EEM 및 EPC로 간헐적 라우팅 프로토콜 플랩 트러블슈팅

목차

소개

이 문서에서는 Cisco IOS® XE with EEM and EPC에서 간헐적 라우팅 프로토콜 플랩 및 BFD 플랩을 해결하는 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

트러블슈팅과 관련된 플랫폼의 EEM(Embedded Event Manager) 및 EPC(Embedded Packet Capture)와 Wireshark의 세부 사항을 잘 알고 있는 것이 좋습니다. 또한 라우팅 프로토콜의 기본 hello 및 keepalive 기능과 BFD(Bidirectional Forwarding Detection)에 대해 잘 아는 것이 좋습니다.

사용되는 구성 요소

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

문제 개요

간헐적인 라우팅 프로토콜 플랩은 프로덕션 네트워크에서 흔히 발생하는 문제이지만, 예측할 수 없는 특성 때문에 실시간 트러블슈팅이 어려울 수 있습니다. EEM은 플랩 발생 시 syslog 문자열로 데이터 캡처를 트리거하여 데이터 수집을 자동화하는 기능을 제공합니다. EEM 및 EPC를 사용하면 인접 디바이스의 양쪽 끝에서 패킷 캡처 데이터를 수집하여 플랩 시간 이전에 잠재적 패킷 손실을 격리할 수 있습니다.

간헐적 라우팅 프로토콜 플랩의 특성은 항상 hello 또는 keepalive 시간 초과 때문입니다(로그에 나타나는 링크 플랩과 같은 명확한 물리적 문제가 아닌 경우). 따라서 이 문서의 논리는 다음과 같습니다.

문제 해결 방법론

라우팅 프로토콜 플랩이 발생하는 시점을 확인하는 데 가장 중요한 것은 문제 발생 시 hello 패킷 또는 keepalive 패킷이 두 디바이스에서 모두 전송 및 수신되었는지 여부입니다. 이 트러블슈팅 방법은 플랩이 발생할 때까지 순환 버퍼에서 연속 EPC를 사용합니다. 이 시점에서 EEM은 관련 syslog 문자열을 사용하여 실행할 명령 집합을 트리거하며, 이 중 하나는 EPC를 중지합니다. 순환 버퍼 옵션을 사용하면 EPC가 버퍼에서 가장 오래된 패킷을 덮어쓰는 동안 계속해서 새 패킷을 캡처할 수 있으므로, 이벤트가 캡처되고 버퍼가 미리 채워지거나 중지되지 않습니다. 그런 다음 패킷 캡처 데이터를 플랩의 타임스탬프와 상호 연결하여 필요한 패킷이 이벤트 이전에 양쪽 끝에서 송수신되었는지 여부를 확인할 수 있습니다.

이 문제는 ISP(Internet Service Provider)와 같은 중간 네트워크를 통해 인접성을 형성하는 디바이스에서 가장 일반적으로 발생하지만, 특정 토폴로지 세부사항에 관계없이 간헐적인 라우팅 프로토콜 플랩 시나리오에 동일한 방법론을 적용할 수 있습니다. 인접 디바이스가 제3자에 의해 관리되고 액세스할 수 없는 경우에도 마찬가지입니다. 이러한 경우 플랩이 발생하기 전에 필요한 패킷을 보내고 받았는지 여부를 증명하기 위해 액세스 가능한 하나의 장치에만 이 문서에 설명된 트러블슈팅 방법을 적용할 수 있습니다. 이를 확인하면 필요한 경우 다른 쪽에서 추가 문제를 해결하기 위해 네이버를 관리하는 상대방에게 데이터를 표시할 수 있습니다.

구성 개요

이 섹션에서는 이러한 자동화된 데이터 캡처를 설정하는 데 사용할 수 있는 컨피그레이션 템플릿 집합을 제공합니다. 필요에 따라 IP 주소, 인터페이스 이름 및 파일 이름을 수정합니다.

ACL 컨피그레이션 템플릿

대부분의 경우 라우팅 인접성의 양쪽 끝에 있는 인터페이스 IP 주소에서 소싱된 트래픽은 라우팅 제어 트래픽 자체뿐입니다. 따라서 로컬 인터페이스 IP 주소와 인접 IP 주소 모두에서 모든 목적지로

