

Overzicht van Keepalive-mechanismen op Cisco IOS

Inhoud

[Inleiding](#)

[Achtergrondinformatie](#)

[Mechanismen voor interfacekaarten](#)

[Ethernet-interfaces](#)

[Seriële interfaces](#)

[HDLC-Keepalives](#)

[PPP-Keepalizers](#)

[GRE-tunnelinterfaces](#)

[Crypto Keepalives](#)

[IKE Keepalives](#)

[NAT-Keepalives](#)

Inleiding

Dit document beschrijft de verschillende beheermechanismen op Cisco IOS®.

Achtergrondinformatie

Levende berichten worden verzonden door een netwerkkapparaat via een fysiek of virtueel circuit om een ander netwerkkapparaat te informeren dat het circuit tussen hen nog steeds functioneert. Voor keepalives om te werken zijn er twee essentiële factoren:

- Het handhaven van het interval is de periode tussen elk keeplevend bericht dat door een netwerkkapparaat wordt verzonden. Dit is altijd aanpasbaar.
- Hiermee wordt het aantal keren herhaald dat het apparaat zonder tegenwerking levendige pakketten blijft verzenden voordat de status wordt gewijzigd in "omlaag". Voor sommige soorten keepalives is dit Configureerbaar, terwijl voor anderen er een standaardwaarde is die niet kan worden gewijzigd.

Mechanismen voor interfacekaarten

Ethernet-interfaces

Op uitzendmedia zoals een Ethernet zijn keepalives iets uniek. Omdat er veel mogelijke burens op Ethernet zijn, is het keeplevend niet ontworpen om te bepalen of het pad naar een bepaalde buurman op de draad beschikbaar is. Het is uitsluitend ontworpen om te controleren of het lokale systeem toegang tot de Ethernet-draad zelf heeft gelezen en geschreven. De router produceert een Ethernet-pakket met zichzelf als bron en bestemming MAC-adres en een speciale Ethernet-type code van 0x9000. De Ethernet hardware stuurt dit pakket naar de Ethernet-draad en ontvangt dan onmiddellijk dit pakket opnieuw. Dit controleert de verzendende en ontvangende hardware op de Ethernet-adapter en de basisintegriteit van de draad.

Source MAC 00-00-0C-04-EF-04	Destination MAC 00-00-0C-04-EF-04	Protocol Type 9000	Data 0000 0100	Layer-2 Padding 0000 ... 0000
---------------------------------	--------------------------------------	-----------------------	-------------------	----------------------------------

Seriële interfaces

De seriële interfaces kunnen verschillende types van insluitingen hebben en elk insluitingstype bepaalt het soort keepalives dat zal worden gebruikt.

Voer de opdracht keeplevend in de interface-configuratiemodus in om de frequentie in te stellen waarmee een router ECHOREQ-pakketten naar zijn peer stuurt:

- Om het systeem terug te zetten naar het standaard-interval van 10 seconden, voert u de opdracht **Keeplevo** in met het **geen**-sleutelwoord.
- Om keepalives uit te schakelen, voert u de opdracht **Keeplive uit**.

Opmerking: Het **keepalive** Deze opdracht is van toepassing op seriële interfaces die gebruik maken van HDLC (High-Level Data Link Control) of PPP-insluiting. Het is niet van toepassing op seriële interfaces die Frame Relay-insluiting gebruiken.

Opmerking: Voor zowel PPP als HDLC insluitingstypen, blokkeert een standhouden van nul keepalives en wordt gerapporteerd in de **show in werking stellen**-configuratie - opdrachtoutput als **keeplevend**.

HDLC-Keepaliven

Een ander bekend houdmechanisme is seriële keepalives voor HDLC. Seriële keepalives worden heen en weer verzonden tussen twee routers en de keepalives worden erkend. Met het gebruik van sequentienummers om elke keeplive te volgen, kan elk apparaat bevestigen of het HDLC peer de verzonden keeplevorming heeft ontvangen. Voor HDLC-insluiting zorgen drie genegerde keepalives ervoor dat de interface wordt verlaagd.

