初心者向けNexusチートシートのトラブルシューティング

内容

概要

このドキュメントでは、問題の診断と修正に使用できるNexus製品のトラブルシューティングに使用できるさまざまなツールについて説明します。

概要

どのツールが利用可能で、どのシナリオで最大の利益のためにそれらを使用するかを理解することが重要です。実際には、特定のツールは、単に他の何かに取り組むように設計されているという理由だけで実現可能ではない場合があります。

次の表に、Nexusプラットフォームでトラブルシューティングを行うさまざまなツールとその機能をまとめます。詳細とCLIの例については、「Nexusツール」の項を参照してください。

ツール	機能	使用例	長所	短所	PERSISTENCE	対象面	使用するCLIコマ
Ethanalyzer	CPU宛またはCPUからのトラフィックをキャプチャする	トラフィック速度の問題、遅延、および輻輳	速度低下、輻輳、および遅延の問題に最適	通常はコントロールプレーントラフィックのみを認識し、レートは制限される	N/A	コントロールプレーン.一部のシナリオ（SPANからCPU	#ethanalyzer loc interface inband #ethanalyzer loc interface [interfa ID] display filter [WORD] 例： #ethanalyzer loc interface Ethern

						）でデータプレーンに使用可能	display filter ICM
SPAN	多数のパケットのキャプチャとミラーリング	Failed（故障）ping s、不正なパケットなど	断続的なトラフィック損失に最適	スニファソフトウェアを実行する外部デバイスが必要TCAMリソースが必要	SPANセッションを設定し、有効/無効にする必要があります	コントロール+データ	#monitor sessio #description [NA #source interfac [port ID] #destin interface [port ID shut
ミラー	Broadcom Nexusデバイスに対してのみ、CPU宛またはCPUからのトラフィックをキャプチャします。	トラフィック速度の問題、遅延、および輻輳	速度低下、輻輳、および遅延の問題に最適	Broadcom Nexusデバイスのみ。レート制限（CloudScale Nexus 9000はSPANからCPUをサポート）	N/A	コントロールプレーン.一部のシナリオではデータプレーンに使用可能	プラットフォーよって異なりまを参照してくだ。ELAMの概要：コ
ELAM	Nexusスイッチに入る（またはNexus 7000の場合は出る）単一のパケットをキャプチャします。	パケットがNexusに到達することを確認し、転送の決定を確認し、パケットの変更を確認し、パケットのインターフェイス/VLANを確認します。	パケットフローおよび転送の問題に最適です。非侵入型	ハードウェアの詳細な理解が必要アーキテクチャ固有の独自のトリガーメカニズムを利用します。検査するトラフィックが分かっている場合にのみ役立ちます	N/A	コントロール+データ	# attach module [MODULE NUM # debug platform internal <>
Nexus 9k Packet Tracer	パケットのパスの検出	接続の問題とパケット損失	断続的/完全な損失に役立つフロー統計情報のカウンタを提供します。TCAMによる彫刻のないラインカードに最	ARPトラフィックをキャプチャできません。Nexus 9000でのみ動作	N/A	データ+コントロール	# test packet-tra src_IP [送信元IP dst_IP [宛先IP] # packet-tracer st test packet-trac stop # test pack tracer show

適

Traceroute	L3ホップに関するパケットのパスを検出する	pingの失敗、ホスト/宛先/インターネットに到達できないなど	パス内のさまざまなホップを検出して、L3障害を切り分けます。	L3境界が壊れている場所のみを特定します（問題自体は特定しません）。	N/A	データ+コントロール	# traceroute [宛 引数は次のとおす。 port、port numb source、interfac vrf、source-inte
ping	ネットワーク内の2点間の接続をテストする	デバイス間の到達可能性のテスト	接続をテストするための迅速でシンプルなツール	ホストが到達可能かどうかを識別するだけです	N/A	データ+コントロール	# ping [宛先IP] 引数は次のとおす。 count、packet-s source interface interval、multica loopback、time
PACL/RACL/VACL	特定のポートまたはVLANに対する入出力トラフィックのキャプチャ	ホスト間の断続的なパケット損失、パケットがNexusに到着またはNexusから出ているかどうかを確認する、など	断続的なトラフィック損失に最適	TCAMリソースが必要です。一部のモジュールでは、手動によるTCAMカービングが必要です	持続性(適用対象 running-設定)	データ+コントロール	# ip access-list [NAME] # ip port access-group [A NAME] # ip acc group [ACL NAM 引数は次のとおす。 deny、fragment no、permit、 remark、show、 statistics、end、 exit、pop、pus where
LogFlash	デバイスのリロードに関係なく、ログアカウント、クラッシュファイル、イベントなど、スイッチの履歴データをグローバルに保存します。	デバイスの突然のリロード/シャットダウン、デバイスのリロード時にログフラッシュデータが分析に役立つ情報を提供する	デバイスのリロード時に情報が保持される（永続的ストレージ）	Nexus 7000の外部=これらのログを収集するには、スーパーバイザプラットフォームにインストールまたは統合する必要があります。（logflashは内部ストレージデバイスのパーティションであるため、conは3K/9Kには適用されません）	Reload-Persistent	データ+コントロール	# dir logflash:
OBFL	障害や環境情報な	デバイスの突然のリロード/シャ	デバイスのリロード時	読み取りと書き込みの数を	Reload-Persistent	データ+コン	# show logging onboard module

	どの特定のモジュールに履歴データを保存	ットダウン、デバイスのリロード時には、ログフラッシュデータが役立つ情報を提供します	に情報が保持される（永続的ストレージ）	制限		トロール	引数は次のとおす。 boot-uptime、ca boot-history、ca first-power-on、 counter-stats、 device-version、 endtime、 environmental- history、error-st exception-log、 internal、interru stats、obfl-histo stat-trace、 starttime、statu # show [PROCE internal event-hi [ARGUMENT]
イベント履歴	現在実行中の特定のプロセスの情報が必要な場合	Nexusのすべてのプロセスには、CDP、STP、OSPF、EIGRP、BGP、vPC、LACPなどの独自のイベント履歴があります	Nexusで実行されている特定のプロセスのトラブルシューティング	デバイスがリロードされると情報が失われる（非永続的）	非永続的	データ+コントロール	引数は次のとおす。 隣接関係， cli， ント，フラッデグ， ha, hello, lo lsa, msgs, objstore，再配布 rib, segrt, spf, sp trigger，統計，
デバッグ	特定のプロセスに対して、より詳細なリアルタイム/ライブ情報が必要な場合	Nexusのすべてのプロセス（CDP、STP、OSPF、IGRP、BGP、vPC、LACPなど）のデバッグを実行できます	Nexusで実行されている特定のプロセスのトラブルシューティングをリアルタイムで行い、より詳細に把握する	ネットワークパフォーマンスに影響を与える可能性がある	非永続的	データ+コントロール	# debug process [PROCESS] 例： # debug ip ospf
ゴールド	ハードウェアコンポーネント（I/Oやスーパーバイザモジュールなど）のブートアップ、ランタイム、オンデマンド診断を提供	USB、ブートフラッシュ、OBFL、ASICメモリ、PCIE、ポートループバック、NVRAMなどのテストハードウェア	リリース6(2)8以降でのみ、ハードウェアの障害を検出し、必要な修正措置を講じることができます。	ハードウェアの問題のみを検出	非永続的	N/A	# show diagnost content module show diagnostic description mod [#] test all
EEM	デバイス	インターフェイ	Pythonスク	EEMを設定す	EEMスク	N/A	変化します。を

のイベントを監視し、必要なアクションを実行する	スのシャットダウン、ファンの誤動作、CPU使用率など、何らかのアクション/回避策/通知を必要とするデバイスのアクティビティ	リプトをサポート	るには、ネットワーク管理者権限が必要です。	リプトとトリガーが設定に存在する	してください。Embedded Ever Managerの設定

Nexusツール

さまざまなコマンドとその構文またはオプションについて詳しく説明する必要がある場合は、 Cisco Nexus 9000シリーズスイッチ – コマンドリファレンス – Cisco.

