

# Cisco IOS 上的保持连接机制的概述

## 目录

[简介](#)

[背景信息](#)

[接口保持连接机制](#)

[以太网接口](#)

[串行接口](#)

[HDLC Keepalive](#)

[PPP Keepalive](#)

[GRE隧道接口](#)

[加密Keepalive](#)

[IKE保活](#)

[NAT Keepalive](#)

## 简介

本文档介绍Cisco IOS®上的各种保持连接<sup>机制</sup>。

## 背景信息

保持连接消息由一台网络设备通过物理或虚电路发送，以便通知另一台网络设备它们之间的电路仍然有效。要使keepalive正常工作，有两个基本因素：

- Keepalive 间隔是网络设备发送的两个 Keepalive 消息之间的时间段。这始终是可配置的。
- keepalive retries是设备在状态更改为“down”之前继续发送保持连接数据包而无响应的次数。对于某些类型的keepalive，这是可配置的，而对于其他类型，默认值是无法更改的。

## 接口保持连接机制

### 以太网接口

在广播介质（如以太网）上，keepalive稍微是唯一的。以太网上可能存在许多邻居，因此，Keepalive 不会确定指向线路上任何特定邻居的路径是否可用。它仅检查本地系统是否拥有对以太网线路的读取和写入访问权限。路由器生成一个以太网数据包，其本身作为源和目的MAC地址，并使用特殊的以太网类型代码0x9000。以太网硬件将此数据包发送到以太网线路，然后立即再次接收该数据包。这样可以检查以太网适配器的发送和接收硬件以及线路的基本完整性。

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
---------------------------------	--------------------------------------	-----------------------	-------------------	----------------------------------

## 串行接口

串行接口可能具有不同的封装类型，每种封装类型确定将使用的keepalive类型。

在接口配置模式下输入**keepalive**命令，以设置路由器向其对等设备发送ECHOREQ数据包的频率：

- 要将系统恢复为默认的keepalive间隔10秒，请输入带有**no**关键字的**keepalive**命令。
- 要禁用keepalive，请输入**keepalive disable**命令。

**注意：**的 **keepalive** 命令适用于使用高级数据链路控制(HDLC)或PPP封装的串行接口。它不适用于使用帧中继封装的串行接口。

**注意：**对于PPP和HDLC封装类型，零的keepalive会禁用keepalive，并在show running-config命令输出中报告为**keepalive disable**。

## HDLC Keepalive

另一个众所周知的keepalive机制是HDLC的串行keepalive。串行 Keepalive 在两个路由器之间来回发送，Keepalive 得到确认。通过使用序列号来跟踪每台保持连接，每台设备都能够确认其HDLC对等体是否收到了发送的保持连接。对于HDLC封装，三个忽略的keepalive会导致接口关闭。

为HDLC连接启用**debug serial interface**命令，以允许用户查看生成和发送的keepalive:

Sample Output:

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

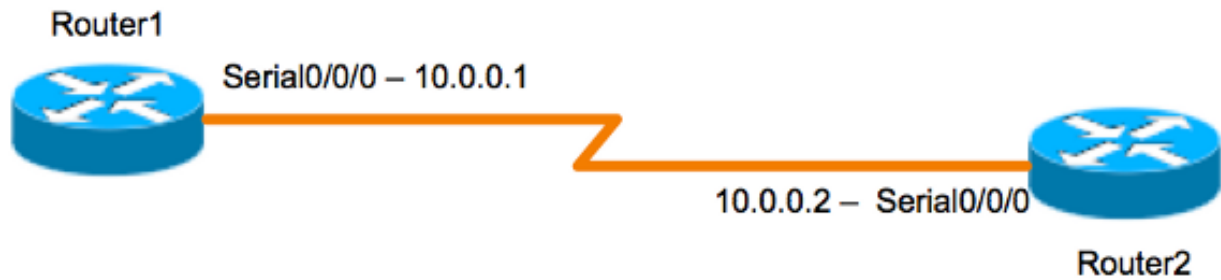
HDLC keepalive包含三个部分，以确定其有效性：

- “myseq”是我们自己的递增数。
- “mineseen”实际上是来自另一端的确认（递增），表示他们希望我们提供此编号。
- “yourseen”，即我们对另一方的认可。

