# Fehlerbehebung: Nexus Kurzreferenz für Anfänger

## Inhalt



## Einleitung

In diesem Dokument werden verschiedene Tools zur Fehlerbehebung bei Nexus-Produkten beschrieben, mit denen Sie Probleme diagnostizieren und beheben können.

## Überblick

Es ist wichtig zu wissen, welche Tools verfügbar sind und in welchem Szenario Sie sie für einen maximalen Gewinn verwenden würden. In der Tat ist manchmal ein bestimmtes Werkzeug nicht machbar, einfach weil es entworfen ist, um an etwas anderem zu arbeiten.

In dieser Tabelle sind die verschiedenen Tools für die Fehlerbehebung auf der Nexus-Plattform und ihre Funktionen zusammengefasst. Details und CLI-Beispiele finden Sie im Abschnitt "Nexus-Tools".

| TOOLS | FUNKTION | BEISPIELE FÜR ANWENDUNGSFÄLLE | PRO | KONTRA | PERSISTENZ | EFFEKTIERTE EBENE | VERWENDETE BEFEHLE |
|---|---|---|---|---|---|---|---|
| Ethanalyzer | Erfassung von Datenverkehr zur oder von der CPU | Probleme mit Verkehrsflaute, Latenz und Staus | Hervorragend bei Problemen mit Langsamkeit, | Erkennt in der Regel nur Steuerungsebenen-Datenverkehr, Übertragungsra | – | Steuern Sie Fläche. In einigen | Integrierte lokale #ethanalyzer Schnittstelle #ethanalyzer lok Schnittstelle [Schnittstellen-I |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | Überlastung und Latenz | te begrenzt | | Szenarien für Datenebene verwendbar (SPAN zu CPU) | Anzeigefilter [WORD] beispiel: #ethanalyzer lok Schnittstelle Eth 6/4 Anzeigefilter ICMP |
| SPAN | Erfassen und Spiegeln einer Reihe von Paketen | Fehlgeschlagen ping s, fehlerhafte Pakete usw. | Hervorragend für unregelmäßigen Datenverlust | Externes Gerät mit Sniffer-Software erforderlich Erfordert TCAM-Ressourcen | SPAN-Sitzung muss konfiguriert und aktiviert/deaktiviert werden. | Steuerung + Daten | #monitor sessio #description [NA #source interfac [Port-ID] #destir interface [Port-I #no shut |
| DMirror | Erfassung von Datenverkehr zur oder von der CPU nur für Broadcom Nexus-Geräte | Probleme mit Verkehrsflaute, Latenz und Staus | Hervorragend bei Problemen mit Langsamkeit, Überlastung und Latenz | Nur für Broadcom Nexus-Geräte. Durchsatzbegrenzung (CloudScale Nexus 9000 verfügt über SPAN-zu-CPU) | – | Steuern Sie Fläche. Kann in einigen Szenarien für die Datenebene verwendet werden | Variiert je nach Plattform, siehe ELAM im Überb Cisco |
| ELAM | Erfasst ein einzelnes Paket, das den Nexus-Switch erreicht (oder verlässt, wenn Nexus 7000) | Überprüfen, ob das Paket den Nexus erreicht, Weiterleitungsentscheidungen, Änderungen am Paket, Schnittstelle/VLAN des Pakets usw. | Hervorragend für Paketfluss- und Weiterleitungsprobleme geeignet. Non-intrusive | Erfordert umfassende Kenntnisse der Hardware. Verwendet spezielle Trigger-Mechanismen, die architekturspezifisch sind. Nur nützlich, wenn Sie wissen, welchen Verkehr Sie überprüfen möchten | – | Steuerung + Daten | # Attach-Modul [MODULE NUM # debug platform internal <> |
| Nexus 9000 | Pfad des Pakets | Verbindungsprobleme und | Stellt einen Zähler für | ARP-Datenverkehr | – | Daten + | # test packet-tra src_IP [SOURC |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Packet Tracer | erkennen | Paketverluste | Flussstatistiken bereit, der für einen vorübergehenden/vollständigen Verlust nützlich ist. Perfekt für Line Cards ohne TCAM-Schnitzereien | kann nicht erfasst werden. Nur für Nexus 9000 geeignet | | Steuerung | dst_IP [DESTINATION test packet-trace start # test pack tracer stop # tes packet-tracer sh |
| Routenverfolgung | Pfad des Pakets in Bezug auf L3-Hops erkennen | Fehlgeschlagene Pings, Host/Ziel/Internet nicht erreichbar usw. | Erkennt die verschiedenen Hops im Pfad, um L3-Fehler zu isolieren. | Identifiziert nur Bereiche, in denen die L3-Grenze überschritten wird (identifiziert nicht das Problem selbst) | – | Daten + Steuerung | # traceroute [ZIE IP] Die Argumente umfassen: Port, Portnumm Quelle, Schnitts VRF, Quellschnittstell |
| Ping | Testen der Verbindung zwischen zwei Punkten im Netzwerk | Erreichbarkeit zwischen Geräten testen | Schnelles und einfaches Tool zum Testen der Konnektivität | Identifiziert nur, ob der Host erreichbar ist oder nicht | – | Daten + Steuerung | # ping [ZIEL-IP] Die Argumente umfassen: Anzahl, Paketgr Quellschnittstell Intervall, Multica Loopback, Time |
| PACL/RACL/VACL | Erfassung des ein- und ausgehenden Datenverkehrs eines bestimmten Ports oder VLANs | Intermittierender Paketverlust zwischen Hosts, Überprüfung, ob Pakete beim Nexus eintreffen/abgehen usw. | Hervorragend für unregelmäßigen Datenverlust | Erfordert TCAM-Ressourcen. Bei einigen Modulen ist eine manuelle TCAM-Partitionierung erforderlich. | Persistent (angewendet auf running-konfiguration) | Daten + Steuerung | # ip access-list [ NAME] # ip port access-group [A NAME] # ip acc group [ACL NAM Die Argumente umfassen: Verweigern, Fragmente, Nei Zulassen, Bemerkung, Anzeigen, Statis Beenden, Been Pop, Push, Wo |
| LogFlash | Globale Speicherung von Verlaufsdaten für den Switch, | Plötzliches Neuladen/Herunterfahren von Geräten. Bei jedem Neuladen eines Geräts liefern Log-Flash- | Informationen bleiben beim erneuten Laden des Geräts erhalten | Extern auf Nexus 7000 = muss auf der Supervisor-Plattform installiert/integriert sein, damit | Dauerhaft neu laden | Daten + Steuerung | # dir logflash: |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | z. B. Protokollkonten, Absturzdateien und Ereignisse, unabhängig vom erneuten Laden des Geräts | Daten Informationen, die für die Analyse hilfreich sein können. | (persistenter Speicher) | diese Protokolle erfasst werden können (con gilt nicht für 3K/9K, da Logflash eine Partition des internen Speichergeräts ist) | | | |
| OBFL | Speicherung von Verlaufsdaten für ein bestimmtes Modul, z. B. Ausfall- und Umgebungsdaten | Plötzliches Neuladen/Herunterfahren von Geräten. Bei jedem Neuladen eines Geräts liefern Log-Flash-Daten hilfreiche Informationen. | Informationen bleiben beim erneuten Laden des Geräts erhalten (persistenter Speicher) | Unterstützt eine begrenzte Anzahl von Lese- und Schreibvorgängen | Dauerhaft neu laden | Daten + Steuerung | # show logging onboard module Die Argumente umfassen: Boot-Uptime, Ca Boot-History, Ca first-power-on, Counterstats, Geräteversion, Endzeit, Umgeb History, Error-St Exception-Log, intern, Interrupt- obfl-history, stat trace, Startzeit, Status |
| Ereignisverlauf | Wenn Sie Informationen für einen bestimmten Prozess benötigen, der derzeit ausgeführt wird | Jeder Prozess im Nexus verfügt über einen eigenen Ereignisverlauf, z. B. CDP, STP, OSPF, EIGRP, BGP, vPC, LACP usw. | Fehlerbehebung bei einem bestimmten Prozess auf Nexus | Informationen gehen verloren, wenn das Gerät neu geladen wird (nicht permanent) | Nicht permanent | Daten + Steuerung | # show [PROCE internal event-hi [ARGUMENT] Die Argumente umfassen: Adjacency, CLI, Event, Flooding hello, ldp, lsa, m objstore, Redistribution, r segrt, spf, spf-tr statistics, te |
| Fehlerbehebung | Wenn Sie für einen bestimmten Prozess detailliertere Echtzeit-/Live- | Für jeden Prozess im Nexus kann eine Fehlersuche durchgeführt werden, z. B. für CDP, STP, OSPF, IGRP, BGP, vPC, LACP usw. | Fehlerbehebung bei einem bestimmten Prozess auf Nexus in Echtzeit für eine detailliertere Analyse | Beeinträchtigung der Netzwerkleistung | Nicht permanent | Daten + Steuerung | # Debug-Prozes [PROCESS] beispiel: # debug ip ospf |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | Informationen benötigen | | | | | | |
| GOLD | Bietet Systemstart-, Laufzeit- und On-Demand-Diagnosen für Hardwarekomponenten (z. B. E/A- und Supervisor-Module) | Hardware wie USB, Bootflash, OBFL, ASIC-Speicher, PCIE, Port-Loopback, NVRAM usw. testen | Erkennen von Fehlern in der Hardware und Durchführen erforderlicher Korrekturmaßnahmen erst ab Version 6(2)8 | Nur Hardwareprobleme erkennen | Nicht permanent | – | # show diagnost content module show diagnostic description mod [#] all testen |
| EEM | Überwachen von Ereignissen auf dem Gerät und Durchführen der erforderlichen Maßnahmen | Alle Geräteaktivitäten, die eine Aktion/Problemumgehung/Benachrichtigung erfordern, z. B. Herunterfahren der Schnittstelle, Fehlfunktion des Lüfters, CPU-Nutzung usw. | Unterstützt Python-Skripte | Für die Konfiguration von EEM sind Netzwerkadministratorberechtigungen erforderlich. | EEM-Skript und -Trigger befinden sich in der Konfiguration | – | Variiert, siehe Konfigurieren de Embedded Even Manager |

