

Hoe GRE-toetsen werken

Inhoud

[Inleiding](#)

[Voorwaarden](#)

[Vereisten](#)

[Gebruikte componenten](#)

[Conventies](#)

[Achtergrondinformatie](#)

[Het mechanisme van de tunnelbehoud](#)

[Functionele beschrijving](#)

[Geheugeneffecten en prestatieeffecten](#)

[Verpakkingsoverwegingen](#)

[Opdrachten en configuratie](#)

[Uitvoer- en schermformaten](#)

[Gerelateerde informatie](#)

[Inleiding](#)

Dit document geeft een overzicht van de manier waarop Generic Routing Encapsulation (GRE) het werk van de hand houdt.

[Voorwaarden](#)

[Vereisten](#)

Lezers van dit document zouden kennis moeten hebben van deze onderwerpen:

- [GRE-tunnelKeepalive](#)
- [Opdrachten voor configuratie van het toetsenbord](#)

[Gebruikte componenten](#)

De informatie in dit document is gebaseerd op de volgende software- en hardware-versies:

- Cisco 7505 router
- Cisco IOS®-software die GRE via IPSec ondersteunt

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u de potentiële impact van elke opdracht begrijpen.

Conventies

Raadpleeg [Cisco Technical Tips Conventions](#) (Conventies voor technische tips van Cisco) voor meer informatie over documentconventies.

Achtergrondinformatie

De GRE keeplevingsfunctie stelt de interface-opdracht behouden voor tunnels in en stelt u in staat om keepalives te configureren voor point-to-point GRE-tunnels. U kunt keepalives configureren met de opdracht **behouden**, en optioneel met de nieuwe extensie.

GRE-tunnels bieden een methode om willekeurige pakketten in een transportprotocol in te sluiten. Ze bieden ook een architectuur aan die is ontworpen om de services te leveren die vereist zijn om een standaard point-to-point insluitingsschema uit te voeren. Hier zijn een paar voordelen van GRE-tunnels:

- GRE-tunnels bieden lokale netwerken met meerdere protocollen via een backbone met één protocol.
- GRE-tunnels bieden werkronde voor netwerken die protocollen met beperkte hoptellingen bevatten.
- GRE-tunnels verbinden ononderbroken subnetwerken.
- GRE-tunnels maken VPN's over WAN's mogelijk.

Bij de huidige implementatie van GRE-tunnels is een geconfigureerde tunnel echter niet in staat het lijnprotocol van een van beide tunneleindpunten omlaag te brengen als het verre uiteinde onbereikbaar is. Dus het verkeer dat uit de tunnel wordt verstuurd is zwart-wit en kan geen andere paden volgen omdat de tunnel altijd omhoog blijft.

Deze situatie is waar voor tunnels die op statische routes of op het routeren van protocollen vertrouwen die routes samenvoegen om een route naar de tunnelbestemming te vinden. Dit geldt ook voor situaties waarin de gegevens in het besturingsplane een ander pad volgen dan de gegevens in het gegevensvlak.

Het mechanisme van de tunnelbehoud

Deze paragraaf biedt een functionele beschrijving van het mechanisme voor het behoud van de tunnelstructuur met behulp van een voorbeeld. In dit gedeelte worden ook de softwareelementen genoemd die door deze functie worden aangepast en wordt het effect op het geheugen en de prestaties besproken.

Functionele beschrijving

Het tunnelsluitmechanisme maakt het mogelijk, breidt en implementeert een interface-specifiek bevel voor tunnelinterfaces, en levert de mogelijkheid om het lijnprotocol van een tunnel omlaag te brengen. Zie het gedeelte [Opdrachten en configuratie](#) voor meer informatie.

Het mechanisme voor het handhaven van de tunneleigenschappen voldoet ook aan deze aanvullende eisen:

- Het mechanisme voor het handhaven van de tunnel functioneert zelfs als het verre

tunneleindpunt geen keepalives ondersteunt.

- Het mechanisme voor tunnelbehoud is ontstaan uit keepalives.
- Het mechanisme voor het behoud van tunneleigenschappen houdt gelijke eigenschappen in stand.
- Het mechanisme voor het bewaren van de tunnel antwoordt om pakketten van het verre eind in leven te houden, zelfs wanneer het lijnprotocol van de tunnel omlaag is.

Hier is een voorbeeld van hoe het mechanisme voor de handhaving van de tunnel werkt (zie [figuur 1](#)):

Afbeelding 1 - Voorbeeld van het Keeplevingsmechanisme van de Tunnel



Uitvoer

```
interface tunnel 0                                interface tunnel 0
ip address 1.1.1.1 255.255.255.240                ip address 1.1.1.2 255.255.255.240
tunnel source 128.8.8.8                          tunnel source 129.9.9.9
tunnel destination 129.9.9.9                    tunnel destination 128.8.8.8
keepalive 5 4                                    keepalive 5 4
interface loopback 0                             interface loopback 0
ip address 128.8.8.8 255.255.255.255            ip address 129.9.9.9 255.255.255.255
```

