# 排除EEM和EPC间歇性路由协议抖动故障

## 目录



## 简介

本文档介绍如何对带有EEM和EPC的Cisco IOS® XE中的间歇性路由协议抖动和BFD抖动进行故障排除。

## 先决条件

### 要求

建议熟悉用于故障排除的平台以及Wireshark的嵌入式事件管理器(EEM)和嵌入式数据包捕获(EPC)的详细信息。此外，建议熟悉路由协议和双向转发检测(BFD)的基本hello和keepalive功能。

### 使用的组件

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原

始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

# 问题概述

间歇性路由协议抖动是生产网络中的常见问题，但由于其不可预知的特性，很难实时对其进行故障排除。EEM能够在发生摆动时通过系统日志字符串触发数据捕获，从而自动进行数据收集。使用EEM和EPC，数据包捕获数据可以从邻接的两端收集，以在抖动发生之前隔离潜在的数据包丢失。

间歇性路由协议摆动的本质是它们始终是由于hello或keepalive超时造成的（除非是明显的物理问题，如日志中会出现链路摆动）。 因此，这是本文档中的逻辑所涵盖的内容。

## 故障排除方法

确定路由协议抖动何时发生的最重要的事情是，在问题发生时，两台设备是否都发送和接收了hello数据包或keepalive数据包。此故障排除方法涉及在循环缓冲区上使用连续的EPC，直到发生摆动，此时EEM使用相关系统日志字符串触发一组命令运行，其中一个命令会停止EPC。循环缓冲区选项允许EPC继续捕获新数据包，同时覆盖缓冲区中最旧的数据包，从而确保捕获事件且缓冲区不会提前填满和停止。然后可以将分组捕获数据与抖动的时间戳相关联，以确定在事件之前两端是否发送和接收了必要的分组。

此问题最常见于通过中间网络（如Internet服务提供商[ISP]）形成邻接关系的设备，但是同样的方法也适用于任何间歇性路由协议抖动场景，无论具体的拓扑细节如何。在邻居设备由第三方管理且无法访问的情况下，也可以执行相同操作。在这种情况下，本文档中描述的故障排除方法可以只应用于可访问的一个设备，以证明该设备在抖动之前是否发送和接收了所需的数据包。确认后，数据可以显示给管理邻居的一方，以便在需要时进一步排除另一端的故障。

# 配置概述

本部分提供了一组配置模板，可用于设置此自动数据捕获。根据需要修改IP地址、接口名称和文件名。

## ACL配置模板

在大多数情况下，路由邻接两端接口的IP地址中唯一发出的流量是路由控制流量本身。因此，允许从本地接口IP地址和邻居IP地址到任何目的地的流量的ACL满足任何路由协议和BFD的要求。如果需要额外的过滤器，也可以根据路由协议或BFD模式指定相关的目标IP。在配置模式下定义ACL参数：

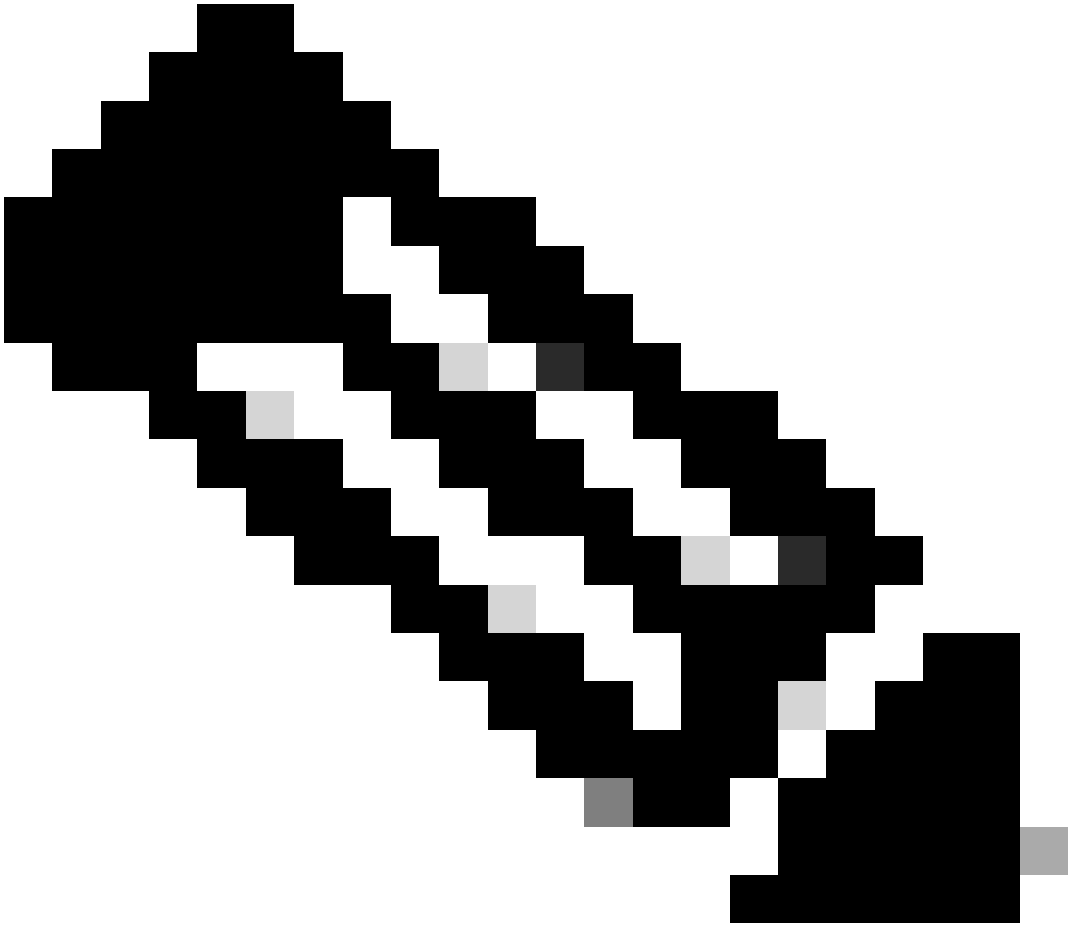
```
config t

ip access-list extended
```

```
permit ip host

 any permit ip host

  any end
```

## EPC参数模板

EPC参数是在特权执行模式而非配置模式下创建的。请务必检查特定于平台的配置指南，以确定EPC是否存在任何限制。为所需接口创建参数，并将其与要过滤所需流量的ACL关联：

- monitor capture <EPC name> interface <interface> both
- monitor capture <EPC name> access-list <ACL name>
- monitor capture <EPC name> buffer size 5 circular



注意：在某些软件版本中，本地生成的流量在接口级EPC中不可见。在这种情况下，可以更改捕获参数以捕获CPU上的两个流量方向：

- monitor capture <EPC name> control-plane both
- monitor capture <EPC name> access-list <ACL name>
- monitor capture <EPC name> buffer size 5 circular