# Nexus-Tools

Weitere Informationen zu verschiedenen Befehlen und deren Syntax bzw. Optionen finden Sie unter Cisco Nexus Switches der Serie 9000 - Befehlsreferenzen - Cisco.

- **Ethanalyzer**

Ethanalyzer ist ein NX-OS-Tool zur Erfassung von CPU-Datenverkehr in Paketen. Alles, was die CPU trifft, egal ob Eingang oder Ausgang, kann mit diesem Tool erfasst werden. Es basiert auf dem weit verbreiteten Open-Source-Netzwerkprotokoll-Analyzer Wireshark. Weitere Informationen zu diesem Tool finden Sie im Ethanalyzer auf Nexus 7000 Troubleshooting Guide - Cisco

Dabei ist zu beachten, dass Ethanalyzer im Allgemeinen den gesamten Datenverkehr vom und zum Supervisor erfasst, d. h. schnittstellenspezifische Erfassungen werden nicht unterstützt. Bestimmte Schnittstellenerweiterungen sind für ausgewählte Plattformen in neueren Codepunkten verfügbar. Außerdem erfasst Ethanalyzer nur Datenverkehr, der über eine CPU geleitet wird, und nicht Datenverkehr, der über eine Hardware geleitet wird. Sie können beispielsweise den Datenverkehr entweder über die In-Band-Schnittstelle, die Verwaltungsschnittstelle oder einen Port an der Vorderseite (sofern unterstützt) erfassen:

```
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
```

