EEMおよびEPCによる断続的なルーティングプロトコルのフラップのトラブルシューティング

内容

はじめに

このドキュメントでは、EEMおよびEPCを搭載したCisco IOS® XEで断続的なルーティングプロトコルフラップおよびBFDフラップをトラブルシューティングする方法について説明します。

前提条件

要件

Wiresharkだけでなく、トラブルシューティングに関係するプラットフォームのEmbedded Event Manager(EEM)およびEmbedded Packet Capture(EPC)の詳細に精通しておくことをお勧めします。さらに、ルーティングプロトコルと双方向フォワーディング検出(BFD)に関する基本的なHelloおよびキープアライブ機能に精通していることが推奨されます。

使用するコンポーネント

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

問題の概要

ルーティングプロトコルの断続的なフラップは、実稼働ネットワークでは一般的な問題ですが、その予測不能な特性により、リアルタイムでのトラブルシューティングが困難な場合があります。EEMは、フラップが発生したときにsyslogストリングを使用してデータキャプチャをトリガーすることで、データ収集を自動化する機能を提供します。EEMとEPCを使用すると、隣接関係の両端からパケットキャプチャデータを収集して、フラップの発生前に潜在的なパケット損失を切り分けることができます。

断続的なルーティングプロトコルのフラップは、helloまたはキープアライブタイムアウトが常に原因で発生します（ただし、リンクのフラップなどの明らかな物理的な問題がログに記録される場合を除きます）。 したがって、このドキュメントのロジックはこの部分を対象としています。

トラブルシューティング手法

ルーティングプロトコルのフラップが発生するタイミングを判断する最も重要なことは、問題発生時に両方のデバイスでhelloパケットまたはキープアライブパケットが送受信されたかどうかです。このトラブルシューティング方法では、フラップが発生するまで循環バッファで連続したEPCを使用します。フラップが発生した時点で、EEMは関連するsyslog文字列を使用して一連のコマンドを実行し、そのうちの1つがEPCを停止します。循環バッファオプションを使用すると、EPCはバッファ内の最も古いパケットを上書きしながら、新しいパケットのキャプチャを続行できます。これにより、イベントがキャプチャされ、バッファがあらかじめ一杯になって停止することがなくなります。次に、パケットキャプチャデータをフラップのタイムスタンプと関連付けて、イベントの発生前に必要なパケットが両端で送受信されたかどうかを判別できます。

この問題は、インターネットサービスプロバイダー(ISP)などの中間ネットワーク上で隣接関係を形成するデバイスで最も一般的に発生しますが、特定のトポロジの詳細に関係なく、断続的に発生するルーティングプロトコルのフラップのシナリオに同じ方法を適用できます。ネイバーデバイスがサードパーティによって管理されていて、アクセスできない場合も同様です。このような場合、このドキュメントで説明するトラブルシューティング方法は、アクセス可能なデバイスがフラップの前に必要なパケットを送受信したかどうかを証明するためだけに適用できます。これが確認されると、必要に応じて相手側でさらにトラブルシューティングを行うために、ネイバーを管理する側にデータを表示できます。

設定の概要

このセクションでは、この自動データキャプチャのセットアップに使用できる一連の設定テンプレートについて説明します。必要に応じて、IPアドレス、インターフェイス名、およびファイル名を変更します。

ACL設定テンプレート

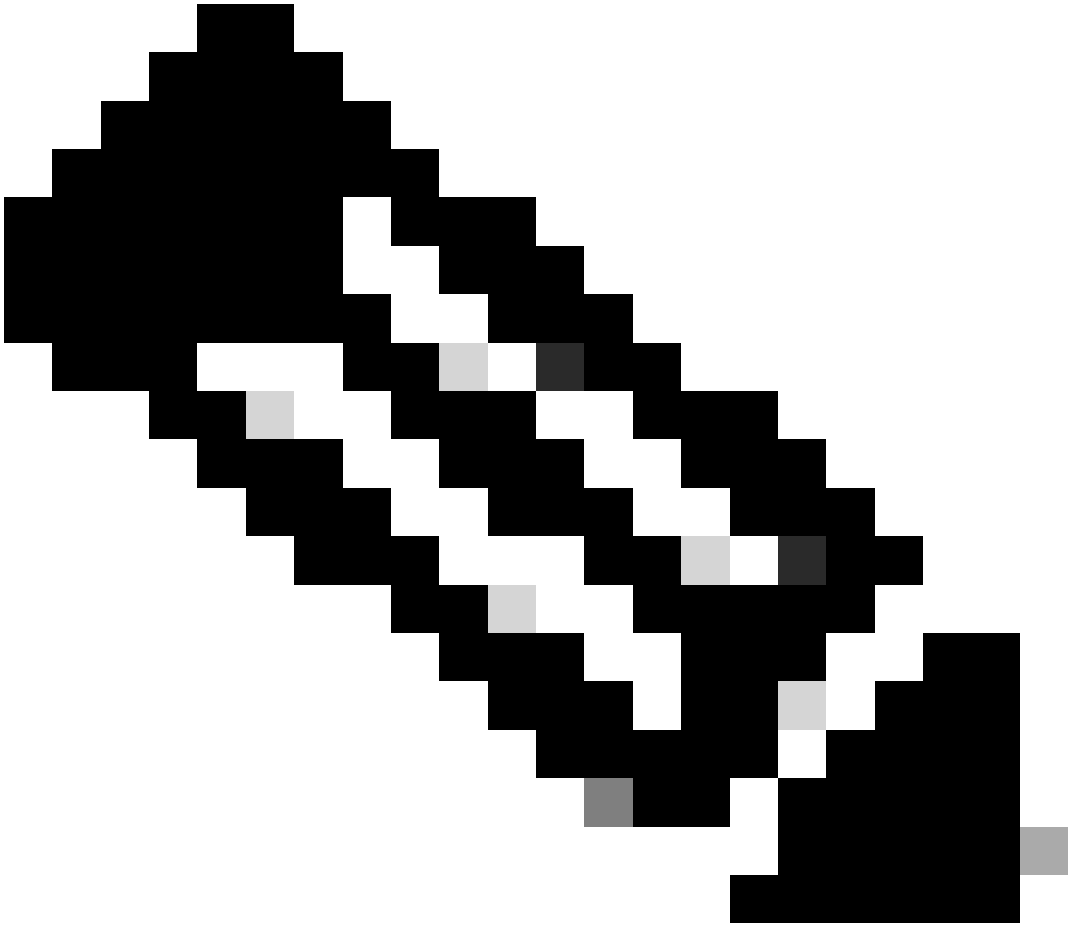
ほとんどの場合、ルーティング隣接関係の両端のインターフェイスIPアドレスから送信されるトラフィックは、ルーティング制御トラフィックそのものです。したがって、ローカルインターフェイスのIPアドレスとネイバーのIPアドレスの両方からあらゆる宛先へのトラフィックを許可するACLは、BFDだけでなく、あらゆるルーティングプロトコルの要件にも対応します。追加のフィルタが必要な場合は、ルーティングプロトコルまたはBFDモードに基づいて、関連する宛先IPを指定することもできます。コンフィギュレーションモードでACLパラメータを定義します。

```
config t

ip access-list extended




      permit ip host




       any permit ip host




        any end
```