配置后，启动EPC:

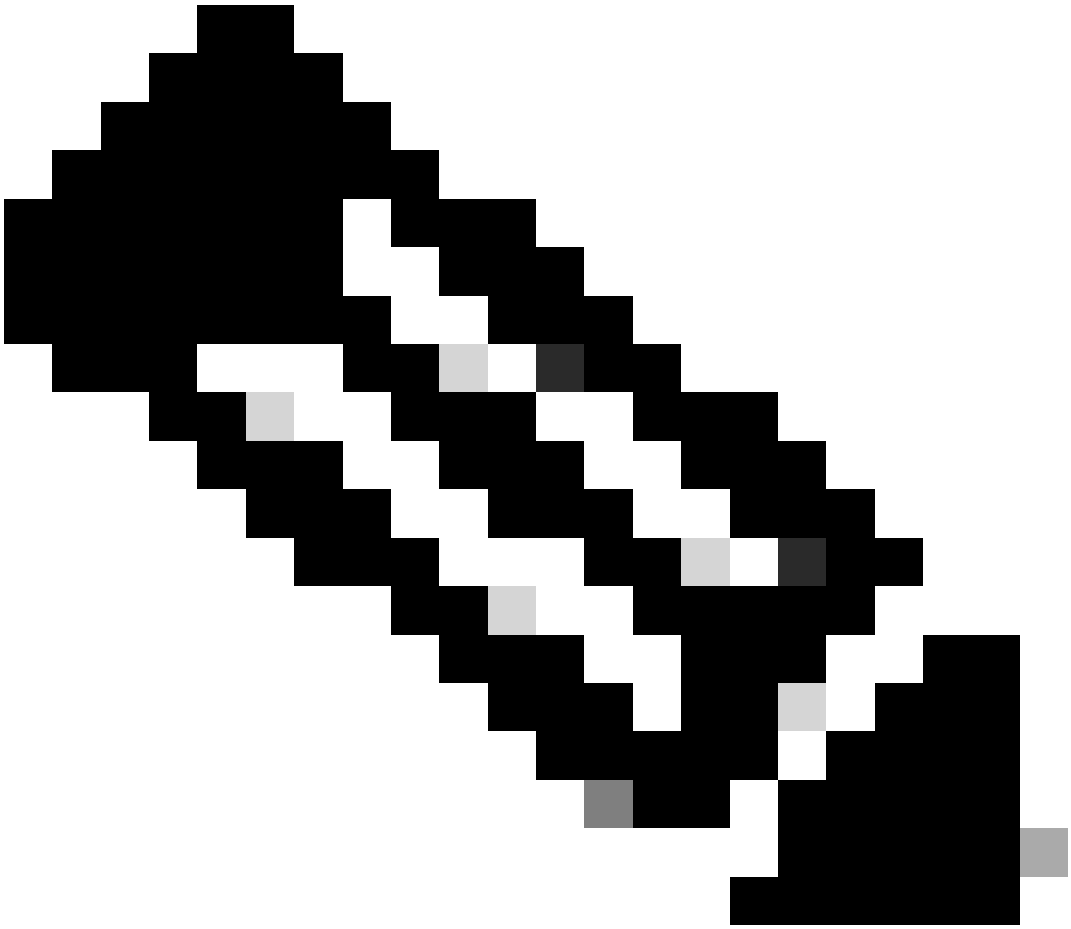
- monitor capture <EPC name> start

EEM设置为在抖动发生时停止捕获。

要确保两个方向都捕获数据包，请检查捕获缓冲区：

```
show monitor capture
```

```
buffer brief
```



注意：Catalyst交换平台（如Cat9k和Cat3k）要求先停止捕获，然后才能查看缓冲区。要确认捕获是否有效，请使用monitor capture stop命令停止捕获，查看缓冲区，然后再次启动以收集数据。

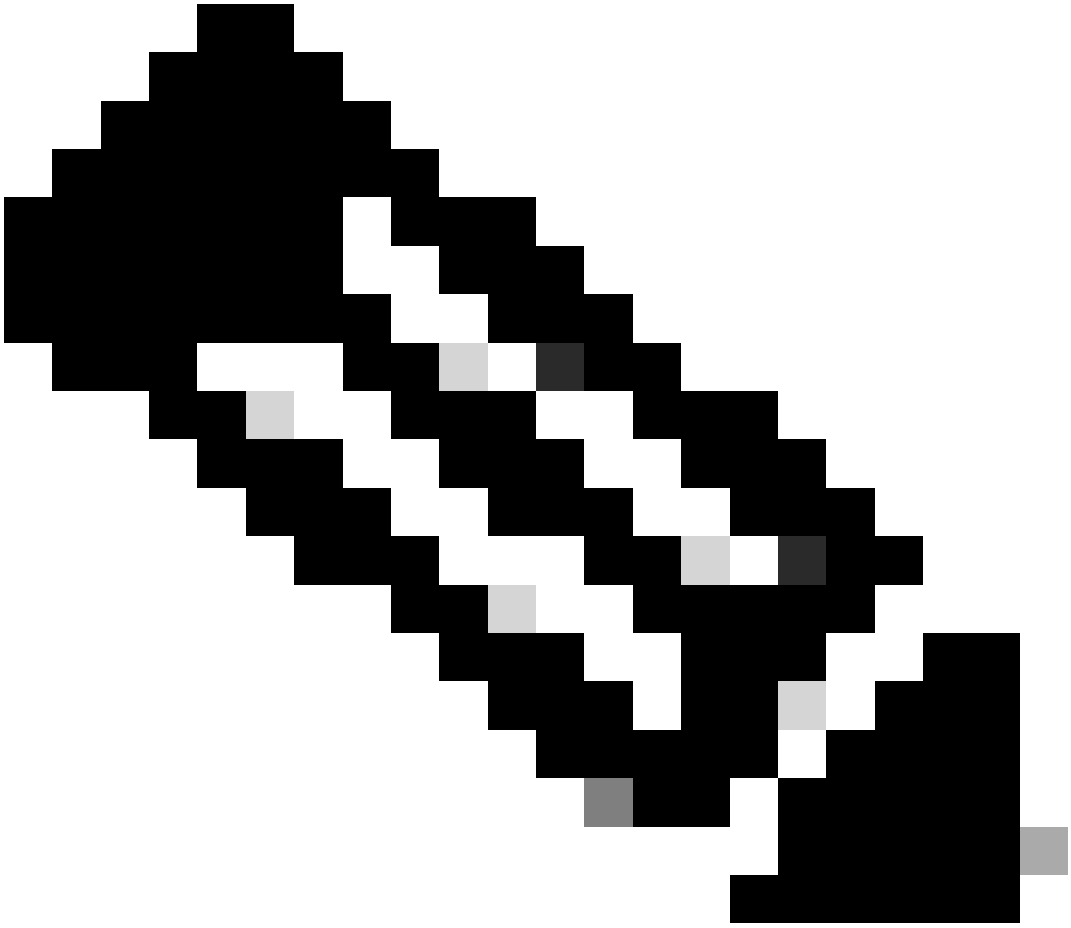
## EEM配置模板

EEM的主要用途是停止数据包捕获，并将其与系统日志缓冲区一起保存。还可以使用其他命令来检查其他因素，例如CPU、接口丢弃或平台特定的资源利用率和丢弃计数器。在配置模式下创建

