

GRE キープアライブの動作方法

内容

[概要](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[表記法](#)

[背景説明](#)

[トンネルのキープアライブ メカニズム](#)

[機能の説明](#)

[メモリおよびパフォーマンスに対する影響](#)

[実装時の注意事項](#)

[コマンドと設定](#)

[出力例と画面形式](#)

[関連情報](#)

概要

このドキュメントでは、Generic Routing Encapsulation (GRE) キープアライブの仕組みについて概説します。

前提条件

要件

このドキュメントの読者は次のトピックについての専門知識を有している必要があります。

- [GRE トンネル キープアライブ](#)
- [キープアライブ設定モードのコマンド](#)

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- Cisco 7505 ルータ
- GRE over IPSec をサポートする Cisco IOS® ソフトウェア

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、初期 (デフォルト) 設定の状態から起動しています。対象のネットワークが実稼働中である場合には、どのようなコマンドについても、その潜在的な影響について確実に理解しておく必要があります。

表記法

ドキュメント表記の詳細は、「[シスコテクニカルティップスの表記法](#)」を参照してください。

背景説明

GRE キープアライブ機能を使用すれば、トンネルで使用する `keepalive` インターフェイス コマンドが有効になり、ポイントツーポイントの GRE トンネルにキープアライブを設定できるようになります。`keepalive` コマンドでキープアライブの設定ができ、さらに、オプションでこのコマンドの新しい拡張機能を使用することもできます。

GRE トンネルには、トランスポート プロトコル内の任意のパケットをカプセル化する方法が用意されています。これらのトンネルでは、任意の標準ポイントツーポイント カプセル化スキームを実装するために必要なサービスを提供する設計になっているアーキテクチャも実現されています。GRE トンネルには次のような利点があります。

- 単一プロトコルのバックボーン上でマルチプロトコルのローカル ネットワークを実現する。
- ホップ カウントが制限されているプロトコルを含んだネットワークに対して回避策を提供する。
- 不連続なサブネットワークを接続する。
- WAN 上で VPN を可能にする。

ただし 現在の GRE トンネルの実装では、遠端に到達不能な場合、設定済みのトンネルにはトンネルのいずれかのエンドポイントの回線プロトコルをダウン状態にする機能がありません。そのため、トンネルから送信されたトラフィックがブラックホール化し、トンネルが常にアップ状態になるために、代替パスを使用できなくなります。

この状況は、スタティック ルートに依存するトンネルや、ルートを集約してトンネルの宛先へのルートを見つけるルーティング プロトコルに依存するトンネルで発生します。また、コントロールプレーンのデータがデータ プレーンのデータとは異なるパスを使用する場合も同様です。

トンネルのキープアライブ メカニズム

このセクションでは、トンネルのキープアライブ メカニズムの機能を、例を使用して説明します。また、このセクションでは、この機能で変更されるソフトウェア要素を列挙して、メモリとパフォーマンスへの影響についても説明しています。

機能の説明

トンネル キープアライブ メカニズムにより、トンネル インターフェイスのインターフェイス特有のコマンドが使用可能となり、これらが機能拡張して実装されるので、トンネルの回線プロトコルをダウン状態にできるようになります。詳細については、「[コマンドと設定](#)」セクションを参照してください。

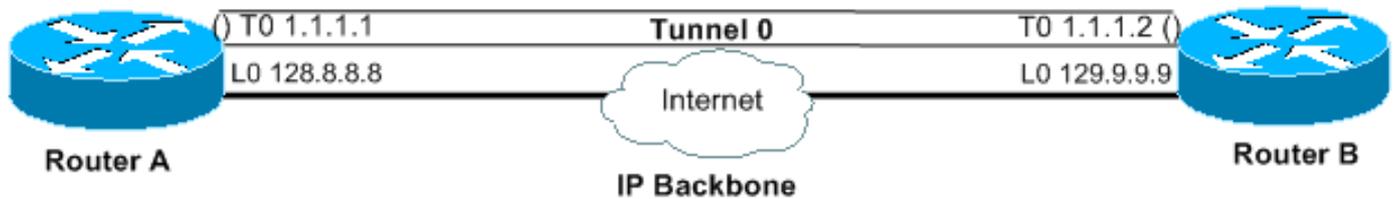
トンネル キープアライブ メカニズムは、次の追加要件にも対処しています。

- トンネルの遠端のエンドポイントでキープアライブがサポートされていない場合でも、トンネル キープアライブ メカニズムが機能する。
- トンネル キープアライブ メカニズムがキープアライブの発信元となる。
- トンネル キープアライブ メカニズムがキープアライブを処理する。

- ・トンネルの回線プロトコルがダウンしている場合でも、トンネル キープアライブ メカニズムは遠端のキープアライブ パケットに応答する。

次にトンネルのキープアライブ メカニズムが機能するしくみについて、例を示します ([図 1 参照](#))。

図 1- トンネル キープアライブ メカニズムの例



出力

```
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 から発信した B へのキープアライブ パケット