EPCパラメータテンプレート

EPCパラメータは、configモードではなく、特権execモードから作成されます。プラットフォーム固有の設定ガイドを確認して、EPCに制限があるかどうかを確認してください。目的のインターフェイスのパラメータを作成し、ACLに関連付けて目的のトラフィックをフィルタリングします。

- monitor capture <EPC name> interface <interface>両方
- monitor capture <EPC名> access-list <ACL名>
- monitor capture <EPC name>バッファサイズ5循環



注：一部のソフトウェアバージョンでは、ローカルで生成されたトラフィックはインターフェイスレベルのEPCでは表示されません。このようなシナリオでは、キャプチャパラメータを変更して、CPUでトラフィックの両方向をキャプチャできます。

- モニタキャプチャ<EPC name>コントロールプレーンboth
- monitor capture <EPC名> access-list <ACL名>
- monitor capture <EPC name>バッファサイズ5循環

設定が完了したら、EPCを起動します。

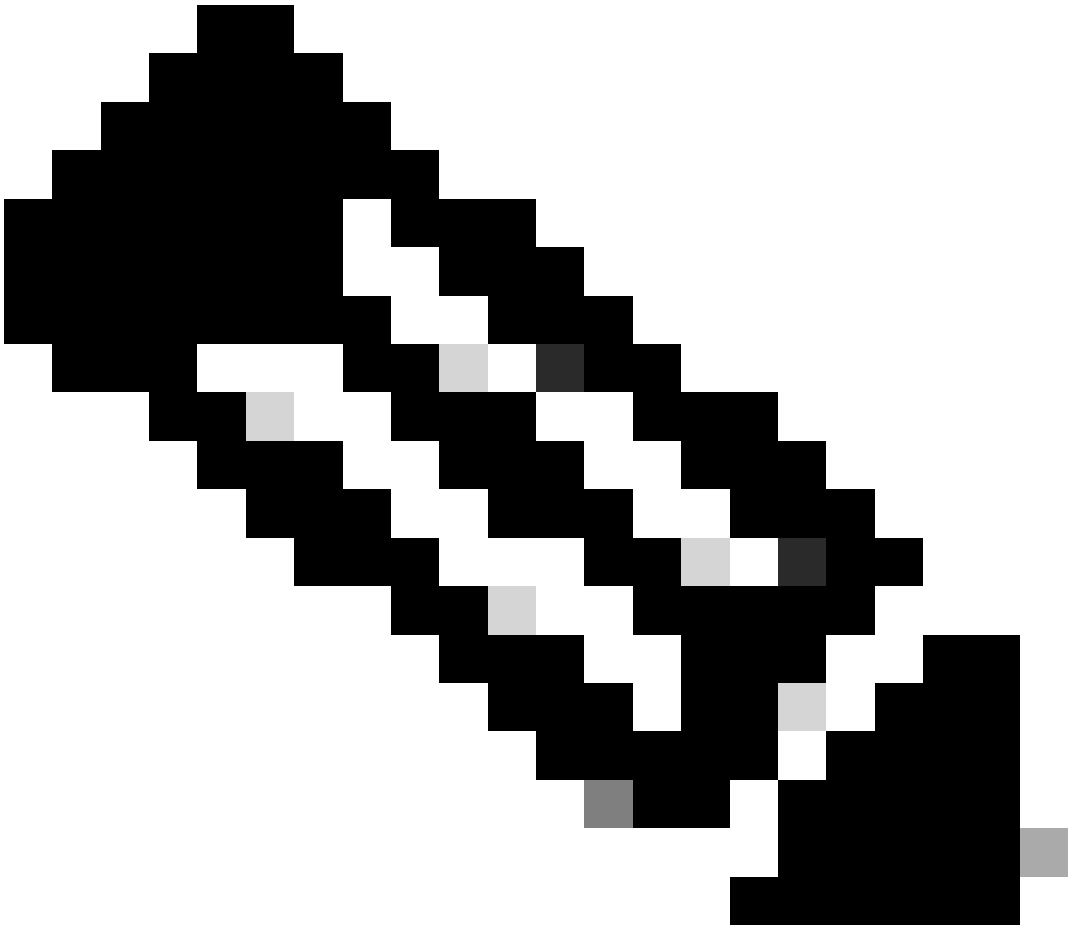
- monitor capture <EPC名> start

EEMは、フラップが発生したときにキャプチャを停止するように設定されています。

