

How GRE Keepalives Work

Document ID: 63760

Introduction

Prerequisites

- Requirements
- Components Used
- Conventions

Background Information

The Tunnel Keepalive Mechanism

- Functional Description
- Memory and Performance Impact
- Packaging Considerations
- Commands and Configuration
- Sample Output and Screen Formats

Related Information

Introduction

This document provides an overview of how Generic Routing Encapsulation (GRE) keepalives work.

Prerequisites

Requirements

Readers of this document should have knowledge of these topics:

- GRE Tunnel Keepalive
- Keepalive Configuration Mode Commands

Components Used

The information in this document is based on these software and hardware versions:

- Cisco 7505 Router
- Cisco IOS® Software that supports GRE over IPSec

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

Background Information

The GRE keepalive feature enables the **keepalive** interface command for tunnels, and allows you to configure keepalives for point-to-point GRE tunnels. You can configure keepalives with the **keepalive** command, and

optionally with its new extension.

GRE tunnels provide a method to encapsulate arbitrary packets inside a transport protocol. They also offer an architecture designed to provide the services required to implement any standard point-to-point encapsulation scheme. Here are some of the advantages of GRE tunnels:

- GRE tunnels provide multi-protocol local networks over a single-protocol backbone.
- GRE tunnels provide workarounds for networks that contain protocols with limited hop counts.
- GRE tunnels connect discontinuous sub-networks.
- GRE tunnels allow VPNs across WANs.

However, in the current implementation of GRE tunnels, a configured tunnel does not have the ability to bring down the line protocol of either tunnel endpoint, if the far end is unreachable. Thus, the traffic sent from the tunnel is black-holed, and it cannot follow alternative paths because the tunnel always stays up.

This situation is true for tunnels that rely on static routes or on routing protocols that aggregate routes to find a route to the tunnel destination. It is also true in situations where the data in the control plane follows a different path from the data in the data plane.

The Tunnel Keepalive Mechanism

This section provides a functional description for the tunnel keepalive mechanism with the help of an example. This section also lists the software elements that this feature modifies, and discusses the impact on memory and performance.

Functional Description

The tunnel keepalive mechanism enables, extends and implements an interface-specific command for tunnel interfaces, and delivers the ability to bring down the line protocol of a tunnel. For more information, see the Commands and Configuration section.

The tunnel keepalive mechanism also addresses these additional requirements:

- The tunnel keepalive mechanism functions even if the far tunnel endpoint does not support keepalives.
- The tunnel keepalive mechanism originates keepalives.
- The tunnel keepalive mechanism processes keepalives.
- The tunnel keepalive mechanism replies to keepalive packets of the far end, even when the line protocol of the tunnel is down.

Here is an example of how the tunnel keepalive mechanism works (see Figure 1):

Figure 1 Example for the Tunnel Keepalive Mechanism



Output

```

interface tunnel 0
ip address 1.1.1.1 255.255.255.240
tunnel source 128.8.8.8
tunnel destination 129.9.9.9
keepalive 5 4
interface loopback 0
ip address 128.8.8.8 255.255.255.255

interface tunnel 0
ip address 1.1.1.2 255.255.255.240
tunnel source 129.9.9.9
tunnel destination 128.8.8.8
keepalive 5 4
interface loopback 0
ip address 129.9.9.9 255.255.255.255

```

A keepalive packet that originates from A to B

```

---outer IP header---'      ---inner IP header---'
=====
|IP | IP src | IP dst | GRE | IP | IP src | IP dst | GRE |
|  |128.8.8.8|129.9.9.9|PT=IP|  |129.9.9.9|128.8.8.8| PT=0|
=====
                        ----'          ---'
                        GRE header      GRE header

```

When you enable keepalives on the tunnel endpoint of Router A, the router at every interval constructs the inner IP header. At the end of the header, the router also appends a GRE header with a Protocol Type (PT) of 0, and no other payload. The router then sends that packet through the tunnel, which results in its encapsulation with the outer IP header, and a GRE header with the PT of IP. The tunnel keepalive counter increments by one. If there is a way to reach the far end tunnel endpoint, and the tunnel line protocol is not down due to other reasons, the packet arrives on Router B. It is then matched against Tunnel 0, is decapsulated, and forwarded to the destination IP, which is the tunnel source, Router A. Upon arrival on Router A, the packet is again decapsulated, and the PT is checked. If the result of the PT check is 0, it signifies that this is a keepalive packet. In such a case, the tunnel keepalive counter is reset to 0, and the packet is discarded.

