Determine What Impacts GRE Tunnel Interface States

Contents

Introduction

This document describes the different conditions that can affect the state of a Generic Routing Encapsulation (GRE) tunnel interface.

Background Information

GRE tunnels are designed to be completely stateless. This means that each tunnel endpoint does not keep any information about the state or availability of the remote tunnel endpoint.

A consequence is that, by default, the local tunnel endpoint router cannot bring the line protocol of the GRE Tunnel interface down if the remote end of the tunnel is unreachable.

The ability to mark an interface as "down" (in this situation) is used in order to remove any static routes in the routing table that use that interface as the outbound interface.

Specifically, if the line protocol for an interface is changed to "down", any static routes indicating that interface are removed from the routing table.

This allows for the installation of an alternate (floating) static route or for Policy Based Routing (PBR) in order to select an alternate next-hop or interface.

Also, there are other applications that trigger when an interface changes state; for example, 'backup interface <b-interface>'.

Four Different Tunnel States

There are four possible states in which a GRE tunnel interface exists:

1. Up/up - This implies that the tunnel is fully functional and passes traffic. It is both administratively up

and its protocol is up as well.
2. Administratively down/down - This implies that the interface has been administratively shut down.
3. Up/down - This implies that, even though the tunnel is administratively up, something causes the line protocol on the interface to be down.
4. Reset/down - This is usually a transient state when the tunnel is reset by software. This usually happens when the tunnel is misconfigured with a Next Hop Server (NHS) that is its own IP address.

When a tunnel interface is first created and no other configuration is applied to it, the interface is not shut by default:

```
<#root>

Router#

show run interface tunnel 1


Building configuration...

Current configuration : 40 bytes
!
interface Tunnel1
 no ip address
end
```

In this state, the interface is always up/down:

```
<#root>

Router(config-if)#

do show ip interface brief


Interface                  IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0         172.16.52.1     YES NVRAM  administratively down down
GigabitEthernet0/1         10.36.128.49    YES NVRAM  down                  down
GigabitEthernet0/2         unassigned      YES NVRAM  down                  down
GigabitEthernet0/3         unassigned      YES NVRAM  down                  down
Loopback1                  192.168.2.1     YES NVRAM  up                    up

Tunnel1                    unassigned      YES unset  up                    down
```

This is because the interface is administratively enabled, but since it does not have a tunnel source or a tunnel destination, the line protocol is down.

In order to make this interface up/up, a valid tunnel source and tunnel destination must be configured:

```
<#root>

Router#

show run interface tunnel  1
```

```
Building configuration...

Current configuration : 113 bytes
!
interface Tunnel1
 ip address 10.1.1.1 255.255.255.0
 tunnel source Loopback1
 tunnel destination 10.0.0.1
end

Router#
```

show ip interface brief

```
Interface                  IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0         172.16.52.1     YES NVRAM  up                    up
GigabitEthernet0/1         10.36.128.49    YES NVRAM  down                  down
GigabitEthernet0/2         unassigned      YES NVRAM  down                  down
GigabitEthernet0/3         unassigned      YES NVRAM  down                  down
Loopback0                  unassigned      YES unset  up                    up
Loopback1                  192.168.2.1     YES manual up                    up
Tunnel1                    10.1.1.1         YES manual up                    up
```

The previous sequence shows that:

- A valid tunnel source consists of any interface that is itself in the up/up state and has an IP address configured on it.
- For example, if the tunnel source was changed to **Loopback0**, the tunnel interface would go down even though **Loopback0** is in the up/up state:

<#root>

```
Router(config)#
```

interface tunnel 1

```
Router(config-if)#
```

tunnel source loopback 0

```
Router(config-if)#
*Sep  6 19:51:31.043: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel1, changed state to do
```

- A valid tunnel destination is one which is routable. However, it does not have to be reachable, which can be seen from this ping test:

<#root>

```
Router#
```

show ip route 10.0.0.1

```
% Network not in table
Router#
```

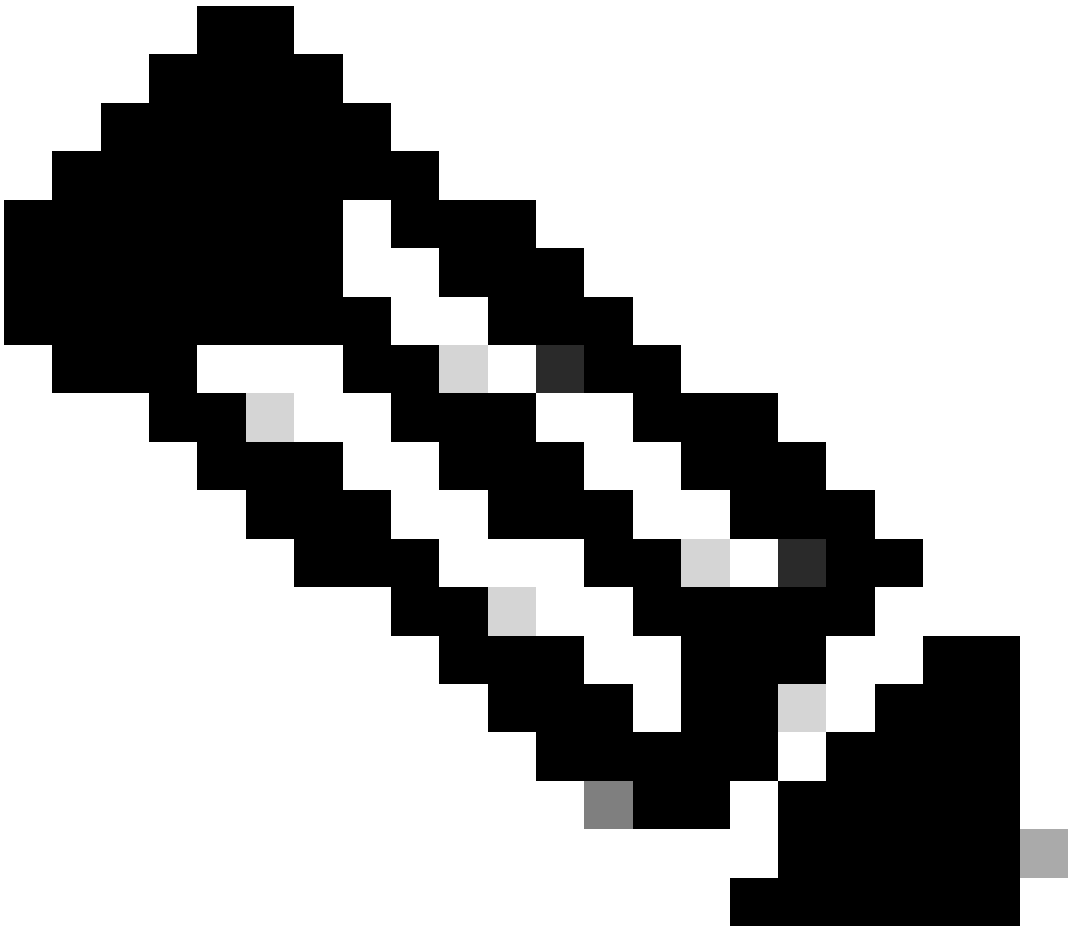
```
show ip route | inc 0.0.0.0


Gateway of last resort is 172.16.52.100 to network 0.0.0.0
S*    0.0.0.0/0 [1/0] via 172.16.52.100
Router#
ping 10.0.0.1


Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

So far, the tunnel has been configured as a point-to-point (P2P) GRE tunnel, which is the default.

If this tunnel were to be changed to a multipoint GRE (mGRE) tunnel, then all that is required for an up state is a valid tunnel source.



Note: An mGRE tunnel can have many tunnel destinations, so that cannot be used to control the

---

tunnel interface state.

<#root>

```
Router#
```

show run interface tunnel 1

```
Building configuration...

Current configuration : 129 bytes
!
interface Tunnel1
 ip address 10.1.1.1 255.255.255.0
 no ip redirects
 tunnel source Loopback1
 tunnel mode gre multipoint
end

Router#
```

show ip interface brief | include Tunnel

```
Tunnel1                    10.1.1.1         YES manual up                    up
```

At any point, if the tunnel interface is administratively shut down, the tunnel immediately goes into an administratively down/down state:

<#root>

```
Router#
```

show run interface tunnel 1

```
Building configuration...

Current configuration : 50 bytes
!
interface Tunnel1
 no ip address
```

shutdown