EEM小程序：

```
config t
event manager applet

      authorization bypass event syslog pattern "

       " maxrun 120 ratelimit 100000 action 000 cli command "enable" action 005 cli command "show clock

        .txt" action 010 cli command "show logging | append bootflash:

          .txt" action 015 cli command "show process cpu sorted | append bootflash:

            .txt" action 020 cli command "show process cpu history | append bootflash:

              .txt" action 025 cli command "show interfaces | append bootflash:

                .txt" action 030 cli command "monitor capture

                 stop" action 035 cli command "monitor capture

                   export bootflash:
```

```
.pcap" action 040 syslog msg "Saved logs to bootflash:


 .txt and saved packet capture to bootflash:


  .pcap" action 045 cli command "end" end
```



注意：在Catalyst交换平台（例如Cat9k和Cat3k）上，导出捕获的命令略有不同。对于这些平台，请修改操作035中使用的CLI命令：

```
action 035 cli command "monitor capture

      export location bootflash:

       .pcap"
```

EEM中的速率限制值以秒为单位，表示必须经过多长时间才能再次运行EEM。在本例中，它被设置为100000秒（27.8小时），以便网络管理员有足够的时间确定它已完成，并在文件再次运行之前从设备中取出文件。如果EEM在此速率限制期后再次自行运行，则不会收集新的数据包捕获数据，因为EPC必须手动启动。但是，新的show命令输出会附加到文本文件中。

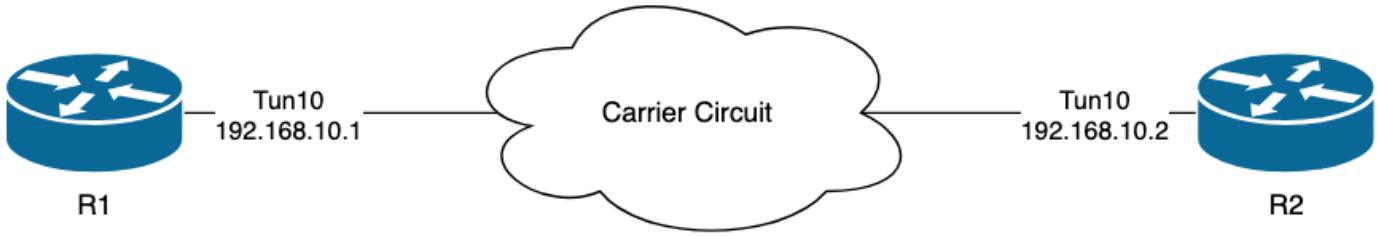
可以根据需要修改EEM以收集平台特定的数据包丢弃信息，并实现您的场景所需的其他功能。

# 排除间歇性路由协议抖动故障

## 示例 — EIGRP

在本示例中，所有计时器均设置为默认值（5秒hello、15秒保持时间）。

拓扑




R1上的日志表明存在间歇性的EIGRP摆动，这些摆动相隔数小时：

```
R1#show logging | i EIGRP
*Jul 16 20:45:08.019: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 192.168.10.2 (Tunnel10) is down: Interf
*Jul 16 20:45:12.919: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 192.168.10.2 (Tunnel10) is up: new adja
*Jul 17 10:25:42.970: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 192.168.10.2 (Tunnel10) is down: holdin
*Jul 17 10:25:59.488: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 192.168.10.2 (Tunnel10) is up: new adja
*Jul 17 14:39:02.970: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 192.168.10.2 (Tunnel10) is down: holdin
*Jul 17 14:39:16.488: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 192.168.10.2 (Tunnel10) is up: new adja
```

数据包丢失可能发生在两个方向上；保持时间到期表示此设备未在保持时间内收到或处理来自对等体的hello，而接口对等体终止表示对等体已终止邻接关系，因为它没有在保持时间内收到或处理hello。