In case Router B is unreachable, Router A continues to construct and send the keepalive packets along with normal traffic. If the line protocol is down, the keepalives do not come back to Router A. Therefore, the keepalive counter continues to increase. The tunnel line protocol stays up only as long as the tunnel keepalive counter remains zero, or less than a configured value. If that condition is not true, the next time you attempt to send a keepalive to Router B, the line protocol is brought down, as soon as the keepalive counter reaches the configured keepalive value. In the up/down state, the tunnel does not forward or process any traffic apart from the keepalive packets. For this to work for keepalive packets only, the tunnel must be forward-and-receive friendly. So the tunnel lookup algorithm must be successful in all cases, and must discard only the data packets if the line protocol is down. When a keepalive packet is received, it implies that the tunnel endpoint is again reachable. The tunnel keepalive counter is then reset to 0, and the line protocol comes back up.

Memory and Performance Impact

The feature places almost no additional demand on the router system memory and performance is expected to remain unaffected by its addition. Keepalive packets are treated as ordinary packets, and so it is possible that they can be dropped under high traffic conditions. For now, you can change the number of retries to deal with this issue. If this proves to be inadequate eventually, you can put locally generated keepalive packets in a high priority queue for transmission. You can then set the TOS value in the IP headers to a more suitable value, other than the default or configured value.

Packaging Considerations

The feature is included in the basic IP tunnel code and in the GRE subsystem. Therefore, it must be available with a basic IP package that has the tunnel and the GRE subsystems.

Commands and Configuration

This section addresses the **keepalive** command enabled and extended by this feature only under Cisco bug ID CSCuk26449. Other commands are documented in the respective *Cisco IOS Configuration Guides and Command References*. The **[no] keepalive <period> <retries>** command is enabled and extended with a second parameter, and is available in Cisco IOS Software Release 12.2(8)T and later. It has also been ported under Cisco bug ID CSCuk29980 and CSCuk29983 to Cisco IOS Software Releases 12.1E and 12.2S.

As **keepalive** is an interface configuration command that enables keepalives on the tunnel interface, only keepalives for the GRE/IP mode are supported currently. The second parameter of the command (*retries*) is visible and available only for tunnel interfaces. Values of the first parameter can range from 1 to 32767. When the value is 0, it is equivalent to "no keepalive". This parameter has a default value of 10. The values for the second parameter can range from 1 to 255, and it indicates the number of keepalives that are sent but not returned, after which the tunnel interface pulls the line protocol down. Keepalives on tunnel interfaces are disabled by default.

Sample Output and Screen Formats

This section provides sample outputs.

```
cisco-7505#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
cisco-7505(config)#interface tunnel 1
cisco-7505(config-if)#?
  access-expression      Build a bridge boolean access expression
  &&&&&&
  keepalive              Enable keepalive                      <=====
  &&&&&&
  timeout               Define timeout values for this interface

cisco-7505(config-if)#keepalive ?                               <=====
  <0-32767>  Keepalive period (default 10 seconds)

cisco-7505(config-if)#keepalive 5 ?                             <=====
  <1-255>    Keepalive retries (default 3 times)
cisco-7505(config-if)#keepalive 5 4                             <=====
cisco-7505(config-if)#end

cisco-7505#show interfaces tunnel 1

Tunnell is up, line protocol is up
  Hardware is Tunnel
  Internet address is 10.1.1.1/24
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec, rely 255/255, load 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive set (5 sec), retries 4                               <=====
  Tunnel source 9.2.2.1, destination 6.6.6.2
  Tunnel protocol/transport GRE/IP, key disabled, sequencing disabled
  Tunnel TOS 0xF, Tunnel TTL 128
  Checksumming of packets disabled, fast tunneling enabled
  Last input never, output 00:57:05, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/0, 1 drops; input queue 0/75, 0 drops
  30 second input rate 0 bits/sec, 0 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
    0 packets input, 0 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
    3 packets output, 1860 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
```

Related Information

- **Generic Routing Encapsulation (GRE) Tunnel Keepalive**
 - **GRE Sample Configurations**
 - **Technical Support and Documentation**
-

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2009 – 2010 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Jun 16, 2006

Document ID: 63760