- **Ethanalyzer**

Ethanalyzerは、パケットのCPUトラフィックをキャプチャするように設計されたNX-OSツールです。このツールを使用すると、入力または出力に関係なく、CPUにヒットするものをすべてキャプチャできます。これは、広く使用されているオープンソースのネットワークプロトコルアナライザWiresharkに基づいています。このツールの詳細については、『Nexus 7000でのEthanalyzerのトラブルシューティングガイド：シスコ』を参照してください。

Ethanalyzerは通常、スーパーバイザとの間でやり取りされるすべてのトラフィックをキャプチャします。つまり、インターフェイス固有のキャプチャをサポートしません。特定のインターフェイスの拡張機能は、より新しいコードポイントの一部のプラットフォームで利用できます。また、Ethanalyzerは、ハードウェアスイッチングではなく、CPUスイッチングされたトラフィックのみをキャプチャします。たとえば、インバンドインターフェイス、管理インターフェイス、またはフロントパネルポート（サポートされている場合）のいずれかでトラフィックをキャプチャできます。

```
Nexus9000_A(config-if-range)# ethanalyzer local interface inband
Capturing on inband
2020-02-18 01:40:55.183177 cc:98:91:fc:55:8b -> 01:80:c2:00:00:00 STP RST. Root =
32768/1/cc:98:91:fc:55:80  Cost = 0  Port = 0x800b
2020-02-18 01:40:55.184031 f8:b7:e2:49:2d:f2 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:40:55.184096 f8:b7:e2:49:2d:f5 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:40:55.184147 f8:b7:e2:49:2d:f4 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:40:55.184190 f8:b7:e2:49:2d:f3 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:40:55.493543 dc:f7:19:1b:f9:85 -> 01:80:c2:00:00:00 STP RST. Root =
32768/1/dc:f7:19:1b:f9:80  Cost = 0  Port = 0x8005
2020-02-18 01:40:56.365722      0.0.0.0 -> 255.255.255.255 DHCP DHCP Discover - Transaction ID
0xc82a6d3
2020-02-18 01:40:56.469094 f8:b7:e2:49:2d:b4 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:40:57.202658 cc:98:91:fc:55:8b -> 01:80:c2:00:00:00 STP RST. Root =
32768/1/cc:98:91:fc:55:80  Cost = 0  Port = 0x800b
2020-02-18 01:40:57.367890      0.0.0.0 -> 255.255.255.255 DHCP DHCP Discover - Transaction ID
0xc82a6d3
10 packets captured
```

```
Nexus9000_A(config-if-range)# ethanalyzer local interface mgmt
Capturing on mgmt0
2020-02-18 01:53:07.055100 cc:98:91:fc:55:94 -> 01:80:c2:00:00:00 STP RST. Root =
32768/46/84:8a:8d:7d:a2:80  Cost = 4  Port = 0x8014
2020-02-18 01:53:09.061398 cc:98:91:fc:55:94 -> 01:80:c2:00:00:00 STP RST. Root =
32768/46/84:8a:8d:7d:a2:80  Cost = 4  Port = 0x8014
2020-02-18 01:53:11.081596 cc:98:91:fc:55:94 -> 01:80:c2:00:00:00 STP RST. Root =
32768/46/84:8a:8d:7d:a2:80  Cost = 4  Port = 0x8014
2020-02-18 01:53:13.080874 cc:98:91:fc:55:94 -> 01:80:c2:00:00:00 STP RST. Root =
32768/46/84:8a:8d:7d:a2:80  Cost = 4  Port = 0x8014
2020-02-18 01:53:15.087361 cc:98:91:fc:55:94 -> 01:80:c2:00:00:00 STP RST. Root =
32768/46/84:8a:8d:7d:a2:80  Cost = 4  Port = 0x8014
2020-02-18 01:53:17.090164 cc:98:91:fc:55:94 -> 01:80:c2:00:00:00 STP RST. Root =
32768/46/84:8a:8d:7d:a2:80  Cost = 4  Port = 0x8014
2020-02-18 01:53:19.096518 cc:98:91:fc:55:94 -> 01:80:c2:00:00:00 STP RST. Root =
32768/46/84:8a:8d:7d:a2:80  Cost = 4  Port = 0x8014
2020-02-18 01:53:20.391215 00:be:75:5b:d9:00 -> 01:00:0c:cc:cc:cc CDP Device ID:
Nexus9000_A(FDO21512ZES)  Port ID: mgmt0
2020-02-18 01:53:21.119464 cc:98:91:fc:55:94 -> 01:80:c2:00:00:00 STP RST. Root =
32768/46/84:8a:8d:7d:a2:80  Cost = 4  Port = 0x8014
2020-02-18 01:53:23.126011 cc:98:91:fc:55:94 -> 01:80:c2:00:00:00 STP RST. Root =
32768/46/84:8a:8d:7d:a2:80  Cost = 4  Port = 0x8014
10 packets captured

Nexus9000-A# ethanalyzer local interface front-panel eth1/1
Capturing on 'Eth1-1'
1 2022-07-15 19:46:04.698201919 28:ac:9e:ad:5c:b8  01:80:c2:00:00:00 STP 53 RST. Root =
32768/1/28:ac:9e:ad:5c:b7 Cost = 0 Port = 0x8001
2 2022-07-15 19:46:04.698242879 28:ac:9e:ad:5c:b8  01:00:0c:cc:cc:cd STP 64 RST. Root =
32768/1/28:ac:9e:ad:5c:b7 Cost = 0 Port = 0x8001
3 2022-07-15 19:46:04.698314467 28:ac:9e:ad:5c:b8  01:00:0c:cc:cc:cd STP 64 RST. Root =
32768/10/28:ac:9e:ad:5c:b7 Cost = 0 Port = 0x8001
4 2022-07-15 19:46:04.698386112 28:ac:9e:ad:5c:b8  01:00:0c:cc:cc:cd STP 64 RST. Root =
32768/20/28:ac:9e:ad:5c:b7 Cost = 0 Port = 0x8001
5 2022-07-15 19:46:04.698481274 28:ac:9e:ad:5c:b8  01:00:0c:cc:cc:cd STP 64 RST. Root =
32768/30/28:ac:9e:ad:5c:b7 Cost = 0 Port = 0x8001
6 2022-07-15 19:46:04.698555784 28:ac:9e:ad:5c:b8  01:00:0c:cc:cc:cd STP 64 RST. Root =
32768/40/28:ac:9e:ad:5c:b7 Cost = 0 Port = 0x8001
7 2022-07-15 19:46:04.698627624 28:ac:9e:ad:5c:b8  01:00:0c:cc:cc:cd STP 64 RST. Root =
32768/50/28:ac:9e:ad:5c:b7 Cost = 0 Port = 0x8001
```

次の出力は、Ethanalyzerでキャプチャできるメッセージの一部を示しています。デフォルトでは、Ethanalyzerは最大10パケットしかキャプチャしないことに注意してください。ただし、このコマンドを使用すると、パケットを無期限にキャプチャするようにCLIに要求できます。Ctrl+Cを使用してキャプチャモードを終了します。

```
Nexus9000_A(config-if-range)# ethanalyzer local interface inband limit-captured-frames 0
Capturing on inband
2020-02-18 01:43:30.542588 f8:b7:e2:49:2d:f2 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:43:30.542626 f8:b7:e2:49:2d:f5 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:43:30.542873 f8:b7:e2:49:2d:f4 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:43:30.542892 f8:b7:e2:49:2d:f3 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:43:31.596841 dc:f7:19:1b:f9:85 -> 01:80:c2:00:00:00 STP RST. Root =
32768/1/dc:f7:19:1b:f9:80  Cost = 0  Port = 0x8005
2020-02-18 01:43:31.661089 f8:b7:e2:49:2d:b2 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:43:31.661114 f8:b7:e2:49:2d:b3 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
```

```
2020-02-18 01:43:31.661324 f8:b7:e2:49:2d:b5 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:43:31.776638 cc:98:91:fc:55:8b -> 01:80:c2:00:00:00 STP RST. Root =
32768/1/cc:98:91:fc:55:80  Cost = 0  Port = 0x800b
2020-02-18 01:43:33.143814 f8:b7:e2:49:2d:b4 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:43:33.596810 dc:f7:19:1b:f9:85 -> 01:80:c2:00:00:00 STP RST. Root =
32768/1/dc:f7:19:1b:f9:80  Cost = 0  Port = 0x8005
2020-02-18 01:43:33.784099 cc:98:91:fc:55:8b -> 01:80:c2:00:00:00 STP RST. Root =
32768/1/cc:98:91:fc:55:80  Cost = 0  Port = 0x800b
2020-02-18 01:43:33.872280 f8:b7:e2:49:2d:f2 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:43:33.872504 f8:b7:e2:49:2d:f5 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
2020-02-18 01:43:33.872521 f8:b7:e2:49:2d:f4 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI
0x00000C (Cisco), PID 0x0134
15 packets captured
```