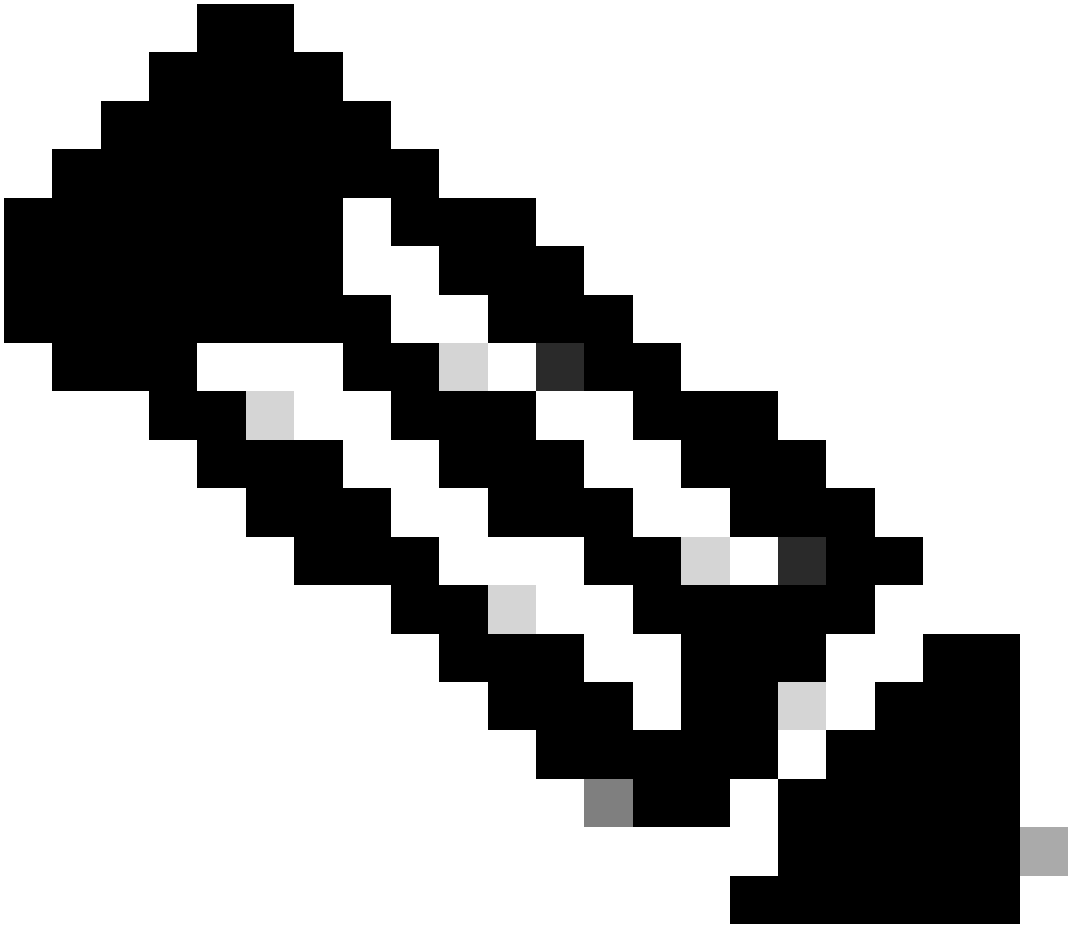
配置

1.使用隧道接口IP地址配置ACL，因为这些地址是hello的源IP地址：

```
R1#conf t
```

```
R1(config)#ip access-list extended FLAP_CAPTURE
R1(config-ext-nacl)#permit ip host 192.168.10.1 any
R1(config-ext-nacl)#permit ip host 192.168.10.2 any
R1(config-ext-nacl)#end
```

---



注意：显示的配置来自R1。R2上的相关接口和EEM的修改文件名也会如此。如果需要其他特殊性，请将EIGRP组播地址224.0.0.10配置为目标IP地址以捕获hello数据包。

---

2.创建EPC并将其与接口和ACL相关联：

```
R1#monitor capture CAP interface Tunnel10 both
R1#monitor capture CAP access-list FLAP_CAPTURE
R1#monitor capture CAP buffer size 5 circular
```

3.启动EPC并确认数据包在两个方向上都捕获：

```
R1#monitor capture CAP start
R1#show monitor capture CAP buffer brief
 ----------------------------------------------------------------------------
 #   size   timestamp     source             destination      dscp    protocol
 ----------------------------------------------------------------------------
   0   74    0.000000   192.168.10.1     ->  224.0.0.10       48 CS6  EIGRP
   1   74    0.228000   192.168.10.2     ->  224.0.0.10       48 CS6  EIGRP
   2   74    4.480978   192.168.10.2     ->  224.0.0.10       48 CS6  EIGRP
   3   74    4.706024   192.168.10.1     ->  224.0.0.10       48 CS6  EIGRP
```

4.配置EEM:

```
R1#conf t
R1(config)#event manager applet R1_EIGRP_FLAP authorization bypass
R1(config-applet)#event syslog pattern "%DUAL-5-NBRCHANGE" maxrun 120 ratelimit 100000
R1(config-applet)#action 000 cli command "enable"
R1(config-applet)#action 005 cli command "show clock | append bootflash:R1_EIGRP_FLAP.txt"
R1(config-applet)#action 010 cli command "show logging | append bootflash:R1_EIGRP_FLAP.txt"
R1(config-applet)#action 015 cli command "show process cpu sorted | append bootflash:R1_EIGRP_FLAP.txt"
R1(config-applet)#action 020 cli command "show process cpu history | append bootflash:R1_EIGRP_FLAP.txt
R1(config-applet)#action 025 cli command "show interfaces | append bootflash:R1_EIGRP_FLAP.txt"
R1(config-applet)#action 030 cli command "monitor capture CAP stop"
R1(config-applet)#action 035 cli command "monitor capture CAP export bootflash:R1_EIGRP_CAP.pcap"
R1(config-applet)#action 040 syslog msg "Saved logs to bootflash:R1_EIGRP_FLAP.txt and saved packet cap
R1(config-applet)#action 045 cli command "end"
R1(config-applet)#end
```

5.等待下一个抖动出现，然后通过首选传输方法从bootflash复制文件进行分析：

```
R1#show logging

      *Jul 17 16:51:47.154: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 192.168.10.2 (Tunnel10) is down:
```

- 路由器上的日志缓冲区表示存在EIGRP抖动，并且文件已由EEM保存。

分析

此时，将日志缓冲区中找到的抖动的时间与收集的数据包捕获相关联，以确定抖动发生时，两端是

否发送和接收了hello数据包。由于在R1上看到收到的Interface PEER-TERMINATION，这意味着R2必须检测到丢失的Hello，因此保持时间已过期，这是从日志文件中看到的情况：

```
*Jul 17 16:51:47.156: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 192.168.10.1 (Tunnel10) is down: holdin
*Jul 17 16:51:51.870: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 192.168.10.1 (Tunnel10) is up: new adja
```

由于R2检测到保持时间已过期，请确认R1在捕获中捕获的抖动之前15秒内是否发送了Hello消息：

| No. | Time | Source | Destination | Protocol | Length | Info | Peer Termination |
|---|---|---|---|---|---|---|---|
| 503 | 2024-07-17 16:51:32.150713 | 192.168.10.1 | 224.0.0.10 | EIGRP | 98 | Hello | |
| 504 | 2024-07-17 16:51:34.293604 | 192.168.10.2 | 224.0.0.10 | EIGRP | 98 | Hello | |
| 505 | 2024-07-17 16:51:36.802191 | 192.168.10.1 | 224.0.0.10 | EIGRP | 98 | Hello | |
| 507 | 2024-07-17 16:51:38.571024 | 192.168.10.2 | 224.0.0.10 | EIGRP | 98 | Hello | |
| 508 | 2024-07-17 16:51:41.456619 | 192.168.10.1 | 224.0.0.10 | EIGRP | 98 | Hello | |
| 510 | 2024-07-17 16:51:43.004216 | 192.168.10.2 | 224.0.0.10 | EIGRP | 98 | Hello | |
| 511 | 2024-07-17 16:51:46.457320 | 192.168.10.1 | 224.0.0.10 | EIGRP | 98 | Hello | |
| 513 | 2024-07-17 16:51:47.154111 | 192.168.10.2 | 224.0.0.10 | EIGRP | 98 | Hello | ✓ |

- 捕获显示192.168.10.1(R1)和192.168.10.2(R2)在R2在16:51:47（数据包513）发送的PEER-TERMINATION hello数据包之前的15秒内hello消息。
- 具体而言，数据包503、505、508和511（用绿色箭头表示）都是由R1在此时间段发送的Hello数据包。

