

# Como as manutenções de atividade de GRE trabalham

## Índice

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenções](#)

[Informações de Apoio](#)

[O mecanismo do keepalive de túnel](#)

[Descrição funcional](#)

[Memória e impacto no desempenho](#)

[Considerações de empacotamento](#)

[Comandos e configuração](#)

[Exemplo de saída e formatos de tela](#)

[Informações Relacionadas](#)

## Introdução

Este documento fornece uma visão geral de como os keepalives do Generic Routing Encapsulation (GRE) funcionam.

## Pré-requisitos

### Requisitos

Os leitores deste documento devem estar cientes destes tópicos:

- [Keepalive do túnel GRE](#)
- [Comandos do modo de configuração de manutenção de atividade](#)

### Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:

- Cisco 7505 Router
- Software de Cisco IOS® que apoia o GRE sobre o IPsec

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma

configuração (padrão) inicial. Se a sua rede estiver ativa, certifique-se de que entende o impacto potencial de qualquer comando.

## [Convenções](#)

Para obter mais informações sobre convenções de documento, consulte as [Convenções de dicas técnicas Cisco](#).

## [Informações de Apoio](#)

A característica do GRE keepalive permite o comando interface do **keepalive** para túneis, e permite que você configure o Keepalives para túneis GRE pontos a ponto. Você pode configurar o Keepalives com o **comando keepalive**, e opcionalmente com sua extensão nova.

Os túneis GRE fornecem um método para encapsular pacotes arbitrários dentro de um protocolo de transporte. Igualmente oferecem uma arquitetura projetada proporcionar os serviços exigidos executar todo o esquema padrão do encapsulamento de Point-to-Point. Estão aqui algumas das vantagens dos túneis GRE:

- Os túneis GRE fornecem redes local do multi-protocol sobre um backbone de protocolo único.
- Os túneis GRE fornecem as soluções para redes que contêm protocolos com contagens de saltos limitados.
- Os túneis GRE conectam sub-redes descontínuas.
- Os túneis GRE permitem VPN através dos WAN.

Contudo, na implementação atual dos túneis GRE, um túnel configurado não tem a capacidade para derrubar o protocolo de linha de um ou outro ponto final de túnel, se a ponta oposta é inacessível. Assim, o tráfego enviado do túnel preto-é furado, e não pode seguir trajetos alternativos porque o túnel fica sempre acima.

Esta situação é verdadeira para os túneis que confiam em rotas estáticas ou nos protocolos de roteamento esses rotas agregadas para encontrar uma rota ao destino de túnel. É igualmente verdadeiro nas situações onde os dados no plano do controle seguem um trajeto diferente dos dados no plano dos dados.

## [O mecanismo do keepalive de túnel](#)

Esta seção fornece uma descrição funcional para o mecanismo do keepalive de túnel com a ajuda de um exemplo. Esta seção igualmente alista os elementos de software que esta característica altera, e discute o impacto na memória e no desempenho.

### [Descrição funcional](#)

O mecanismo do keepalive de túnel permite, estende e executa um comando interface-specific para interfaces de túnel, e entrega a capacidade para derrubar o protocolo de linha de um túnel. Para mais informação, veja os [comandos e a seção de configuração](#).

O mecanismo do keepalive de túnel igualmente endereça estas exigências adicionais:

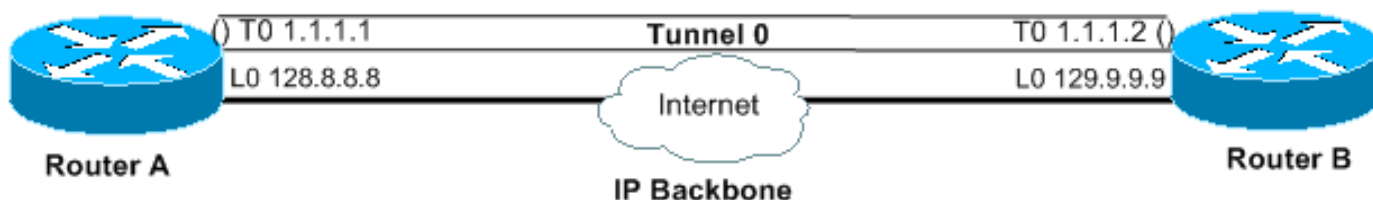
- O mecanismo do keepalive de túnel funciona mesmo se o ponto final de túnel distante não

suporta o Keepalives.

- O mecanismo do keepalive de túnel origina o Keepalives.
- O mecanismo do keepalive de túnel processa o Keepalives.
- O mecanismo do keepalive de túnel responde aos pacotes keepalive da ponta oposta, mesmo quando o protocolo de linha do túnel está para baixo.