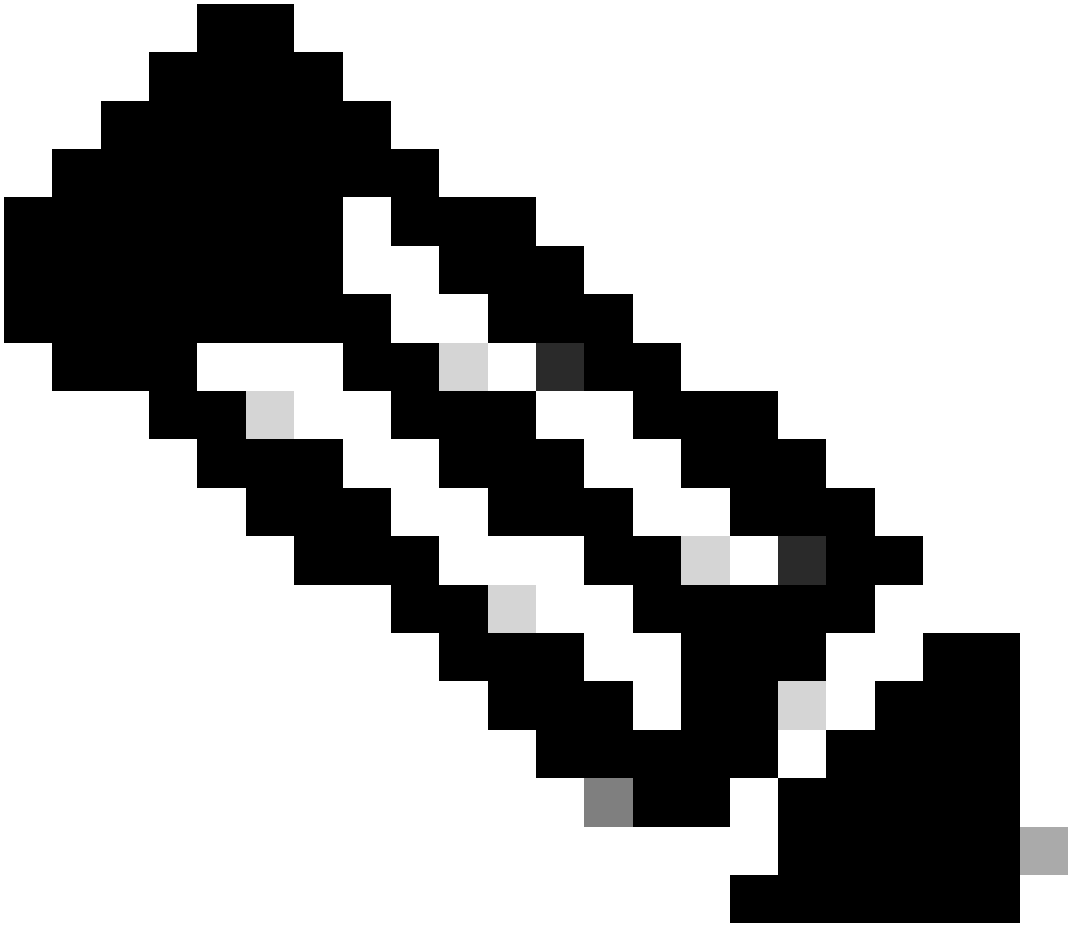
의 트래픽을 허용하는 ACL은 라우팅 프로토콜과 BFD에 대한 요구 사항을 다룹니다. 추가 필터가 필요한 경우 라우팅 프로토콜 또는 BFD 모드 기반의 관련 대상 IP도 지정할 수 있습니다. 컨피그레이션 모드에서 ACL 매개변수를 정의합니다.

```
config t

ip access-list extended




      permit ip host




       any permit ip host




        any end
```

EPC 매개변수 템플릿

EPC 매개변수는 컨피그레이션 모드가 아닌 권한 실행 모드에서 생성됩니다. 플랫폼별 컨피그레이션 가이드를 확인하여 EPC에 제한이 있는지 확인하십시오. 원하는 인터페이스에 대한 매개변수를 생성하고 이를 ACL과 연결하여 원하는 트래픽에 대해 필터링합니다.

- monitor capture <EPC name> interface <interface> 모두
- monitor capture <EPC name> access-list <ACL name>
- monitor capture <EPC name> 버퍼 크기 5 circular



참고: 일부 소프트웨어 버전에서는 로컬에서 생성된 트래픽이 인터페이스 레벨 EPC에 표시되지 않습니다. 이러한 시나리오에서는 CPU에서 트래픽의 양방향을 캡처하도록 캡처 매개변수를 변경할 수 있습니다.

- 두 컨트롤 플레인 모두 모니터링
- monitor capture <EPC name> access-list <ACL name>
- monitor capture <EPC name> 버퍼 크기 5 circular

구성이 완료되면 EPC를 시작합니다.