Ethanalyzerでフィルタを使用して、特定のトラフィックに焦点を当てることもできます。ethanalzyerで使用できるフィルタには、キャプチャフィルタと表示フィルタという2種類があります。キャプチャフィルタは、キャプチャフィルタで定義された基準に一致するトラフィックのみをキャプチャします。表示フィルタはすべてのトラフィックをキャプチャしますが、表示フィルタで定義された基準に一致するトラフィックだけが表示されます。

```
Nexus9000_B# ping 10.82.140.106 source 10.82.140.107 vrf management count 2
PING 10.82.140.106 (10.82.140.106) from 10.82.140.107: 56 data bytes
64 bytes from 10.82.140.106: icmp_seq=0 ttl=254 time=0.924 ms
64 bytes from 10.82.140.106: icmp_seq=1 ttl=254 time=0.558 ms

Nexus9000_A(config-if-range)# ethanalyzer local interface mgmt display-filter icmp
Capturing on mgmt0
2020-02-18 01:58:04.403295 10.82.140.107 -> 10.82.140.106 ICMP Echo (ping) request
2020-02-18 01:58:04.403688 10.82.140.106 -> 10.82.140.107 ICMP Echo (ping) reply
2020-02-18 01:58:04.404122 10.82.140.107 -> 10.82.140.106 ICMP Echo (ping) request
2020-02-18 01:58:04.404328 10.82.140.106 -> 10.82.140.107 ICMP Echo (ping) reply

4 packets captured
```

また、詳細オプションを使用してパケットをキャプチャし、Wiresharkと同様に端末で表示することもできます。これにより、パケットのDissector結果に基づいて完全なヘッダー情報を確認できます。たとえば、フレームが暗号化されている場合、暗号化されたペイロードは表示されません。例：

```
Nexus9000_A(config-if-range)# ethanalyzer local interface mgmt display-filter icmp detail
Capturing on mgmt0
Frame 2 (98 bytes on wire, 98 bytes captured)
    Arrival Time: Feb 18, 2020 02:02:17.569801000
    [Time delta from previous captured frame: 0.075295000 seconds]
    [Time delta from previous displayed frame: 0.075295000 seconds]
    [Time since reference or first frame: 0.075295000 seconds]
    Frame Number: 2
    Frame Length: 98 bytes
    Capture Length: 98 bytes
    [Frame is marked: False]
    [Protocols in frame: eth:ip:icmp:data]
Ethernet II, Src: 00:be:75:5b:de:00 (00:be:75:5b:de:00), Dst: 00:be:75:5b:d9:00
(00:be:75:5b:d9:00)
    Destination: 00:be:75:5b:d9:00 (00:be:75:5b:d9:00)
        Address: 00:be:75:5b:d9:00 (00:be:75:5b:d9:00)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
```

```
    Type: IP (0x0800)
>>>>>>>Output Clipped
```

Ethanalyzerを使用すると、次のことができます。

- 出力（PCAPファイル）をさまざまなターゲットファイルシステム上の指定したファイル名に書き込みます。ブートフラッシュ、ログフラッシュ、USBなど 保存したファイルをデバイスの外部に転送し、必要に応じてWiresharkで表示できます。
- ブートフラッシュからファイルを読み取り、端末に表示します。CPUインターフェイスから直接読み取るのと同様に、detailキーワードを使用すると、完全なパケット情報を表示することもできます。

さまざまなインターフェイスソースと出力オプションについては、この例を参照してください。

```
Nexus9000_A# ethanalyzer local interface mgmt capture-filter "host 10.82.140.107" write
bootflash:TEST.PCAP
Capturing on mgmt0
10
Nexus9000_A# dir bootflash:
       4096    Feb 11 02:59:04 2020  .rpmstore/
       4096    Feb 12 02:57:36 2020  .swtam/
       2783    Feb 17 21:59:49 2020  09b0b204-a292-4f77-b479-1ca1c4359d6f.config
       1738    Feb 17 21:53:50 2020  20200217_215345_poap_4168_init.log
       7169    Mar 01 04:41:55 2019  686114680.bin
       4411    Nov 15 15:07:17 2018  EBC-SC02-M2_303_running_config.txt
   13562165    Oct 26 06:15:35 2019  GBGBLD4SL01DRE0001-CZ07-
        590    Jan 10 14:21:08 2019  MDS20190110082155835.lic
       1164    Feb 18 02:18:15 2020  TEST.PCAP
>>>>>>>Output Clipped

Nexus9000_A# copy bootflash: ftp:
Enter source filename: TEST.PCAP
Enter vrf (If no input, current vrf 'default' is considered): management
Enter hostname for the ftp server: 10.122.153.158
Enter username: calo
Password:
***** Transfer of file Completed Successfully *****
Copy complete, now saving to disk (please wait)...
Copy complete.

Nexus9000_A# ethanalyzer local read bootflash:TEST.PCAP
2020-02-18 02:18:03.140167 10.82.140.107 -> 10.82.140.106 ICMP Echo (ping) request
2020-02-18 02:18:03.140563 10.82.140.106 -> 10.82.140.107 ICMP Echo (ping) reply
2020-02-18 02:18:15.663901 10.82.140.107 -> 10.82.140.106 ICMP Echo (ping) request
2020-02-18 02:18:15.664303 10.82.140.106 -> 10.82.140.107 ICMP Echo (ping) reply
2020-02-18 02:18:15.664763 10.82.140.107 -> 10.82.140.106 ICMP Echo (ping) request
2020-02-18 02:18:15.664975 10.82.140.106 -> 10.82.140.107 ICMP Echo (ping) reply
2020-02-18 02:18:15.665338 10.82.140.107 -> 10.82.140.106 ICMP Echo (ping) request
2020-02-18 02:18:15.665536 10.82.140.106 -> 10.82.140.107 ICMP Echo (ping) reply
2020-02-18 02:18:15.665864 10.82.140.107 -> 10.82.140.106 ICMP Echo (ping) request
2020-02-18 02:18:15.666066 10.82.140.106 -> 10.82.140.107 ICMP Echo (ping) reply

RTP-SUG-BGW-1# ethanalyzer local interface front-panel eth1-1 write bootflash:e1-1.pcap
Capturing on 'Eth1-1'
10

RTP-SUG-BGW-1# ethanalyzer local read bootflash:e1-1.pcap detail
Frame 1: 53 bytes on wire (424 bits), 53 bytes captured (424 bits) on interface Eth1-1, id 0
    Interface id: 0 (Eth1-1)
        Interface name: Eth1-1
    Encapsulation type: Ethernet (1)
```

```
Arrival Time: Jul 15, 2022 19:59:50.696219656 UTC
[Time shift for this packet: 0.000000000 seconds]
Epoch Time: 1657915190.696219656 seconds
[Time delta from previous captured frame: 0.000000000 seconds]
[Time delta from previous displayed frame: 0.000000000 seconds]
[Time since reference or first frame: 0.000000000 seconds]
Frame Number: 1
Frame Length: 53 bytes (424 bits)
Capture Length: 53 bytes (424 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: eth:llc:stp]
```

- **SPAN**

SPANはSwitchPort Analyzer（スイッチポートアナライザ）の略で、インターフェイスからのすべてのトラフィックをキャプチャし、そのトラフィックを宛先ポートにミラーリングするために使用されます。宛先ポートは通常、ネットワークアナライザツール（Wiresharkを実行しているPCなど）に接続し、これらのポートを通過するトラフィックを分析できるようにします。単一のポートからのトラフィック、または複数のポートとVLANからのトラフィックに対してSPANを実行できます。

SPANセッションには、送信元ポートと宛先ポートが含まれます。送信元ポートは、イーサネットポート（サブインターフェイスなし）、ポートチャネル、スーパーバイザのインバンドインターフェイスにすることができ、同時に宛先ポートにすることはできません。さらに、9300および9500プラットフォームなどの一部のデバイスでは、FEX（ファブリックエクステンダ）ポートもサポートされています。宛先ポートには、イーサネットポート（アクセスまたはトランク）、ポートチャネル（アクセスまたはトランク）を使用できます。9300アップリンクポートなどの一部のデバイスもサポートされますが、FEXポートは宛先をサポートしません。

複数のSPANセッションを入力/出力/両方に設定できます。個々のデバイスがサポートできるSPANセッションの総数には制限があります。たとえば、Nexus 9000では最大32のセッションをサポートできますが、Nexus 7000では16のセッションしかサポートできません。これをCLIで確認するか、使用する製品のSPAN設定ガイドを参照してください。

NX-OSリリースと製品タイプごとに、サポートされるインターフェイスのタイプと機能が異なることに注意してください。使用する製品とバージョンの最新の設定ガイドラインと制限事項を参照してください。Nexus 9000とNexus 7000のリンクは次のとおりです。