パケットが両方向でキャプチャされることを確認するには、キャプチャバッファを確認します。

```
show monitor capture
```

```
buffer brief
```



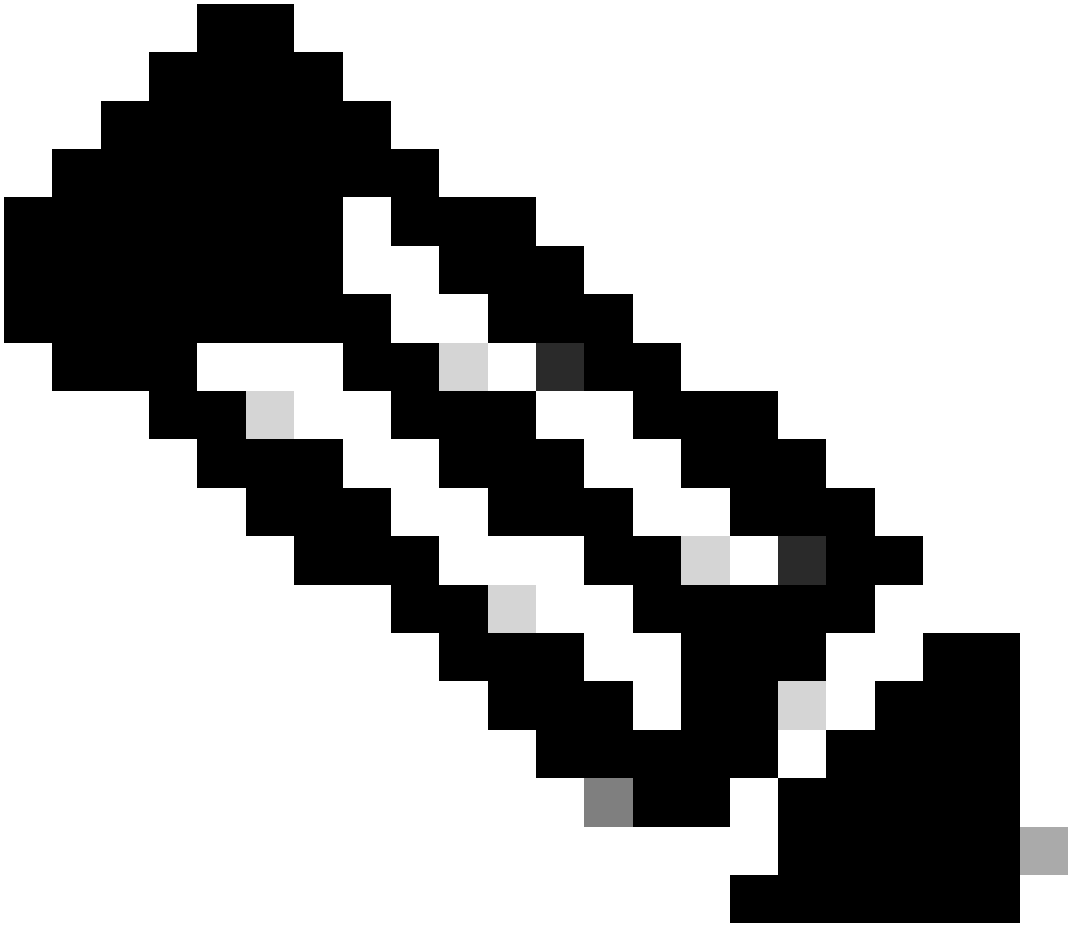
注:Catalystスイッチングプラットフォーム（Cat9kやCat3kなど）では、バッファを表示する前にキャプチャを停止する必要があります。キャプチャが正しく動作していることを確認するには、monitor capture stopコマンドでキャプチャを停止し、バッファを表示してから再開してデータを収集します。

EEM設定テンプレート

EEMの主な目的は、パケットキャプチャを停止し、syslogバッファとともに保存することです。CPU、インターフェイスの廃棄、プラットフォーム固有のリソース使用率と廃棄カウンタなど、他の要因を確認するために、追加のコマンドを組み込むことができます。設定モードでEEMアプレットを作成します。

```
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:Catalystスイッチングプラットフォーム（Cat9kやCat3kなど）では、キャプチャをエクスポートするコマンドは若干異なります。これらのプラットフォームでは、action 035で使用されているCLIコマンドを変更します。

```
action 035 cli command "monitor capture

      export location bootflash:

       .pcap"
```

EEMのratelimit値は秒単位で、EEMを再実行できるまでに経過する必要がある時間を示します。この例では、100000秒（27.8時間）に設定されているため、ネットワーク管理者がファイルの完了を確認し、再実行する前にデバイスからファイルを取り出すのに十分な時間を確保できます。このレート制限期間後にEEMが自動的に再実行される場合、EPCを手動で開始する必要があるため、新しいパケットキャプチャデータは収集されません。ただし、新しいshowコマンド出力がテキストファイルに追加されます。

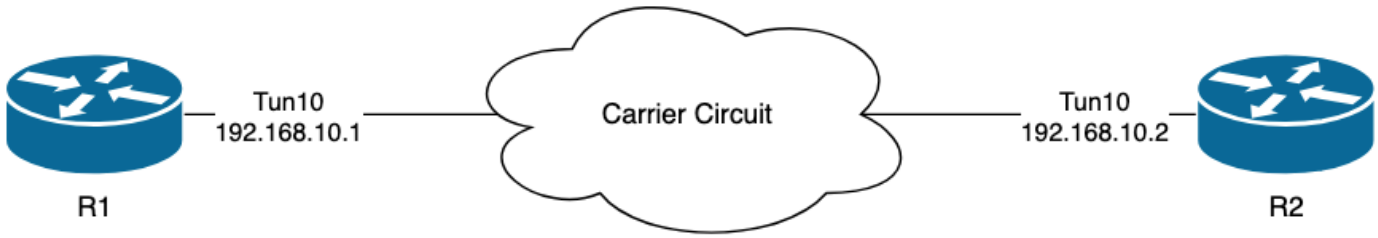
プラットフォーム固有のパケットドロップ情報を収集し、シナリオに必要な追加機能を実現するために、必要に応じてEEMを変更できます。

断続的なルーティングプロトコルフラップのトラブルシューティング

例：EIGRP

この例では、すべてのタイマーがデフォルトに設定されています（5秒間のhello、15秒間のホールドタイム）。

トポロジ



R1のログは、相互に数時間離れた場所で断続的なEIGRPフラップが発生したことを示しています。

```
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
```

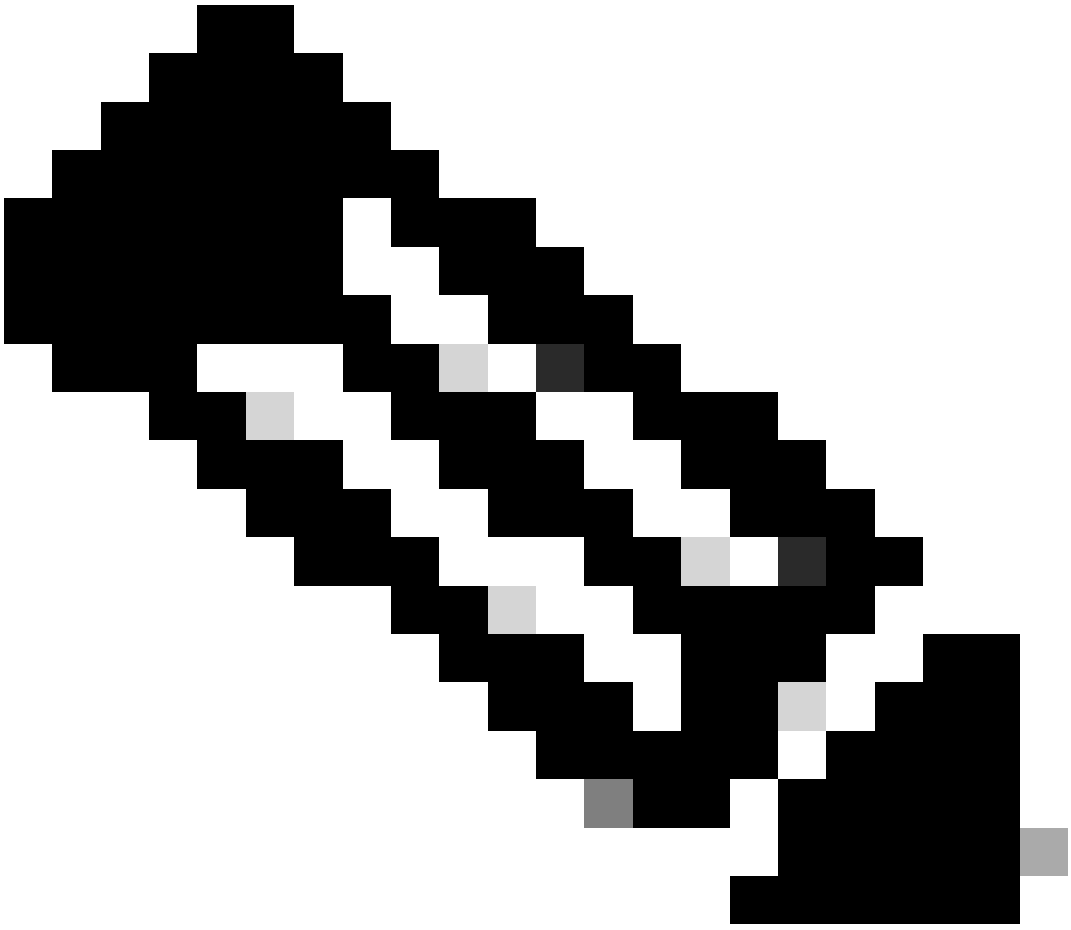
パケット損失は双方向になる可能性があります。holding time expiredは、このデバイスがホールドタイム内にピアからhelloを受信しなかったか、または処理しなかったことを示します。

Interface PEER-TERMINATION receivedは、ホールドタイム内にhelloを受信しなかったか、または処理しなかったために、ピアが隣接関係を終了したことを示します。

コンフィギュレーション

1. トンネルインターフェイスのIPアドレスを使用してACLを設定します。これらはhelloの送信元IPアドレスです。

