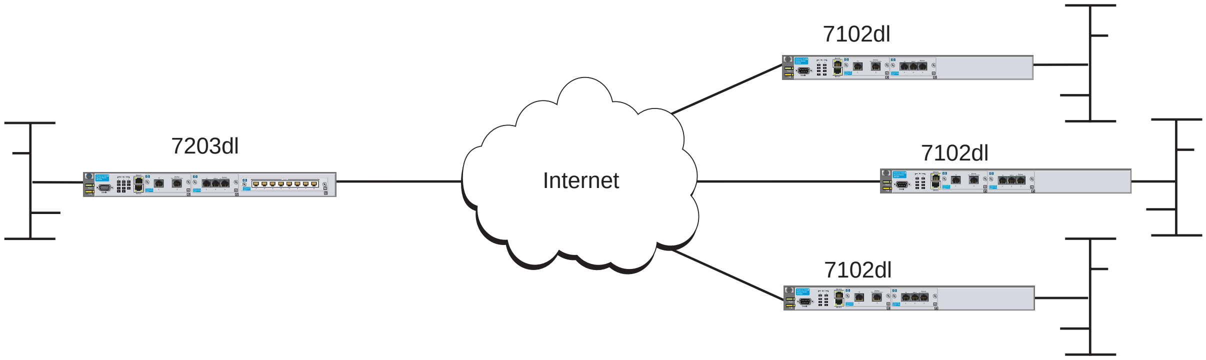


Figure 1. Non-heterogeneous Corporate Network

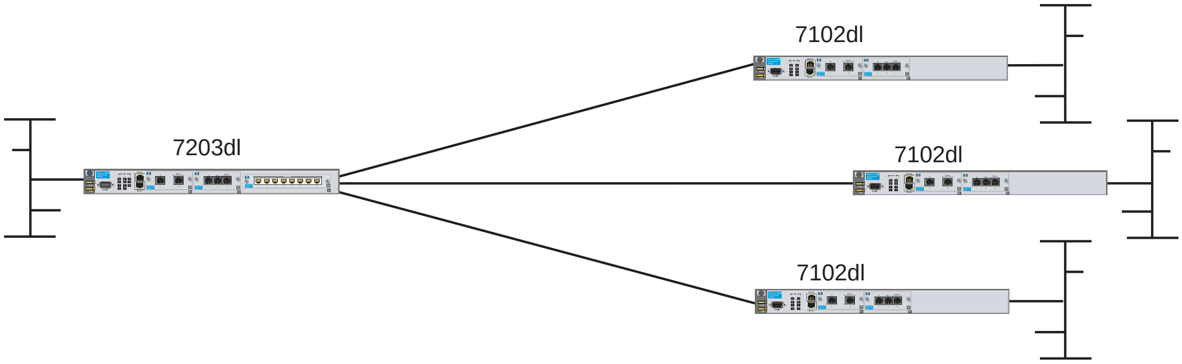
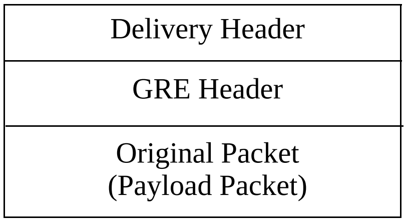


Figure 2. Traditional Point-to-Point Corporate Network

Overview of GRE

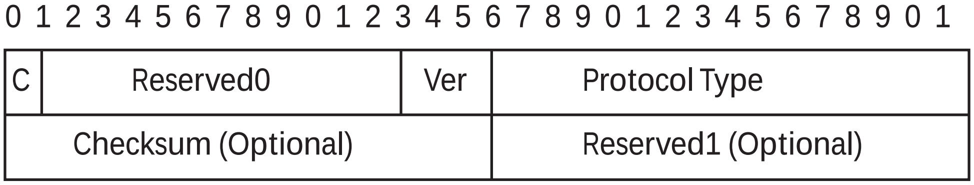
GRE in its simplest form provides a way to encapsulate any network layer protocol over any other network layer protocol. For the SROS implementation, GRE is used to transport IP packets with private IP addresses over the public internet to a remote private IP network. GRE allows routers to act as if they have a virtual point-to-point connection to each other. GRE tunnels allow routing protocols (like RIP and OSPF) to be forwarded to another router across the Internet. In addition, GRE tunnels can encapsulate multicast data streams for transmission over the Internet.