下一步是确认R1发送的所有问候消息在R2之前是否都已收到，因此必须检查从R2收集的捕获：

| No. | Time | Source | Destination | Protocol | Length | Info | Peer Termination |
|---|---|---|---|---|---|---|---|
| 498 | 2024-07-17 16:51:32.154320 | 192.168.10.1 | 224.0.0.10 | EIGRP | 98 | Hello | |
| 499 | 2024-07-17 16:51:34.296179 | 192.168.10.2 | 224.0.0.10 | EIGRP | 98 | Hello | |
| 500 | 2024-07-17 16:51:38.573467 | 192.168.10.2 | 224.0.0.10 | EIGRP | 98 | Hello | |
| 501 | 2024-07-17 16:51:43.006794 | 192.168.10.2 | 224.0.0.10 | EIGRP | 98 | Hello | |
| 502 | 2024-07-17 16:51:47.156716 | 192.168.10.2 | 224.0.0.10 | EIGRP | 98 | Hello | ✓ |

```
> Internet Protocol Version 4, Src: 192.168.10.2, Dst: 224.0.0.10
v Cisco EIGRP
    Version: 2
    Opcode: Hello (5)
    Checksum: 0xdfd1 [correct]
    [Checksum Status: Good]
  > Flags: 0x00000000
    Sequence: 0
    Acknowledge: 0
    Virtual Router ID: 0 (Address-Family)
    Autonomous System: 1
  v Parameters: Peer Termination
```

- 捕获显示从192.168.10.1(R1)收到的最后一个hello时间为16:51:32（以绿色箭头表示）。 此后，接下来的15秒仅显示R2发送的hello数据包（以红色框表示）。 来自R1的捕获中的数据包505、508和511不会出现在R2的捕获中。这会导致R2检测保持计时器超时，并在16:51:47发送PEER-TERMINATION hello数据包（数据包502）。

从这些数据可以得出结论，丢包发生在R1和R2之间的运营商网络中。在这种情况下，丢包发生在R1到R2的方向。为了进一步调查，需要让运营商检查路径是否存在丢包。

## OSPF

同样的逻辑也可用于排除间歇性OSPF抖动故障。本部分介绍这些关键因素，这些因素在计时器、

IP地址过滤器和日志消息方面与其他路由协议不同。

- 默认计时器为10秒hello和40秒dead计时器。在对死计时器过期抖动进行故障排除时，请始终确认网络中正在使用的计时器。
- Hello数据包源自接口IP地址。如果需要其他ACL特异性，则OSPF Hello的组播目的地址为224.0.0.5。
- 设备上的日志消息略有不同。与EIGRP相反，OSPF没有对等终止消息的概念。相反，检测已过期dead计时器的设备将此记录为抖动原因，然后其发送的Hello不再包含对等体的路由器ID，这将导致对等体进入INIT状态。当再次检测到Hello时，邻接关系会转换到全状态。例如：

R1检测到失效计时器已过期：

```
R1#show logging | i OSPF
*Jul 30 15:29:14.027: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.30.2 on Tunnel20 from FULL to DOWN, Neighbo
*Jul 30 15:32:30.278: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.30.2 on Tunnel20 from LOADING to FULL, Loa
*Jul 30 16:33:19.841: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.30.2 on Tunnel20 from FULL to DOWN, Neighbo
*Jul 30 16:48:10.504: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.30.2 on Tunnel20 from LOADING to FULL, Loa
```

但是，R2只在OSPF恢复为FULL时显示日志消息。当状态更改为INIT时，不会显示日志消息：

```
R2#show logging | i OSPF
*Jul 30 16:32:30.279: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.30.1 on Tunnel20 from LOADING to FULL, Loa
*Jul 30 16:48:10.506: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.30.1 on Tunnel20 from LOADING to FULL, Loa
```

要在两台设备上触发EEM，请使用“%OSPF-5-ADJCHG”作为系统日志模式。这可以确保EEM在关闭并恢复运行期间在两个设备上触发。配置的速率限制值可确保当看到带有此字符串的多个日志时，它不会在短时间内触发两次。关键在于确认两端的数据包捕获中是否发送和接收hello数据包。

## 调试输出中显示“BGP

相同的逻辑可用于排除间歇性BGP摆动故障。本部分介绍这些关键因素，这些因素在计时器、IP地址过滤器和日志消息方面与其他路由协议不同。

- 默认计时器为60秒keepalive和180秒保持时间。在对保持时间过期的抖动进行故障排除时，请始终确认网络中正在使用的计时器。
- 保持连接数据包在邻居IP地址之间单播发送到TCP目标端口179。如果需要其他ACL特殊性，请允许从源IP地址到目标TCP端口179的TCP流量。
- 两台设备上的BGP日志消息看起来相似，但检测到保持时间过期的设备显示它向邻居发送了通知，而另一台则表示它收到了通知消息。例如：

R1检测到保持时间已过，并将通知发送到R2:

```
R1#show logging | i BGP
*Jul 30 17:49:23.730: %BGP-3-NOTIFICATION: sent to neighbor 192.168.30.2 4/0 (hold time expired) 0 byte
*Jul 30 17:49:23.731: %BGP-5-NBR_RESET: Neighbor 192.168.30.2 reset (BGP Notification sent)
*Jul 30 17:49:23.732: %BGP-5-ADJCHANGE: neighbor 192.168.30.2 Down BGP Notification sent
*Jul 30 17:49:23.732: %BGP_SESSION-5-ADJCHANGE: neighbor 192.168.30.2 IPv4 Unicast topology base remove
```

R2收到来自R1的通知，因为R1检测到保持时间已过期：

```
R2#show logging | i BGP
*Jul 30 17:49:23.741: %BGP-3-NOTIFICATION: received from neighbor 192.168.30.1 4/0 (hold time expired)
*Jul 30 17:49:23.741: %BGP-5-NBR_RESET: Neighbor 192.168.30.1 reset (BGP Notification received)
*Jul 30 17:49:23.749: %BGP-5-ADJCHANGE: neighbor 192.168.30.1 Down BGP Notification received
*Jul 30 17:49:23.749: %BGP_SESSION-5-ADJCHANGE: neighbor 192.168.30.1 IPv4 Unicast topology base remove
```

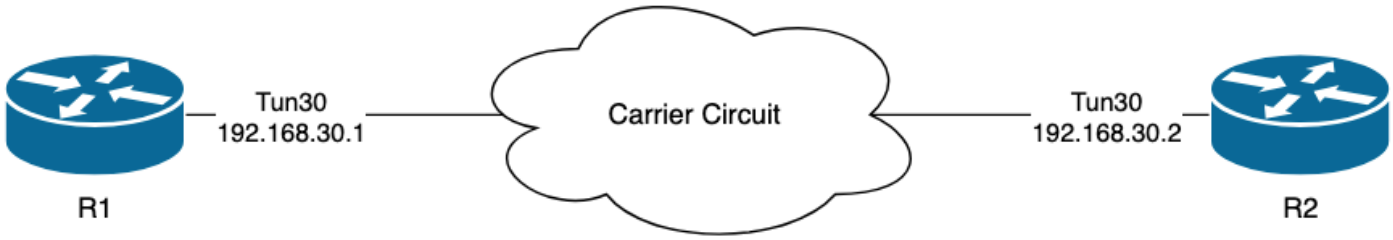
要触发BGP抖动的EEM，请使用“%BGP_SESSION-5-ADJCHANGE”作为系统日志模式。在抖动后还记录的任何其他“%BGP”系统日志消息也可用于触发EEM。