```
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：示されている設定はR1のものです。関連するインターフェイスに対してR2で同じ操作を行い、EEMのファイル名を変更します。さらに詳細な指定が必要な場合は、helloを

キャプチャする宛先IPアドレスとしてEIGRPマルチキャストアドレス224.0.0.10を使用してACLを設定します。

2. EPCを作成し、インターフェイスおよびACLに関連付けます。

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

3. EPCを起動し、パケットが両方向でキャプチャされたことを確認します。

```
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. EEMを設定します。

```
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. 次のフラップが発生するのを待ち、分析のために必要な転送方式でブートフラッシュからファイルをコピーします。

```
R1#show logging
```

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

- ルータのログバッファは、EIGRPフラップが発生したこと、およびファイルがEEMによって保存されたことを示します。

分析

この時点で、ログバッファで見つかったフラップの時間を収集されたパケットキャプチャと関連付けて、フラップ発生時に両端でhelloパケットが送受信されたかどうかを確認します。インターフェイスPEER-TERMINATIONの受信がR1で確認されたため、これはR2が失われたhelloを検出し、保持時間が期限切れになったことを意味します。これはログファイルに表示されます。

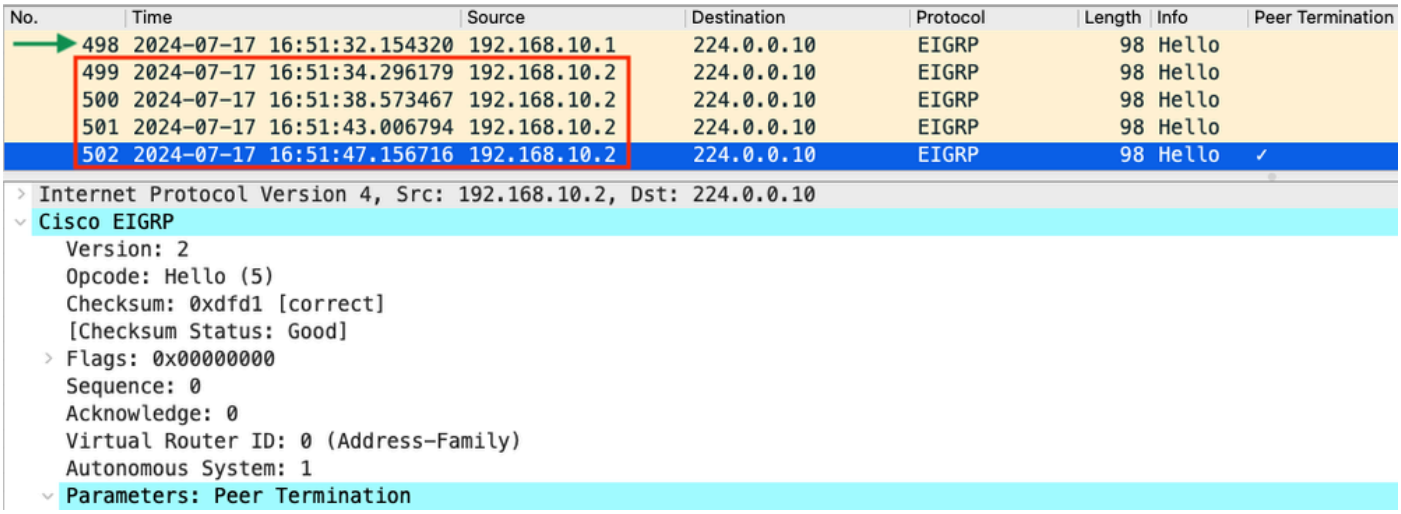
```
*Jul 17 16:51:47.156: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 192.168.10.1 (Tunnel10) is down: holdin
*Jul 17 16:51:51.870: %DUAL-5-NBRCHANGE: EIGRP-IPv4 1: Neighbor 192.168.10.1 (Tunnel10) is up: new adja
```

R2が保留時間の期限切れを検出したため、R1で収集されたキャプチャのフラップの前の15秒間にR1によってhelloが送信されたかどうかを確認します。

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	✓

- キャプチャは、R2が16:51:47（パケット513）に送信するPEER-TERMINATION helloパケットの15秒前の192.168.10.1(R1)と192.168.10.2(R2)の両方からのhelloを示します。
- 具体的には、パケット503、505、508、および511（緑色の矢印で示されている）は、この期間にR1によって送信されたすべてのhelloです。

次のステップでは、R1によって送信されたすべてのhelloがその時点でR2によって受信されたかどうかを確認します。そのため、R2から収集されたキャプチャを確認する必要があります。