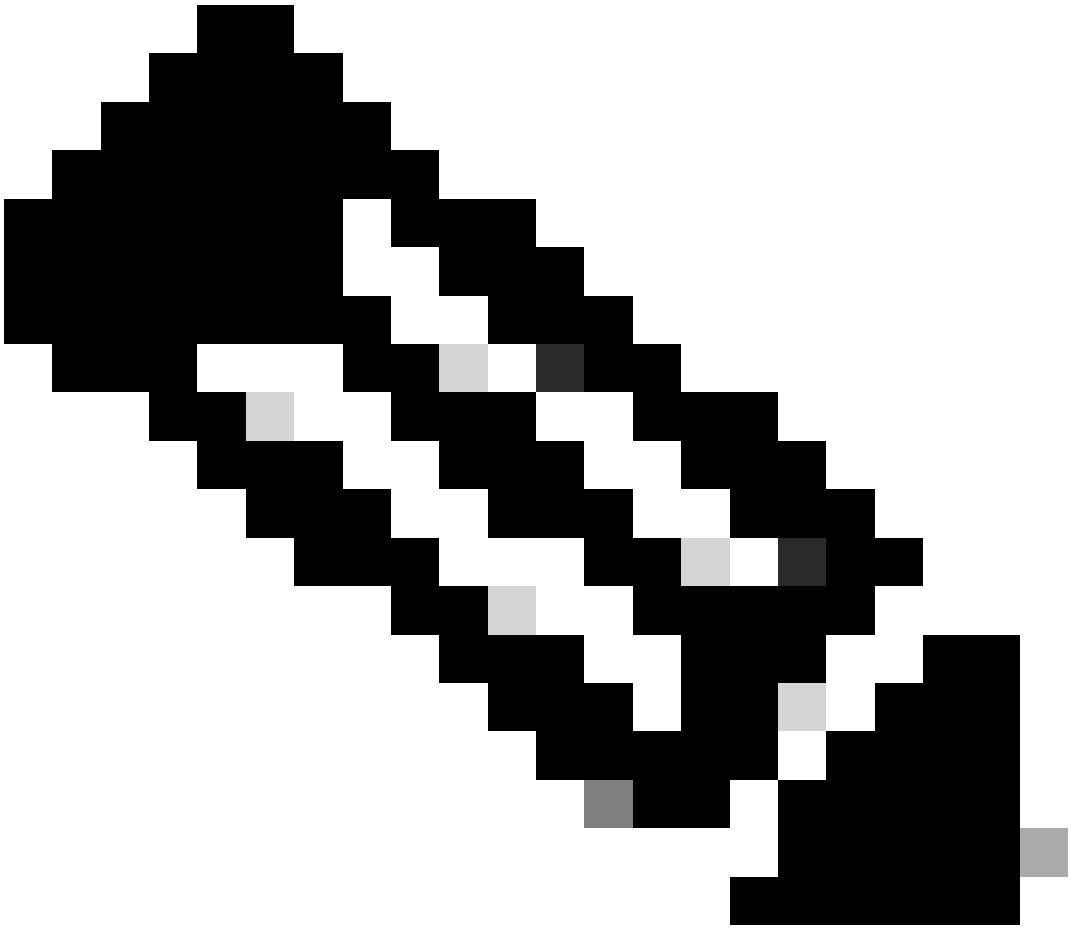
- 모니터링 캡처 <EPC name> 시작

EEM은 플랩이 발생하면 캡처를 중지하도록 설정됩니다.

패킷이 양방향으로 캡처되었는지 확인하려면 캡처 버퍼를 확인합니다.

```
show monitor capture
```

```
buffer brief
```



참고: Catalyst 스위칭 플랫폼(예: Cat9k 및 Cat3k)에서는 버퍼를 보기 전에 캡처를 중지해야 합니다. 캡처가 작동하는지 확인하려면 monitor capture stop 명령으로 캡처를 중지하고 버퍼를 본 다음 다시 시작하여 데이터를 수집합니다.

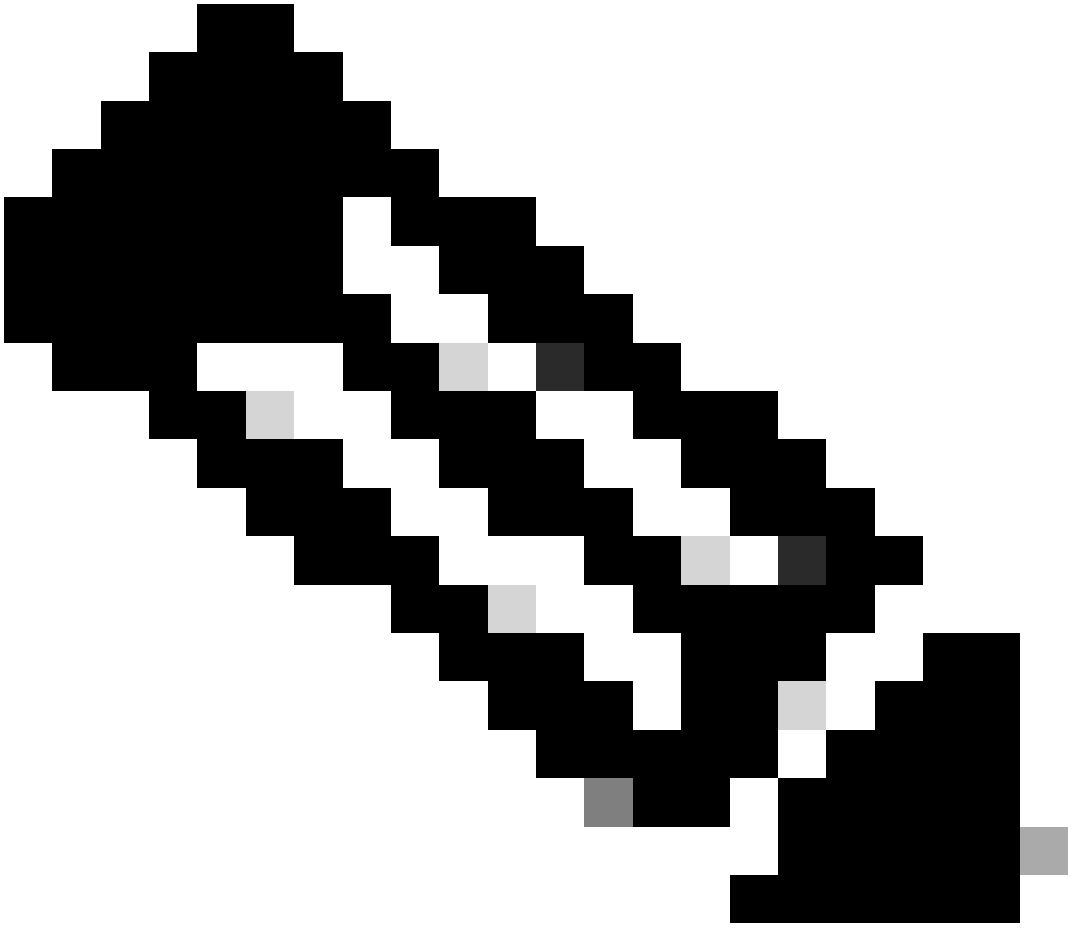
EEM 컨피그레이션 템플릿

EEM의 주요 목적은 패킷 캡처를 중지하고 syslog 버퍼와 함께 저장하는 것입니다. CPU, 인터페이

스 삭제 또는 플랫폼별 리소스 사용률 및 삭제 카운터와 같은 다른 요소를 확인하기 위해 추가 명령을 포함할 수 있습니다. 컨피그레이션 모드에서 EEM 애플릿을 생성합니다.

```
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
```



참고: Catalyst 스위칭 플랫폼(예: Cat9k 및 Cat3k)에서 캡처를 내보내는 명령은 약간 다릅니다. 이러한 플랫폼의 경우 035 단계에서 사용된 CLI 명령을 수정합니다.

```
action 035 cli command "monitor capture

      export location bootflash:

       .pcap"
```

EEM의 속도 제한 값은 초 단위이며 EEM을 다시 실행할 수 있을 때까지 경과해야 하는 시간을 나타냅니다. 이 예에서는 네트워크 관리자가 완료되었음을 확인하고 디바이스에서 파일을 가져온 다음 다시 실행하기 전에 해당 파일을 가져올 수 있는 충분한 시간을 허용하기 위해 100000초(27.8시간)로 설정됩니다. 이 속도 제한 기간 후에 EEM이 자체적으로 다시 실행되면 EPC를 수동으로 시작해야 하므로 새 패킷 캡처 데이터가 수집되지 않습니다. 그러나 새 show 명령 출력이 텍스트 파일에 추가됩니다.