# 排除间歇性BFD抖动故障

同样的方法也可用于对间歇性BFD摆动进行故障排除，但有些细微的差异可用于分析。本节介绍一些基本的BFD功能，并举例说明如何使用EEM和EPC进行故障排除。有关BFD故障排除的详细信息，请参阅排除Cisco IOS XE中的双向转发检测。

在本示例中，BFD计时器设置为300ms，乘数为3，这意味着每300ms发送一次回声，并且当一行中有3个回声数据包未返回时（等于900ms保持时间）检测到回声故障。

## 拓扑




## 示例 — BFD回声模式

在BFD回应模式（默认模式）下，发送BFD回应数据包时会将本地接口IP作为源和目标。这样，邻居可以在数据平面中处理数据包，并将其返回给源设备。每个BFD回声与BFD回声消息报头中的回声ID一起发送。这些参数可用于确定发送的BFD回应数据包是否被接收回，因为如果任何给定的BFD回应数据包确实被邻居返回，则肯定有两次出现该数据包。用于控制BFD会话状态的BFD控制数据包在接口IP地址之间单播发送。

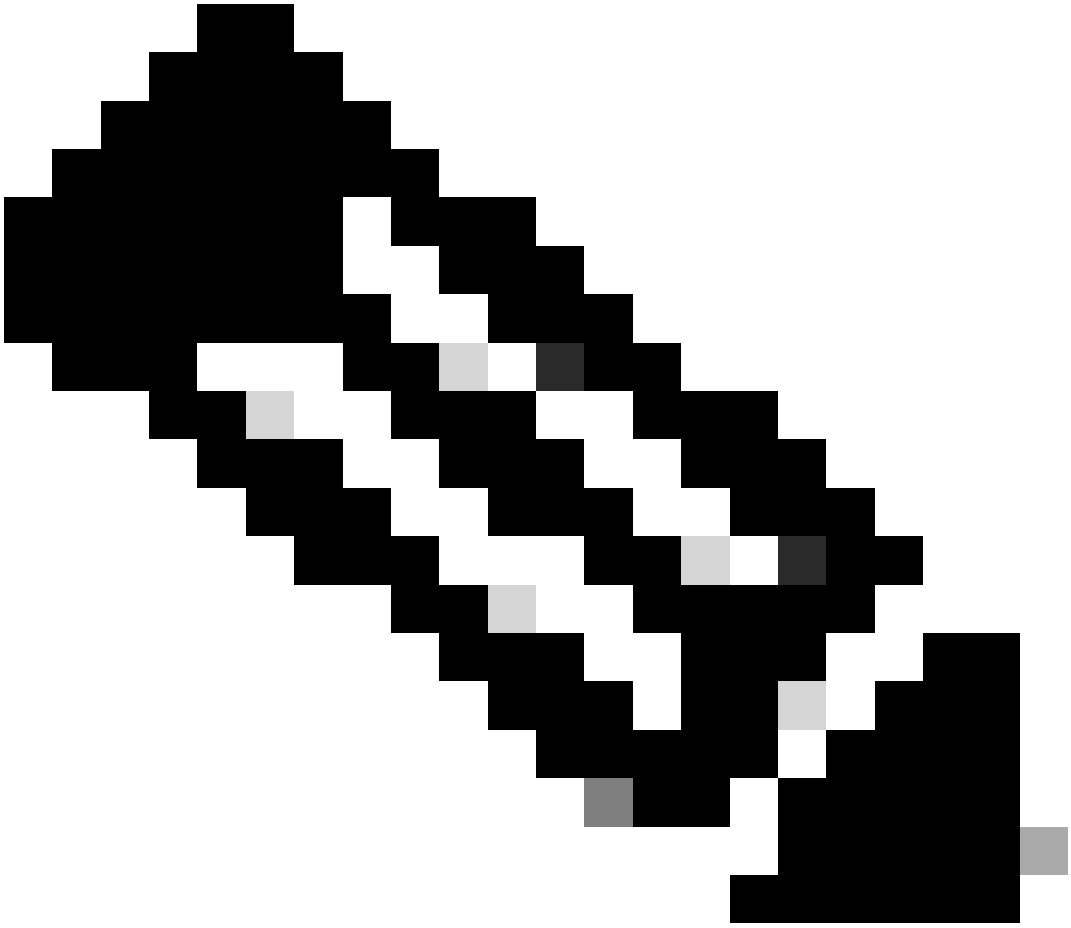
来自R1的日志表明BFD邻接因回声故障而多次中断，这意味着在这些间隔期间，R1没有收到或处理来自R2的3个回声数据包。

```
R1#show logging | i BFD
*Jul 18 13:41:09.007: %BFDFSM-6-BFD_SESS_DOWN: BFD-SYSLOG: BFD session ld:4097 handle:1,is going Down R
*Jul 18 13:41:09.009: %BGP-5-NBR_RESET: Neighbor 192.168.30.2 reset (BFD adjacency down)
*Jul 18 13:41:09.010: %BGP-5-ADJCHANGE: neighbor 192.168.30.2 Down BFD adjacency down
*Jul 18 13:41:09.010: %BGP_SESSION-5-ADJCHANGE: neighbor 192.168.30.2 IPv4 Unicast topology base remove
*Jul 18 13:41:09.010: %BFD-6-BFD_SESS_DESTROYED: BFD-SYSLOG: bfd_session_destroyed,  ld:4097 neigh proc
*Jul 18 13:41:13.335: %BFDFSM-6-BFD_SESS_UP: BFD-SYSLOG: BFD session ld:4097 handle:1 is going UP
*Jul 18 13:41:18.576: %BFD-6-BFD_SESS_CREATED: BFD-SYSLOG: bfd_session_created, neigh 192.168.30.2 proc
*Jul 18 13:41:19.351: %BFDFSM-6-BFD_SESS_UP: BFD-SYSLOG: BFD session ld:4097 handle:1 is going UP
*Jul 18 15:44:08.360: %BFDFSM-6-BFD_SESS_DOWN: BFD-SYSLOG: BFD session ld:4097 handle:1,is going Down R
*Jul 18 15:44:08.362: %BGP-5-NBR_RESET: Neighbor 192.168.30.2 reset (BFD adjacency down)
*Jul 18 15:44:08.363: %BGP-5-ADJCHANGE: neighbor 192.168.30.2 Down BFD adjacency down
*Jul 18 15:44:08.363: %BGP_SESSION-5-ADJCHANGE: neighbor 192.168.30.2 IPv4 Unicast topology base remove
*Jul 18 15:44:08.363: %BFD-6-BFD_SESS_DESTROYED: BFD-SYSLOG: bfd_session_destroyed,  ld:4097 neigh proc
*Jul 18 15:44:14.416: %BFDFSM-6-BFD_SESS_UP: BFD-SYSLOG: BFD session ld:4097 handle:1 is going UP
*Jul 18 15:44:14.418: %BFD-6-BFD_SESS_DESTROYED: BFD-SYSLOG: bfd_session_destroyed,  ld:4097 neigh proc
*Jul 18 15:44:18.315: %BFD-6-BFD_SESS_CREATED: BFD-SYSLOG: bfd_session_created, neigh 192.168.30.2 proc
```