Cisco Nexus 9000シリーズNX-OSシステム管理設定ガイド、リリース9.3(x):SPANの設定[Cisco Nexus 9000シリーズスイッチ] : シスコ

Cisco Nexus 7000シリーズNX-OSシステム管理設定ガイド – SPANの設定[Cisco Nexus 7000シリーズスイッチ] – シスコ

SPANセッションにはさまざまなタイプがあります。一般的なタイプの一部を次に示します。

- ローカルSPAN：送信元ホストと宛先ホストの両方がスイッチに対してローカルであるSPANセッションのタイプ。つまり、SPANセッションの設定に必要なすべての設定は、単一のスイッチ、つまり送信元と宛先のホストポートが存在するスイッチに適用されます。
- リモートSPAN(RSPAN)：送信元ホストと宛先ホストがスイッチに対してローカルでないSPANセッションのタイプ。つまり、1つのスイッチに送信元RSPANセッションを設定し、宛先スイッチに宛先RSPANを設定して、RSPAN VLANを使用して接続を拡張します。

注：RSPANはNexusではサポートされていません

- 拡張リモートSPAN(ERSPAN):スイッチは、コピーされたフレームをGRE(Generic Routing Encapsulation)トンネルヘッダーでカプセル化し、パケットを設定された宛先にルーティングします。カプセル化スイッチとカプセル化解除スイッチ（2つの異なるデバイス）に送信元セッションと宛先セッションを設定します。 これにより、レイヤ3ネットワーク上でトラフィックをSPANできます。
- SPAN-to-CPU：宛先ポートがスーパーバイザまたはCPUである特別なタイプのSPANセッションに与えられる名前。これはローカルSPANセッションの形式であり、標準SPANセッションを使用できない場合に使用できます。一般的な原因には次のものがあります。使用可能または適切なSPAN宛先ポートがない、サイトにアクセスできない、またはサイトが管理されていない、SPAN宛先ポートに接続できるデバイスがない、などがあります。詳細については、次のリンクを参照してください。Nexus 9000 Cloud Scale ASIC NX-OS SPAN-to-CPU Procedure - Cisco。SPANからCPUへのレートはCoPP（コントロールプレーンポリシング）によって制限されるため、注意が必要です sniffing ポリサーを超える1つ以上の送信元インターフェイスでは、SPANからCPUへのセッションでドロップが発生する可能性があります。この場合、データは回線上にあるものの100 %反射されないため、SPANからCPUへの変換は、高いデータレートや断続的な損失を伴うシナリオのトラブルシューティングには必ずしも適切ではありません。SPANをCPUセッションに設定し、管理上これを有効にしたら、Ethanalyzerを実行して、CPUに送信されるトラフィックを確認し、それに応じて分析を実行する必要があります。

次に、Nexus 9000スイッチで簡単なローカルSPANセッションを設定する方法の例を示します。

```
Nexus9000_A(config-monitor)# monitor session ?

*** No matching command found in current mode, matching in (config) mode ***
  <1-32>
  all     All sessions

Nexus9000_A(config)# monitor session 10
Nexus9000_A(config-monitor)#?
  description  Session description (max 32 characters)
  destination  Destination configuration
  filter       Filter configuration
  mtu          Set the MTU size for SPAN packets
  no           Negate a command or set its defaults
  show         Show running system information
  shut         Shut a monitor session
  source       Source configuration
  end          Go to exec mode
  exit         Exit from command interpreter
  pop          Pop mode from stack or restore from name
  push         Push current mode to stack or save it under name
  where        Shows the cli context you are in

Nexus9000_A(config-monitor)# description Monitor_Port_e1/1
Nexus9000_A(config-monitor)# source interface ethernet 1/1
Nexus9000_A(config-monitor)# destination interface ethernet 1/10
Nexus9000_A(config-monitor)# no shut
```

次の例は、起動されたSPAN to CPUセッションの設定を示し、Ethanalyzerを使用してトラフィックをキャプチャします。

```
N9000-A#show run monitor

monitor session 1
source interface Ethernet1/7 rx
```

```
destination interface sup-eth0 << this is what sends the traffic to CPU
no shut

RTP-SUG-BGW-1# ethanalyzer local interface inband mirror limit-c 0
Capturing on 'ps-inb'
2020-02-18 02:18:03.140167 10.82.140.107 -> 10.82.140.106 ICMP Echo (ping) request
2020-02-18 02:18:15.663901 10.82.140.107 -> 10.82.140.106 ICMP Echo (ping) request
```

- **Dmirror**

Dmirrorは、BroadcomベースのNexusプラットフォーム用のSPAN-TO-CPUセッションの一種です。概念はSPAN-to-CPUと同じで、レートは50 pps（パケット/秒）に制限されています。この機能は、bcm-shell CLIを使用して内部データパスをデバッグするために実装されました。関連する制限事項があるため、ユーザがSupにSPANセッションを設定できるNX-OS CLIはありません。これは、制御トラフィックに影響を与え、CoPPクラスを消費する可能性があるためです。

- **ELAM**

ELAMはEmbedded Logic Analyzer Moduleの略です。ASICを調べ、単一の**パケット**に対してどのような転送が決定されるかを判断する機能を提供します。そのため、ELAMを使用すると、パケットがフォワーディングエンジンに到達するかどうか、およびどのポート/VLAN情報に到達するかを特定できます。また、L2 ~ L4のパケット構造と、パケットに変更が加えられたかどうかを確認できます。

ELAMはアーキテクチャに依存し、パケットをキャプチャする手順は内部アーキテクチャに基づいてプラットフォームごとに異なることを理解することが重要です。ツールを正しく適用するには、ハードウェアのASICマッピングを知っている必要があります。Nexus 7000では、1つのパケットに対して2つのキャプチャが取得されます。1つは**データパス(DBUS)**の決定前、もう1つは**結果パス(RBUS)**の決定後です。 DBUS情報を表示すると、パケットが受信された場所とレイヤ2 ~ 4の情報を確認できます。RBUSの結果には、パケットの転送先と、フレームが変更されたかどうかが表示されます。DBUSとRBUSのトリガーを設定し、準備が整っていることを確認してから、パケットをリアルタイムでキャプチャする必要があります。各種ラインカードの手順は次のとおりです。

さまざまなELAM手順の詳細については、次の表のリンクを参照してください。

ELAM の概要	ELAMの概要：シスコ
Nexus 7K F1モジュール	Nexus 7000 F1モジュールのELAM手順：シスコ
Nexus 7K F2モジュール	Nexus 7000 F2モジュールのELAM手順：シスコ
Nexus 7K F3モジュール	F3- ELAMの例
Nexus 7K Mモジュール	Nexus 7000 Mシリーズモジュールの ELAM手順：シスコ
Nexus 7K M1/M2およびF2モジュール	M1/M2、F2およびEthanalyzer向けNexus 7K ELAM
Nexus 7K M3モジュール	Nexus 7000 M3モジュールのELAM手順：シスコ

Nexus 7000向けELAM - M1/M2（Eurekaプラットフォーム）

- **show module**コマンドを使用して、モジュール番号を確認します。
- **attach module x**（xはモジュール番号）を使用してモジュールに接続します。
- **show hardware internal dev-port-map**コマンドを使用して内部ASICマッピングを確認し、

