

Fonctionnement des Keepalives GRE

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Conventions](#)

[Informations générales](#)

[Le mécanisme de keepalive de tunnel](#)

[Description fonctionnelle](#)

[Mémoire et incidence des performances](#)

[Considérations de emballage](#)

[Commandes et configuration](#)

[Sortie témoin et masques d'écran](#)

[Informations connexes](#)

Introduction

Ce document fournit un aperçu de la façon dont fonctionnent les keepalives GRE (Generic Routing Encapsulation).

Conditions préalables

Conditions requises

Les lecteurs de ce document devraient avoir connaissance des sujets suivants :

- [Keepalive de tunnel GRE](#)
- [Commandes de mode de configuration de keepalive](#)

Composants utilisés

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Routeur Cisco 7505
- Logiciel de Cisco IOS® qui prend en charge GRE au-dessus d'IPSec

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-

vous que vous comprenez l'effet potentiel de toute commande.

[Conventions](#)

Pour plus d'informations sur les conventions de documents, reportez-vous à [Conventions relatives aux conseils techniques Cisco](#).

[Informations générales](#)

La caractéristique de keepalive GRE active la commande d'interface de **keepalive** pour des tunnels, et te permet pour configurer le Keepalives pour les tunnels point par point GRE. Vous pouvez configurer le Keepalives avec la commande de **keepalive**, et sur option avec sa nouvelle extension.

Les tunnels GRE fournissent une méthode pour encapsuler les paquets arbitraires à l'intérieur d'un protocole de transport. Ils offrent également une architecture conçue pour fournir les services exigés pour implémenter n'importe quel schéma point par point standard d'encapsulation. Voici certains des avantages des tunnels GRE :

- Les tunnels GRE fournissent les réseaux locaux multiprotocole au-dessus d'un circuit principal à protocole unique.
- Les tunnels GRE fournissent des contournements pour les réseaux qui contiennent des protocoles avec le saut limité comptent.
- Les tunnels GRE connectent des sous-réseaux discontinus.
- Les tunnels GRE permettent des VPN à travers des WAN.

Cependant, dans l'implémentation en cours de GRE perce un tunnel, un tunnel configuré n'a pas la capacité de réduire la ligne protocole de l'un ou l'autre de périphérique du tunnel, si l'extrémité est inaccessible. Ainsi, le trafic envoyé du tunnel noir-est troué, et il ne peut pas suivre des chemins alternatifs parce que le tunnel reste toujours.

Cela situation vaut pour les tunnels qui se fondent sur les artères statiques ou sur les protocoles de routage qui agrègent des artères pour trouver une artère à la destination de tunnel. Il est également vrai dans les situations où les données dans l'avion de contrôle suivent un différent chemin des données dans le plan de données.

[Le mécanisme de keepalive de tunnel](#)

Cette section fournit une description fonctionnelle pour le mécanisme de keepalive de tunnel avec l'aide d'un exemple. Cette section répertorie également les éléments de logiciel que cette caractéristique modifie, et discute l'incidence sur la mémoire et la représentation.

[Description fonctionnelle](#)

Le mécanisme de keepalive de tunnel active, étend et implémente une commande d'interface-particularité pour des interfaces de tunnel, et fournit la capacité de réduire la ligne protocole d'un tunnel. Le pour en savoir plus, voient la section de [commandes et de configuration](#).

Le mécanisme de keepalive de tunnel adresse également ces conditions requises supplémentaires :

- Le mécanisme de keepalive de tunnel fonctionne même si le périphérique du tunnel lointain ne prend en charge pas le Keepalives.
- Le mécanisme de keepalive de tunnel lance le Keepalives.
- Le mécanisme de keepalive de tunnel traite le Keepalives.
- Le mécanisme de keepalive de tunnel répond dans des paquets keepalive de l'extrémité, même lorsque la ligne protocole du tunnel est en baisse.

Voici un exemple de la façon dont le mécanisme de keepalive de tunnel fonctionne (voir le [schéma 1](#)) :

Figure 1 – Exemple pour le mécanisme de keepalive de tunnel
Sortie

```
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
```