```
end

Router#
```

show ip interface brief | include Tunnel

```
Tunnel1                    unassigned      YES unset
```

administratively down down

P2P GRE Tunnel State

A P2P GRE Tunnel interface usually comes up as soon as it is configured with a valid tunnel source address or interface which is up **and** a tunnel destination IP address which is routable (shown previously).

Line Protocol Down Locally on the Router

Under normal circumstances, there are only three reasons for a GRE tunnel to be in the up/down state:

- There is no route, which includes the default route, to the tunnel destination address.
- The interface that anchors the tunnel source is down.
- The route to the tunnel destination address is through the tunnel itself, which results in recursion.

These three rules (missing, route, interface down, and misrouted tunnel destination) are problems local to the router at the tunnel endpoints.

They do not cover problems in the intervening network or other features related to the GRE tunnel that can be configured. This document describes scenarios where other factors can influence the state of the GRE tunnel.

GRE Tunnel Keepalives

The basic rules do not cover the case in which the GRE tunneled packets are successfully forwarded yet lost before they reach the other end of the tunnel.

This causes data packets that go through the GRE tunnel to be lost, even though an alternate route that uses PBR or a floating static route via another interface is potentially available.

Keepalives on the GRE tunnel interface are used in order to solve this issue in the same way keepalives are used on physical interfaces.

With Cisco IOS® Software Release 12.2(8)T, it is possible to configure keepalives on a P2P GRE tunnel interface. With this change, the tunnel interface dynamically shuts down if the keepalives fail for a certain period of time.

In order to better understand how GRE tunnel keepalives work, refer to GRE Tunnel Keepalives.

Note: GRE tunnel keepalives are only valid and have an effect on P2P GRE tunnels; they are not valid and do not have any effect on mGRE tunnels.

GRE Tunnels with Tunnel Protection

In Cisco IOS® Software Releases 15.4(3)M/15.4(3)S and later, the GRE tunnel line protocol state follows the IPsec Security Association (SA) state. Therefore, the line protocol can remain down until the IPsec session is fully established.

This was committed with Cisco bug ID CSCum34057 (initial attempt with Cisco bug ID CSCuj29996 and then backed out with Cisco bug ID CSCuj99287).

Multipoint GRE (mGRE) Tunnel Interfaces

For mGRE tunnel interfaces, some of the previous checks for P2P tunnels are not applicable (because there is no fixed tunnel destination).

Here are the reasons an mGRE tunnel line protocol can be in a down state:

- The tunnel source interface is in a down state.
- If the Interface State Control feature is enabled for Dynamic Multipoint VPN (DMVPN) and no NHSs respond, then the line protocol is put in a down state.
- For details on the Interface State Control Feature, see the DMVPN Tunnel Health Monitoring and Recovery Configuration Guide.

Dependencies on Redundancy State

When a tunnel source IP address is configured as a redundancy IP address (for example, a Hot Standby Router Protocol Virtual IP (HSRP VIP) address), then the tunnel interface state tracks the redundancy state.

This adds another check which keeps such tunnel interfaces in the line protocol down state until the redundancy state changes to ACTIVE.

In this example, a misconfigured **ipc zone default** configuration causes redundancy to be in the NEGOTIATION state and keeps such tunnel interfaces in a down state:

```
<#root>

Router#

show redundancy state


       my state = 3  -

NEGOTIATION


     peer state = 1  -DISABLED
           Mode = Simplex
        Unit ID = 0

     Maintenance Mode = Disabled
    Manual Swact = disabled (system is simplex (no peer unit))
 Communications = Down      Reason: Simplex mode

   client count = 16
 client_notification_TMR = 60000 milliseconds
           RF debug mask = 0x0



<#root>

Router#

show interface tunnel100


Tunnel100 is up, line protocol is down


  Hardware is Tunnel
```

```
  Internet address is 172.16.1.100/24
  MTU 17912 bytes, BW 100 Kbit/sec, DLY 50000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 10.122.162.254 (GigabitEthernet0/1)
   Tunnel Subblocks:
      src-track:
         Tunnel100 source tracking subblock associated with GigabitEthernet0/1
          Set of tunnels with source GigabitEthernet0/1, 2 members (includes
          iterators), on interface <OK>
  Tunnel protocol/transport multi-GRE/IP
<SNIP>
```

Troubleshoot

In addition to the reasons previously outlined, the tunnel line state evaluation for the tunnel down reason can be seen with the **show tunnel interface tunnel x** hidden command:

<#root>

```
Router#
```

`show tunnel interface tunnel 100`

```
Tunnel100
   Mode:multi-GRE/IP, Destination UNKNOWN, Source GigabitEthernet0/1
   Application ID 1: unspecified
   Tunnel Subblocks:
      src-track:
         Tunnel100 source tracking subblock associated with GigabitEthernet0/1
          Set of tunnels with source GigabitEthernet0/1, 2 members (includes
          iterators), on interface <OK>
   Linestate - current down
```

`Internal linestate - current down, evaluated down - interface not up`

```
   Tunnel Source Flags: Local
   Transport IPv4 Header DF bit cleared
   OCE: IP tunnel decap
   Provider: interface Tu100, prot 47
     Performs protocol check [47]
     Performs Address save check
     Protocol Handler: GRE: key 0x64, opt 0x2000
       ptype: ipv4 [ipv4 dispatcher: drop]
       ptype: ipv6 [ipv6 dispatcher: drop]
       ptype: mpls [mpls dispatcher: drop]
       ptype: otv [mpls dispatcher: drop]
       ptype: generic [mpls dispatcher: drop]
```

Note: There is an open enhancement to make the tunnel down reason more explicit in order to indicate that it is due to the redundancy state because it is not active. This is tracked by Cisco bug ID CSCuq31060.

Related Information

- **RFC 1701, Generic Router Encapsulation (GRE)**
- **RFC 2890, Key and Sequence Number Extensions to GRE**
- **Generic Routing Encapsulation (GRE) Tunnel Keepalive**
- **IP Fragmentation and PMTUD**
- **Technical Support & Documentation - Cisco Systems**