L2LKPとL3LKPを確認します。

```
Nexus7000(config)#show module
Mod  Ports  Module-Type                        Model              Status
---  -----  ---------------------------------- ------------------ ----------
1    0      Supervisor Module-2                N7K-SUP2E          active *
2    0      Supervisor Module-2                N7K-SUP2E          ha-standby
3    48     1/10 Gbps Ethernet Module          N7K-F248XP-25E     ok
4    24     10 Gbps Ethernet Module            N7K-M224XP-23L     ok


Nexus7000(config)# attach module 4
Attaching to module 4 ...
To exit type 'exit', to abort type '$.'
Last login: Fri Feb 14 18:10:21 UTC 2020 from 127.1.1.1 on pts/0

module-4# show hardware internal dev-port-map
--------------------------------------------------------------
CARD_TYPE:       24 port 10G
>Front Panel ports:24
--------------------------------------------------------------
 Device name             Dev role              Abbr num_inst:
--------------------------------------------------------------
> Skytrain               DEV_QUEUEING           QUEUE  4
> Valkyrie               DEV_REWRITE            RWR_0  4
> Eureka                 DEV_LAYER_2_LOOKUP     L2LKP  2
> Lamira                 DEV_LAYER_3_LOOKUP     L3LKP  2
> Garuda                 DEV_ETHERNET_MAC       MAC_0  2
> EDC                    DEV_PHY                PHYS   6
> Sacramento Xbar ASIC   DEV_SWITCH_FABRIC      SWICHF 1
+-----------------------------------------------------------------------+
+----------------+++FRONT PANEL PORT TO ASIC INSTANCE MAP+++------------+
+-----------------------------------------------------------------------+
FP port |  PHYS | SECUR | MAC_0 | RWR_0 | L2LKP | L3LKP | QUEUE |SWICHF
   1       0       0       0       0,1     0       0       0,1     0
   2       0       0       0       0,1     0       0       0,1     0
   3       0       0       0       0,1     0       0       0,1     0
   4       0       0       0       0,1     0       0       0,1     0
   5       1       0       0       0,1     0       0       0,1     0
   6       1       0       0       0,1     0       0       0,1     0
   7       1       0       0       0,1     0       0       0,1     0
   8       1       0       0       0,1     0       0       0,1     0
   9       2       0       0       0,1     0       0       0,1     0
   10      2       0       0       0,1     0       0       0,1     0
   11      2       0       0       0,1     0       0       0,1     0
   12      2       0       0       0,1     0       0       0,1     0
   13      3       1       1       2,3     1       1       2,3     0
   14      3       1       1       2,3     1       1       2,3     0
   15      3       1       1       2,3     1       1       2,3     0
   16      3       1       1       2,3     1       1       2,3     0
   17      4       1       1       2,3     1       1       2,3     0
   18      4       1       1       2,3     1       1       2,3     0
   19      4       1       1       2,3     1       1       2,3     0
   20      4       1       1       2,3     1       1       2,3     0
   21      5       1       1       2,3     1       1       2,3     0
   22      5       1       1       2,3     1       1       2,3     0
   23      5       1       1       2,3     1       1       2,3     0
   24      5       1       1       2,3     1       1       2,3     0
+-----------------------------------------------------------------------+
+-----------------------------------------------------------------------+
```

- まず、L2でパケットをキャプチャし、転送の決定が正しいかどうかを確認します。これを行

うには、L2LKPマッピングカラムを調べて、ポートに対応するASICインスタンス#を特定します。
- 次に、**elam asic eureka instance** xコマンドを使用して、このインスタンスでELAMを実行しますxはASICインスタンス番号で、DBUSとRBUSのトリガーを設定します。**status**コマンドを使用してトリガーのステータスを確認し、トリガーが設定されていることを確認します。

```
module-4(eureka-elam)# trigger dbus dbi ingress ipv4 if source-ipv4-address 192.0.2.2
destination-ipv4-address 192.0.2.4 rbi-corelate
module-4(eureka-elam)# trigger rbus rbi pb1 ip if cap2 1

module-4(eureka-elam)# status

Slot: 4, Instance: 1
EU-DBUS: Configured
trigger dbus dbi ingress ipv4 if source-ipv4-address 192.168.10.1
EU-RBUS: Configured
trigger rbus rbi pb1 ip if cap2 1
```

- **start**コマンドを使用してトリガーをアクティブ化し、**status**コマンドを使用してトリガーのステータスを確認し、トリガーが武装していることを確認します。

```
module-4(eureka-elam)# start
module-4(eureka-elam)# status

Slot: 4, Instance: 1 EU-DBUS: Armed <<<<<<<<<<
trigger dbus dbi ingress ipv4 if source-ipv4-address 192.168.10.1
EU-RBUS: Armed <<<<<<<<<<
trigger rbus rbi pb1 ip if cap2 1
```

- トリガーが武装していることをステータスが示すと、キャプチャの準備が整います。この時点で、トラフィックを送信し、トリガーが実際にトリガーされたかどうかを確認するためにステータスを再度チェックする必要があります。

```
module-4(eureka-elam)# status

Slot: 4, Instance: 1
EU-DBUS: Triggered    <<<<<<<<<<trigger dbus dbi ingress ipv4 if source-ipv4-address
192.168.10.1 EU-RBUS: Triggered <<<<<<<<<<
trigger rbus rbi pb1 ip if cap2 1
```

- トリガーされたら、rbusとdbusの両方のパケットシーケンス番号をチェックして、両方が同じパケットをキャプチャしたことを確認します。これは、**show dbus | i seq ;show rbus | i seq**。シーケンス番号が一致する場合、dbusとrbusの内容を表示できます。そうでない場合は、同じパケットをキャプチャできるようになるまで、キャプチャを再実行します。

  注：より正確に処理するには、ELAMを常に複数回実行して、転送の問題を確認します。

- rbusとdbusの内容は、**show dbus**コマンドと**show rbus**コマンドで表示できます。キャプチャで重要なのは、シーケンス番号と送信元/宛先インデックスです。Dbusは、パケットを受信したポートを示すソースインデックスを表示します。Rbusは、パケットの転送先ポートの宛先インデックスを示します。また、送信元と宛先のIP/MACアドレスおよびVLAN情報も確認できます。
- 送信元と宛先のインデックス（LTLインデックスとも呼ばれます）を使用して、**show system internal pixm info ltl #**コマンドで関連する前面パネルポートを確認できます。

Nexus 7000向けELAM - M1/M2（Lamiraプラットフォーム）

手順はLamiraプラットフォームでも同じですが、いくつかの違いがあります。

- キーワードLamira elam **asic** lamiraインスタンスx**を指定してELAMを実行します。**
- ELAMをトリガーするコマンドは次のとおりです。

```
module-4(lamira-elam)#trigger dbus ipv4 if source-ipv4-address 192.0.2.2 destination-ipv4-
address 192.0.2.4
module-4(lamira-elam)# trigger rbus
```

- **status**コマンドを使用してステータスをチェックし、トラフィックを送信する前に状態がArmedであり、トラフィックをキャプチャした後にトリガーされることを確認します。
- その後、dbusとshow busの出力を、Eurekaで示されているのと同じように解釈できます。

Nexus 7000向けELAM:F2/F2E(Clipper Platform)

ここでも、手順は似ていますが、トリガーのみが異なります。相違点は次のとおりです。

- キーワードClipper **elam asic clipper instance x**を指定してELAMを実行し、レイヤ2またはレイヤ3モードを指定します。

```
module-4# elam asic  clipper instance 1
module-4(clipper-elam)#
```

- ELAMをトリガーするコマンドは次のとおりです。

```
module-4(clipper-l2-elam)# trigger dbus ipv4 ingress if source-ipv4-address 192.0.2.3
destination-ipv4-address 192.0.2.2
module-4(clipper-l2-elam)# trigger rbus ingress if trig
```

- **status**コマンドを使用してステータスをチェックし、トラフィックを送信する前に状態がArmedであり、トラフィックをキャプチャした後にトリガーされることを確認します。
- その後、dbusとshow busの出力を、Eurekaで示されているのと同じように解釈できます。

Nexus 7000向けELAM - F3（フランカープラットフォーム）

ここでも、手順は似ていますが、トリガーのみが異なります。相違点は次のとおりです。

- キーワードFlanker **elam asic flanker instance x**を指定してELAMを実行し、レイヤ2またはレイヤ3モードを指定します。

```
module-4# elam asic flanker instance 1
module-4(flanker-elam)#
```

- ELAMをトリガーするコマンドは次のとおりです。

```
module-9(fln-l2-elam)# trigger dbus ipv4 if destination-ipv4-address 10.1.1.2
module-9(fln-l2-elam)# trigger rbus ingress if trig
```

- **status**コマンドを使用してステータスをチェックし、トラフィックを送信する前に状態がArmedであり、トラフィックをキャプチャした後にトリガーされることを確認します。
- その後、Eurekaに示すように、dbusとrbusの出力を同様に解釈できます。

Nexus 9000向けELAM（Tahoeプラットフォーム）

Nexus 9000では、手順はNexus 7000とは少し異なります。Nexus 9000については、「Nexus 9000 Cloud Scale ASIC (Tahoe) NX-OS ELAM - Cisco