**注意：**当myseq和mineen字段中的值差超过路由器2上的3时，线路将关闭，接口将重置。

由于HDLC保活是ECHOREQ类型的保活，因此保持活频率很重要，建议它们在两端精确匹配。如果计时器不同步，则序列号开始无序。例如，如果将一端设置为10秒，将另一端设置为25秒，则只要频率差不足以导致序列号的差值为3，接口仍将保持打开状态。

作为HDLC保持连接工作原理的示例，路由器1和路由器2分别通过Serial0/0和Serial2/0直接连接。为了说明如何使用失败的HDCL Keepalive来跟踪接口状态，Router 1上的Serial 0/0将关闭。



## 路由器 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
```

## 路由器 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 Keepalive

PPP keepalive与HDLC keepalive略有不同。与HDLC不同，PPP keepalive更像ping。双方可以在闲暇时互相执行ping操作。协商的正确操作是始终响应此“ping”。因此，对于PPP Keepalive，频率或计时器值仅与本地相关，对另一端没有影响。即使一端关闭keepalive，它仍会响应来自具有keepalive计时器的一端的回应请求。但是，它永远不会发起任何自己的行动。

为PPP连接启用debug ppp packet命令，以允许用户查看发送的PPP Keepalive:

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

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

PPP Keepalive包含三个部分：

- ID号 — 用于标识对等体响应的ECHOREQ。
- Keepalive类型 — ECHOREQ是始发设备发送的Keepalive，ECHOREP是对等设备发送的响应。
- 幻数 — 通知包括服务器和远程客户端的幻数。对等体验证LCP Echo-Request数据包中的幻数，并传输包含路由器协商的幻数的相应LCP Echo-Reply数据包。

对于PPP封装，五个忽略的keepalive会导致接口关闭

## GRE隧道接口

GRE 隧道 Keepalive 机制对于以太网或串行接口稍有不同。它使一端能够与远程路由器之间发起和接收 Keepalive 数据包（即使远程路由器不支持 GRE Keepalive）。GRE 是 IP 内隧道传输 IP 的数据包隧道机制，因此，GRE IP 隧道数据包可以在另一个 GRE IP 隧道数据包里构建。对于 GRE Keepalive，发送方在原始 Keepalive 请求包中预先构建 Keepalive 响应数据包，远程端只需对外部 GRE IP 报头执行标准 GRE 解封，然后转发内部 IP GRE 数据包。通过这种机制，可以将 Keepalive 响应转发出物理接口而不是隧道接口。有关GRE隧道Keepalive的工作的详细信息，请参阅[GRE Keepalive如何工作](#)。

## 加密Keepalive

## IKE保活

互联网密钥交换(IKE)保活是一种机制，用于确定VPN对等体是否已启用并能够接收加密流量。除接口keepalive外，还需要单独的加密keepalive，因为VPN对等体通常从不背对背连接，因此接口keepalive无法提供有关VPN对等体状态的足够信息。

在Cisco IOS设备上，IKE保活通过使用称为失效对等体检测(DPD)的专有方法启用。要允许网关将DPD发送到对等设备，请在全局配置模式下输入以下命令：

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

要禁用keepalive，请使用此命令的“no”形式。有关此命令中每个关键字的功能的详细信息，请参阅[crypto isakmp keepalive](#)。为了更精细，也可以在ISAKMP配置文件下配置keepalive。有关详细信息，请参阅[ISAKMP配置文件概述\[Cisco IOS IPsec\]](#)。

## NAT Keepalive

在网络地址转换(NAT)后面有一个VPN对等体的情况下，NAT遍历用于加密。但是，在空闲期间，上游设备上的NAT条目可能超时。当您启动隧道且NAT不是双向时，这可能会导致问题。启用NAT保持连接，以便在两个对等体之间连接期间保持动态NAT映射的活动状态。NAT Keepalive是未加密负载为一个字节的UDP数据包。虽然当前DPD实施与NAT Keepalive类似，但有一点差别——如果IPsec实体在指定时间段内未发送或接收数据包，则DPD用于检测对等体状态，同时发送NAT Keepalive。有效范围为5到3600秒。

**提示：**如果启用了NAT保活(通过crypto isamkp nat keepalive 命令)，则用户应确保空闲值比NAT映射到期时间(20秒)短。

有关此功能的详细信息，请参见[IPsec NAT透明度](#)。