- キャプチャは、192.168.10.1(R1)から受信した最後のhelloが16:51:32（緑色の矢印で示されている）であることを示しています。 この後、次の15秒では、R2によって送信されたhello（赤いボックスで示されています）のみが表示されます。 R1からのキャプチャのパケット505、508、および511は、R2のキャプチャには表示されません。これにより、R2はHolding timer expiredを検出し、16:51:47（パケット502）にPEER-TERMINATION helloパケットを送信します。

このデータの結論は、パケット損失がR1とR2の間のキャリアネットワークのどこかにあるということです。この例では、損失はR1からR2の方向でした。さらに詳しく調査するには、キャリアを関与させてパスのドロップをチェックする必要があります。

OSPF

同じロジックを使用して、断続的なOSPFフラップのトラブルシューティングを行うことができます。このセクションでは、タイマー、IPアドレスフィルタ、およびログメッセージに関して、他のルーティングプロトコルと区別する主な要因について説明します。

- デフォルトのタイマーは、10秒のhelloと40秒のdeadタイマーです。dead timer expired flapsのトラブルシューティングを行う際には、ネットワークで使用されているタイマーを必ず確認してください。
- Helloパケットの送信元はインターフェイスのIPアドレスです。追加のACL特異性が必要な場合、OSPF helloのマルチキャスト宛先アドレスは224.0.0.5です。
- デバイスのログメッセージは若干異なります。EIGRPとは異なり、OSPFにはピア終了メッセージの概念はありません。むしろ、deadタイマーの期限切れを検出したデバイスでは、これをフラップの理由としてログに記録し、そのデバイスから送信されるhelloにはピアのルータIDが含まれなくなるため、ピアはINIT状態に移行します。helloが再度検出されると、隣接関係はFULL状態に達するまで遷移します。例：

R1がdead timer expiredを検出します。

```
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, Neighb
*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, Neighb
```

```
*Jul 30 16:48:10.504: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.30.2 on Tunnel20 from LOADING to FULL, Loa
```

ただし、R2では、OSPFがFULLに戻ったときにだけログメッセージが表示されます。状態がINITに変わると、ログメッセージは表示されません。

```
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
```

両方のデバイスでEEMをトリガーするには、syslogパターンとして「%OSPF-5-ADJCHG」を使用します。これにより、EEMがダウンしてアップ状態に戻る限り、両方のデバイスで確実にEEMがトリガーされます。ratelimitの値を設定することで、この文字列を持つ複数のログが表示された場合に、短時間で2回トリガーされないようになります。キーは、Helloが両側のパケットキャプチャで送受信されるかどうかを確認することです。

BGP

断続的なBGPフラップのトラブルシューティングにも、同じロジックを使用できます。このセクションでは、タイマー、IPアドレスフィルタ、およびログメッセージに関して、他のルーティングプロトコルと区別する主な要因について説明します。

- デフォルトのタイマーは60秒のキープアライブと180秒のホールドタイムです。ホールドタイム期限切れフラップのトラブルシューティングを行う際は、ネットワークで使用されているタイマーを必ず確認してください。
- キープアライブパケットは、ネイバーIPアドレス間でTCP宛先ポート179にユニキャストで送信されます。追加のACL特異性が必要な場合、送信元IPアドレスから宛先TCPポート179へのTCPトラフィックを許可します。
- BGPのログメッセージは両方のデバイスで似ていますが、ホールドタイムの期限切れを検出したデバイスは、ネイバーに通知を送信したことを示し、他方は通知メッセージを受信したことを示します。例：

R1は保留時間の期限切れを検出し、R2に通知を送信します。

```
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
```

R1が保留時間の期限切れを検出したため、R2はR1から通知を受信します。

```
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
```

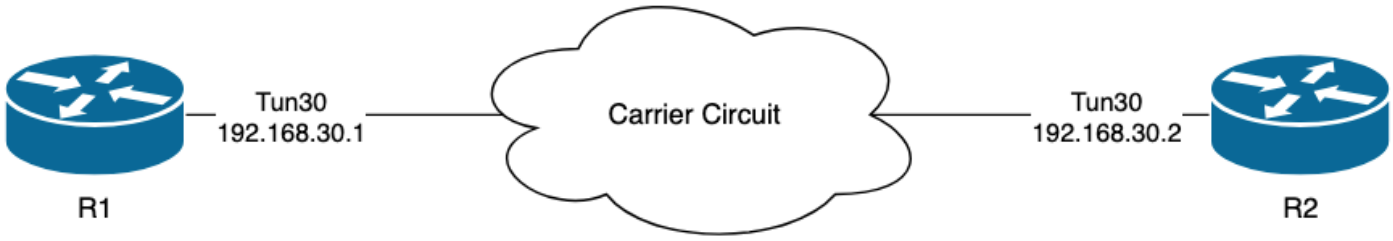
BGPフラップに対してEEMをトリガーするには、syslogパターンとして「%BGP_SESSION-5-ADJCHANGE」を使用します。フラップ後に記録されたその他の「%BGP」syslogメッセージも、EEMのトリガーに使用できます。

断続的なBFDフラップのトラブルシューティング

断続的なBFDフラップのトラブルシューティングにも同じ方法を適用できますが、分析に適用できるマイナーな違いがいくつかあります。このセクションでは、基本的なBFD機能について説明し、EEMとEPCを使用したトラブルシューティングの例を示します。BFDトラブルシューティングの詳細については、「Cisco IOS XEの双方向フォワーディング検出のトラブルシューティング」を参照してください。