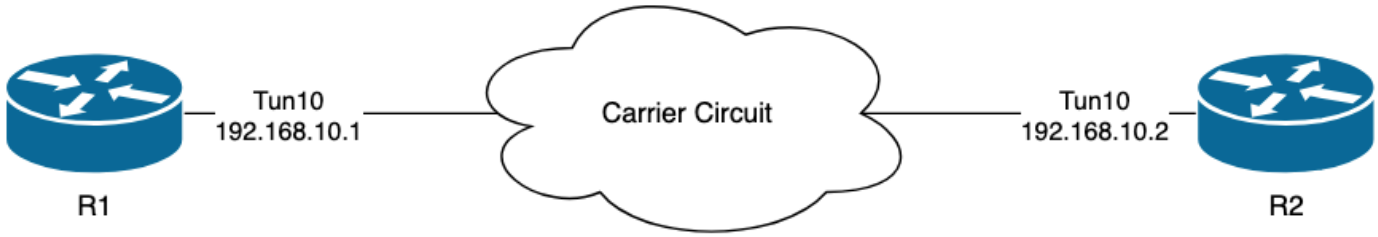
필요에 따라 EEM을 수정하여 플랫폼별 패킷 삭제 정보를 수집하고 시나리오에 필요한 추가 기능을 구현할 수 있습니다.

간헐적 라우팅 프로토콜 플랩 문제 해결

예 - EIGRP

이 예에서는 모든 타이머가 기본값으로 설정됩니다(5초 Hello, 15초 Holding Time).

토폴로지



R1의 로그는 여러 시간 간격으로 간헐적인 EIGRP 플랩이 발생했음을 나타냅니다.

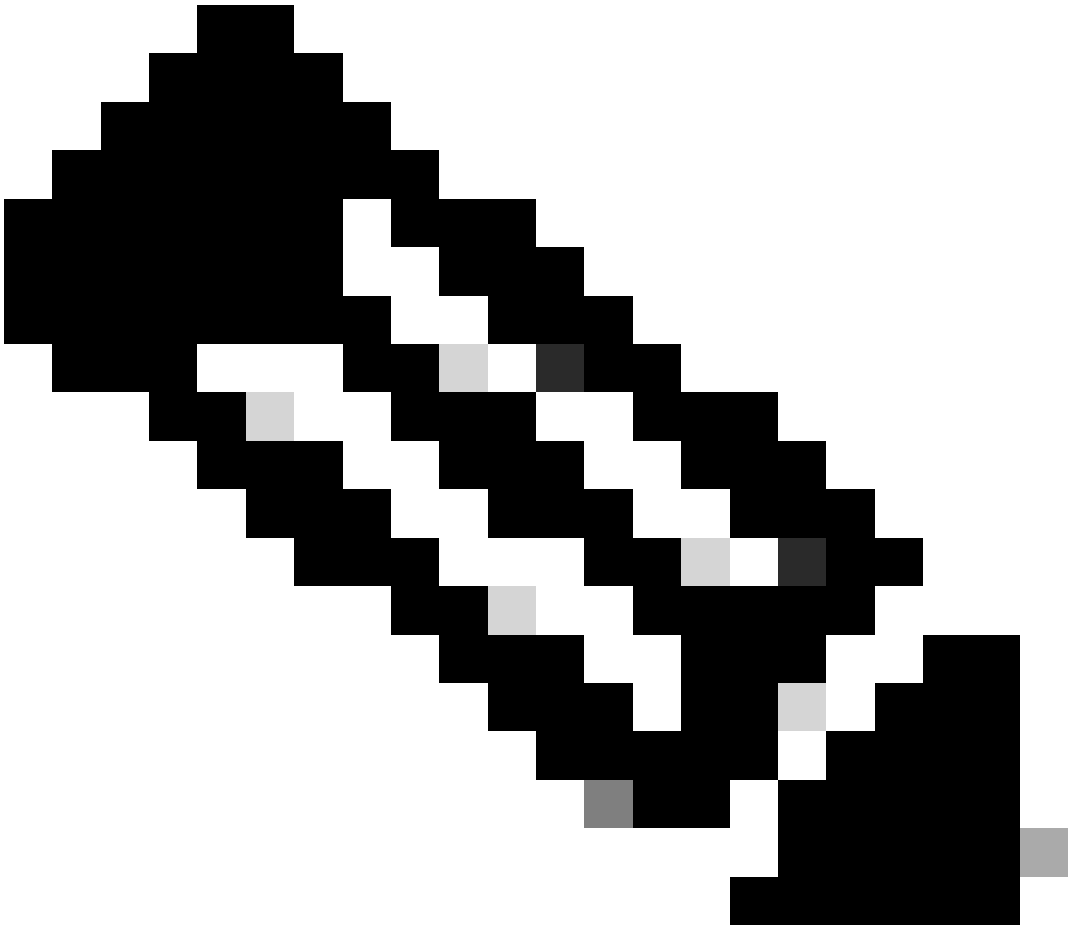
```
R1#show logging | i EIGRP
*Jul 16 20:45:08.019: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 192.168.10.2 (Tunnel10) is down: Interf
*Jul 16 20:45:12.919: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 192.168.10.2 (Tunnel10) is up: new adja
*Jul 17 10:25:42.970: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor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

패킷 손실은 양방향일 수 있습니다. holding time expired는 이 장치가 보류 시간 내에 피어에서 hello를 수신 또는 처리하지 않았음을 나타내고, Interface PEER-TERMINATION received는 보류 시간 내에 hello를 수신 또는 처리하지 않았기 때문에 피어가 인접성을 종료했음을 나타냅니다.

설정

1. 터널 인터페이스의 소스 IP 주소이므로 터널 인터페이스 IP 주소로 ACL을 구성합니다.