Schakel het opdracht **debug seriële interface** in voor een HDLC-verbinding om de gebruiker toe te staan keepalives te zien die gegenereerd en verzonden worden:

Sample Output:

```
17:21:09.685: Serial0/0: HDLC myseq 0, mineseen 0*, yourseen 1, line up
```

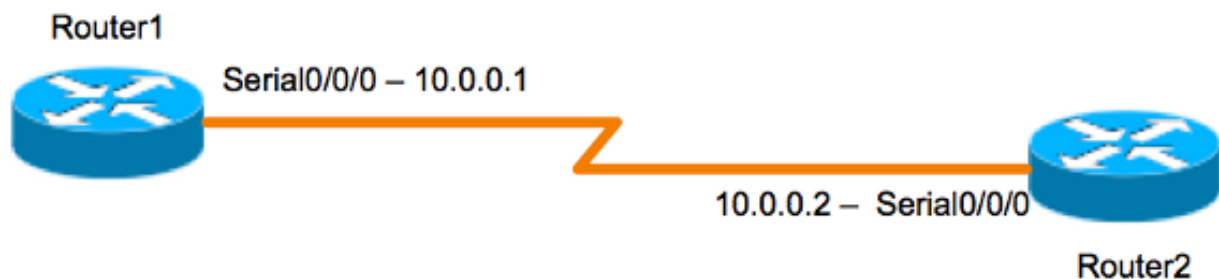
HDLC-keepalives bevatten drie stukken om te bepalen dat het werkt:

- De "myseq" is ons eigen groeiend getal.
- De "mineseen" die in feite een erkenning is van de andere kant (verhoogd), die zegt dat ze dit getal van ons verwachten.
- De "yourseen" die onze erkenning aan de andere kant is.

Opmerking: Wanneer het verschil in de waarden in de myseq en de mineseen velden drie op router 2 overschrijdt, gaat de lijn omlaag en wordt de interface gereset.

Aangezien HDLC-keepalives ECHOREQ-type keepalives zijn, is de frequentie van het behouden van de dieren belangrijk en wordt aanbevolen deze aan beide zijden exact op elkaar af te stemmen. Als de timers niet zijn gesynchroniseerd, beginnen de sequentienummers uit orde te raken. Bijvoorbeeld, als je de ene kant op 10 seconden instelt en de andere op 25 seconden, dan zal de interface blijven bestaan zolang het verschil in frequentie niet voldoende is om de sequentienummers met een verschil van drie uit te zetten.

Als illustratie van hoe HDLC het werk in stand houdt, worden router 1 en router 2 direct aangesloten via respectievelijk Seriele0/0 en Serial2/0. Om aan te geven hoe de defecte HDCL-keepalives worden gebruikt om de interfacestaten te volgen, zal Seriele 0/0 op Router 1 worden uitgeschakeld.



router 1

```
Router1#show interfaces serial 0/0/0
Serial0/0/0 is up, line protocol is up (connected)
Hardware is HD64570
Internet address is 10.0.0.1/8
MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 1/255
Encapsulation HDLC, loopback not set, keepalive set (10 sec)
[output is omitted]

17:21:09.685: Serial0/0: HDLC myseq 0, mineseen 0*, yourseen 1, line up
17:21:19.725: Serial0/0: HDLC myseq 1, mineseen 1*, yourseen 2, line up
17:21:29.753: Serial0/0: HDLC myseq 2, mineseen 2*, yourseen 3, line up
17:21:39.773: Serial0/0: HDLC myseq 3, mineseen 3*, yourseen 4, line up
17:21:49.805: Serial0/0: HDLC myseq 4, mineseen 4*, yourseen 5, line up
17:21:59.837: Serial0/0: HDLC myseq 5, mineseen 5*, yourseen 6, line up
17:22:09.865: Serial0/0: HDLC myseq 6, mineseen 6*, yourseen 7, line up
17:22:19.905: Serial0/0: HDLC myseq 7, mineseen 7*, yourseen 8, line up
17:22:29.945: Serial0/0: HDLC myseq 8, mineseen 8*, yourseen 9, line up
Router1 (config-if)#shut
17:22:39.965: Serial0/0: HDLC myseq 9, mineseen 9*, yourseen 10, line up
17:22:42.225: %LINK-5-CHANGED: Interface Serial0/0, changed state
to administratively down
```

```
17:22:43.245: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0,  
changed state to down
```