- 最初に、**show hardware internal tah interface #**コマンドを使用して、インターフェイスのマッピングを確認します。この出力で最も重要な情報は、**ASIC #**、**Slice #**、および**source ID (srcid) #**です。
- また、**show system internal ethpm info interface #**コマンドを使用して、この情報を再確認することもできます**| i src**。ここで重要なことは、前述した値に加えて、dpidとdmodの値です。
- **show module**コマンドを使用して、モジュール番号を確認します。
- **attach module x**（xはモジュール番号）を使用してモジュールに接続します。
- **module-1# debug platform internal tah elam asic #**コマンドを使用して、モジュールでELAMを実行します。
- キャプチャするトラフィックの種類（L2、L3、GREやVXLANなどのカプセル化されたトラフィックなど）に基づいて、内側または外側のトリガーを設定します。

```
Nexus9000(config)# attach module 1
module-1# debug platform internal tah elam asic 0
module-1(TAH-elam)# trigger init asic # slice # lu-a2d 1 in-select 6 out-select 0 use-src-id #
module-1(TAH-elam-insel6)# reset
module-1(TAH-elam-insel6)# set outer ipv4 dst_ip 192.0.2.1 src_ip 192.0.2.2
```

- トリガーが設定されたら、**start**コマンドでELAMを開始し、トラフィックを送信し、**report**コマンドで出力を表示します。レポートの出力には、発信インターフェイスと着信インターフェイス、およびVLAN ID、送信元と宛先のIP/MACアドレスが表示されます。

```
SUGARBOWL ELAM REPORT SUMMARY
slot - 1, asic - 1, slice - 1
==============================


Incoming Interface: Eth1/49
Src Idx : 0xd, Src BD : 10
Outgoing Interface Info: dmod 1, dpid 14
Dst Idx : 0x602, Dst BD : 10

Packet Type: IPv4
Dst MAC address: CC:46:D6:6E:28:DB
Src MAC address: 00:FE:C8:0E:27:15
.1q Tag0 VLAN: 10,  cos = 0x0
Dst IPv4 address: 192.0.2.1
Src IPv4 address: 192.0.2.2
```

```
Ver     =  4, DSCP    =    0, Don't Fragment = 0 Proto   =  1, TTL     =   64, More Fragments =
0 Hdr len = 20, Pkt len =   84, Checksum       = 0x667f
```

Nexus 9000向けELAM（NorthStarプラットフォーム）

NorthStarプラットフォームの手順はTahoeプラットフォームと同じですが、唯一の違いは、ELAMモードに入るとキーワード**ns**が**tah**の代わりに使用されることです。

```
module-1#debug platform internal ns elam asic 0
```

- **N9K Packet Tracer**

Nexus 9000 packet tracerツールを使用してパケットのパスを追跡できます。フロー統計情報のカウンタが組み込まれているため、断続的または完全なトラフィック損失シナリオに役立ちます。これは、TCAMリソースが制限されているか、他のツールを実行できない場合に非常に役立ちます。さらに、このツールではARPトラフィックをキャプチャできず、Wiresharkなどのパケットコンテンツの詳細は表示されません。

パケットトレーサを設定するには、次のコマンドを使用します。

```
N9K-9508#test packet-tracer src_ip


               <==== provide your src and dst ip
N9K-9508# test packet-tracer start                                  <==== Start packet tracer
N9K-9508# test packet-tracer stop                                   <==== Stop packet tracer
N9K-9508# test packet-tracer show                                   <==== Check for packet
matches
```

詳細については、「Nexus 9000:Packet Tracerツールの説明 – シスコ

- **tracerouteとping**

これらのコマンドは、接続の問題を迅速に特定できる、最も有用な2つのコマンドです。

PingはInternet Control Message Protocol(ICMP)を使用して特定の宛先にICMPエコーメッセージを送信し、その宛先からのICMPエコー応答を待ちます。ホスト間のパスが問題なく正常に動作している場合は、応答が返され、pingが成功することを確認できます。デフォルトでは、pingコマンドは5x ICMPエコーメッセージ（両方向で同じサイズ）を送信します。すべてが正常に動作している場合は、5x ICMPエコー応答を確認できます。Address Resolution Protocol（ARP；アドレス解決プロトコル）要求中にスイッチがMACアドレスを学習すると、初期エコー要求が失敗することがあります。その後すぐにpingを再実行すると、最初のping損失はありません。さらに、次のキーワードを使用して、pingの数、パケットサイズ、送信元、送信元インターフェイス、およびタイムアウト間隔を設定することもできます。

```
F241.04.25-N9K-C93180-1# ping 10.82.139.39 vrf management
PING 10.82.139.39 (10.82.139.39): 56 data bytes
36 bytes from 10.82.139.38: Destination Host Unreachable
Request 0 timed out
64 bytes from 10.82.139.39: icmp_seq=1 ttl=254 time=23.714 ms
64 bytes from 10.82.139.39: icmp_seq=2 ttl=254 time=0.622 ms
64 bytes from 10.82.139.39: icmp_seq=3 ttl=254 time=0.55 ms
64 bytes from 10.82.139.39: icmp_seq=4 ttl=254 time=0.598 ms


F241.04.25-N9K-C93180-1# ping 10.82.139.39 ?
  <CR>
```

```
  count             Number of pings to send
  df-bit            Enable do not fragment bit in IP header
  interval          Wait interval seconds between sending each packet
  packet-size       Packet size to send
  source            Source IP address to use
  source-interface  Select source interface
  timeout           Specify timeout interval
  vrf               Display per-VRF information
```

tracerouteは、パケットが宛先に到達する前に、パケットがとるさまざまなホップを識別するために使用されます。これは、障害が発生しているL3境界を特定するのに役立つため、非常に重要なツールです。次のキーワードを使用して、ポート、送信元、および送信元インターフェイスを使用することもできます。

```
F241.04.25-N9K-C93180-1# traceroute 10.82.139.39 ?
  <CR>
  port              Set destination port
  source            Set source address in IP header
  source-interface  Select source interface
  vrf               Display per-VRF information
```

```
Nexus_1(config)# traceroute 192.0.2.1
traceroute to 192.0.2.1 (192.0.2.1), 30 hops max, 40 byte packets
 1 198.51.100.3 (198.51.100.3)  1.017 ms  0.655 ms  0.648 ms
 2 203.0.113.2 (203.0.113.2)  0.826 ms  0.898 ms  0.82 ms
 3 192.0.2.1 (192.0.2.1)  0.962 ms  0.765 ms  0.776 ms
```

- PACL/RACL/VACL

ACLはAccess Control Listの略です。関連する定義済みの基準に基づいてトラフィックをフィルタリングできる重要なツールです。ACLに一致基準のエントリが入力されると、着信トラフィックまたは発信トラフィックのいずれかをキャプチャするために適用できます。ACLの重要な側面は、フロー統計情報のカウンタを提供する機能です。PACL/RACL/VACLという用語は、これらのACLのさまざまな実装を指し、特に断続的なトラフィック損失に対して、ACLを強力なトラブルシューティングツールとして使用できます。ここでは、これらの用語について簡単に説明します。

- PACLはPort Access Control Listの略です。L2スイッチポート/インターフェイスにアクセスリストを適用すると、そのアクセスリストはPACLと呼ばれます。
- RACLはRouter Access Control Listの略です。アクセスリストをL3ルーテッドポート/インターフェイスに適用すると、そのアクセスリストはRACLと呼ばれます。
- VACLはVLAN Access Control Listの略です。VACLは、VLANに出入りするパケットや、VLAN内でブリッジされるすべてのパケットに適用するように設定できます。VACLは、セキュリティパケットフィルタ専用であり、トラフィックを特定の物理インターフェイスにリダイレクトします。VACLは方向（入力または出力）によって定義されません。

次の表に、ACLのバージョン間の比較を示します。

ACL タイプ	PACL	RACL	VACL
機能	L2インターフェイスで受信したトラフィックをフィルタリングします。	L3インターフェイスで受信したトラフィックをフィルタリングする	vLANトラフィックのフィルタリング
適用先	- L2インターフェイス/ポート。 - L2ポートチャネルインターフェイス。 – トランクポートに適用すると	-VLAN インターフェイス. – 物理L3インターフェイス。 - L3サブインターフェイス。 - L3ポートチャネルインターフェイス。	イネーブルにすると、ACのVLANのすべてのポートランクポートを含む）に適されます。

	、ACLはそのトランクポートで許可されているすべてのVLANのトラフィックをフィルタリングします。	– 管理インターフェイス。	
適用方向	インバウンドのみ。	着信または発信	-