GRE tunneling is accomplished by creating routable tunnel endpoints that operate on top of existing physical and/or other logical endpoints. By design, GRE tunnels connect A to B and provide a clear data path between them. This data path is not secure. IPSec must be used on the GRE tunnels to ensure data security. Data is routed by the system to the GRE endpoint using routes established in the route table. (Note that these routes can be manually established or dynamically learned using routing protocols such as RIP or OSPF.) Once a data packet is received by the GRE endpoint, it is encapsulated in a GRE header and routed again using the endpoint configuration (destination address of the tunnel); therefore each data packet traveling over the GRE tunnel gets routed through the system twice. An IP packet routed through a GRE tunnel looks like this:



GRE Header

The GRE packet header varies in length (depending on options) and can be 32 to 160 bits long. A standard GRE header has the form:



Name	Bit Position	Description
Checksum Present (C)	Bit 0	Specifies whether there is a checksum in the GRE header. When set to 1, the checksum and reserved1 fields (bits 32 to 63) must contain appropriate information. When set to 0, the checksum and reserved1 fields are not necessary.
Reserved0	Bits 1 to 12	Contains several bits to indicate the presence of optional fields in the header (per RFC2890).
Version (Ver)	Bits 13 to 15	Must contain 000 for GRE.
Protocol Type	2 octets	Specifies the protocol type of the original payload packet as defined in RFC1700. For SROS GRE applications, this will always be 0x800 (IP).

Name	Bit Position	Description
Checksum	2 octets	When the checksum present bit is set to 1, the checksum octets contain the IP checksum of the GRE header and payload packet.
Reserved1	2 octets	Reserved for future use. Only present when the checksum present bit is set to 1.

Additional optional fields specified by RFC2890 (key and sequence number) are not discussed in this guide.

GRE Packet Flow

Tracing a packet destined for a network available through the GRE tunnel will provide a better functional understanding of how the SROS GRE implementation works. Using the network diagram in Figure 3 as our example, trace a packet from a node on the 192.168.1.0 network (192.168.1.10) to a node on the 192.168.2.0 network (192.168.2.15).

