# Présentation des mécanismes Keepalive sur Cisco IOS

## Table des matières

[Introduction](#)
[Informations générales](#)
[Mécanismes de maintien de connexion](#)
[Interfaces Ethernet](#)
[Interfaces série](#)
[Keepalives HDLC](#)
[Keepalives PPP](#)
[Interfaces de tunnel GRE](#)
[Crypto Keepalives](#)
[IKE Keepalives](#)
[NAT Keepalives](#)

## Introduction

Ce document décrit les différents mécanismes de test d'activité sur Cisco IOS®.

## Informations générales

Les messages Keepalive sont envoyés par un périphérique réseau via un circuit physique ou virtuel afin d'informer un autre périphérique réseau que le circuit entre eux fonctionne toujours. Pour que les keepalives fonctionnent, il faut tenir compte de deux facteurs essentiels :

- L'intervalle de test d'activité est l'intervalle de temps entre chaque message de test d'activité envoyé par un périphérique réseau. Cette option est toujours configurable.
- Le nombre de nouvelles tentatives de test d'activité correspond au nombre de fois où le périphérique continue d'envoyer des paquets de test d'activité sans réponse avant que l'état ne passe à « inactif ». Pour certains types de keepalive, cette option est configurable, alors que pour d'autres, il existe une valeur par défaut qui ne peut pas être modifiée.

## Mécanismes de maintien de connexion

### Interfaces Ethernet

Sur un support de diffusion tel qu'un réseau Ethernet, les messages de veille sont légèrement uniques. Comme il existe de nombreux voisins possibles sur l’Ethernet, le keepalive n’est pas conçu pour déterminer si le chemin vers un voisin particulier sur le câble est disponible. Il est

uniquement conçu pour vérifier que le système local dispose d’un accès en lecture et en écriture au câble Ethernet lui-même. Le routeur produit un paquet Ethernet avec lui-même comme adresse MAC source et de destination et un code de type Ethernet spécial de 0x9000. Le matériel Ethernet envoie ce paquet sur le câble Ethernet, puis le reçoit immédiatement à nouveau. Cela permet de vérifier le matériel émetteur et récepteur sur l'adaptateur Ethernet et l'intégrité de base du câble.

## Interfaces série

Les interfaces série peuvent avoir différents types d’encapsulation et chaque type d’encapsulation détermine le type de keepalive qui sera utilisé.

Entrez la commande keepalive en mode de configuration d'interface afin de définir la fréquence à laquelle un routeur envoie des paquets ECHOREQ à son homologue :

- Afin de restaurer le système à l'intervalle keepalive par défaut de 10 secondes, entrez la commande keepalive avec le mot clé no.
- Afin de désactiver les keepalives, entrez la commande keepalive disable.

---

Remarque : La keepalive commande s’applique aux interfaces série qui utilisent l’encapsulation HDLC (High-Level Data Link Control) ou PPP. Elle ne s’applique pas aux interfaces série qui utilisent l’encapsulation Frame Relay.

---

---

Remarque : Pour les types d'encapsulation PPP et HDLC, un keepalive de zéro désactive les keepalives et est signalé dans le résultat de la commande show running-config comme keepalive disable.

---

### Keepalives HDLC

Un autre mécanisme de keepalive bien connu est les keepalives série pour HDLC. Les messages de test d’activité en série sont envoyés entre deux routeurs et les messages de test d’activité sont confirmés. Avec l'utilisation de numéros de séquence pour suivre chaque keepalive, chaque périphérique est en mesure de confirmer si son homologue HDLC a reçu le keepalive qu'il a envoyé. Pour l’encapsulation HDLC, trois messages de test d’activité ignorés provoquent l’arrêt de l’interface.

Activez la commande debug serial interface pour une connexion HDLC afin de permettre à l'utilisateur de voir les messages de test d'activité qui sont générés et envoyés :

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

Les keepalives HDLC contiennent trois éléments afin de déterminer son fonctionnement :

- Le "myseq" qui est notre propre nombre incrémentant.
- Le "vu des mines" qui est en fait un accusé de réception de l'autre côté (incrémenté) qui dit qu'ils attendent ce nombre de nous.
- Le "yourseen" qui est notre reconnaissance à l'autre côté.

---

Remarque : Lorsque la différence entre les valeurs des champs myseq et mineseen dépasse trois sur le routeur 2, la ligne s'arrête et l'interface est réinitialisée.

---

Puisque les keepalives HDLC sont des keepalives de type ECHOREQ, la fréquence de keepalive est importante et il est recommandé qu'ils correspondent exactement des deux côtés.  Si les minuteurs ne sont pas synchronisés, les numéros d'ordre commencent à être désordonnés. Par exemple, si vous définissez un côté sur 10 secondes et l'autre sur 25 secondes, l'interface restera active tant que la différence de fréquence n'est pas suffisante pour que les numéros de séquence soient désactivés d'une différence de trois.

Pour illustrer le fonctionnement des keepalives HDLC, les routeurs 1 et 2 sont connectés directement via Serial0/0 et Serial2/0, respectivement. Afin d'illustrer comment les keepalives HDCL défaillants sont utilisés pour suivre les états d'interface, Serial 0/0 sera arrêté sur le Routeur 1.