Está aqui um exemplo de como o mecanismo do keepalive de túnel trabalha (veja [figura 1](#)):

Figura 1 – Exemplo para o mecanismo do keepalive de túnel



### Saída

```
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
```

### Um pacote keepalive que origine à 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
```

Quando você permite o Keepalives no ponto final de túnel do roteador A, o roteador em cada intervalo constrói o cabeçalho IP interno. Na extremidade do encabeçamento, o roteador igualmente não adiciona um cabeçalho de GRE com um tipo de protocolo (PT) de 0, e nenhum outro payload. O roteador envia então esse pacote através do túnel, que conduz a seu encapsulamento com o cabeçalho IP exterior, e de um cabeçalho de GRE com o PT do IP. Os incrementos do contador do keepalive de túnel por um. Se há uma maneira de alcançar o ponto final de túnel da ponta oposta, e o protocolo de linha do túnel não é abaixo de devido a outras razões, o pacote chega no roteador B. É combinado então contra o tunnel0, é o descapsulado, e enviado ao IP de destino, que é o origem de túnel, a chegada de A. Em cima do roteador no roteador A, o pacote é outra vez descapsulado, e o PT é verificado. Se o resultado da verificação PT é 0, significa que este é um pacote keepalive. Em tal caso, o contador do keepalive de túnel é restaurado a 0, e o pacote é rejeitado.

Caso que o roteador B é inacessível, o roteador A continua a construir e enviar os pacotes keepalive junto com o tráfego normal. Se o protocolo de linha está inativo, o Keepalives não vem para trás ao roteador A. Consequentemente, o contador de keepalive continua a aumentar. O protocolo de linha do túnel fica acima somente enquanto o contador do keepalive de túnel permanece zero, ou menos do que um valor configurado. Se essa circunstância não é verdadeira, a próxima vez que você tenta enviar um keepalive ao roteador B, o protocolo de linha está derrubado, assim que o contador de keepalive alcançar o valor de keepalive configurado. No

estado up/down, o túnel não envia nem processa nenhum tráfego independentemente dos pacotes keepalive. Para que isto trabalhe para pacotes keepalive somente, o túnel deve ser dianteiro-e-recebe amigável. Assim o algoritmo da consulta do túnel deve ser bem sucedido em todos os casos, e deve rejeitar somente os pacotes de dados se o protocolo de linha está inativo. Quando um pacote keepalive é recebido, implica que o ponto final de túnel é outra vez alcançável. O contador do keepalive de túnel é restaurado então a 0, e o protocolo de linha vem apoio.

## Memória e impacto no desempenho

A característica não coloca quase nenhuma procura adicional na memória de sistema do roteador e o desempenho é esperado permanecer não afetado por sua adição. Os pacotes keepalive são tratados como pacotes ordinários, e assim que é possível que podem ser deixados cair sob condições de tráfego elevado. Por agora, você pode mudar o número de novas tentativas para tratar esta edição. Se isto prova ser inadequado eventualmente, você pode pôr pacotes keepalive localmente gerados em uma fila de alta prioridade para a transmissão. Você pode então ajustar o valor TOS nos cabeçalhos IP a um valor mais apropriado, a não ser o padrão ou o valor configurado.

## Considerações de empacotamento

A característica é incluída no código básico do túnel IP e no subsistema GRE. Consequentemente, deve estar disponível com um pacote básico IP que tenha o túnel e os subsistemas GRE.

## Comandos e configuração

Esta seção endereça o **comando keepalive** permitido e estendido por esta característica somente sob a identificação de bug Cisco CSCuk26449. Outros comandos são documentados nos *guias e nas referências de comandos de configuração do IOS* do respectivo Cisco. O **<period>** os **<retries>** o comando do **keepalive do [no]** são permitidos e estendidos com um segundo parâmetro, e estão disponíveis no Cisco IOS Software Release 12.2(8)T e Mais Recente. Foi movido igualmente sob a identificação de bug Cisco CSCuk29980 e CSCuk29983 aos Cisco IOS Software Releases 12.1E e 12.2S.

Porque o **keepalive** é um comando interface configuration que permita o Keepalives na interface de túnel, simplesmente o Keepalives para o modo GRE/IP é apoiado atualmente. O segundo parâmetro do comando (*novas tentativas*) está visível e disponível somente para interfaces de túnel. Os valores do primeiro parâmetro podem variar de 1 a 32767. Quando o valor é 0, é equivalente a “nenhum keepalive”. Este parâmetro tem um valor padrão do 10. Os valores para o segundo parâmetro podem variar de 1 a 255, e indica o número de Keepalives que é enviado mas não retornado, depois do qual a interface de túnel puxa o protocolo de linha para baixo. O Keepalives em interfaces de túnel é desabilitado à revelia.

## Exemplo de saída e formatos de tela

Esta seção fornece exemplos de saída.

```
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 Tunnel1 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
```

## [Informações Relacionadas](#)

- [Keepalive do túnel de encapsulamento de roteamento genérico \(GRE\)](#)
- [Configurações de amostra GRE](#)
- [Suporte técnico e documentação](#)