## 配置

1.使用隧道接口IP地址配置ACL，因为这些地址是BFD回应数据包和控制数据包的源IP地址：

```
R1#conf t
R1(config)#ip access-list extended FLAP_CAPTURE
R1(config-ext-nacl)#permit ip host 192.168.30.1 any
R1(config-ext-nacl)#permit ip host 192.168.30.2 any
```



注意：显示的配置来自R1。R2上的相关接口和EEM的修改文件名也会如此。如果需要其他特殊性，请使用目标端口3785（回应数据包）和3784（控制数据包）配置UDP的ACL。

---

2.创建EPC并将其与接口和ACL相关联：

```
R1#monitor capture CAP interface Tunnel30 both
R1#monitor capture CAP access-list FLAP_CAPTURE
R1#monitor capture CAP buffer size 5 circular
```

3.启动EPC并确认数据包在两个方向上都捕获：

```
R1#monitor capture CAP start
R1#show monitor capture CAP buff brief
 ----------------------------------------------------------------------------
 #   size   timestamp     source             destination      dscp    protocol
```

```
------------------------------------------------------------------------------
  0   54    0.000000   192.168.30.2     ->  192.168.30.2     48 CS6  UDP
  1   54    0.000000   192.168.30.2     ->  192.168.30.2     48 CS6  UDP
  2   54    0.005005   192.168.30.1     ->  192.168.30.1     48 CS6  UDP
  3   54    0.005997   192.168.30.1     ->  192.168.30.1     48 CS6  UDP
```

4.配置EEM:

```
R1#conf t
R1(config)#event manager applet R1_BFD_FLAP authorization bypass
R1(config-applet)#event syslog pattern "%BFDFSM-6-BFD_SESS_DOWN" maxrun 120 ratelimit 100000
R1(config-applet)#action 000 cli command "enable"
R1(config-applet)#action 005 cli command "show clock | append bootflash:R1_BFD_FLAP.txt"
R1(config-applet)#action 010 cli command "show logging | append bootflash:R1_BFD_FLAP.txt"
R1(config-applet)#action 015 cli command "show process cpu sorted | append bootflash:R1_BFD_FLAP.txt"
R1(config-applet)#action 020 cli command "show process cpu history | append bootflash:R1_BFD_FLAP.txt"
R1(config-applet)#action 025 cli command "show interfaces | append bootflash:R1_BFD_FLAP.txt"
R1(config-applet)#action 030 cli command "monitor capture CAP stop"
R1(config-applet)#action 035 cli command "monitor capture CAP export bootflash:R1_BFD_CAP.pcap"
R1(config-applet)#action 040 syslog msg "Saved logs to bootflash:R1_BFD_FLAP.txt and saved packet captu
R1(config-applet)#action 045 cli command "end"
R1(config-applet)#end
```

5.等待下一个抖动出现，然后通过首选传输方法从bootflash复制文件进行分析：

```
R1#show logging
```

```
      *Jul 18 19:09:47.482: %BFDFSM-6-BFD_SESS_DOWN: BFD-SYSLOG: BFD session ld:4097 handle:1,is going
```

- 日志缓冲区表示在19:09:47出现BFD抖动，并且文件已由EEM保存。

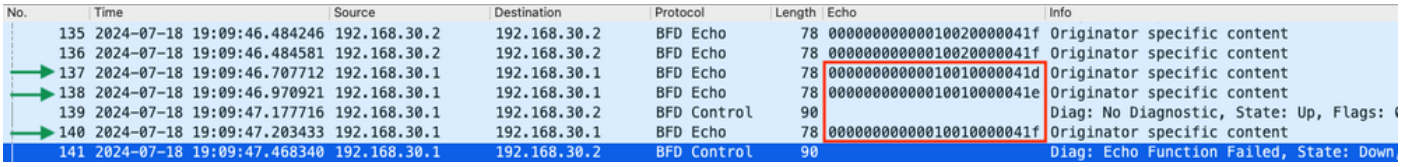
分析

此时，将日志缓冲区中发现的抖动的时间与收集的数据包捕获相关联，以便确定问题发生时两端是否发送和接收BFD回声。由于R1上的抖动原因为ECHO FAILURE，这意味着它也会向R2发送控制数据包以终止BFD会话，这反映在R2收集的日志文件中，其中显示BFD关闭原因RX DOWN:

```
*Jul 18 19:09:47.468: %BFDFSM-6-BFD_SESS_DOWN: BFD-SYSLOG: BFD session ld:4098 handle:2,is going Down R
```

```
*Jul 18 19:09:47.470: %BGP-5-NBR_RESET: Neighbor 192.168.30.1 reset (BFD adjacency down)
*Jul 18 19:09:47.471: %BGP-5-ADJCHANGE: neighbor 192.168.30.1 Down BFD adjacency down
*Jul 18 19:09:47.471: %BGP_SESSION-5-ADJCHANGE: neighbor 192.168.30.1 IPv4 Unicast topology base remove
*Jul 18 19:09:47.471: %BFD-6-BFD_SESS_DESTROYED: BFD-SYSLOG: bfd_session_destroyed,  ld:4098 neigh proc
```