Diese Ausgabe zeigt einige der Nachrichten, die mit Ethanalyzer erfasst werden können. Bitte beachten Sie, dass Ethanalyzer standardmäßig nur bis zu 10 Pakete erfasst. Sie können diesen Befehl jedoch verwenden, um die CLI aufzufordern, Pakete unbegrenzt zu erfassen. Beenden Sie den Erfassungsmodus mit STRG+C.

```
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
```

Sie können mit Ethanalyzer auch Filter verwenden, um sich auf bestimmten Datenverkehr zu konzentrieren. Es gibt zwei Arten von Filtern, die Sie mit ethanalzyer verwenden können. Sie werden als Capture-Filter und Display-Filter bezeichnet. Ein Erfassungsfilter erfasst nur den Datenverkehr, der den im Erfassungsfilter definierten Kriterien entspricht. Ein Anzeigefilter erfasst weiterhin den gesamten Datenverkehr, es wird jedoch nur der Datenverkehr angezeigt, der den im Anzeigefilter definierten Kriterien entspricht.

```
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
```

Sie können Pakete auch mit der Detailoption erfassen und in Ihrem Terminal anzeigen, ähnlich wie in Wireshark. Auf diese Weise können Sie die vollständigen Header-Informationen basierend auf dem Paketzerlegungsergebnis anzeigen. Wenn beispielsweise ein Frame verschlüsselt ist, können Sie die verschlüsselte Nutzlast nicht sehen. Siehe folgendes Beispiel:

```
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
```

Mit Ethanalyzer können Sie:

- Schreiben Sie die Ausgabe (eine PCAP-Datei) in den angegebenen Dateinamen auf verschiedenen Zieldateisystemen: bootflash, logflash, USB, etc... Anschließend können Sie die gespeicherte Datei an eine Stelle außerhalb des Geräts übertragen und bei Bedarf in Wireshark anzeigen.
- Lesen Sie eine Datei vom Bootflash und zeigen Sie sie auf Ihrem Terminal an. Genau wie beim direkten Lesen von der CPU-Schnittstelle können Sie auch die vollständigen Paketinformationen anzeigen, wenn Sie das Stichwort detail verwenden.

Beispiele für verschiedene Schnittstellenquellen und Ausgabeoptionen:

```
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 steht für SwitchPort Analyzer und wird verwendet, um den gesamten Datenverkehr von einer Schnittstelle zu erfassen und diesen Datenverkehr an einen Zielport zu spiegeln. Der Zielport stellt in der Regel eine Verbindung zu einem Netzwerk-Analysetool her (z. B. einem PC mit Wireshark), mit dem Sie den Datenverkehr analysieren können, der diese Ports durchquert. Sie können SPAN für den Datenverkehr von einem einzelnen Port oder mehreren Ports und VLANs verwenden.

SPAN-Sitzungen umfassen einen Quell- und einen Ziel-Port. Ein Quell-Port kann ein Ethernet-Port (ohne Subschnittstellen), Port-Channels oder Supervisor Inband-Schnittstellen sein und nicht gleichzeitig ein Ziel-Port sein. Darüber hinaus werden für einige Geräte wie die 9300- und 9500-

Plattform auch FEX-Ports (Fabric Extender) unterstützt. Ein Ziel-Port kann ein Ethernet-Port (Access oder Trunk), ein Port-Channel (Access oder Trunk) sein. Bei einigen Geräten wie dem 9300 werden Uplink-Ports ebenfalls unterstützt, während FEX-Ports nicht als Ziel unterstützt werden.

Sie können mehrere SPAN-Sitzungen als Eingangs-/Ausgangs-/beides konfigurieren. Die Gesamtzahl der SPAN-Sitzungen, die ein einzelnes Gerät unterstützen kann, ist begrenzt. Ein Nexus 9000 kann beispielsweise bis zu 32 Sitzungen unterstützen, während ein Nexus 7000 nur 16 Sitzungen unterstützt. Sie können dies in der CLI überprüfen oder die SPAN-Konfigurationsanleitungen für das von Ihnen verwendete Produkt lesen.

Beachten Sie, dass sich die unterstützten Schnittstellentypen und -funktionen für jede NX-OS-Version und den Produkttyp unterscheiden. Beachten Sie die aktuellen Konfigurationsrichtlinien und -einschränkungen für das Produkt und die Version, die Sie verwenden. Hier sind die Links für Nexus 9000 und Nexus 7000:

Cisco Nexus Serie 9000 NX-OS - Systemmanagement-Konfigurationsleitfaden, Version 9.3(x) - Konfigurieren von SPAN [Cisco Nexus Switches der Serie 9000] - Cisco

Cisco Nexus Serie 7000 NX-OS - Systemmanagement - Konfigurationsanleitung - Konfigurieren von SPAN [Cisco Nexus Switches der Serie 7000] - Cisco

Es gibt verschiedene Arten von SPAN-Sitzungen. Einige der gebräuchlichsten Typen sind hier aufgelistet:

- Local SPAN (Lokales SPAN): Ein SPAN-Sitzungstyp, bei dem sich der Quell- und der Ziel-Host lokal im Switch befinden. Mit anderen Worten: Die gesamte zum Einrichten der SPAN-Sitzung erforderliche Konfiguration wird auf einen einzigen Switch angewendet, d. h. auf denselben Switch, auf dem sich die Quell- und Ziel-Host-Ports befinden.
- Remote SPAN (RSPAN): Ein SPAN-Sitzungstyp, bei dem sich der Quell- und der Ziel-Host nicht lokal am Switch befinden. Mit anderen Worten: Sie konfigurieren Quell-RSPAN-Sitzungen auf einem Switch und Ziel-RSPAN auf dem Ziel-Switch und erweitern die Verbindung mit dem RSPAN-VLAN.

Anmerkung: RSPAN wird auf Nexus nicht unterstützt.

- Extended Remote SPAN (ERSPAN): Der Switch kapselt den kopierten Frame mit einem GRE-Tunnel-Header (Generic Routing Encapsulation) und leitet das Paket an das konfigurierte Ziel weiter. Sie konfigurieren Quell- und Zielsitzungen auf den Kapselungs- und Entkapselungsschaltern (zwei verschiedene Geräte). So können wir den SPAN-Verkehr über ein Layer-3-Netzwerk leiten.
- SPAN-zu-CPU: Ein Name, der einem speziellen Typ einer SPAN-Sitzung zugewiesen wird, bei der der Ziel-Port der Supervisor oder die CPU ist. Dies ist eine Form der lokalen SPAN-Sitzung und kann in Fällen verwendet werden, in denen eine Standard-SPAN-Sitzung nicht genutzt werden kann. Einige der häufigsten Gründe sind: keine verfügbaren oder geeigneten SPAN-Zielports, kein Zugriff auf den Standort oder ein nicht verwalteter Standort, kein Gerät verfügbar, das eine Verbindung zum SPAN-Zielport herstellen kann usw. Weitere Informationen finden Sie unter diesem Link Nexus 9000 Cloud Scale ASIC NX-OS SPAN-to-CPU Procedure - Cisco. Beachten Sie, dass die Übertragungsrate der SPAN-zu-CPU durch CoPP (Control Plane Policing) begrenzt ist. sniffing  eine oder mehrere Quellschnittstellen, die die Richtlinie überschreiten, können zu Unterbrechungen für die SPAN-CPU-Sitzung führen.

Wenn dies geschieht, spiegeln die Daten nicht 100 % des Inhalts des Kabels wider, sodass SPAN zu CPU nicht immer für Fehlerbehebungsszenarien mit hoher Datenrate und/oder vorübergehendem Verlust geeignet ist. Sobald Sie eine SPAN-zu-CPU-Sitzung konfigurieren und administrativ aktivieren, müssen Sie Ethanalyzer ausführen, um den Datenverkehr anzuzeigen, der an die CPU gesendet wird, um die Analyse entsprechend durchzuführen.

Dies ist ein Beispiel dafür, wie Sie eine einfache lokale SPAN-Sitzung auf einem Nexus 9000-Switch konfigurieren können:

```
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
```

Dieses Beispiel zeigt die Konfiguration einer SPAN-zu-CPU-Sitzung, die gestartet wurde, und dann die Verwendung von Ethanalyzer zur Erfassung des Datenverkehrs:

```
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
```

- Entspiegeln

Dmirror ist eine SPAN-TO-CPU-Sitzung für Broadcom-basierte Nexus-Plattformen. Das Konzept ist dasselbe wie für SPAN-to-CPU und die Übertragungsrate auf 50 pps (Pakete pro Sekunde) beschränkt. Die Funktion wurde implementiert, um den internen Datenpfad mit der CLI bcm-shell zu debuggen. Aufgrund der damit verbundenen Einschränkungen gibt es keine NX-OS-CLI, mit der Benutzer SPAN-Sitzungen für den Sup konfigurieren können, da dies den Steuerungsdatenverkehr beeinflussen und CoPP-Klassen verwenden kann.

- **ELAM**

ELAM steht für Embedded Logic Analyzer Module. Sie bietet die Möglichkeit, den ASIC zu untersuchen und zu bestimmen, welche Weiterleitungsentscheidungen für ein **EINZELNES** Paket getroffen werden. Mithilfe von ELAM können Sie ermitteln, ob das Paket die Weiterleitungs-Engine erreicht, und welche Ports/VLAN-Informationen zur Verfügung stehen. Sie können auch überprüfen, ob die Paketstruktur von L2 nach L4 geändert wurde und ob Änderungen am Paket vorgenommen wurden.

Es ist wichtig zu verstehen, dass ELAM architekturabhängig ist und das Verfahren zur Paketerfassung je nach interner Architektur von Plattform zu Plattform variiert. Sie müssen die ASIC-Zuordnungen der Hardware kennen, um das Tool korrekt anwenden zu können. Für den Nexus 7000 werden zwei Erfassungen für ein einzelnes Paket durchgeführt, eine vor der Entscheidung für den **Datenbus (DBUS)** und eine andere nach der Entscheidung für den **Ergebnisbus (RBUS)**. Wenn Sie die DBUS-Informationen anzeigen, können Sie sehen, was/wo das Paket empfangen wurde, sowie die Layer-2- bis Layer-4-Informationen. Ergebnisse im RBUS können Ihnen zeigen, wohin das Paket weitergeleitet wird und ob der Frame geändert wurde. Sie müssen Trigger für DBUS und RBUS einrichten, sicherstellen, dass sie bereit sind, und dann versuchen, das Paket in Echtzeit zu erfassen. Die Vorgehensweise für verschiedene Linecards ist wie folgt:

Einzelheiten zu den verschiedenen ELAM-Verfahren finden Sie unter den Links in dieser Tabelle:

| | |
|---|---|
| ELAM - ÜBERBLICK | ELAM im Überblick - Cisco |
| Nexus 7K F1-Modul | ELAM-Verfahren für Nexus 7000 F1-Module - Cisco |
| Nexus 7K F2-Modul | ELAM-Verfahren für Nexus 7000 F2-Module - Cisco |
| Nexus 7K F3-Modul | F3 - ELAM-Beispiel |
| Nexus 7K M-Modul | ELAM-Verfahren für Nexus 7000-Module der M-Serie - Cisco |
| Nexus 7K M1/M2- und F2-Modul | Nexus 7K ELAM für M1/M2 und F2 und Ethanalyzer |
| Nexus 7K M3-Modul | ELAM-Verfahren für Nexus 7000 M3-Module - Cisco |

### ELAM für Nexus 7000 - M1/M2 (Eureka-Plattform)

- Überprüfen Sie die Modulnummer mit dem Befehl **show module**.
- Am Modul mit **Anschlussmodul x anhängen**, wobei x die Modulnummer ist.
- Überprüfen Sie mit dem Befehl **show hardware internal dev-port-map** die interne ASIC-Zuordnung, und prüfen Sie, ob L2LKP und L3LKP vorhanden sind.