この例では、BFDタイマーが300ミリ秒に設定され、乗数が3になっています。これは、300ミリ秒ごとにエコーが送信されることを意味します。連続する3つのエコーパケットが返されない場合（900ミリ秒のホールドタイムに相当）は、エコー障害が検出されます。

トポロジ



例 – BFDエコーモード

BFDエコーモード（デフォルトモード）では、BFDエコーパケットは送信元および宛先としてローカルインターフェイスIPを使用して送信されます。これにより、ネイバーはデータプレーンでパケットを処理し、送信元デバイスに返すことができます。各BFDエコーは、BFDエコーメッセージヘッダー内のエコーIDで送信されます。これらは、送信されたBFDエコーパケットが戻って受信されたかどうかを判断するために使用できます。これは、ネイバーによって実際に返された場合は、任意のBFDエコーパケットが2回発生している必要があるためです。BFD制御パケットは、BFDセッションの状態を制御するために使用され、インターフェイスIPアドレス間でユニキャストで送信されます。

R1からのログは、BFD隣接関係がエコー障害により複数回ダウンしたことを示しています。これは、これらのインターバルの間、R1がR2から戻された独自のエコーパケットの3つを受信または処理しなかったことを意味します。

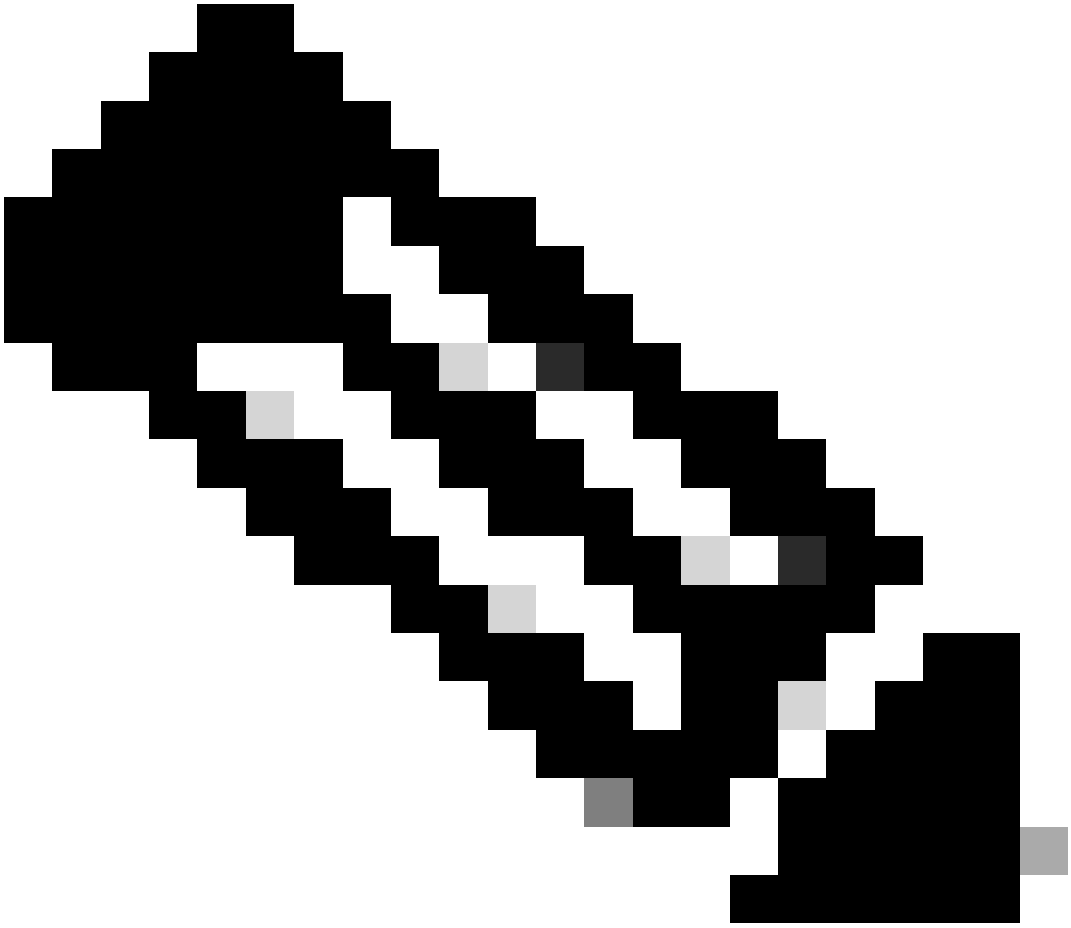
```
R1#show logging | i BFD
```

```
*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
```

コンフィギュレーション

1. トンネルインターフェイスのIPアドレスを使用してACLを設定します。これは、BFDエコーパケットと制御パケットの送信元IPアドレスであるためです。

```
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：示されている設定はR1のものです。関連するインターフェイスに対してR2で同じ操作を行い、EEMのファイル名を変更します。追加の特異性が必要な場合は、宛先ポート3785（エコーパケット）および3784（制御パケット）を使用してUDPのACLを設定します。

2. EPCを作成し、インターフェイスおよびACLに関連付けます。

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

3. EPCを起動し、パケットが両方向でキャプチャされたことを確認します。

```
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. EEMを設定します。

```
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. 次のフラップが発生するのを待ち、分析のために必要な転送方式でブートフラッシュからファイルをコピーします。