Een Keeplevlek-pakket dat afkomstig is van A tot 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
```

Wanneer u keepalives op het tunneleindpunt van router A toelaat, bouwt de router bij elk interval de binnenste IP header. Aan het eind van de header voegt de router ook een GRE-header toe met een Protocol Type (PT) van 0, en geen andere lading. De router stuurt dan dat pakket door de tunnel, wat in zijn insluiting resulteert met de buitenste IP-header, en een GRE-header met PT van IP. De tunnel houdt toonaangevende stappen bij één. Als er een manier is om het verre eindtunneleindpunt te bereiken, en het protocol van de tunnellijn is niet door andere redenen gedaald, komt het pakket op router B aan. Het wordt dan gematcht tegen Tunnel 0, wordt gedecapsuleerd en doorgestuurd naar de bestemming IP, die de tunnelbron, router A is. Na aankomst op router A, wordt het pakket opnieuw gedecapsuleerd, en PT wordt gecontroleerd. Als het resultaat van de PT controle 0 is, betekent het dat dit een keeplevend pakket is. In zo'n geval wordt de teller voor het bewaren van de tunnel teruggezet op 0, en wordt het pakje verworpen.

Indien router B onbereikbaar is, blijft router A de keeplevendige pakketten samen met normaal verkeer bouwen en verzenden. Als het lijnprotocol omlaag is, komen de keepalives niet terug naar router A. Daarom blijft de keepleventeller stijgen. Het protocol van de tunnellijn blijft omhoog slechts zolang de teller van de tunnelhandhaving nul blijft, of minder dan een geconfigureerde

waarde. Als die voorwaarde niet waar is, de volgende keer dat u probeert om een keeplevend naar router B te verzenden, wordt het lijnprotocol teruggebracht, zodra de keeplepel de geconfigureerde keeplevewaarde bereikt. In de omhoog/omlaag staat, voorwaarts of verwerkt de tunnel geen verkeer behalve de overlevingspakketten. Om dit te doen om alleen voor levendige pakketten te werken moet de tunnel voorwaarts- en ontvangstvriendelijk zijn. Dus moet het algoritme van de tunnelraadpleging in alle gevallen succesvol zijn, en moet slechts de gegevenspakketten weggooien als het lijnprotocol omlaag is. Wanneer een pakketje met behoud wordt ontvangen, impliceert dit dat het tunneleindpunt opnieuw bereikbaar is. De teller van de tunnelhandhaving wordt dan teruggezet op 0, en het lijnprotocol komt terug.

Geheugeneffecten en prestatieeffecten

Deze optie stelt bijna geen extra vraag op het geheugen van het routersysteem en de prestaties zullen naar verwachting niet worden beïnvloed door de toevoeging ervan. Levende pakketten worden als gewone pakketten behandeld en het is dus mogelijk dat ze onder hoge verkeersomstandigheden kunnen worden gedropt. Op dit moment kunt u het aantal pogingen wijzigen om deze kwestie aan te pakken. Als dit uiteindelijk ontoereikend blijkt, kunt u lokaal gegenereerde overlevingspakketten in een rij met hoge prioriteit plaatsen voor transmissie. U kunt de TOS-waarde in de IP-headers dan instellen op een geschiktere waarde dan de standaard of ingestelde waarde.

Verpakkingsoverwegingen

Deze functie is opgenomen in de basiscode voor IP-tunnels en in het subsysteem GRE. Daarom moet het beschikbaar zijn met een basisIP-pakket dat de tunnel en de GRE-subsysteem bevat.

Opdrachten en configuratie

Deze sectie richt zich op de opdracht Levensonderhoud die door deze functie is ingeschakeld en uitgebreid alleen onder Cisco bug-ID CSCuk2649. Andere opdrachten zijn gedocumenteerd in de *respectievelijke Cisco IOS Configuration Guides en Opdrachtreferenties*. De **[no]-standby <punt> <>**opdracht is ingeschakeld en uitgebreid met een tweede parameter en is beschikbaar in Cisco IOS-software release 12.2(8)T en hoger. Het is ook geëxporteerd onder Cisco bug-ID CSCuk2980 en CSCuk29983 naar Cisco IOS-software releases 12.1E en 12.2S.

Aangezien **Keepalive** een opdracht voor de interfaceconfiguratie is die keepalives op de tunnelinterface toelaat, worden alleen keepalives voor de GRE/IP-modus momenteel ondersteund. De tweede parameter van de opdracht (*opnieuw proberen*) is alleen zichtbaar en beschikbaar voor tunnelinterfaces. Waarden van de eerste parameter kunnen variëren van 1 tot 32767. Als de waarde 0 is, is het gelijk aan "geen keeplevend". Deze parameter heeft een standaardwaarde van 10. De waarden voor de tweede parameter kunnen van 1 tot 255 variëren, en het geeft het aantal keepalives aan die worden verzonden maar niet teruggegeven, waarna de tunnelinterface het lijnprotocol naar beneden trekt. Keepalives op tunnelinterfaces worden standaard uitgeschakeld.

Uitvoer- en schermformaten

In dit deel worden de steekproefuitgangen vermeld.

```
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
    0 output buffer failures, 0 output buffers swapped out

```

[Gerelateerde informatie](#)

- [Generic Routing Encapsulation \(GRE\) Tunnel Keeplive](#)
- [GRE-voorbeeldconfiguratie](#)
- [Technische ondersteuning en documentatie](#)