```
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
```



참고: 표시된 컨피그레이션은 R1에서 가져온 것입니다. 관련 인터페이스의 R2 및 EEM의 수정된 파일 이름에서도 마찬가지입니다. 추가 특수성이 필요한 경우 EIGRP 멀티캐스트 주소 224.0.0.10을 대상 IP 주소로 사용하는 ACL을 구성하여 Hello를 캡처합니다.

---

2. EPC를 생성하고 인터페이스 및 ACL과 연결합니다.

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

3. EPC를 시작하고 패킷이 양방향으로 캡처되었는지 확인합니다.

```
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. EEM을 구성합니다.

```
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. 다음 플랩이 발생할 때까지 기다렸다가 분석을 위해 원하는 전송 방법을 통해 bootflash에서 파일을 복사합니다.

```
R1#show logging

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

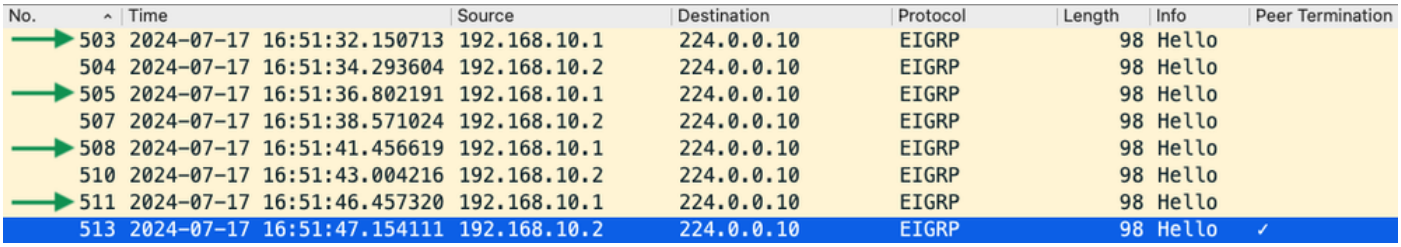
- 라우터의 로그 버퍼는 EIGRP 플랩이 있었으며 EEM에서 파일을 저장했음을 나타냅니다.

분석

이때 로그 버퍼에 있는 플랩의 시간과 수집한 패킷 캡처의 상관관계를 분석하여 플랩이 발생할 때 hello 패킷이 양 끝에서 전송 및 수신되었는지 확인합니다. 수신된 인터페이스 PEER-TERMINATION이 R1에 표시되었으므로, 이는 R2가 손실된 Hello를 탐지했어야 하며 따라서 유지 시간이 만료되었음을 의미하며, 이는 로그 파일에 표시됩니다.

```
*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
```

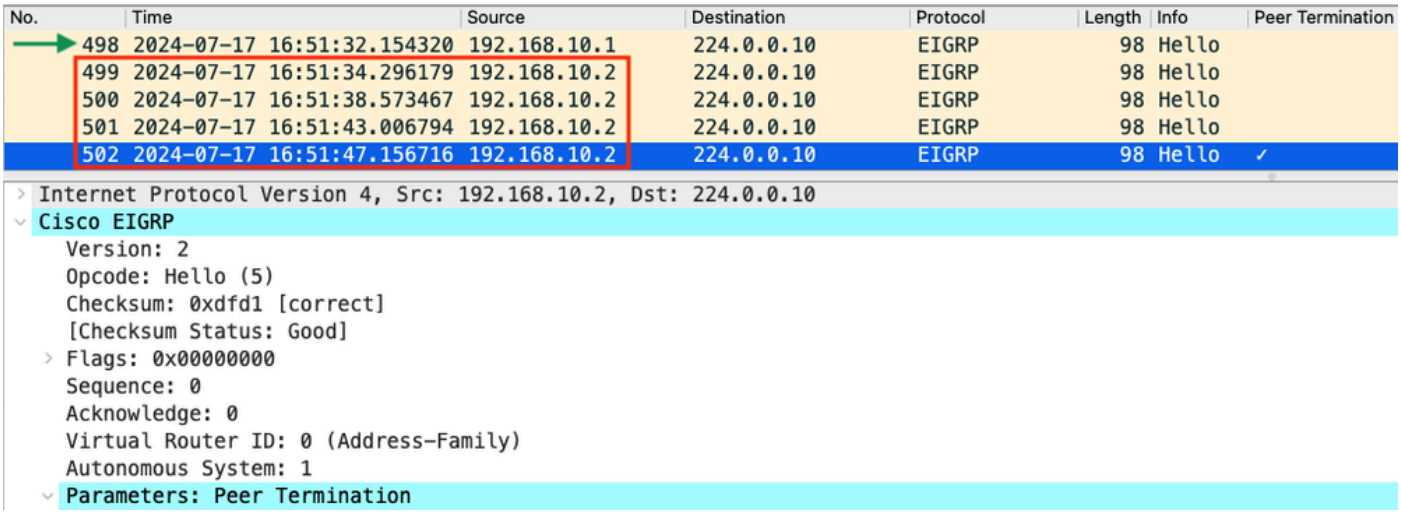
R2에서 대기 시간이 만료된 것을 감지했기 때문에 R1에 수집된 캡처의 플랩이 있기 15초 전에 R1이 보낸 Hello가 있는지 확인합니다.