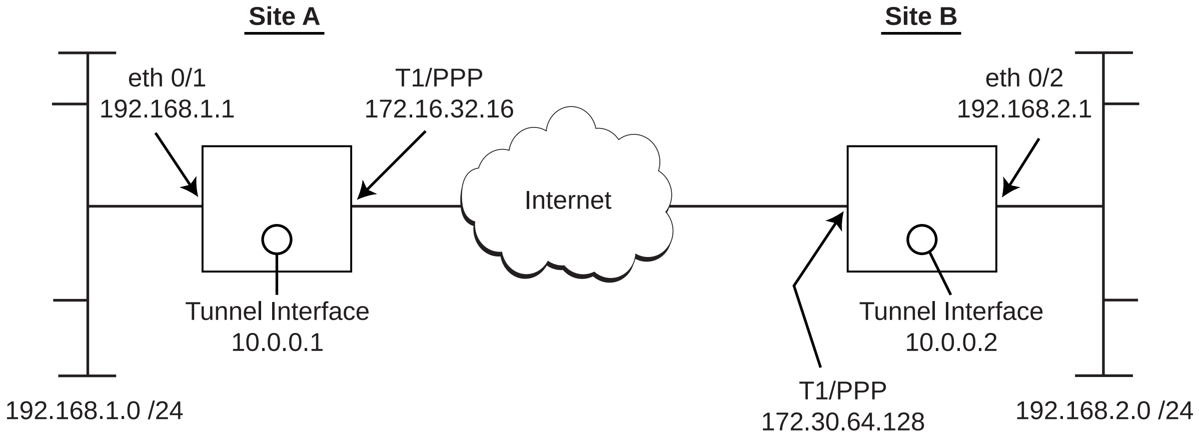


Figure 3. Network Example for GRE Packet Flow in SROS

Step	Description
1	Packet originates from 192.168.1.10.
2	Packet received by the router at Site A on eth 0/1 (192.168.1.1).
3	SROS checks route table for routing information (destination 192.168.2.15) and determines the destination network is available through the tunnel interface.
4	Packet is encased in GRE header with source IP (172.16.32.16) and destination IP (172.30.64.128) and routed back through the route stack.
5	SROS checks route table for routing information for destination IP 172.30.64.128 and routes the packet out the WAN interface.
6	The router at Site B receives the packet on the PPP interface (172.30.64.128). The system recognizes that there is a GRE header on the packet and sends it to the tunnel interface associated with the source and destination IP addresses (172.16.32.16 and 172.30.64.128).
7	GRE header is stripped and the packet is routed through the route stack with a destination IP of 192.168.2.15.
8	Packet is routed out eth 0/2 for delivery.

Configuring GRE

Basic GRE tunnels are created using the following steps:

1. Create a tunnel interface and assign an IP address.
2. Configure the tunnel source and destination IP addresses.
3. Configure additional GRE parameters (optional) such as keepalives, tunnel keys, and checksum verification.
4. Configure the routing information (static routes, RIP, OSPF).

This document provides a discussion of these configuration steps. Each step provides a brief discussion of available settings. In addition, each step provides a sample command listing for a generic configuration. Specific example configurations (with configuration scripts) are provided at the end of this document. For detailed information regarding GRE configuration parameters, refer to the documentation provided on your *ProCurve SROS System Documentation* CD.

Creating Tunnel Interfaces and Assigning an IP Address

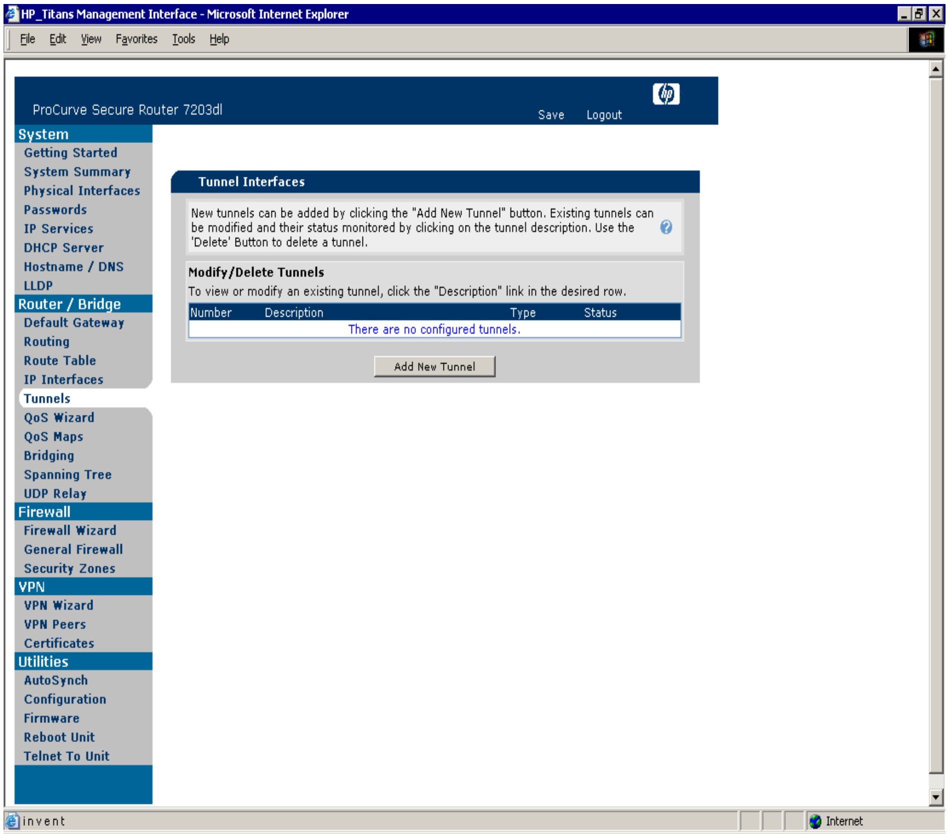
GRE tunnel interfaces are virtual interfaces created within the SROS software. Each tunnel interface must have a numerical label identifier. It is a good idea to include an alphanumeric description string (using the **description** command) on each tunnel interface to act as an identifier. GRE tunnel interfaces need an assigned IP address so that the interface can be added to the routable network. Both endpoints of a GRE tunnel should be given IP addresses on the same subnet.