router 2

```
Router2#show interfaces serial 0/0/0  
Serial0/0/0 is up, line protocol is up (connected)  
Hardware is HD64570  
Internet address is 10.0.0.2/8  
MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 1/255  
Encapsulation HDLC, loopback not set, keepalive set (10 sec)  
[output is omitted]
```

```
17:21:04.929: Serial2/0: HDLC myseq 0, mineseen 0, yourseen 0, line up  
17:21:14.941: Serial2/0: HDLC myseq 1, mineseen 1*, yourseen 1, line up  
17:21:24.961: Serial2/0: HDLC myseq 2, mineseen 2*, yourseen 2, line up  
17:21:34.981: Serial2/0: HDLC myseq 3, mineseen 3*, yourseen 3, line up  
17:21:45.001: Serial2/0: HDLC myseq 4, mineseen 4*, yourseen 4, line up  
17:21:55.021: Serial2/0: HDLC myseq 5, mineseen 5*, yourseen 5, line up  
17:22:05.041: Serial2/0: HDLC myseq 6, mineseen 6*, yourseen 6, line up  
17:22:15.061: Serial2/0: HDLC myseq 7, mineseen 7*, yourseen 7, line up  
17:22:25.081: Serial2/0: HDLC myseq 8, mineseen 8*, yourseen 8, line up  
17:22:35.101: Serial2/0: HDLC myseq 9, mineseen 9*, yourseen 9, line up  
17:22:45.113: Serial2/0: HDLC myseq 10, mineseen 10*, yourseen 10, line up  
17:22:55.133: Serial2/0: HDLC myseq 11, mineseen 10, yourseen 10, line up  
17:23:05.153: HD(0): Reset from 0x203758  
17:23:05.153: HD(0): Asserting DTR  
17:23:05.153: HD(0): Asserting DTR and RTS  
17:23:05.153: Serial2/0: HDLC myseq 12, mineseen 10, yourseen 10, line up  
17:23:15.173: HD(0): Reset from 0x203758  
17:23:15.173: HD(0): Asserting DTR  
17:23:15.173: HD(0): Asserting DTR and RTS  
17:23:15.173: Serial2/0: HDLC myseq 13, mineseen 10, yourseen 10, line down  
17:23:16.201: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0,  
changed state to down  
Router2#  
17:23:25.193: Serial2/0: HDLC myseq 14, mineseen 10, yourseen 10, line down
```

PPP-Keepalizers

PPP keepalives zijn een beetje anders dan HDLC-keepalives. In tegenstelling tot HDLC lijken PPP-keepalives meer op pings. Beide kanten kunnen elkaar in hun vrije tijd pingelen. De juiste stap in de onderhandelingen is om **ALTIJD** te reageren op deze 'ping'. Dus voor PPP keepalives zijn de frequentie of de timer waarde slechts lokaal relevant en hebben deze geen impact op de andere kant. Zelfs als de ene kant de keepalives uitzet, zal het nog steeds reageren op die echo-verzoeken van de kant die een sluitingtimer hebben. Maar het zal nooit een eigen initiatief nemen.

Schakel het opdracht **debug ppp** voor een PPP-verbinding in om de gebruiker in staat te stellen de PPP-toetsen te zien die worden verzonden:

```
17:00:11.412: Se0/0/0 LCP-FS: I ECHOREQ [Open] id 32 len 12 magic 0x4234E325  
en ontvangen reacties:
```

```
17:00:11.412: Se0/0/0 LCP-FS: O ECHOREP [Open] id 32 len 12 magic 0x42345A4D
```

PPP keepalives bevatten drie stukken:

- ID - nummer - gebruikt om te identificeren op welke ECHOREQ de peer reageert.
- Type - ECHOREQ is keepaliefaten die door het oorspronkelijke apparaat worden verstuurd en ECHOREP is een reactie die door het peer wordt verstuurd.
- Magische getallen - de meldingen bevatten de magische nummers van zowel de server als de externe client. De peer bevestigt het magische nummer in het pakket LCP Echo-request en geeft het corresponderende LCP Echo-antwoord pakket over dat het magische nummer bevat dat door de router onderhandeld is.