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	✓

- 캡처는 R2가 16:51:47(패킷 513)에 전송하는 PEER-TERMINATION hello 패킷 이전 15초 동안 192.168.10.1(R1) 및 192.168.10.2(R2)의 Hello를 보여줍니다.
- 구체적으로, 패킷 503, 505, 508 및 511(녹색 화살표로 표시됨)은 이 기간 동안 R1에 의해 전송된 모든 헬로였다.

다음 단계는 R1이 전송한 모든 Hello를 당시 R2가 수신했는지 확인하는 것이므로 R2에서 수집한 캡처를 확인해야 합니다.



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
```

- 캡처는 192.168.10.1(R1)에서 받은 마지막 hello가 16:51:32(녹색 화살표로 표시)에 있음을 보여 줍니다. 이 후 다음 15초에는 R2(빨간색 상자로 표시됨)가 보낸 Hello만 표시됩니다. R1의 캡처에 있는 패킷 505, 508 및 511은 R2의 캡처에 나타나지 않습니다. 따라서 R2는 보류 타이머가 만료되었음을 탐지하고 16:51:47에 PEER-TERMINATION hello 패킷을 보냅니다(패킷

502).

이 데이터의 결론은 패킷 손실이 R1과 R2 사이의 캐리어 네트워크 어딘가에 있다는 것입니다. 이 경우 손실은 R1에서 R2 방향으로 발생했습니다. 더 자세히 조사하려면 삭제의 경로를 확인하기 위해 캐리어를 참여시켜야 합니다.

OSPF

동일한 논리를 사용하여 간헐적인 OSPF 플랩을 해결할 수 있습니다. 이 섹션에서는 타이머, IP 주소 필터 및 로그 메시지와 관련하여 다른 라우팅 프로토콜과 구별되는 주요 요소에 대해 설명합니다.

- 기본 타이머는 10초 Hello와 40초 Dead 타이머입니다. 데드 타이머 만료 플랩 트러블슈팅 시 네트워크에서 사용 중인 타이머를 항상 확인합니다.
- Hello 패킷은 인터페이스 IP 주소에서 소싱됩니다. 추가 ACL 특수성이 필요한 경우 OSPF Hello의 멀티캐스트 대상 주소는 224.0.0.5입니다.
- 디바이스의 로그 메시지는 약간 다릅니다. EIGRP와 달리 OSPF를 사용하는 피어 종료 메시지의 개념은 없습니다. 대신 만료된 데드 타이머를 탐지하는 디바이스는 이를 플랩 사유로 로깅한 다음 전송하는 hello에 피어의 라우터 ID가 더 이상 포함되지 않으므로 피어가 INIT 상태로 이동합니다. Hello가 다시 탐지되면 FULL 상태에 도달할 때까지 인접성이 전환됩니다. 예를 들면 다음과 같습니다.

R1은 만료된 데드 타이머를 탐지합니다.

```
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
```

그러나 R2는 OSPF가 FULL로 다시 이동할 때만 로그 메시지를 표시합니다. 상태가 INIT로 변경되면 로그 메시지가 표시되지 않습니다.

```
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
```

두 디바이스에서 EEM을 트리거하려면 syslog 패턴으로 "%OSPF-5-ADJCHG"를 사용합니다. 이렇게 하면 EEM이 중단되었다가 다시 가동되는 한 두 디바이스에서 모두 트리거됩니다. 구성된 ratelimit 값은 이 문자열의 여러 로그가 표시될 때 짧은 기간 내에 두 번 트리거되지 않도록 합니다. 양쪽의 패킷 캡처에서 Hello가 전송 및 수신되는지 확인하는 것이 핵심입니다.

BGP

동일한 논리를 사용하여 간헐적인 BGP 플랩을 해결할 수 있습니다. 이 섹션에서는 타이머, IP 주소 필터 및 로그 메시지와 관련하여 다른 라우팅 프로토콜과 구별되는 주요 요소에 대해 설명합니다.

- 기본 타이머는 60초 keepalive와 180초 hold time입니다. 보류 시간이 만료된 플랩 트러블슈팅 시 항상 네트워크에서 사용 중인 타이머를 확인합니다.
- 킵얼라이브 패킷은 인접 IP 주소 간에 유니캐스트로 TCP 대상 포트 179로 전송됩니다. 추가 ACL 특수성이 필요한 경우 소스 IP 주소에서 대상 TCP 포트 179로의 TCP 트래픽을 허용합니다.
- BGP에 대한 로그 메시지는 두 디바이스에서 비슷하지만, 보류 시간이 만료됨을 감지한 디바이스는 네이버에 알림을 보낸 것으로 표시되고, 다른 디바이스는 알림 메시지를 받았음을 나타냅니다. 예를 들면 다음과 같습니다.

R1은 만료된 보류 시간을 탐지하고 R2에 알림을 보냅니다.

```
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
```

R1에서 탐지된 보류 시간이 만료되었으므로 R2는 R1에서 알림을 수신합니다.