Un paquet keepalive qui provient d'A 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
```

Quand vous activez le Keepalives sur le périphérique du tunnel du routeur A, le routeur à chaque intervalle construit l'en-tête IP intérieure. À l'extrémité de l'en-tête, le routeur n'ajoute également une en-tête GRE avec un type de Protocol (pinte) de 0, et aucune autre charge utile. Le routeur envoie alors ce paquet par le tunnel, qui a comme conséquence son encapsulation avec l'en-tête IP externe, et une en-tête GRE avec la pinte d'IP. Les incréments de compteur de keepalive de tunnel par un. S'il y a une manière d'atteindre le périphérique du tunnel d'extrémité, et la ligne protocole de tunnel n'est pas en bas d'en raison d'autres raisons, le paquet arrive sur le routeur B. Il est alors apparié contre le tunnel 0, est désencapsulé, et expédié à l'IP de destination, qui est la source du tunnel, l'arrivée d'A. Upon de routeur sur le routeur A, le paquet est de nouveau désencapsulé, et la pinte est vérifiée. Si le résultat du contrôle pinte est 0, il signifie que c'est un paquet keepalive. En pareil cas, le compteur de keepalive de tunnel est remis à l'état initial à 0, et le paquet est jeté.

Au cas où le routeur B serait inaccessible, le routeur A continue à construire et envoyer les paquets keepalive avec le trafic normal. Si la ligne protocole est en baisse, le Keepalives ne revient pas au routeur A. Par conséquent, le compteur de keepalive continue à augmenter. La ligne protocole de tunnel reste seulement tant que le compteur de keepalive de tunnel demeure zéro, ou moins qu'une valeur configurée. Si cette condition n'est pas vraie, la prochaine fois que vous tentez d'envoyer une keepalive au routeur B, la ligne protocole est réduite, dès que le compteur de keepalive atteindra la valeur configurée de keepalive. Dans l'état haut/bas, le tunnel n'expédie ou traite aucun trafic indépendamment des paquets keepalive. Pour que ceci fonctionne pour des paquets keepalive seulement, le tunnel doit être en avant-et-reçoivent amical. Ainsi l'algorithme de consultation de tunnel doit être réussi dans des tous les cas, et doit jeter seulement les paquets de données si la ligne protocole est en baisse. Quand un paquet keepalive est reçu, il implique que le périphérique du tunnel est de nouveau accessible. Le compteur de keepalive de tunnel est alors remis à l'état initial à 0, et la ligne protocole se réactive.

Mémoire et incidence des performances

La caractéristique ne place presque aucune demande supplémentaire sur la mémoire système de routeur et on s'attend à ce que la représentation demeure inchangée par son ajout. Des paquets keepalive sont traités en tant que paquets ordinaires, et ainsi il est possible qu'ils puissent être lâchés dans des états du trafic élevé. Pour l'instant, vous pouvez changer le nombre de relances pour traiter cette question. Si ceci s'avère insuffisant par la suite, vous pouvez mettre les paquets keepalive localement générés dans une file d'attente prioritaire pour la transmission. Vous pouvez alors placer la valeur de TOS dans les en-têtes IP à une valeur plus appropriée, autre que le par défaut ou la valeur configurée.

Considérations de empaquetage

La caractéristique est incluse dans le code de base de tunnel IP et dans le sous-système GRE. Par conséquent, il doit être disponible avec un module de base IP qui a le tunnel et les sous-systèmes GRE.

Commandes et configuration

Cette section adresse la commande **enabled** de **keepalive** et étende par cette caractéristique seulement sous l'ID de bogue Cisco CSCuk26449. D'autres commandes sont documentées dans les *guides de configuration et des références de commandes IOS* de respectiveCisco. **[Non] le <period > les <retries > la commande de keepalive est activé et étendu avec un deuxième paramètre, et est disponible dans le Logiciel Cisco IOS version 2.2(8)T et plus tard. Il a été également mis en communication sous l'ID de bogue Cisco CSCuk29980 et CSCuk29983 aux versions du logiciel Cisco IOS 12.1E et 12.2S.**

Car la **keepalive** est une commande de configuration d'interface qui active le Keepalives sur l'interface de tunnel, seulement le Keepalives pour le mode GRE/IP est pris en charge actuellement. Le deuxième paramètre de la commande (*relances*) est visible et disponible seulement pour des interfaces de tunnel. Les valeurs du premier paramètre peuvent s'étendre de 1 à 32767. Quand la valeur est 0, elle est équivalente à « aucune keepalive ». Ce paramètre a une valeur par défaut de 10. Les valeurs pour le deuxième paramètre peuvent s'étendre de 1 à 255, et il indique le nombre de Keepalives qui sont envoyés mais pas retournés, après quoi l'interface de tunnel abaisse la ligne protocole. Le Keepalives sur des interfaces de tunnel est désactivé par défaut.

Sortie témoin et masques d'écran

Cette section fournit des sorties témoin.

```
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

Informations connexes

- [**Keepalive de tunnel GRE \(Generic Routing Encapsulation\)**](#)
- [**Configurations d'échantillon GRE**](#)
- [**Soutien technique et documentation**](#)