

# GRE Keepalive 작동 방식

## 목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[표기규칙](#)

[배경 정보](#)

[터널 연결 메커니즘](#)

[기능 설명](#)

[메모리 및 성능 영향](#)

[패키징 고려 사항](#)

[명령 및 구성](#)

[샘플 출력 및 화면 형식](#)

[관련 정보](#)

## 소개

이 문서에서는 GRE(Generic Routing Encapsulation) keepalives의 작동 방식을 간략하게 설명합니다.

## [사전 요구 사항](#)

### [요구 사항](#)

이 문서의 독자는 다음 주제에 대해 알고 있어야 합니다.

- [GRE 터널 keepalive](#)
- [Keepalive 컨피그레이션 모드 명령](#)

### [사용되는 구성 요소](#)

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- Cisco 7505 Router
- GRE over IPSec을 지원하는 Cisco IOS® 소프트웨어

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우, 모든 명령어의 잠재적인 영향을 미리 숙지하시기 바랍니다.

## [표기 규칙](#)

문서 규칙에 대한 자세한 내용은 [Cisco 기술 팁 표기 규칙](#)을 참조하십시오.

## [배경 정보](#)

GRE keepalive 기능은 터널에 **keepalive** interface 명령을 활성화하고 포인트-투-포인트 GRE 터널에 대한 keepalive를 구성할 수 있습니다. keepalive 명령을 사용하여 **keepalive** 및 선택적으로 새 확장자로 구성할 수 있습니다.

GRE 터널은 전송 프로토콜 내에 임의의 패킷을 캡슐화하는 방법을 제공합니다. 또한 표준 포인트 투 포인트 캡슐화 체계를 구현하는 데 필요한 서비스를 제공하도록 설계된 아키텍처를 제공합니다. 다음은 GRE 터널의 몇 가지 장점입니다.

- GRE 터널은 단일 프로토콜 백본을 통해 다중 프로토콜 로컬 네트워크를 제공합니다.
- GRE 터널은 제한된 홉의 수가 있는 프로토콜을 포함하는 네트워크에 대한 해결 방법을 제공합니다.
- GRE 터널은 비연속 하위 네트워크를 연결합니다.
- GRE 터널은 WAN 전체에서 VPN을 허용합니다.

그러나 현재 GRE 터널 구현에서 구성된 터널은 먼 끝에 도달할 수 없는 경우 두 터널 엔드포인트의 회선 프로토콜을 다운할 수 없습니다. 따라서 터널에서 전송된 트래픽은 블랙홀이며 터널이 항상 작동 상태이므로 대체 경로를 따를 수 없습니다.

이 상황은 고정 경로를 사용하는 터널 또는 터널 대상에 대한 경로를 찾기 위해 경로를 집계하는 라우팅 프로토콜에 의존하는 터널에 적용됩니다. 컨트롤 플레인의 데이터가 데이터 플레인의 데이터와 다른 경로를 따르는 경우에도 마찬가지입니다.

## [터널 연결 메커니즘](#)

이 섹션에서는 예제의 도움말과 함께 터널 keepalive 메커니즘에 대한 기능 설명을 제공합니다. 이 섹션에서는 이 기능이 수정하는 소프트웨어 요소를 나열하며 메모리 및 성능에 미치는 영향에 대해 설명합니다.

### [기능 설명](#)

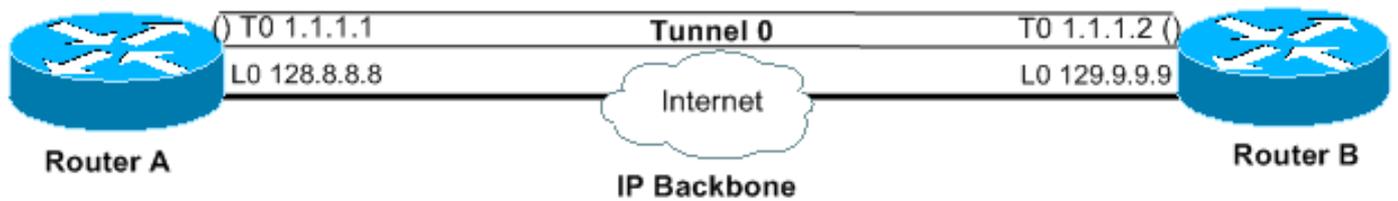
tunnel keepalive 메커니즘은 터널 인터페이스에 대해 interface-specific 명령을 활성화, 확장 및 구현하며 터널의 회선 프로토콜을 다운하는 기능을 제공합니다. 자세한 내용은 명령 [및 구성](#) 섹션을 참조하십시오.

tunnel keepalive 메커니즘은 다음과 같은 추가 요구 사항도 해결합니다.

- far 터널 끝점이 keepalive를 지원하지 않는 경우에도 tunnel keepalive 메커니즘이 작동합니다.
- tunnel keepalive 메커니즘이 keepalive를 시작합니다.
- tunnel keepalive 메커니즘은 keepalive 프로세스를 처리합니다.
- 터널 keepalive 메커니즘은 터널의 회선 프로토콜이 다운된 경우에도 먼 끝의 keepalive 패킷에 응답합니다.

다음은 터널 keepalive 메커니즘의 작동 방식의 예입니다([그림 1](#) 참조).