The following commands create a standard GRE tunnel interface and assign a tunnel IP address:

```
ProCurve>enable
ProCurve#config terminal
ProCurve(config)#interface tunnel 1
ProCurve(config-tunnel 1)#description TunnelToSiteA
ProCurve(config-tunnel 1)#ip address 10.0.0.1 255.255.255.0
ProCurve(config-tunnel 1)#no shutdown
```

Note *All SROS interfaces (physical and virtual) are disabled by default. Use the **no shutdown** command to activate the interface to pass data.*

To create a new GRE tunnel configuration using the Web GUI, select **Tunnels** from the menu list (located on the left side of the screen) and click **Add New Tunnel** to activate the tunnel configuration screen.



Tunnel Interfaces

New tunnels can be added by clicking the "Add New Tunnel" button. Existing tunnels can be modified and their status monitored by clicking on the tunnel description. Use the 'Delete' Button to delete a tunnel.

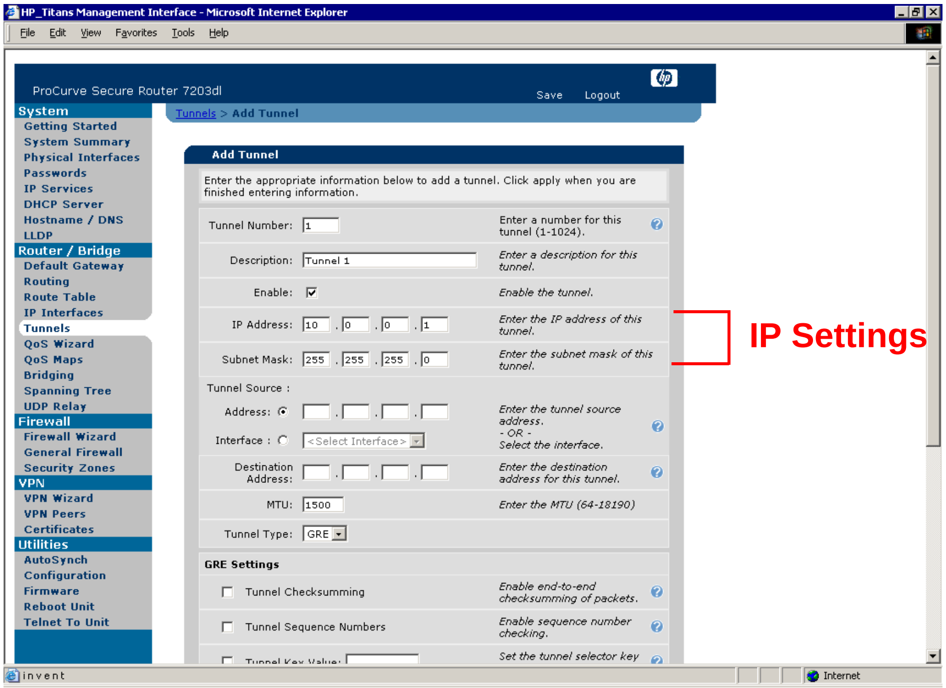
Modify/Delete Tunnels

To view or modify an existing tunnel, click the "Description" link in the desired row.

Number	Description	Type	Status
There are no configured tunnels.			

Add New Tunnel

Enter the IP address and subnet mask for the tunnel. Continue with the next step to configure the tunnel source and destination IP addresses.



Add Tunnel

Enter the appropriate information below to add a tunnel. Click apply when you are finished entering information.