アクセスリストの設定方法の例を次に示します。詳細については、『Cisco Nexus 9000シリーズNX-OSセキュリティコンフィギュレーションガイド、リリース9.3(x):IP ACLの設定[Cisco Nexus 9000シリーズスイッチ] - Cisco

```
Nexus93180(config)# ip access-list




Nexus93180(config-acl)# ?
  <1-4294967295>  Sequence number
  deny            Specify packets to reject
  fragments       Optimize fragments rule installation
  no              Negate a command or set its defaults
  permit          Specify packets to forward
  remark          Access list entry comment
  show            Show running system information
  statistics      Enable per-entry statistics for the ACL
  end             Go to exec mode
  exit            Exit from command interpreter
  pop             Pop mode from stack or restore from name
  push            Push current mode to stack or save it under name
  where           Shows the cli context you are in

Nexus93180(config)# int e1/1

Nexus93180(config-if)# ip port access-group



         >>>>>> When you configure ACL like this, it is PACL.
in  Inbound packets
Nexus93180(config-if)# ip access-group



              >>>>>> When you configure ACL like this, it is RACL.
  in   Inbound packets
  out  Outbound packets
```

- LOGFLASH

LogFlashは、Nexusプラットフォームで外部コンパクトフラッシュ、USBデバイス、またはスーパーバイザの内蔵ディスクとして使用できるパーシステントストレージの一種です。スイッチから取り外すと、LogFlashが見つからないことをユーザに定期的に通知します。Logflashはスーパーバイザにインストールされ、アカウンティングログ、syslogメッセージ、デバッグ、Embedded Event Manager(EEM)出力などの履歴データを保持します。EEMについては、この記事の後半で説明します。次のコマンドを使用して、LogFlashの内容を確認できます。

```
Nexus93180(config)# dir logflash:
          0    Nov 14 04:13:21 2019  .gmr6_plus
```

```
      20480    Feb 18 13:35:07 2020  ISSU_debug_logs/
         24    Feb 20 20:43:24 2019  arp.pcap
         24    Feb 20 20:36:52 2019  capture_SYB010L2289.pcap
       4096    Feb 18 17:24:53 2020  command/
       4096    Sep 11 01:39:04 2018  controller/
       4096    Aug 15 03:28:05 2019  core/
       4096    Feb 02 05:21:47 2018  debug/
    1323008    Feb 18 19:20:46 2020  debug_logs/
       4096    Feb 17 06:35:36 2020  evt_log_snapshot/
       4096    Feb 02 05:21:47 2018  generic/
       1024    Oct 30 17:27:49 2019  icamsql_1_1.db
      32768    Jan 17 11:53:23 2020  icamsql_1_1.db-shm
     129984    Jan 17 11:53:23 2020  icamsql_1_1.db-wal
       4096    Feb 14 13:44:00 2020  log/
      16384    Feb 02 05:21:44 2018  lost+found/
       4096    Aug 09 20:38:22 2019  old_upgrade/
       4096    Feb 18 13:40:36 2020  vdc_1/

Usage for logflash://sup-local
 1103396864 bytes used
 7217504256 bytes free
 8320901120 bytes total
```

ユーザがデバイスをリロードした場合、またはイベントが原因でデバイスが突然勝手にリロードした場合、ログ情報はすべて失われます。このようなシナリオでは、LogFlashは履歴データを提供します。履歴データは、問題の原因を特定するために確認できます。もちろん、このイベントが再び発生した場合に何を探すべきかのヒントを提供する根本原因を特定するには、さらなるデューデリジェンスが必要です。

デバイスにlogflashをインストールする方法については、「Nexus 7000のロギング機能：シスコ」リンクを参照してください。

- **OBFL**

OBFLはOnBoard Failure Loggingの略です。Nexusトップオブラックスイッチとモジュラスイッチの両方で使用できる永続的なストレージのタイプです。LogFlashと同様に、デバイスがリロードされると情報が保持されます。OBFLは、障害や環境データなどの情報を保存します。情報はプラットフォームやモジュールごとに異なりますが、Nexus 93108プラットフォームのモジュール1（つまり、1つのモジュールのみを備えた固定シャーシ）の出力例を次に示します。

```
Nexus93180(config)# show logging onboard module 1 ?
*** No matching command found in current mode, matching in (exec) mode ***
  <CR>
  >                     Redirect it to a file
  >>                    Redirect it to a file in append mode
  boot-uptime           Boot-uptime
  card-boot-history     Show card boot history
  card-first-power-on   Show card first power on information
  counter-stats         Show OBFL counter statistics
  device-version        Device-version
  endtime               Show OBFL logs till end time mm/dd/yy-HH:MM:SS
  environmental-history Environmental-history
  error-stats           Show OBFL error statistics
  exception-log         Exception-log
  internal              Show Logging Onboard Internal
  interrupt-stats       Interrupt-stats
  obfl-history          Obfl-history
  stack-trace           Stack-trace
  starttime             Show OBFL logs from start time mm/dd/yy-HH:MM:SS
  status                Status
  |                     Pipe command output to filter
```

```
Nexus93180(config)# show logging onboard module 1 status
----------------------------
OBFL Status
----------------------------
    Switch OBFL Log:                                        Enabled
    Module:  1 OBFL Log:                                    Enabled
    card-boot-history                                       Enabled
    card-first-power-on                                     Enabled
    cpu-hog                                                 Enabled
    environmental-history                                   Enabled
    error-stats                                             Enabled
    exception-log                                           Enabled
    interrupt-stats                                         Enabled
    mem-leak                                                Enabled
    miscellaneous-error                                     Enabled
    obfl-log (boot-uptime/device-version/obfl-history)  Enabled
    register-log                                            Enabled
    system-health                                           Enabled
    temp Error                                              Enabled
    stack-trace                                             Enabled
```

この情報は、ユーザが意図的にデバイスをリロードした場合や、リロードをトリガーしたイベントが原因でデバイスがリロードされた場合に役立ちます。この場合、OBFL情報は、ラインカードの観点から何が問題となったのかを特定するのに役立ちます。show logging onboardコマンドを使用すると、作業を開始できます。必要なものすべてを取得するには、モジュールコンテキスト内からキャプチャする必要があることに注意してください。show logging onboard module xまたはattach mod xを使用していることを確認します。show logging onboard

- **イベント履歴**

イベント履歴は、Nexus上で実行されるプロセスで発生するさまざまなイベントに関する情報を提供できる強力なツールの1つです。つまり、Nexusプラットフォーム上で実行されるすべてのプロセスは、バックグラウンドで実行されるイベント履歴を持ち、そのプロセスのさまざまなイベントに関する情報を保存します（これらは常に実行されるデバッグと考えてください）。 これらのイベント履歴は永続的ではなく、保存されているすべての情報はデバイスのリロード時に失われます。これらは、特定のプロセスに関する問題を特定し、そのプロセスのトラブルシューティングを行う場合に非常に便利です。たとえば、OSPFルーティングプロトコルが正しく動作しない場合、OSPFに関連付けられたイベント履歴を使用して、OSPFプロセスが失敗した場所を特定できます。CDP/STP、UDLD、LACP/OSPF、EIGRP/BGPなど、Nexusプラットフォームのほぼすべてのプロセスに関連するイベント履歴を見つけることができます。

これは、通常、参照例を使用してプロセスのイベント履歴を確認する方法です。すべてのプロセスには複数のオプションがあるため、?プロセスの下で使用可能なさまざまなオプションを確認する。

```
Nexus93180(config)# show
```

```
Nexus93180# show ip ospf event-history ?
  adjacency       Adjacency formation logs
  cli             Cli logs
  event           Internal event logs
  flooding        LSA flooding logs
  ha              HA and GR logs
```

```
  hello           Hello related logs
  ldp             LDP related logs
  lsa             LSA generation and databse logs
  msgs            IPC logs
  objstore        DME OBJSTORE related logs
  redistribution  Redistribution logs
  rib             RIB related logs
  segrt           Segment Routing logs
  spf             SPF calculation logs
  spf-trigger     SPF TRIGGER related logs
  statistics      Show the state and size of the buffers
  te              MPLS TE related logs

Nexus93180# show spanning-tree internal event-history ?
  all      Show all event historys
  deleted  Show event history of deleted trees and ports
  errors   Show error logs of STP
  msgs     Show various message logs of STP
  tree     Show spanning tree instance info
  vpc      Show virtual Port-channel event logs
```

- **デバッグ**

デバッグはNX-OSの強力なツールで、リアルタイムのトラブルシューティングイベントを実行し、ファイルに記録したり、CLIで表示したりできます。デバッグ出力はCPUのパフォーマンスに影響を与えるため、ファイルに記録することを強く推奨します。CLIでデバッグを直接実行する前には注意が必要です。

通常、デバッグが実行されるのは、問題が単一のプロセスであることが判明し、ネットワーク内の実際のトラフィックに対して、このプロセスがどのようにリアルタイムで動作するかを確認する場合だけです。定義したユーザアカウント権限に基づいて、デバッグ機能を有効にする必要があります。