```
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
```

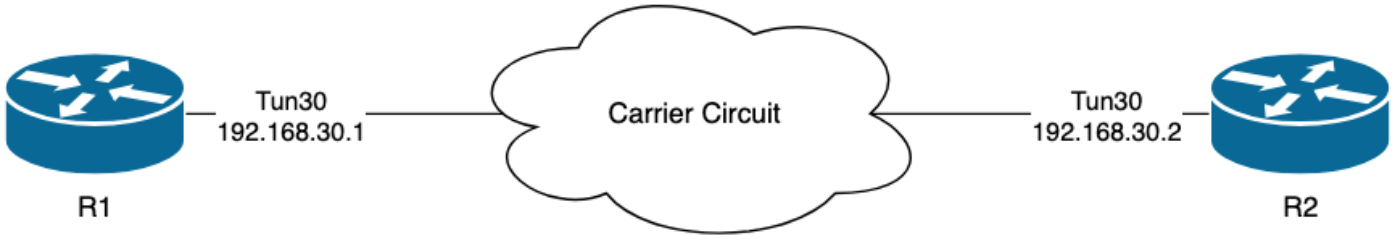
BGP 플랩에 대한 EEM을 트리거하려면 syslog 패턴으로 "%BGP_SESSION-5-ADJCHANGE"를 사용합니다. 플랩 이후에 로깅되는 다른 "%BGP" syslog 메시지도 EEM을 트리거하는 데 사용할 수 있습니다.

간헐적 BFD 플랩 문제 해결

동일한 방법론을 적용하여 간헐적인 BFD 플랩을 해결할 수 있으며, 분석에 적용할 몇 가지 사소한 차이점이 있습니다. 이 섹션에서는 몇 가지 기본적인 BFD 기능에 대해 설명하고 EEM 및 EPC를 사용하여 문제를 해결하는 방법의 예를 제공합니다. 자세한 BFD 문제 해결 정보는 Cisco IOS XE에서 Troubleshoot Bidirectional Forwarding Detection을 참조하십시오.

이 예에서 BFD 타이머는 승수 3으로 300ms로 설정되며, 이는 에코는 300ms마다 전송되며, 에코 실패는 행에서 3개의 에코 패킷이 반환되지 않을 때 탐지됩니다(900ms 대기 시간과 같음).

토폴로지



예 - BFD 에코 모드

BFD 에코 모드(기본 모드)에서는 BFD 에코 패킷이 로컬 인터페이스 IP를 소스 및 대상으로 전송합니다. 그러면 네이버가 데이터 플레인에서 패킷을 처리하여 소스 디바이스로 반환할 수 있습니다. 각 BFD 에코는 BFD 에코 메시지 헤더에 에코 ID를 포함하여 전송됩니다. 전송된 BFD 에코 패킷이 다시 수신되었는지 확인하는 데 사용할 수 있습니다. 인접 디바이스에서 실제로 반환한 경우 지정된 BFD 에코 패킷이 두 번 발생해야 하기 때문입니다. BFD 세션의 상태를 제어하는 데 사용되는 BFD 제어 패킷은 인터페이스 IP 주소 간에 유니캐스트로 전송됩니다.

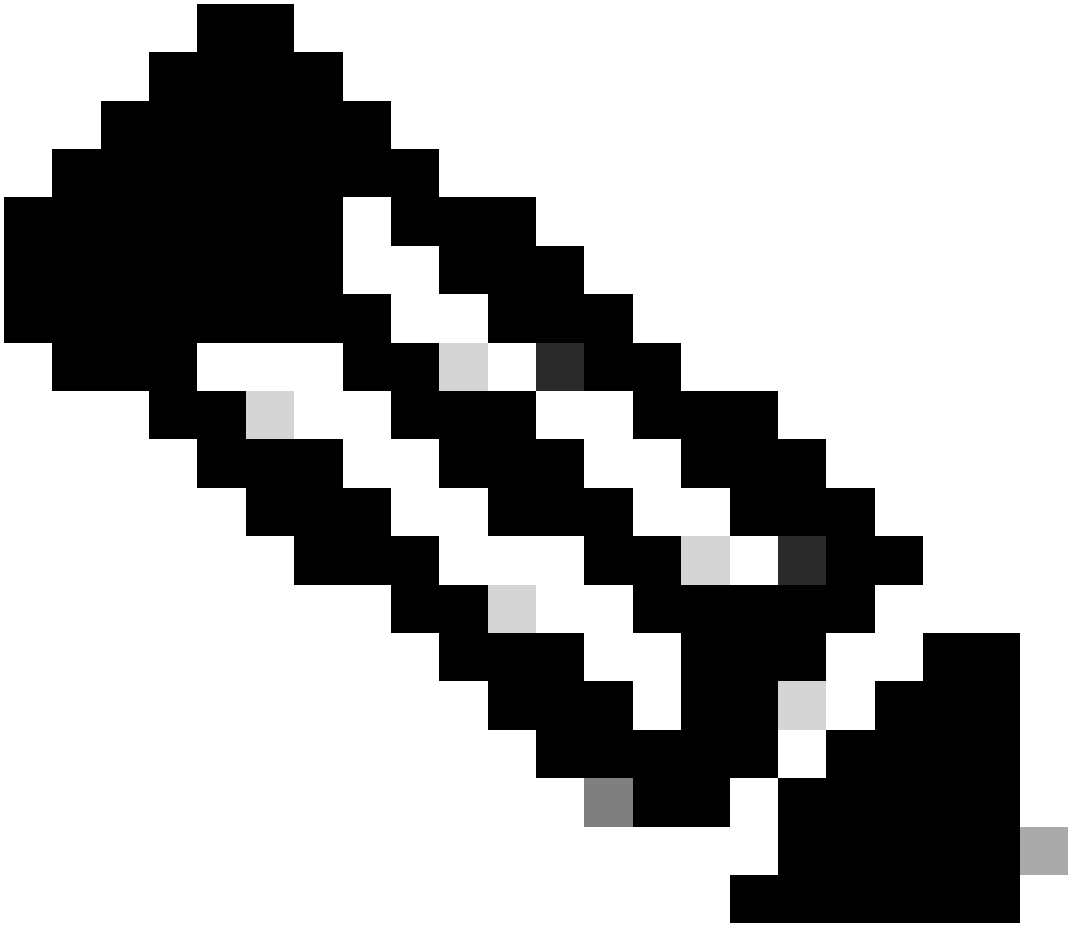
R1의 로그는 ECHO FAILURE로 인해 BFD 인접성이 여러 번 다운되었음을 나타냅니다. 즉, 해당 간격 동안 R1은 R2에서 자체 에코 패킷의 3을 수신 또는 처리하지 않았습니다.