Tunnel Number:	1	*Enter a number for this tunnel (1-1024).*
Description:	Tunnel 1	*Enter a description for this tunnel.*
Enable:	☑	*Enable the tunnel.*
IP Address:	10 . 0 . 0 . 1	*Enter the IP address of this tunnel.*
Subnet Mask:	255 . 255 . 255 . 0	*Enter the subnet mask of this tunnel.*
Tunnel Source :		
Address: ●	. . .	*Enter the tunnel source address.* *- OR -* *Select the interface.*
Interface : ○	<Select Interface>	
Destination Address:	. . .	*Enter the destination address for this tunnel.*
MTU:	1500	*Enter the MTU (64-18190)*
Tunnel Type:	GRE	

GRE Settings

□ Tunnel Checksumming	*Enable end-to-end checksumming of packets.*
□ Tunnel Sequence Numbers	*Enable sequence number checking.*
□ Tunnel Key Value:	*Set the tunnel selector key*

IP Settings

Configuring the Tunnel Source and Destination Addresses

The GRE tunnel source address is the IP address (or interface name, such as **ppp 1**) for the public connection on the local router. Tunnel source addresses are used for the source address of the delivery header and should be addresses assigned to you by your Internet service provider. If you choose to use an interface (such as **ppp 1**) for your tunnel source address, the IP address assigned to that interface is placed in the source address of the delivery header.

Note *If you use an interface (such as **ppp 1**) for the configured tunnel source address, the specified interface must be active for the tunnel to negotiate. If the specified interface is not active, no data will be passed over the GRE tunnel.*

The following command assigns a tunnel source address to the **tunnel 1** interface:

```
ProCurve(config-tunnel 1)#tunnel source 172.16.32.16
```

The following command assigns a tunnel source using a configured PPP interface:

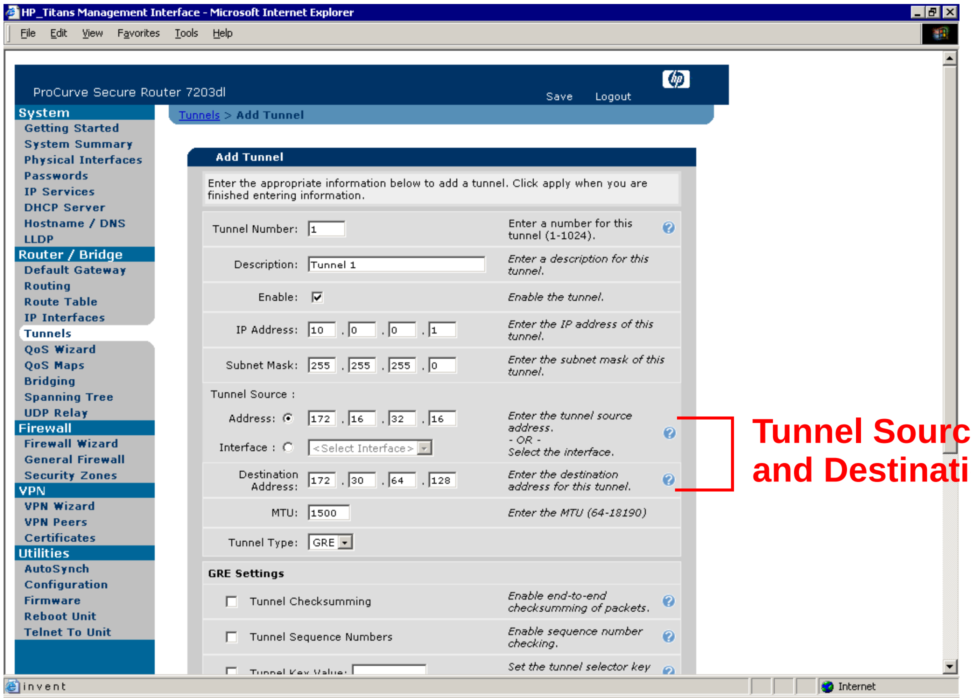
```
ProCurve(config-tunnel 1)#tunnel source ppp 1
```

The GRE tunnel destination address is the IP address for the public connection on the remote router. Tunnel destination addresses are used for the destination address of the delivery header and should be addresses assigned for the remote router by the Internet service provider.