```
---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
```

ルータ A のトンネルのエンドポイントでキープアライブを有効にすると、ルータではインターバルごとに内側の IP ヘッダーが作成されます。ヘッダーの最後には、Protocol Type (PT) に 0 が設定された GRE ヘッダーもルータで追加されますが、他のペイロードはありません。次に、ルータではパケットがトンネルを通して送信されます。その結果、パケットは外部 IP ヘッダーとともにカプセル化されて、PT に IP が設定された GRE ヘッダーが付けられます。トンネル キープアライブ カウンタは 1 増加します。トンネルの遠端のエンドポイントに到達する方法があり、トンネルの回線プロトコルが他の理由によるダウン状態になっていない場合は、パケットはルータ B に到達します。次に、パケットはトンネル 0 と照合されて、カプセル化が解除され、宛先 IP つまりトンネルの発信元であるルータ A に転送されます。ルータ A にパケットが到達すると、再びカプセル化が解除されて、PT が調べられます。PT を調べた結果が 0 であれば、このパケットはキープアライブ パケットであることとなります。その場合、トンネル キープアライブ カウンタが 0 にリセットされた上で、パケットは廃棄されます。

ルータ B に到達できない場合は、ルータ A では、通常のトラフィックとともにキープアライブ パケットが引き続き作成されて送信されます。回線プロトコルがダウンしている場合、キープアライブはルータ A に戻りません。したがって、キープアライブ カウンタは増加し続けます。トンネルのキープアライブ カウンタが 0 のままか設定値未満の場合は、トンネルの回線プロトコルがアップ状態になっています。そのような値ではない場合、次にキープアライブをルータ B へ送信するときに、キープアライブ カウンタが設定済みのキープアライブ値に達するとすぐに回線プロトコルはダウン状態になります。アップ状態でもダウン状態でも、このトンネルはキープアライブ パケット以外のトラフィックの転送または処理を行いません。キープアライブ パケットのみに

対して機能するには、トンネルが転送と受信に影響を与えてはなりません。そのため、トンネル参照アルゴリズムはどんな場合でも成功し、回線プロトコルがダウンしているときは、データパケットだけが廃棄されることとなります。キープアライブパケットが受信されると、トンネルのエンドポイントが再び到達可能になったことを示しています。その場合、トンネルキープアライブカウンタが0にリセットされて、回線プロトコルが再びアップ状態になります。

メモリおよびパフォーマンスに対する影響

この機能では、ルータシステムのメモリとパフォーマンスにはほとんど追加の負荷がかからないので、この機能の追加による影響はありません。キープアライブパケットは通常のパケットとして扱われるため、トラフィックが多い場合にはドロップされる可能性があります。今のところは、リトライの回数を変更してこの問題に対処することができます。ただし、結果としてこの対処方法では不十分な場合は、ローカルに生成されたキープアライブパケットを、伝送用高プライオリティキューに入れることができます。次に、IPヘッダーのTOS値を、デフォルト値や設定値以外のより適切な値に設定できます。

実装時の注意事項

この機能は、基本的なIPトンネルコードとGREサブシステムに含まれています。そのため、トンネルとGREサブシステムを備えた基本IPパッケージで提供されることとなります。

コマンドと設定

このセクションでは、Cisco Bug ID CSCuk26449でのみ、この機能によって有効および拡張されるkeepaliveコマンドについて説明します。その他のコマンドについては、対応するCisco IOSコンフィギュレーションガイドおよびコマンドリファレンスに記載されています。[no] keepalive <period> <retries> コマンドは第2パラメータが有効になり拡張されました。これは、Cisco IOS ソフトウェア リリース 12.2(8)T 以降で使用することができます。また、これは Cisco Bug ID CSCuk29980 および CSCuk29983 において、Cisco IOS ソフトウェア リリース 12.1E と 12.2S にも移植されました。

keepalive はトンネル インターフェイスでキープアライブを有効にするインターフェイス設定コマンドであるため、現在は GRE/IP モードに対するキープアライブのみがサポートされています。コマンドの第2パラメータ (retries) は、トンネル インターフェイスからのみ認識可能であり、使用できます。最初のパラメータの値の範囲は1 ~ 32767です。値が0の場合、「no keepalive」と同等です。このパラメータのデフォルト値は10です。2番目のパラメータの値は1 ~ 255の範囲で指定でき、送信されても返されないキープアライブの数を示します。その後、トンネルインターフェイスが回線プロトコルをダウンさせます。トンネル インターフェイス上のキープアライブは、デフォルトでは無効になっています。

出力例と画面形式

このセクションでは、出力例を示します。

```
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
```

[関連情報](#)

- [総称ルーティング カプセル化 \(GRE \) トンネルのキープアライブ](#)
- [GRE の設定例](#)
- [テクニカル サポートとドキュメント](#)