```
R1#show logging

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

- ログバッファは、19:09:47にBFDフラップが発生し、ファイルがEEMによって保存されたことを示します。

分析

この時点で、ログバッファで見つかったフラップの時間と収集されたパケットキャプチャを関連

付けて、問題が発生したときに両端でBFDエコーが送受信されたかどうかを確認します。R1のフラップ理由はECHO FAILUREであるため、これは、BFDセッションを終了するために制御パケットもR2に送信したことを意味します。これは、BFDダウン理由RX DOWNが表示されるR2から収集したログファイルに反映されます。

```
*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
```

R1はエコー障害を検出したため、R1で収集されたパケットキャプチャを調べて、フラップの前に900ミリ秒でBFDエコーを送受信したかどうかを確認します。

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	0000000000001002000041f	Originator specific content
136	2024-07-18 19:09:46.484581	192.168.30.2	192.168.30.2	BFD Echo	78	0000000000001002000041f	Originator specific content
137	2024-07-18 19:09:46.707712	192.168.30.1	192.168.30.1	BFD Echo	78	0000000000001001000041d	Originator specific content
138	2024-07-18 19:09:46.970921	192.168.30.1	192.168.30.1	BFD Echo	78	0000000000001001000041e	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	0000000000001001000041f	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

- キャプチャは、R1がフラップの時点までBFDエコーパケットをアクティブに送信したが、R2から返されなかったことを示しています。そのため、R1は制御パケットを送信して、セッションを19:09:47.468で終了します。
- これは、パケット137、138、および140（緑色の矢印で示される）がキャプチャ内で1回だけ表示され、BFDエコーID（赤いボックス内）から確認できることを示しています。 エコーが返されていれば、同じBFDエコーIDを持つ各パケットの2つ目のコピーが存在することになります。IPヘッダーのIP Identificationフィールド（この図には示されていません）を使用して、同様に確認できます。
- このキャプチャは、パケット136の後にR2からBFDエコーを受信しなかったことも示しています。これは、R2からR1への方向のパケット損失のもう1つの兆候です。

次のステップは、R1によって送信されたすべてのBFDエコーパケットがR2によって受信され、返されたかどうかを確認することです。そのため、R2から収集されたキャプチャを確認する必要があります。

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	0000000000001001000041d	Originator specific content
108	2024-07-18 19:09:46.708430	192.168.30.1	192.168.30.1	BFD Echo	78	0000000000001001000041d	Originator specific content
110	2024-07-18 19:09:46.774829	192.168.30.2	192.168.30.2	BFD Echo	78	0000000000001002000041d	Originator specific content
111	2024-07-18 19:09:46.971240	192.168.30.1	192.168.30.1	BFD Echo	78	0000000000001001000041e	Originator specific content
112	2024-07-18 19:09:46.971542	192.168.30.1	192.168.30.1	BFD Echo	78	0000000000001001000041e	Originator specific content
113	2024-07-18 19:09:47.015058	192.168.30.2	192.168.30.2	BFD Echo	78	0000000000001002000041e	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	0000000000001001000041f	Originator specific content
117	2024-07-18 19:09:47.204021	192.168.30.1	192.168.30.1	BFD Echo	78	0000000000001001000041f	Originator specific content
118	2024-07-18 19:09:47.286688	192.168.30.2	192.168.30.2	BFD Echo	78	0000000000001002000041f	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

- このキャプチャは、R1によって送信されたすべてのBFDエコーがR2によって受信され、返されたことを示しています（緑色の矢印で示されています）。パケット107と108は同じBFDエコー、パケット111と112は同じBFDエコー、パケット116と117は同じBFDエコーです。
- このキャプチャは、R2がエコーパケット（赤いボックスで示されている）をアクティブに

送信したことも示しています。これらはR1のキャプチャには表示されておらず、R2からR1の方向にあるデバイス間のパケット損失を示しています。

このデータの結論は、パケット損失はR1とR2の間のキャリアネットワークのどこかにあり、この時点での証拠はすべて、損失の方向がR2からR1であることを示しています。さらに詳しく調査するには、キャリアを関与させてパスのドロップをチェックする必要があります。

BFD非同期モード

BFD非同期モードを使用している場合(echo function disabled)も同様の方法を適用できます。また、EEMとEPCの設定を同じままにできます。非同期モードの違いは、一般的なルーティングプロトコルの隣接関係に似た方法で、デバイスが互いにキープアライブとしてユニキャストBFD制御パケットを送信することです。これは、UDPポート3784のパケットだけが送信されることを意味します。このシナリオでは、必要なインターバルの間にネイバーからBFDパケットを受信している限り、BFDはアップ状態のままです。これが発生しない場合、障害の原因はDETECT TIMER EXPIREDであり、ルータはピアに制御パケットを送信してセッションをダウンさせます。

障害を検出したデバイス上のキャプチャを分析するには、フラップの直前にピアから受信したユニキャストBFDパケットを探します。たとえば、TX間隔が3の倍数で300ミリ秒に設定されていて、フラップの前の900ミリ秒でBFDパケットが受信されない場合は、パケット損失の可能性があることを示しています。EEMを介してネイバーから収集されたキャプチャでは、この同じタイムウィンドウを確認します。その時間内にパケットが送信された場合は、デバイス間のどこかで損失が発生していることを確認します。

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。