```
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
+---------------------------------------------------------------------+
+---------------------------------------------------------------------+
```

- Zunächst erfassen Sie das Paket in L2 und prüfen, ob die Weiterleitungsentscheidung richtig ist. Sehen Sie dazu in der Spalte für L2LKP-Zuordnungen nach, und geben Sie die ASIC-Instanznummer an, die dem Port entspricht.
- Als Nächstes starten Sie ELAM auf dieser Instanz mit dem Befehl **elam asic eureka instance x**wobei x die ASIC-Instanznummer ist und die Trigger für DBUS und RBUS konfiguriert werden. Überprüfen Sie den Status der Trigger mit dem **Befehlsstatus**, und bestätigen Sie, dass die Trigger konfiguriert wurden.

```
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
```

- Aktivieren Sie die Trigger mit dem Befehl **start,** und überprüfen Sie, ob der Status der Trigger mit dem Befehl **status** bestätigt, dass die Trigger aktiviert sind.

```
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
```

- Sobald der Status zeigt, dass die Auslöser bewaffnet sind, sind sie bereit, erfasst werden. Zu diesem Zeitpunkt müssen Sie den Datenverkehr weiterleiten und den Status erneut überprüfen, um festzustellen, ob die Trigger tatsächlich ausgelöst wurden.

```
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
```

- Überprüfen Sie nach der Auslösung die Paketsequenznummer für rbus und dbus, um sicherzustellen, dass beide dasselbe Paket erfasst haben. Dies kann mit dem Befehl **show dbus | i seq ; Show-Rbus | i seq**. Wenn die Sequenznummer übereinstimmt, können Sie den Inhalt von dbus und rbus sehen. Wenn nicht, führen Sie die Erfassung erneut aus, bis Sie dasselbe Paket erfassen können.

  **Anmerkung:** Führen Sie ELAM immer mehrmals aus, um Weiterleitungsprobleme zu bestätigen.

- Sie können den Inhalt von rbus und dbus mit den Befehlen **show dbus** und **show rbus anzeigen**. Wichtig bei der Erfassung sind die Sequenznummer und der Quell-/Zielindex. Dbus zeigt Ihnen den Quellindex, der Ihnen den Port mitteilt, auf dem das Paket empfangen wurde. Rbus zeigt den Zielindex des Ports an, an den das Paket weitergeleitet wird. Zusätzlich können Sie auch Quell- und Ziel-IP/MAC-Adressen sowie VLAN-Informationen anzeigen.
- Mit dem Quell- und Zielindex (auch LTL-Index genannt) können Sie den zugehörigen Port an der Vorderseite mit dem Befehl **show system internal pixm info ltl #** überprüfen.

### ELAM für Nexus 7000 - M1/M2 (Lamira-Plattform)

Die Vorgehensweise ist auch für die Lamira-Plattform identisch, es gibt jedoch einige Unterschiede:

- Sie führen ELAM mit dem Schlüsselwort Lamira **elam asic lamira instance x aus**.
- Die Befehle zum Auslösen des ELAM sind wie folgt:

```
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
```

- Sie überprüfen den Status mit dem **Status-**Befehl und stellen sicher, dass sie bewaffnet sind, bevor Sie Datenverkehr senden und ausgelöst werden, nachdem Sie ihn erfasst haben.
- Sie können dann die Ausgänge von dbus und show bus in ähnlicher Weise interpretieren, wie für Eureka gezeigt.

### ELAM für Nexus 7000 - F2/F2E (Clipper-Plattform)

Auch hier ist das Verfahren ähnlich, nur die Trigger sind unterschiedlich. Es gibt folgende Unterschiede:

- Sie führen ELAM mit dem Schlüsselwort Clipper **elam asic clipper instance x aus** und geben den Layer 2- oder Layer 3-Modus an.

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

- Die Befehle zum Auslösen des ELAM sind wie folgt:

```
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
```

- Sie überprüfen den Status mit dem **Status-**Befehl und stellen sicher, dass sie bewaffnet sind, bevor Sie Datenverkehr senden und ausgelöst werden, nachdem Sie ihn erfasst haben.
- Sie können dann die Ausgänge von dbus und show bus in ähnlicher Weise interpretieren, wie für Eureka gezeigt.

### ELAM für Nexus 7000 - F3 (Flanker-Plattform)

Auch hier ist das Verfahren ähnlich, nur Trigger sind unterschiedlich. Es gibt folgende Unterschiede:

- Sie führen ELAM mit dem Schlüsselwort Flanker **elam asic flanker instance x aus** und geben den Layer 2- oder Layer 3-Modus an.

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

- Die Befehle zum Auslösen des ELAM sind wie folgt:

```
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
```

- Mit dem Befehl **status** überprüfen Sie den Status und stellen sicher, dass sie bewaffnet sind, bevor Sie Datenverkehr senden und ausgelöst werden, nachdem Sie ihn erfasst haben.
- Sie können dann die Ausgänge von dbus und rbus in ähnlicher Weise interpretieren, wie für Eureka gezeigt.

### ELAM für Nexus 9000 (Tahoe-Plattform)

Beim Nexus 9000 unterscheidet sich das Verfahren geringfügig vom Nexus 7000. Informationen zu Nexus 9000 finden Sie unter dem Link Nexus 9000 Cloud Scale ASIC (Tahoe) NX-OS ELAM - Cisco

- Überprüfen Sie zunächst mithilfe des Befehls **show hardware internal tah interface #** die Schnittstellenzuordnung. Die wichtigsten Informationen in dieser Ausgabe sind **ASIC, Slice und Quell-ID (srcid).**
- Zusätzlich können Sie diese Informationen mit dem Befehl **show system internal ethpm info interface # | i i src**. Wichtig ist hier neben dem, was zuvor aufgeführt wurde, der dpid- und der dmod-Wert.
- Überprüfen Sie die Modulnummer mit dem Befehl **show module**.
- Am Modul mit **Anschlussmodul x anhängen**, wobei x die Modulnummer ist.
- Führen Sie ELAM auf dem Modul mit dem Befehl **module-1# debug platform internal tah elam asic #**
- Konfigurieren Sie den inneren oder äußeren Trigger auf Basis der Art des Datenverkehrs, den Sie erfassen möchten (L2, L3, gekapselten Datenverkehr wie GRE oder VXLAN usw.):

```
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
```

- Sobald die Trigger festgelegt sind, starten Sie ELAM mit dem Befehl **start**, senden Datenverkehr und zeigen die Ausgabe mit dem Befehl **report an**. In der Berichtausgabe werden die ausgehenden und eingehenden Schnittstellen sowie die VLAN-ID, die Quell- und Ziel-IP/MAC-Adresse angezeigt.

```
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
```

### ELAM für Nexus 9000 (NorthStar-Plattform)

Das Verfahren für die NorthStar-Plattform ist das gleiche wie die Tahoe-Plattform, der einzige Unterschied besteht darin, dass das Schlüsselwort **ns** anstelle von **tah** verwendet wird, wenn der ELAM-Modus eingegeben wird:

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

- ### N9K Packet Tracer

Das Nexus 9000 Packet Tracer-Tool kann verwendet werden, um den Pfad des Pakets zu verfolgen, und mit seinen integrierten Zählern für Flussstatistiken ist es ein nützliches Tool für Szenarien mit vorübergehenden/vollständigen Datenverlusten. Diese Vorgehensweise ist sehr nützlich, wenn die TCAM-Ressourcen für die Ausführung anderer Tools eingeschränkt oder nicht verfügbar sind. Darüber hinaus kann dieses Tool ARP-Datenverkehr nicht erfassen und zeigt keine Details zum Paketinhalt wie Wireshark an.

Verwenden Sie folgende Befehle, um die Paketverfolgung zu konfigurieren:

```
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
```

Weitere Informationen finden Sie unter dem Link Nexus 9000: Erläuterung des Packet Tracer-Tools - Cisco

- ### Routenverfolgung und Pings

Diese Befehle sind die zwei nützlichsten Befehle, mit denen Sie Verbindungsprobleme schnell identifizieren können.

Ping verwendet Internet Control Message Protocol (ICMP), um ICMP-Echo-Nachrichten an das jeweilige Ziel zu senden, und wartet auf ICMP-Echo-Antworten von diesem Ziel. Wenn der Pfad zwischen den Hosts problemlos funktioniert, ohne dass Probleme auftreten, können Sie sehen, dass die Antworten zurückkommen und die Pings erfolgreich sind. Der Befehl ping sendet standardmäßig 5x ICMP-Echo-Nachrichten (gleiche Größe in beide Richtungen), und wenn alles gut funktioniert, können Sie 5x ICMP-Echo-Antworten sehen. Manchmal schlägt die erste Echoanfrage fehl, wenn Switches die MAC-Adresse während der ARP-Anfrage (Address Resolution Protocol) ermitteln. Wenn Sie den Ping direkt danach wiederholen, gibt es keinen anfänglichen Pingverlust. Darüber hinaus können Sie die Anzahl der Pings, die Paketgröße, die Quelle, die Quellschnittstelle und die Zeitüberschreitungsintervalle mit den folgenden Schlüsselwörtern festlegen:

```
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
```

Mit der Traceroute werden die verschiedenen Hops identifiziert, die ein Paket benötigt, bevor es sein Ziel erreicht. Es ist ein sehr wichtiges Tool, da es dabei hilft, die L3-Grenze zu identifizieren, an der ein Fehler auftritt. Sie können den Port, die Quell- und die Quell-Schnittstelle auch mit den folgenden Schlüsselwörtern verwenden:

```
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 steht für Access Control List. Es handelt sich um ein wichtiges Tool, mit dem Sie den Datenverkehr anhand eines bestimmten Kriteriums filtern können. Sobald die ACL mit Einträgen für Abgleichkriterien gefüllt ist, kann sie angewendet werden, um eingehenden oder ausgehenden Datenverkehr zu erfassen. Ein wichtiger Aspekt von ACL ist die Möglichkeit, Zähler für Flow-Statistiken bereitzustellen. Die Begriffe PACL/RACL/VACL beziehen sich auf verschiedene Implementierungen dieser ACLs, die Ihnen die Verwendung von ACL als leistungsstarkes Fehlerbehebungstool speziell für den zeitweiligen Verlust von Datenverkehr ermöglichen. Diese Begriffe werden hier kurz beschrieben:

- PACL steht für Port Access Control List: Wenn Sie eine Zugriffsliste auf einen L2-Switch-Port bzw. eine L2-Switch-Schnittstelle anwenden, wird diese Zugriffsliste als PACL bezeichnet.
- RACL steht für Router Access Control List: Wenn Sie eine Zugriffsliste auf einen gerouteten L3-Port/eine geroutete Schnittstelle anwenden, wird diese Zugriffsliste als RACL bezeichnet.
- VACL steht für VLAN Access Control List: Sie können VACLs so konfigurieren, dass sie für alle Pakete gelten, die in ein VLAN oder aus einem VLAN heraus geroutet oder innerhalb eines VLAN überbrückt werden. VACLs dienen ausschließlich der Paketfilterung und der Umleitung des Datenverkehrs an bestimmte physische Schnittstellen. VACLs werden nicht nach Richtung definiert (Eingang oder Ausgang).

In dieser Tabelle werden die ACL-Versionen verglichen.

| ACL-TYP | PACL | RACL | VACL |
| --- | --- | --- | --- |
| FUNKTION | An einer L2-Schnittstelle empfangenen Datenverkehr filtern. | An einer L3-Schnittstelle empfangenen Datenverkehr filtern | Filtern des vLAN-Datenverkehrs |
| ANGEWENDET AUF | - L2-Schnittstellen/-Ports<br>- L2-Port-Channel-Schnittstellen<br>- Wird sie auf einen Trunk-Port angewendet, filtert die ACL den Datenverkehr in allen VLANs, die auf diesem Trunk-Port zulässig sind. | - VLAN-Schnittstellen<br>- Physische L3-Schnittstellen<br>- L3-Subschnittstellen<br>- L3-Port-Channel-Schnittstellen<br>- Management-Schnittstellen | Nach der Aktivierung wird ACL auf alle Ports in dies VLAN angewendet (einschließlich Trunk-Port |
| ANGEWANDTE RICHTUNG | Nur eingehend. | Eingehend oder ausgehend | - |

Hier ist ein Beispiel, wie Sie eine Zugriffsliste konfigurieren können. Weitere Informationen finden Sie im Link [Cisco Nexus 9000 Serie NX-OS Security Configuration Guide, Release 9.3(x) - Configuring IP ACLs [Cisco Nexus Switches der Serie 9000] - 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 ist eine Art von persistentem Speicher, der auf Nexus-Plattformen entweder als externer Compact Flash-Speicher, als USB-Gerät oder als integrierter Datenträger im Supervisor verfügbar ist. Wenn der Switch entfernt wird, benachrichtigt das System den Benutzer regelmäßig, dass LogFlash fehlt. Logflash wird auf dem Supervisor installiert und enthält Verlaufsdaten wie Accounting-Protokolle, Syslog-Meldungen, Fehlerbehebungen und Embedded Event Manager (EEM)-Ausgaben. EEM wird später in diesem Artikel besprochen. Sie können den Inhalt von LogFlash mit dem folgenden Befehl überprüfen:

```
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
```

Falls ein Benutzer das Gerät neu lädt oder es aufgrund eines Ereignisses plötzlich neu lädt, gehen alle Protokollinformationen verloren. In solchen Szenarien kann LogFlash historische Daten liefern, die überprüft werden können, um eine wahrscheinliche Ursache des Problems zu identifizieren. Natürlich ist weitere Sorgfalt erforderlich, um die Ursache zu identifizieren, die Ihnen Hinweise darauf gibt, wonach Sie suchen sollten, falls dieses Ereignis erneut auftritt.

Informationen zur Installation von Logflash auf dem Gerät finden Sie unter dem Link Nexus 7000-Protokollierungsfunktionen - Cisco.

- OBFL

OBFL steht für OnBoard Failure Logging. Es handelt sich um eine Art von persistentem Speicher, der sowohl für Nexus Top-of-Rack- als auch für modulare Switches verfügbar ist. Wie der LogFlash werden die Informationen nach dem erneuten Laden des Geräts beibehalten. OBFL speichert Informationen wie Ausfälle und Umgebungsdaten. Die Informationen variieren je nach Plattform und Modul. Hier sehen Sie jedoch eine Beispielausgabe für Modul 1 der Nexus 93108-Plattform (d. h. ein festes Chassis mit nur einem Modul):

```
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
```

Auch diese Informationen sind nützlich, wenn ein Gerät entweder vom Benutzer absichtlich oder aufgrund eines Ereignisses, das ein erneutes Laden ausgelöst hat, neu geladen wird. In diesem Fall können OBFL-Informationen helfen, aus Sicht einer Line Card zu erkennen, was schief gelaufen ist. Der Befehl **show logging onboard** ist ein guter Ausgangspunkt. Denken Sie daran, dass Sie alles, was Sie brauchen, aus dem Modulkontext heraus erfassen müssen. Stellen Sie sicher, dass Sie **show logging on board module x** verwenden oder **mod x anhängen. Anmeldung an Bord anzeigen.**

- **Ereignisverlauf**

Der Ereignisverlauf ist eines der leistungsstarken Tools, mit dem Sie Informationen zu verschiedenen Ereignissen erhalten, die bei einem auf Nexus ausgeführten Prozess auftreten. Mit anderen Worten: Jeder Prozess, der auf einer Nexus-Plattform ausgeführt wird, verfügt über Ereignisverlaufslisten, die im Hintergrund ausgeführt werden und Informationen zu verschiedenen Ereignissen dieses Prozesses speichern (diese stellen sich als ständig ausgeführte Debugs vor). Diese Ereignisverlaufsdaten sind nicht persistent und alle gespeicherten Informationen gehen beim Neuladen des Geräts verloren. Diese sind sehr nützlich, wenn Sie ein Problem mit einem bestimmten Prozess identifiziert haben und diesen Prozess beheben möchten. Wenn beispielsweise das OSPF-Routing-Protokoll nicht ordnungsgemäß funktioniert, können Sie mit OSPF verknüpfte Ereignisverlaufshistorien verwenden, um den Fehler beim OSPF-Prozess zu identifizieren. Auf der Nexus-Plattform finden Sie Ereignisprotokolle zu nahezu allen Prozessen, z. B. CDP/STP, UDLD, LACP/OSPF, EIGRP/BGP usw.