```
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
```

설정

1. BFD 에코 패킷 및 제어 패킷의 소스 IP 주소이므로 터널 인터페이스 IP 주소로 ACL을 구성합니다.

```
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
```



참고: 표시된 컨피그레이션은 R1에서 가져온 것입니다. 관련 인터페이스의 R2 및 EEM의 수정된 파일 이름에서도 마찬가지입니다. 추가 지정이 필요한 경우 대상 포트 3785(에코 패킷) 및 3784(제어 패킷)를 사용하여 UDP에 대한 ACL을 구성합니다.

---

2. EPC를 생성하고 인터페이스 및 ACL과 연결합니다.

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

3. EPC를 시작하고 패킷이 양방향으로 캡처되었는지 확인합니다.

```
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. EEM을 구성합니다.

```
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. 다음 플랩이 발생할 때까지 기다렸다가 분석을 위해 원하는 전송 방법을 통해 bootflash에서 파일을 복사합니다.

```
R1#show logging
```

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

- 로그 버퍼는 19:09:47에 BFD 플랩이 있었으며 파일이 EEM에 의해 저장되었음을 나타냅니다 .

분석

이때 로그 버퍼에서 발견된 플랩의 시간을 수집된 패킷 캡처와 상호 연결하여 문제가 발생할 때 BFD 에코가 양쪽에서 전송 및 수신되었는지 확인합니다. R1의 플랩 사유가 ECHO FAILURE이므로, 이는 또한 BFD 세션을 종료하기 위해 제어 패킷을 R2에 보냈을 것이며, 이는 BFD 다운 사유

RX DOWN이 표시되는 R2에서 수집한 로그 파일에 반영됩니다.

```
*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
```

R1이 ECHO FAILURE를 감지했으므로 R1에서 수집한 패킷 캡처를 확인하여 플랩 전 900ms의 BFD 에코를 보내고 받았는지 확인합니다.

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

- 캡처는 R1이 플랩 시간까지 BFD 에코 패킷을 적극적으로 전송했지만 R2에 의해 반환되지 않았음을 보여줍니다. 따라서 R1은 제어 패킷을 전송하여 19:09:47.468에 세션을 종료합니다.
- 이는 패킷 137, 138 및 140(녹색 화살표로 표시됨)이 캡처에서 단 한 번만 보인다는 사실로부터 명백하며, 이는 BFD 에코 ID(빨간색 상자)로부터 결정될 수 있다. 에코가 반환된 경우 동일한 BFD 에코 ID를 가진 각 패킷의 두 번째 복사본이 있습니다. IP 헤더의 IP Identification 필드(여기서는 그림 참조)를 사용하여 이를 확인할 수도 있습니다.
- 이 캡처는 또한 패킷 136 이후에 R2로부터 수신된 BFD 에코가 없음을 보여주는데, 이는 R2에서 R1로의 방향의 패킷 손실의 또 다른 표시이다.

다음 단계는 R1에서 보낸 모든 BFD 에코 패킷이 R2에서 수신되고 반환되었는지 확인하는 것입니다. 따라서 R2에서 수집한 캡처를 확인해야 합니다.

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

- 이 캡처는 R1에 의해 전송된 모든 BFD 에코가 R2에 의해 수신되고 반환되었음을 보여줍니다(녹색 화살표로 표시됨). 패킷(107, 108)은 동일한 BFD 에코이고, 패킷(111, 112)은 동일한 BFD 에코이며, 패킷(116, 117)은 동일한 BFD 에코이다.
- 이 캡처는 또한 R2가 R1의 캡처에서 보이지 않는 에코 패킷(빨간색 상자로 표시됨)을 활발하게 전송했음을 보여주며, 이는 R2에서 R1 방향으로 디바이스 간 패킷 손실을 추가로 나타냅니다.

이 데이터의 결론은 패킷 손실이 R1과 R2 사이의 캐리어 네트워크 어딘가에 있다는 것입니다. 이 시점에서 모든 증거는 손실의 방향이 R2에서 R1으로 변경된다는 것을 나타냅니다. 더 자세히 조사하기 위해, 캐리어는 삭제의 경로를 확인하는 데 관여해야 합니다.

BFD 비동기 모드

BFD 비동기 모드가 사용 중인 경우(에코 기능 비활성화)에도 동일한 방법을 적용할 수 있으며, EEM과 EPC 설정을 동일하게 유지할 수 있습니다. 비동기 모드의 차이점은 디바이스가 일반적인 라우팅 프로토콜 인접성과 유사하게 유니캐스트 BFD 제어 패킷을 keepalive로 서로 전송한다는 것입니다. 즉, UDP 포트 3784 패킷만 전송됩니다. 이 시나리오에서, BFD는 BFD 패킷이 필요한 간격 내에 인접 디바이스로부터 수신되는 한 업(up) 상태를 유지한다. 이러한 상황이 발생하지 않으면 오류 원인은 DETECT TIMER EXPIRED이며 라우터는 세션을 종료하기 위해 제어 패킷을 피어로 전송합니다.

장애를 탐지한 디바이스의 캡처를 분석하려면 플랩 직전 시간 동안 피어에서 수신한 유니캐스트 BFD 패킷을 확인합니다. 예를 들어, TX 간격이 300ms로 설정되고 승수가 3인 경우 플랩 이전 900ms에서 수신된 BFD 패킷이 없는 경우 잠재적인 패킷 손실이 있음을 나타냅니다. EEM을 통해 네이버에서 수집한 캡처에서 이 동일한 타임 윈도우를 확인합니다. 패킷이 그 시간 동안 전송된 경우 디바이스 간 어딘가에 손실이 있음을 확인합니다.

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.