イベント履歴と同様に、CDP/STP、UDLD、LACP/OSPF、EIGRP/BGPなどのNexusデバイスのすべてのプロセスに対してデバッグを実行できます。

通常、プロセスのデバッグは次のように実行します。すべてのプロセスには複数のオプションがあるため、?プロセスの下で使用可能なさまざまなオプションを確認する。

```
Nexus93180# debug
```

```
Nexus93180# debug spanning-tree ?
  all      Configure all debug flags of stp
  bpdu_rx  Configure debugging of stp bpdu rx
  bpdu_tx  Configure debugging of stp bpdu tx
  error    Configure debugging of stp error
  event    Configure debugging of Events
  ha       Configure debugging of stp HA
  mcs      Configure debugging of stp MCS
  mstp     Configure debugging of MSTP
  pss      Configure debugging of PSS
  rstp     Configure debugging of RSTP
  sps      Configure debugging of Set Port state batching
  timer    Configure debugging of stp Timer events
  trace    Configure debugging of stp trace
```

```
  warning  Configure debugging of stp warning

Nexus93180# debug ip ospf ?
  adjacency                Adjacency events
  all                      All OSPF debugging
  database                 OSPF LSDB changes
  database-timers          OSPF LSDB timers
  events                   OSPF related events
  flooding                 LSA flooding
  graceful-restart         OSPF graceful restart related debugs
  ha                       OSPF HA related events
  hello                    Hello packets and DR elections
  lsa-generation           Local OSPF LSA generation
  lsa-throttling           Local OSPF LSA throttling
  mpls                     OSPF MPLS
  objectstore              Objectstore Events
  packets                  OSPF packets
  policy                   OSPF RPM policy debug information
  redist                   OSPF redistribution
  retransmission           OSPF retransmission events
  rib                      Sending routes to the URIB
  segrt                    Segment Routing Events
  snmp                     SNMP traps and request-response related events
  spf                      SPF calculations
  spf-trigger              Show SPF triggers
```

- **ゴールド**

GOLDはGeneric OnLine診断を表します。名前が示すように、これらのテストは通常システムのヘルスチェックとして使用され、問題のハードウェアのチェックまたは確認に使用されます。さまざまなオンラインテストが実行され、使用中のプラットフォームに基づいて実行されます。これらのテストの中には、中断するものもあれば、中断しないものもあります。これらのオンラインテストは、次のように分類できます。

- **ブートアップ診断:**これらのテストは、デバイスの起動時に実行されるテストです。また、すべてのASICのデータとコントロールプレーン間の接続を含む、スーパーバイザとモジュール間の接続も確認します。ManagementPortLoopbackやEOBCLoopbackなどのテストは中断しますが、OBFLおよびUSBのテストは中断しません。
- **ランタイムまたはヘルスモニタリング診断**：これらのテストでは、デバイスの状態に関する情報が得られます。これらのテストは無停止で実行され、ハードウェアの安定性を確保するためにバックグラウンドで実行されます。必要に応じて、またはトラブルシューティングの目的で、これらのテストを有効または無効にできます。
- **オンデマンド診断**：問題をローカライズするために、記載されているすべてのテストをオンデマンドで再実行できます。

次のコマンドを使用して、スイッチで使用可能なさまざまなタイプのオンラインテストを確認できます。

```
Nexus93180(config)# show diagnostic content module all
Diagnostics test suite attributes:
B/C/* - Bypass bootup level test / Complete bootup level test / NA
P/*   - Per port test / NA
M/S/* - Only applicable to active / standby unit / NA
D/N/* - Disruptive test / Non-disruptive test / NA
H/O/* - Always enabled monitoring test / Conditionally enabled test / NA
F/*   - Fixed monitoring interval test / NA
X/*   - Not a health monitoring test / NA
E/*   - Sup to line card test / NA
L/*   - Exclusively run this test / NA
```

```
T/*   - Not an ondemand test / NA
A/I/* - Monitoring is active / Monitoring is inactive / NA

Module 1: 48x10/25G + 6x40/100G Ethernet Module (Active)

                                                    Testing Interval
ID     Name                                Attributes      (hh:mm:ss)
____   ___________________________________ ____________    _____________
 1)    USB--------------------------->     C**N**X**T*      -NA-
 2)    NVRAM------------------------->     ***N******A      00:05:00
 3)    RealTimeClock----------------->     ***N******A      00:05:00
 4)    PrimaryBootROM---------------->     ***N******A      00:30:00
 5)    SecondaryBootROM-------------->     ***N******A      00:30:00
 6)    BootFlash--------------------->     ***N******A      00:30:00
 7)    SystemMgmtBus----------------->     **MN******A      00:00:30
 8)    OBFL-------------------------->     C**N**X**T*      -NA-
 9)    ACT2-------------------------->     ***N******A      00:30:00
10)    Console----------------------->     ***N******A      00:00:30
11)    FpgaRegTest------------------->     ***N******A      00:00:30
12)    Mce--------------------------->     ***N******A      01:00:00
13)    AsicMemory-------------------->     C**D**X**T*      -NA-
14)    Pcie-------------------------->     C**N**X**T*      -NA-
15)    PortLoopback------------------>     *P*N**XE***      -NA-
16)    L2ACLRedirect----------------->     *P*N***E**A      00:01:00
17)    BootupPortLoopback------------>     CP*N**XE*T*      -NA-
```

上記の17の各テストの動作を表示するには、次のコマンドを使用します。

```
Nexus93180(config)#show diagnostic description module 1 test all
USB :
        A bootup test that checks the USB controller initialization
        on the module.

NVRAM :
        A health monitoring test, enabled by default that checks the
        sanity of the NVRAM device on the module.

RealTimeClock :
        A health monitoring test, enabled by default that verifies
        the real time clock on the module.

PrimaryBootROM :
        A health monitoring  test that verifies the primary BootROM
        on the module.

SecondaryBootROM :
        A health monitoring  test that verifies the secondary BootROM
        on the module.

BootFlash :
        A Health monitoring test, enabled by default, that verifies
        access to the internal compactflash devices.

SystemMgmtBus :
        A Health monitoring test, enabled by default, that verifies
        the standby System Bus.

OBFL :
        A bootup test that checks the onboard flash used for failure
        logging (OBFL) device initialization on the module.

ACT2 :
```

```
        A Health monitoring test, enabled by default, that verifies
        access to the ACT2 device.

Console :
        A health monitoring test,enabled by default that checks health
        of console device.

FpgaRegTest :
        A health monitoring test,enabled by default that checks read/write
        access to FPGA scratch registers on the module.

Mce :
        A Health monitoring test, enabled by default, that check for
        machine errors on sup.

AsicMemory :
        A bootup test that checks the asic memory.

Pcie :
        A bootup test that tests pcie bus of the module

PortLoopback :
        A health monitoring test that tests the packet path from
        the Supervisor card to the physical port in ADMIN DOWN state
        on Linecards.

L2ACLRedirect :
        A health monitoring test, enabled by default, that does a
        non disruptive loopback for TAHOE asics to check the ACL Sup
        redirect with the CPU port.

BootupPortLoopback :
        A Bootup test that tests the packet path from the Supervisor
        card to all of the physical ports at boot time.
```

- **EEM**

EEMはEmbedded Event Managerの略です。特定のイベントが発生した場合に特定のタスクを実行するようにデバイスをプログラムできる強力なツールです。デバイス上のさまざまなイベントを監視し、問題をトラブルシューティングして回復するために必要なアクションを実行します。EEMは3つの主要なコンポーネントで構成されており、それぞれについて簡単に説明します。

- **イベントステートメント**：これらのイベントを監視し、Nexusに特定のアクション（回避策の実行、SNMPサーバへの通知、CLIログの表示など）を実行させる必要があります。
- **アクションステートメント**：これらは、イベントがトリガーされたときにEEMが実行する手順です。これらのアクションは、単にインターフェイスを無効にしたり、いくつかのshowコマンドを実行して出力をftpサーバ上のファイルにコピーしたり、電子メールを送信したりすることなどです。
- **ポリシー**:基本的には、CLIまたはbashスクリプトを使用してスーパーバイザに設定できる1つ以上のaction文と組み合わせたイベントです。Pythonスクリプトを使用してEEMを起動することもできます。スーパーバイザでポリシーが定義されると、そのポリシーは該当するモジュールにプッシュされます。

EEMの詳細については、『Cisco Nexus 9000 Series NX-OS System Management Configuration Guide, Release 9.2(x) - Configuring the Embedded Event Manager [Cisco Nexus 9000 Series Switches] - Cisco』を参照してください。

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。