The following command assigns a tunnel source address to the **tunnel 1** interface:

```
ProCurve(config-tunnel 1)#tunnel destination 172.30.64.128
```

The following example configures the tunnel source and destination addresses using the Web GUI. After configuring the tunnel source and destination addresses, click **Apply** (at the bottom of the page) to create the tunnel interface.



Add Tunnel

Enter the appropriate information below to add a tunnel. Click apply when you are finished entering information.

Tunnel Number:	1	Enter a number for this tunnel (1-1024).
Description:	Tunnel 1	*Enter a description for this tunnel.*
Enable:	☑	*Enable the tunnel.*
IP Address:	10 . 0 . 0 . 1	*Enter the IP address of this tunnel.*
Subnet Mask:	255 . 255 . 255 . 0	*Enter the subnet mask of this tunnel.*
Tunnel Source :		
Address: ●	172 . 16 . 32 . 16	*Enter the tunnel source address. - OR - Select the interface.*
Interface : ○	<Select Interface>	
Destination Address:	172 . 30 . 64 . 128	*Enter the destination address for this tunnel.*
MTU:	1500	*Enter the MTU (64-18190)*
Tunnel Type:	GRE	

GRE Settings

□ Tunnel Checksumming	*Enable end-to-end checksumming of packets.*
□ Tunnel Sequence Numbers	*Enable sequence number checking.*
□ Tunnel Key Value:	*Set the tunnel selector key*

Tunnel Source and Destination

Configuring Additional GRE Parameters

GRE Keepalives

GRE tunnels can use periodic status messages, known as keepalives, to verify the integrity of the tunnel from end to end. By default, GRE tunnel keepalives are disabled. Use the **keepalive** command to enable this feature. Keepalives do not have to be configured on both ends of the tunnel in order to work; a tunnel is not aware of incoming keepalive packets.The **keepalive** command has the following syntax:

```
ProCurve(config-tunnel 1)#keepalive <period> <retries>
```

<period>
Defines the time interval (in seconds) between transmitted keepalive packets. Enter a number from 1 to 32,767 seconds.

Note *Each transmitted keepalive packet requires Internet bandwidth for transmission. Though keepalive packets are small, the lower the period, the greater the number of transmitted keepalives and the more bandwidth required.*

<retries>
Defines the number of times to retry after failed keepalives before determining that the tunnel endpoint is down. Enter a number from 1 to 255 times.

The following command activates GRE tunnel keepalives, configures a period of 1 minute, and specifies that the tunnel should be considered down after 5 failed keepalives.

```
ProCurve(config-tunnel 1)#keepalive 60 5
```

Tunnel Key

A tunnel key is shared by both endpoints of the tunnel to help delineate between tunnels with the same source and destination addresses. When enabled, the tunnel key information is stored in the GRE header and the key present bit is set to indicate to the receiver that a tunnel key was used. A matching tunnel key value must be used on both ends of the tunnel or received tunnel packets are discarded.

Note *Tunnel keys do not provide security for the tunnel. Key information is sent (in clear text) in each GRE packet. To secure the tunnel information, you must use IPSec.*

The **tunnel key** command has the following syntax:

```
ProCurve(config-tunnel 1)#tunnel key <value>
```

<value>
Defines the key value for this tunnel. Enter a number from 1 to 4,294,967,294.

Checksum

Tunnel checksums are used to verify the integrity of incoming GRE packets. Checksums are not intended as a security method for data; rather they help recognize unintentional errors created in data during transmission. When enabled, the tunnel checksum is calculated for each outgoing GRE packet and the result is stored in the GRE header. The checksum present bit is set in the GRE header to indicate to the router at the other end of the tunnel that a checksum was used. Both ends of the tunnel must have **tunnel checksum** enabled for proper operation. When both endpoints have **tunnel checksum** enabled, a packet with an incorrect checksum is dropped. If the endpoints differ in their checksum configurations, all packets flow without checksum verification. The **tunnel checksum** command has the following syntax:

```
ProCurve(config-tunnel 1)#tunnel checksum
```