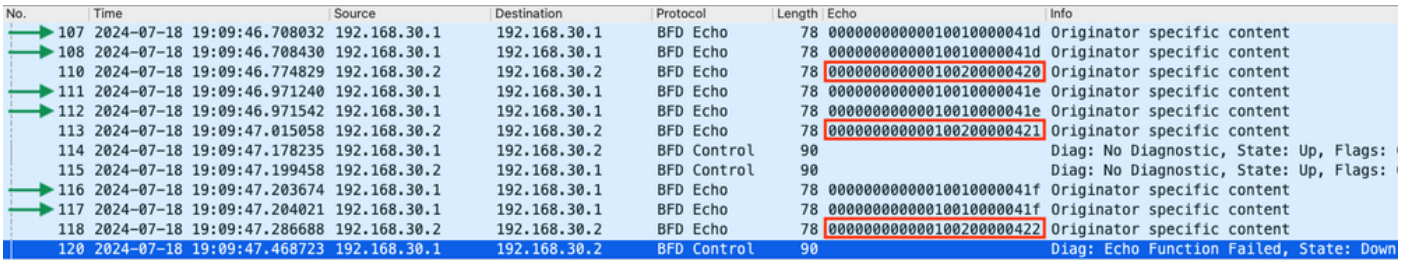
由于R1检测到ECHO FAILURE，因此请检查R1上收集的数据包捕获情况，看它是否在抖动之前的900ms内发送和接收BFD应答。



| No. | Time | Source | Destination | Protocol | Length | Echo | Info |
|---|---|---|---|---|---|---|---|
| 135 | 2024-07-18 19:09:46.484246 | 192.168.30.2 | 192.168.30.2 | BFD Echo | 78 | 0000000000000100200000041f | Originator specific content |
| 136 | 2024-07-18 19:09:46.484581 | 192.168.30.2 | 192.168.30.2 | BFD Echo | 78 | 0000000000000100200000041f | Originator specific content |
| 137 | 2024-07-18 19:09:46.707712 | 192.168.30.1 | 192.168.30.1 | BFD Echo | 78 | 0000000000000100100000041d | Originator specific content |
| 138 | 2024-07-18 19:09:46.970921 | 192.168.30.1 | 192.168.30.1 | BFD Echo | 78 | 0000000000000100100000041e | Originator specific content |
| 139 | 2024-07-18 19:09:47.177716 | 192.168.30.1 | 192.168.30.2 | BFD Control | 90 | | Diag: No Diagnostic, State: Up, Flags: |
| 140 | 2024-07-18 19:09:47.203433 | 192.168.30.1 | 192.168.30.1 | BFD Echo | 78 | 0000000000000100100000041f | Originator specific content |
| 141 | 2024-07-18 19:09:47.468340 | 192.168.30.1 | 192.168.30.2 | BFD Control | 90 | | Diag: Echo Function Failed, State: Down |

- 捕获表明R1一直主动发送BFD回应数据包，一直到抖动发生时，但R2未返回这些数据包，因此R1发送控制数据包以在19:09:47.468终止会话。
- 这一点从以下事实中明显可见：数据包137、138和140（用绿色箭头表示）在捕获中仅出现一次，这可以从BFD回声ID（在红色框中）确定。 如果已返回回应，则具有相同BFD回应ID的每个数据包都有第二个副本。IP报头中的IP Identification（IP标识）字段（此处未图片）也可用于验证这一点。
- 此捕获还显示，在数据包136之后，没有从R2收到BFD响应，这是数据包在R2到R1的方向上丢失的另一个迹象。

下一步是确认R1发送的所有BFD回应数据包是否都已接收并由R2返回，因此必须检查从R2收集的捕获：



| No. | Time | Source | Destination | Protocol | Length | Echo | Info |
|---|---|---|---|---|---|---|---|
| 107 | 2024-07-18 19:09:46.708032 | 192.168.30.1 | 192.168.30.1 | BFD Echo | 78 | 0000000000000100100000041d | Originator specific content |
| 108 | 2024-07-18 19:09:46.708430 | 192.168.30.1 | 192.168.30.1 | BFD Echo | 78 | 0000000000000100100000041d | Originator specific content |
| 110 | 2024-07-18 19:09:46.774829 | 192.168.30.2 | 192.168.30.2 | BFD Echo | 78 | 0000000000000100200000420 | Originator specific content |
| 111 | 2024-07-18 19:09:46.971240 | 192.168.30.1 | 192.168.30.1 | BFD Echo | 78 | 0000000000000100100000041e | Originator specific content |
| 112 | 2024-07-18 19:09:46.971542 | 192.168.30.1 | 192.168.30.1 | BFD Echo | 78 | 0000000000000100100000041e | Originator specific content |
| 113 | 2024-07-18 19:09:47.015058 | 192.168.30.2 | 192.168.30.2 | BFD Echo | 78 | 0000000000000100200000421 | Originator specific content |
| 114 | 2024-07-18 19:09:47.178235 | 192.168.30.1 | 192.168.30.2 | BFD Control | 90 | | Diag: No Diagnostic, State: Up, Flags: |
| 115 | 2024-07-18 19:09:47.199458 | 192.168.30.2 | 192.168.30.1 | BFD Control | 90 | | Diag: No Diagnostic, State: Up, Flags: |
| 116 | 2024-07-18 19:09:47.203674 | 192.168.30.1 | 192.168.30.1 | BFD Echo | 78 | 0000000000000100100000041f | Originator specific content |
| 117 | 2024-07-18 19:09:47.204021 | 192.168.30.1 | 192.168.30.1 | BFD Echo | 78 | 0000000000000100100000041f | Originator specific content |
| 118 | 2024-07-18 19:09:47.286688 | 192.168.30.2 | 192.168.30.2 | BFD Echo | 78 | 0000000000000100200000422 | Originator specific content |
| 120 | 2024-07-18 19:09:47.468723 | 192.168.30.1 | 192.168.30.2 | BFD Control | 90 | | Diag: Echo Function Failed, State: Down |

- 此捕获显示R1发送的所有BFD响应均由R2接收并返回（标有绿色箭头）；数据包107和108是相同的BFD回应，数据包111和112是相同的BFD回应，数据包116和117是相同的BFD回应。
- 此捕获还显示R2主动发送了R1上捕获不到的回应数据包（以红色方框表示），进一步表明从R2到R1的方向上设备之间的数据包丢失。

从这些数据可以得出结论，丢包发生在R1和R2之间的运营商网络中，此时的所有证据都表明丢包方向是从R2到R1。为了进一步调查，需要让运营商检查路径是否存在丢包。

## BFD异步模式

当使用BFD异步模式（回声功能被禁用）时，可以使用相同的方法，并且EEM和EPC配置可以保持不变。异步模式的区别在于，设备将单播BFD控制数据包作为keepalive发送到彼此，类似于典型的路由协议邻接。这意味着只发送UDP端口3784数据包。在这种情况下，只要在所需间隔内收到来自邻居的BFD数据包，BFD就会保持运行状态。当这种情况不发生时，故障原因为DETECT TIMER

EXPIRED，路由器会向对等设备发送控制数据包以关闭会话。

要分析检测到故障的设备上的捕获，请查找在抖动之前的时间从对等设备接收的单播BFD数据包。例如，如果TX间隔设置为300ms，乘数为3，则如果在抖动之前的900ms中没有收到BFD数据包，则表示可能存在数据包丢失。在通过EEM从邻居收集的捕获中，选中此同一时间段；如果数据包是在此时间内发送的，则确认设备之间的某个位置存在丢失。

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。