Routeur 1

<#root>

```
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 omited]

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
```

Routeur 2

<#root>

```
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 omited]


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
```

Keepalives PPP

Les keepalives PPP sont légèrement différents des keepalives HDLC. Contrairement à HDLC, les keepalives PPP ressemblent davantage à des requêtes ping. Les deux parties peuvent s'envoyer des requêtes ping quand elles le souhaitent. Le mouvement négocié approprié est de TOUJOURS répondre à cette « requête ping ». Ainsi, pour les keepalives PPP, la fréquence ou la valeur de temporisateur sont uniquement pertinentes localement et n'ont aucun impact de l'autre côté.

Même si un côté désactive les messages de veille, il répond toujours aux requêtes d'écho du côté qui dispose d'un compteur de messages de veille. Cependant, il n'initiera jamais rien de son propre.

Activez la commande debug ppp packet pour une connexion PPP afin de permettre à l'utilisateur de voir les messages de test d'activité PPP qui sont envoyés :

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

et les réponses reçues :

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

Les keepalives PPP contiennent trois éléments :

- ID number : utilisé pour identifier l'ECHOREQ auquel l'homologue répond.
- Type de test d'activité : ECHOREQ sont des tests d'activité envoyés par le périphérique d'origine et ECHOREP sont des réponses envoyées par l'homologue.
- Numéros magiques : les notifications incluent les numéros magiques du serveur et du client distant. L'homologue valide le nombre magique dans le paquet LCP Echo-Request et transmet le paquet LCP Echo-Reply correspondant qui contient le nombre magique négocié par le routeur.

Pour l'encapsulation PPP, cinq keepalives ignorés provoquent l'arrêt de l'interface

### Interfaces de tunnel GRE

Le mécanisme de keepalive du tunnel GRE est légèrement différent de celui des interfaces Ethernet ou série. Il permet à un côté d'émettre et de recevoir des paquets de test d'activité vers et depuis un routeur distant, même si ce dernier ne prend pas en charge les tests d'activité GRE. Puisque GRE est un mécanisme de tunneling de paquets pour le tunneling IP dans IP, un paquet de tunnel IP GRE peut être construit dans un autre paquet de tunnel IP GRE. Pour les keepalives GRE, l'expéditeur préconstruit le paquet de réponse keepalive à l'intérieur du paquet de demande de keepalive d'origine de sorte que l'extrémité distante n'ait besoin que d'effectuer une décapsulation GRE standard de l'en-tête IP GRE externe, puis de transférer le paquet GRE IP interne. Ce mécanisme fait que la réponse keepalive transfère l'interface physique plutôt que l'interface de tunnel. Pour plus de détails sur le fonctionnement des keepalives de tunnel GRE, consultez Comment fonctionnent les keepalives de GRE.

## Crypto Keepalives

## IKE Keepalives

Les keepalives IKE (Internet Key Exchange) sont un mécanisme utilisé pour déterminer si un homologue VPN est actif et capable de recevoir du trafic chiffré. Des keepalives de chiffrement séparés sont requis en plus des keepalives d'interface car les homologues VPN ne sont généralement jamais connectés dos à dos, de sorte que les keepalives d'interface ne fournissent pas suffisamment d'informations sur l'état de l'homologue VPN.

Sur les périphériques Cisco IOS, les keepalives IKE sont activés par l'utilisation d'une méthode propriétaire appelée Dead Peer Detection (DPD). Afin de permettre à la passerelle d'envoyer des DPD à l'homologue, entrez cette commande en mode de configuration globale :

<#root>

**crypto**

**isakmp**

**keepalive**

*seconds* *[retry-seconds]* [

**periodic**

**| on-demand**

]

Afin de désactiver les keepalives, utilisez la forme "no" de cette commande. Pour plus d'informations sur ce que fait chaque mot clé dans cette commande, consultez crypto isakmp keepalive. Pour plus de granularité, les keepalives peuvent également être configurés sous le profil ISAKMP. Pour plus de détails, consultez Vue d'ensemble du profil ISAKMP [Cisco IOS IPsec].

## NAT Keepalives

Dans les cas où un homologue VPN est derrière une traduction d'adresses de réseau (NAT), NAT-Traversal est utilisé pour le chiffrement. Cependant, pendant les périodes d'inactivité, il est possible que l'entrée NAT sur le périphérique en amont expire. Cela peut causer des problèmes lorsque vous activez le tunnel et que la NAT n'est pas bidirectionnelle. Les keepalives NAT sont activés afin de maintenir le mappage NAT dynamique actif pendant une connexion entre deux homologues. Les keepalives NAT sont des paquets UDP avec une charge utile non chiffrée d'un octet. Bien que l'implémentation DPD actuelle soit similaire aux keepalives NAT, il existe une légère différence : DPD est utilisé pour détecter l'état de l'homologue tandis que les keepalives NAT sont envoyés si l'entité IPsec n'a pas envoyé ou reçu le paquet à une période spécifiée. La plage valide est comprise entre 5 et 3 600 secondes.

Conseil : Si les keepalives NAT sont activés (via la commande crypto isamkp nat keepalive), les utilisateurs doivent s'assurer que la valeur d'inactivité est plus courte que le temps d'expiration du mappage NAT de 20 secondes.

Pour plus d'informations sur cette fonctionnalité, consultez Transparence NAT IPsec.

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.