Sequence Datagrams

Tunnel sequence datagrams activate sequence number checking on incoming GRE packets, allowing the tunnel to drop packets arriving out of order. Both ends of the tunnel must have sequence numbering enabled. When both endpoints have sequence numbering enabled, a packet arriving with a sequence number less than the current expected value will be dropped. If the endpoints differ in their sequence numbering configuration, all packets will still flow without any sequence number verification. Be careful enabling sequence number verification on a tunnel because out-of-order sequences can occur due to network conditions outside the tunnel endpoints. It may be difficult to establish a successful traffic flow after an out-of-sequence condition occurs.The **tunnel sequence-datagrams** command has the following syntax:

```
ProCurve(config-tunnel 1)#tunnel sequence-datagrams
```

Configuring Routing Information

SROS products support various routing protocols including static routes, RIP, OSPF, and Border Gateway Protocol (BGP). RIP, OPSF, and BGP are all routing protocols which allow routers to share the information contained in their route tables with other routers in the network. These routing protocols are generally used on networks that frequently change or contain a large number of nodes. For small applications, manually adding static routes to the router’s route table is the easiest method of configuration.

Recursive Routing

Recursive routing can be a problem in GRE applications utilizing dynamic routing protocols such as RIP and OSPF. Typical two-site GRE applications involve each site learning at least two routes between them. The first is the route to the remote-site LAN through the GRE tunnel. The second is the route to the remote-site public IP address through the normal gateway. In instances where the redistribute connected command is used, sites can add incorrect information to the route table by adding a route to the remote-site public IP address through the GRE tunnel. When this occurs, a recursive routing loop is formed. When packets from the local network are destined for the remote-site LAN, the route table tells us that the packet should be encapsulated in a GRE header and sent through the tunnel. This encapsulated packet is then sent back through the routing mechanism with the remote-site public IP address as the destination. The route table uses the incorrect routing information to route the packet through the tunnel interface again. This situation provides the potential for the packet to be encapsulated for the GRE tunnel an infinite number of times. To avoid this situation, if an SROS GRE tunnel endpoint receives an IP packet to encapsulate that already contains a GRE header, the tunnel is automatically shut down.

Static Route Example

Manually adding static routes to the route table requires two steps.

1. Determine the routes needed (destination address and subnet mask, as well as the next-hop address or forwarding interface). Be sure to plan the default route.
2. Use the **ip route** command in the Global configuration mode to add the route to the route table.

The following lists the complete syntax for the **ip route** command:

ip route *<ip address> <subnet mask> <interface or ip address> <administrative distance>*

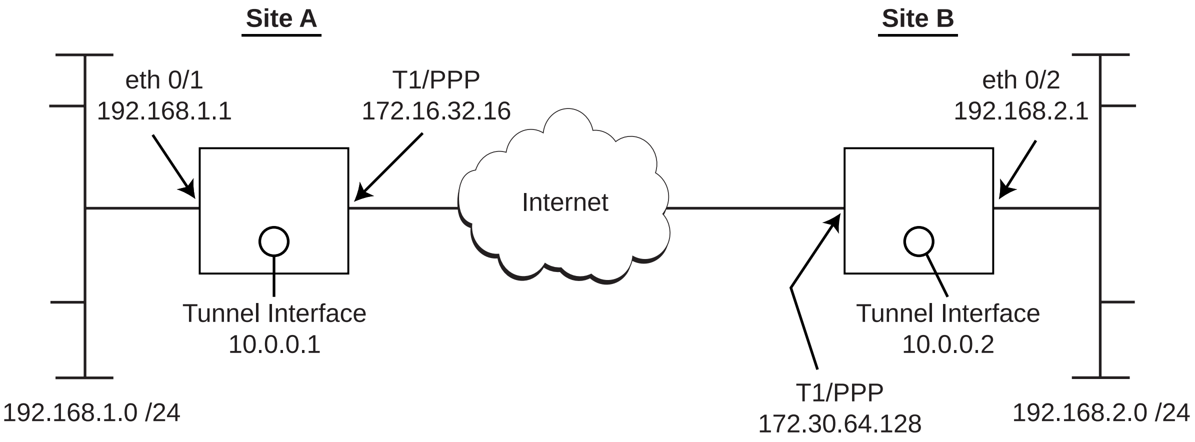
<ip address>
Specifies the network address (in dotted decimal notation) to add to the route table.