### [그림 1 - 터널 keepalive 메커니즘의 예](#)



## 출력

```

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

```

### A에서 B로 시작되는 keepalive 패킷

```

---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

```

라우터 A의 터널 엔드포인트에서 keepalive를 활성화하면 간격마다 라우터가 내부 IP 헤더를 구성합니다.헤더 끝에 라우터는 PT(Protocol Type)가 0이고 다른 페이로드가 없는 GRE 헤더도 추가합니다.그런 다음 라우터가 터널을 통해 해당 패킷을 전송하여 외부 IP 헤더와 GRE 헤더를 캡슐화하고 PT of IP가 포함된 GRE 헤더를 생성합니다.tunnel keepalive 카운터는 1씩 증가합니다.먼 엔드 터널 엔드포인트에 연결하는 방법이 있고 다른 이유로 터널 회선 프로토콜이 다운되지 않은 경우 패킷이 라우터 B에 도착합니다.그런 다음 터널 0에 대해 매칭되고, 역캡슐화되고, 터널 소스인 라우터 A인 목적지 IP에 전달됩니다. 라우터 A에 도착하면 패킷이 다시 역캡슐화되며 PT가 확인됩니다. PT 확인 결과가 0이면 keepalive 패킷임을 나타냅니다.이 경우 tunnel keepalive 카운터는 0으로 재설정되고 패킷이 삭제됩니다.

라우터 B에 연결할 수 없는 경우 라우터 A는 정상 트래픽과 함께 keepalive 패킷을 계속해서 구성하고 전송합니다.라인 프로토콜이 다운되면 keepalive가 라우터 A로 돌아가지 않으므로 keepalive 카운터가 계속 증가합니다.tunnel keepalive 카운터가 0으로 유지되거나 구성된 값보다 작으면 터널 회선 프로토콜은 작동 상태를 유지합니다.이 조건이 true가 아니면 다음에 keepalive를 라우터 B로 전송하려고 할 때 keepalive 카운터가 구성된 keepalive 값에 도달하자마자 회선 프로토콜이 중단됩니다.up/down 상태에서는 터널이 keepalive 패킷과 별도로 트래픽을 전달하거나 처리하지 않습니다.keepalive 패킷에 대해서만 작동하려면 터널이 전달 및 수신 친화적이어야 합니다.따라서 터널 조회 알고리즘은 모든 경우에 성공해야 하며, 라인 프로토콜이 다운된 경우 데이터 패킷만 삭제해야 합니다.keepalive 패킷이 수신되면 터널 엔드포인트에 다시 연결할 수 있음을 의미합니다.그런 다음 tunnel keepalive 카운터가 0으로 재설정되고 회선 프로토콜이 다시 작동합니다.

## 메모리 및 성능 영향

이 기능은 라우터 시스템 메모리에 대한 추가 수요를 거의 발생시키지 않으며, 추가해도 성능에 영향을 미치지 않을 것으로 예상됩니다.keepalive 패킷은 일반 패킷으로 처리되므로 트래픽이 많은 경우 삭제할 수 있습니다.지금은 이 문제를 처리하기 위해 재시도 횟수를 변경할 수 있습니다.이것이 결국 부적격한 것으로 판명될 경우, 로컬로 생성된 keepalive 패킷을 전송 우선 순위가 높은 대기열

에 넣을 수 있습니다.그런 다음 IP 헤더의 TOS 값을 기본값 또는 구성된 값 이외의 보다 적합한 값으로 설정할 수 있습니다.

## 패키징 고려 사항

이 기능은 기본 IP 터널 코드와 GRE 하위 시스템에 포함되어 있습니다.따라서 터널과 GRE 하위 시스템이 있는 기본 IP 패키지와 함께 사용할 수 있어야 합니다.

## 명령 및 구성

이 섹션에서는 **keepalive** 명령을 Cisco 버그 ID CSCuk26449에서만 사용하도록 설정하고 이 기능으로 확장했습니다. 다른 명령은 해당 *Cisco IOS 구성 가이드 및 명령 참조에 설명되어 있습니다*. **[no] keepalive <period> <retries>** 명령은 두 번째 매개 변수로 활성화되고 확장되며 Cisco IOS Software 릴리스 12.2(8)T 이상에서 사용할 수 있습니다.또한 Cisco 버그 ID CSCuk29980 및 CSCuk29983에서 Cisco IOS Software 릴리스 12.1E 및 12.2S로 이동되었습니다.

keepalive는 터널 인터페이스에서 keepalive를 활성화하는 인터페이스 컨피그레이션 명령이므로 현재 GRE/IP 모드에 대한 keepalive만 지원됩니다.명령의 두 번째 매개변수(재시도 횟수)는 볼 수 있으며 터널 인터페이스에만 사용할 수 있습니다.첫 번째 매개 변수의 값은 1부터 32767까지입니다. 값이 0이면 "keepalive 없음"과 같습니다. 이 매개 변수의 기본값은 10입니다. 두 번째 매개 변수의 값은 1에서 255까지입니다. 이 값은 전송되었지만 반환되지 않은 keepalive 수를 나타내며, 터널 인터페이스에서 라인 프로토콜을 아래로 가져옵니다.터널 인터페이스의 keepalive는 기본적으로 비활성화되어 있습니다.

## 샘플 출력 및 화면 형식

이 섹션에서는 샘플 출력을 제공합니다.

```
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
```

## 관련 정보

- [GRE\(Generic Routing Encapsulation\) 터널 Keepalive](#)
- [GRE 샘플 구성](#)
- [기술 지원 및 문서](#)