Voor PPP-insluiting zorgen vijf genegeerde keepalives ervoor dat de interface wordt verlaagd

GRE-tunnelinterfaces

Het GRE-tunnelbeheermechanisme is iets anders dan voor Ethernet- of seriële interfaces. Het geeft de mogelijkheid voor één kant om levendige pakketten aan en van een verre router te produceren en te ontvangen zelfs als de afstandsrouter geen GRE keepalives ondersteunt. Aangezien GRE een pakketunnelmechanisme is voor het tunnelen van IP in IP, kan een GRE IP-tunnelpakket in een ander GRE IP-tunnelpakket worden gebouwd. Voor GRE keepalives bouwt de afzender het pakket van de keeplevende reactie binnen het originele pakket van de keeplevende om te vragen zodat het verre eind slechts standaard GRE decapsulation van de buitenste GRE IP header moet uitvoeren en dan het binnenste IP GRE-pakket door sturen. Dit mechanisme veroorzaakt de aanhoudende respons om de fysieke interface uit te sturen in plaats van de tunnelinterface. Voor meer informatie over het werken van GRE-tunneleigenschappen, zie [hoe GRE Keepalives werkt](#).

Crypto Keepalives

IKE Keepalives

De keepalives van Internet Key Exchange (IKE) zijn een mechanisme dat wordt gebruikt om te bepalen of een VPN-peer in staat is om versleuteld verkeer te ontvangen. Afzonderlijke crypto keepalives zijn vereist naast interface keepalives omdat VPN peers over het algemeen nooit terug naar rug worden aangesloten, zodat de interface keepalives niet genoeg informatie over staat van de VPN peer verstrekken.

Op Cisco IOS-apparaten worden IKE-keepalives geactiveerd door het gebruik van een eigen methode die Dead Peer Detection (DPD) wordt genoemd. Om de gateway toe te staan om DPDs naar de peer te verzenden, voer deze opdracht in globale configuratiewijze:

```
crypto isakmp keepalive seconds [retry-seconds] [ periodic | on-demand ]
```

Gebruik het "nee" formulier van deze opdracht om keepalives uit te schakelen. Voor meer informatie over wat elk sleutelwoord in deze opdracht doet, zie [crypto isakmp keeplevend](#). Voor een grotere granulariteit kunnen de keepalives ook worden geconfigureerd onder het ISAKMP-profiel. Zie [ISAKMP Profile - Overzicht \[Cisco IOS IPsec\]](#) voor meer informatie.

NAT-Keepalives

In geval van scenario's waar één VPN-peer achter een Network adresomzetting (NAT) zit, wordt NAT-Traversal gebruikt voor encryptie. Tijdens stationaire periodes is het echter mogelijk dat het NAT-punt op het upstream-apparaat uitvalt. Dit kan problemen veroorzaken als je de tunnel omhoog brengt en NAT is niet tweerichtings. NAT-keepalives zijn ingeschakeld om de dynamische NAT-omzetting levend te houden tijdens een verbinding tussen twee peers. NAT-keepalives zijn UDP-pakketten met een niet-versleutelde lading van één byte. Hoewel de huidige DPD-implementatie gelijk is aan NAT-keepalives, is er een klein verschil - DPD wordt gebruikt om peer status te detecteren terwijl NAT-keepalives worden verzonden als de IPsec-entiteit het pakket niet op een bepaalde tijd heeft verzonden of ontvangen. Het geldige bereik ligt tussen 5 en 3600 seconden.

Tip: Als NAT keepalives zijn ingeschakeld (door de opdracht **crypto-isakmp nat keeplevendig**), dienen gebruikers ervoor te zorgen dat de stationaire waarde korter is dan de verlooptijd van NAT-mapping van 20 seconden.

Zie [IPsec NAT Transparency](#) voor meer informatie over deze optie.