So überprüfen Sie normalerweise anhand von Referenzbeispielen den Ereignisverlauf für einen Prozess. Jeder Prozess hat mehrere Optionen, also verwenden **?** um zu prüfen, ob verschiedene Optionen unter einem Prozess verfügbar sind.

```
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
```

- ## Fehlerbehebung

Debugging-Programme sind leistungsstarke Tools in NX-OS, mit denen Sie Fehlerbehebungsereignisse in Echtzeit ausführen und in einer Datei oder einer Anzeige in der CLI protokollieren können. Es wird dringend empfohlen, die Debug-Ausgaben in einer Datei zu protokollieren, da sie die CPU-Leistung beeinträchtigen. Seien Sie vorsichtig, bevor Sie ein Debugging direkt über die CLI ausführen.

Debugs werden in der Regel nur dann ausgeführt, wenn Sie ein Problem als einen einzigen Prozess identifiziert haben und in Echtzeit überprüfen möchten, wie sich dieser Prozess mit echtem Datenverkehr im Netzwerk verhält. Sie müssen eine Debugfunktion aktivieren, die auf den definierten Benutzerkontoberechtigungen basiert.

Ebenso wie der Ereignisverlauf können Sie auf einem Nexus-Gerät Debug-Vorgänge für jeden Prozess ausführen, z. B. für CDP/STP, UDLD, LACP/OSPF, EIGRP/BGP usw.

So führen Sie normalerweise ein Debuggen für einen Prozess aus. Jeder Prozess hat mehrere Optionen, also verwenden **?** um zu prüfen, ob verschiedene Optionen unter einem Prozess verfügbar sind.

```
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
```

- GOLD

GOLD steht für Generic OnLine Diagnostics. Wie der Name schon sagt, werden diese Tests im Allgemeinen als Systemstatusprüfung verwendet und zur Überprüfung oder Verifizierung der betreffenden Hardware verwendet. Es gibt verschiedene Online-Tests, die auf der verwendeten Plattform basieren. Einige dieser Tests sind unterbrechungsfrei, andere dagegen nicht. Diese Online-Tests lassen sich wie folgt kategorisieren:

- **Systemstart-Diagnose:** Diese Tests werden beim Hochfahren des Geräts ausgeführt. Darüber hinaus wird die Verbindung zwischen dem Supervisor und den Modulen geprüft. Dies schließt die Verbindung zwischen Daten- und Kontrollebene für alle ASICs ein. Tests wie ManagementPortLoopback und EOBCLoopback führen zu Unterbrechungen, OBFL- und USB-Tests dagegen nicht.
- **Laufzeitdiagnose oder Diagnose der Systemüberwachung:** Diese Tests liefern Informationen über den Zustand des Geräts. Diese Tests sind unterbrechungsfrei und werden im

Hintergrund ausgeführt, um die Stabilität der Hardware zu gewährleisten. Sie können diese Tests nach Bedarf oder zur Fehlerbehebung aktivieren/deaktivieren.

- **On-Demand-Diagnose:** Alle genannten Tests können bei Bedarf wiederholt werden, um ein Problem zu lokalisieren.

Mit dem folgenden Befehl können Sie überprüfen, ob verschiedene Online-Tests für Ihren Switch verfügbar sind:

```
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-
```

Verwenden Sie den folgenden Befehl, um anzuzeigen, was die 17 genannten Tests tun:

```
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 steht für Embedded Event Manager. Es ist ein leistungsstarkes Tool, mit dem Sie Ihr Gerät so programmieren können, dass es bestimmte Aufgaben für den Fall eines bestimmten Ereignisses ausführt. Es überwacht verschiedene Ereignisse auf dem Gerät und ergreift dann die erforderlichen Maßnahmen, um das Problem zu beheben und möglicherweise wiederherzustellen.

EEM besteht aus drei Hauptkomponenten, die hier jeweils kurz beschrieben werden:

- **Event-Anweisung:** Dies sind die Ereignisse, die überwacht werden sollen, damit Nexus eine bestimmte Aktion ausführt, z. B. eine Problemumgehung durchführen, einen SNMP-Server benachrichtigen oder ein CLI-Protokoll anzeigen usw.
- **Aktionshinweise:** Dies sind die Schritte, die EEM nach der Auslösung eines Ereignisses durchführt. Diese Aktionen können einfach sein, eine Schnittstelle zu deaktivieren oder einige show-Befehle auszuführen und Ausgaben in eine Datei auf dem FTP-Server zu kopieren, eine E-Mail zu senden usw.
- **Richtlinien:** Es handelt sich im Grunde um ein Ereignis in Kombination mit einer oder mehreren action-Anweisungen, die Sie auf dem Supervisor über CLI oder ein Bash-Skript konfigurieren können. Sie können EEM auch mit einem Python-Skript aufrufen. Sobald die Richtlinie auf dem Supervisor definiert wurde, wird sie an das entsprechende Modul weitergeleitet.

Einzelheiten zu EEM finden Sie unter dem Link Cisco Nexus 9000 Serie NX-OS System Management Configuration Guide, Release 9.2(x) - Configuring the Embedded Event Manager [Cisco Nexus Switches der Serie 9000] - Cisco.

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.