<subnet mask>
Specifies the subnet mask (in dotted decimal notation) associated with the listed network IP address.

<interface or ip address>
Specifies the gateway peer IP address (in dotted decimal notation) or a configured interface in the unit.

<administrative distance>
Specifies an administrative distance associated with this router. Valid range is 1 to 255. The administrative distance provides a way for a router to determine the best route when multiple routes to the same destination exist. The smaller the administrative distance, the more reliable the route.

Let's review the following example to illustrate the static route creation process.



The following table outlines the static routes needed in the Site A router.

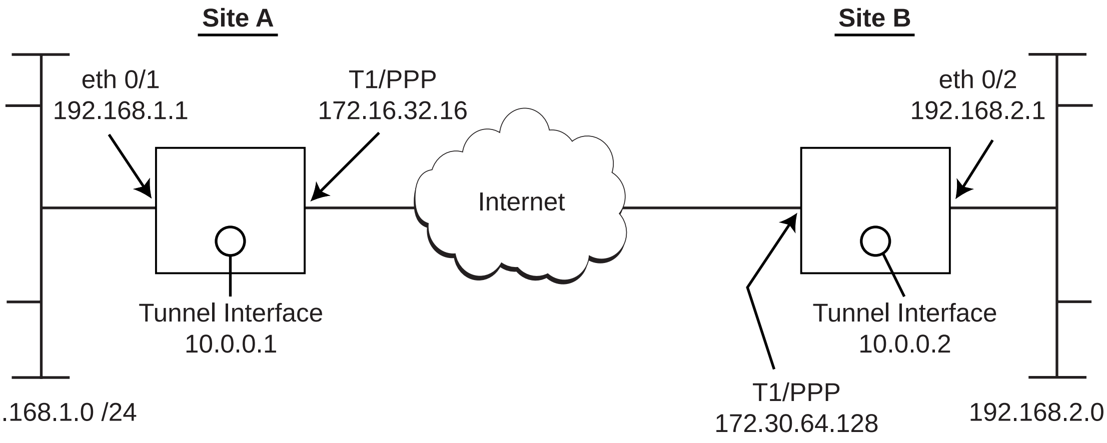
Destination Address	Subnet Mask	Next-Hop Address/Forwarding Interface
192.168.2.0	255.255.255.0	tunnel 1
Default Route (0.0.0.0)	0.0.0.0	Public Internet Connection (ppp 1 or 172.16.32.16)

Remote end tunnel interfaces do not need to be manually added to the route table. These interfaces show up as being directly connected to the local unit.

The following commands add the necessary static routes to Site A router’s route table:

```
ProCurve>enable
ProCurve#config terminal
ProCurve(config)#ip route 192.168.2.0 255.255.255.0 tunnel 1
ProCurve(config)#ip route 0.0.0.0 0.0.0.0 172.16.32.16
```

Configuration Example



Example Script

```
!
!
hostname "Site A Router"
enable password md5 encrypted 7f1a02ddd2cf3df129eb99c8408a5e28
!
ip firewall
no ip firewall alg h323
ip firewall alg sip
!
!
!
interface eth 0/1
  ip address  192.168.1.1  255.255.255.0
  access-policy NAT
  no shutdown
!
!
interface t1 3/1
  tdm-group 1 timeslots 1-24 speed 64
  no shutdown
!
!
interface ppp 1
  ip address  64.128.32.16  255.255.255.0
  no shutdown
  bind 1 t1 3/1 1 ppp 1
```

```
!
!
interface tunnel 1
  description TunnelToSiteB
  ip address  10.0.0.1  255.255.255.0
  tunnel mode gre
  tunnel source 64.128.32.16
  tunnel destination 64.128.32.56
  mtu 1476
  bandwidth 1536
  no shutdown
!
!
ip access-list extended TRUSTED
  permit ip 192.168.1.0 0.0.0.255  any
!
ip policy-class NAT
  nat source list TRUSTED interface ppp 1 overload
!
!
!
ip route 172.10.10.0 255.255.255.0 tunnel 1
ip route 192.168.2.0 255.255.255.0 tunnel 1
!
!
end
```