

# 保活机制概述在Cisco IOS的

## 目录

[简介](#)

[背景信息](#)

[接口keepalive机制](#)

[以太网接口](#)

[串行接口](#)

[HDLC Keepalive](#)

[PPP Keepalive](#)

[GRE隧道接口](#)

[crypto Keepalive](#)

[IKE Keepalive](#)

[NAT Keepalive](#)

## 简介

本文描述在Cisco IOS的多种保活机制。

## 背景信息

保活信息由一个网络设备传送通过一条物理或虚拟电路为了仍然通知该另一个的网络设备在他们之间的电路功能。为了使工作的Keepalive有两个重要要素：

- Keepalive 间隔是网络设备发送的两个 Keepalive 消息之间的时间段。这总是可配置。
- Keepalive重试次数是设备继续发送keepalive数据包，不用答复的次数，在状态更改对“下来”前。对于Keepalive一些类型这可配置，而其他的有不可能更改的默认值。

## 接口keepalive机制

### 以太网接口

在广播价质例如以太网，Keepalive是轻微唯一。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，将自己作为源和目标 MAC 地址，并使用特殊的以太网类型代码 0x9000。以太网硬件将该数据包发送到以太网线路，随后立即接收该数据包。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

Serial interfaces能有不同种类的封装，并且每种封装类型确定将使用的这Keepalive。

输入在接口配置模式的**保活命令**为了设置路由器发送ECHOREQ数据包给其对等体的频率：

- 为了恢复系统到默认Keepalive间隔10秒，请输入与**没有关键字的保活命令**。
- 为了禁用Keepalive，请输入**disable命令的Keepalive**。

**注意：** **keepalive** 命令适用对使用高级数据链路Control的serial interfaces (HDLC)或PPP封装。它不适用于使用帧中继封装的serial interfaces。

**注意：** 对于PPP和HDLC封装类型，Keepalive零在**show running-config命令**输出中禁用Keepalive和报告作为**Keepalive禁用**。

## HDLC Keepalive

另一个著名的保活机制是HDLC的序列Keepalive。串行 Keepalive 在两个路由器之间来回发送，Keepalive 得到确认。使用使用序号跟踪每Keepalive，每个设备能确认发送的它是否是HDLC对等体接收Keepalive。对于HDLC封装，三忽略Keepalive促成接口被建立下来。

启用**debug serial interface命令**HDLC连接的为了允许用户发现生成并且被发送的Keepalive：

Sample Output:

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

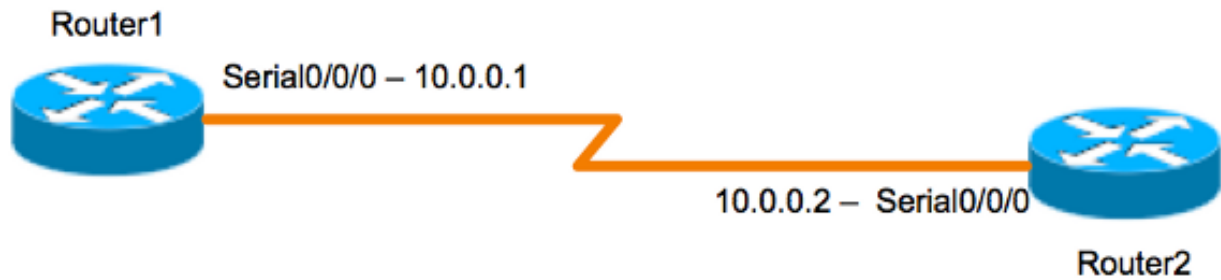
HDLC Keepalive包含三个片段为了确定它工作：

- “myseq”是我们自己增加的编号。
- “mineseen”实际上是从说的另一侧的一确认(被增加)他们预计从我们的此编号。
- “yourseen”是我们的确认对另一侧。

**注意：** 当差异按值在myseq和mineseen字段超过三在Router2时，线路断开，并且接口重置。

因为HDLC Keepalive是ECHOREQ类型Keepalive，保活频率是重要，并且推荐他们在两边正确地配合。如果计时器是不同步的，离开的序号开始秩序。例如，如果集一端到10秒和其他对25秒，它将允许接口保持，只要在频率的差异不是满足造成序号关闭由差异三。

作为例证HDLC Keepalive如何工作，路由器1和Router2通过各自Serial0/0和Serial2/0直接地连接。为了说明失败的HDCL Keepalive如何用于跟踪接口状态，Serial0/0在路由器1.将被关闭。



## 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是稍微与HDLC Keepalive不同。不同于HDLC，PPP Keepalive是更多类似ping。两边能互相ping在他们的休闲。适当的经过协商的移动是总是响应对此“ping”。因此对于PPP Keepalive，频率或计时器值是只本地相关的并且没有在另一侧的影响。即使一端关闭Keepalive，将回答从有保活计时器的侧的那些ECHO请求。然而，它不会启动其自己中的任一。

启用debug ppp packet命令PPP连接的为了允许用户发现被发送的PPP Keepalive：

```
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
```

并且接收的答复：

```
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包含三个片段：

- ID号码-曾经识别哪ECHOREQ对等体回应。
- 保活类型- ECHOREQ是始发设备发送的Keepalive，并且ECHOREP是对等体发送的答复。
- 幻数-通知包括服务器和远程客户端幻数。对等体验证在LCP ECHO请求数据包的幻数，并且发送包含路由器协商的幻数的对应的LCP ECHO回复数据包。

对于PPP封装，五忽略Keepalive促成接口被建立下来

## 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。欲了解更详细的信息在工作GRE隧道 Keepalive，请参阅[GRE Keepalive如何工作](#)。

## crypto Keepalive

### IKE Keepalive

Internet Key Exchange (IKE) Keepalive是用于的机制确定VPN对等项是否上和能收到加密流量。分开的crypto Keepalive要求除接口keepalive之外，因为VPN对等体通常紧接从未连接，因此接口

keepalive不提供关于VPN对等项的状态的足够的信息。

在Cisco IOS设备上，IKE Keepalive启用使用呼叫对端死机检测(DPD)的专有方法。为了允许网关发送DPDs对对等体，请输入此in命令全局配置模式：

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

为了禁用Keepalive，请使用此命令no表。关于什么的更多信息在此命令的每个关键字执行，请参阅[crypto isakmp keepalive](#)。对于更多粒度，Keepalive可能也配置在ISAKMP简档下。欲了解更详细的信息，请参阅[ISAKMP简档概述\[Cisco IOS IPsec\]](#)。

## NAT Keepalive

在一VPN对等项是在网络地址转换(NAT)后的方案的情况下，NAT遍历使用加密。然而，在空闲期间很可能，在上行设备的NAT条目也许计时。这能引起问题，当您启动通道时，并且NAT不双向。NAT Keepalive启用为了保持动态NAT映射运行在两对等体之间的一连接时。NAT Keepalive是有一个字节一未加密有效负载的UDP数据包。虽然当前DPD实施类似于NAT Keepalive，有轻微的区别-DPD用于检测对等体状态，当NAT Keepalive被发送时，如果IPsec实体没有发送也没有收到数据包在特定的时间。有效范围是在5到3600秒之间。

**提示：**如果NAT Keepalive启用(通过**crypto isakmp nat保活命令**)，用户应该保证空闲值比映射有效期20秒的NAT短。

关于此功能的更多信息，请参阅[IPsec NAT透明度](#)。