

針對初學者的Nexus備忘稿故障排除

目錄

[簡介](#)

[概觀](#)

[Nexus工具](#)

[Ethanalyzer](#)

[範圍](#)

[Dmirror](#)

[伊蘭](#)

[N9K Packet Tracer](#)

[Traceroute和Ping](#)

[PACL/RACL/VACL](#)

[OBFL](#)

[事件記錄](#)

[調試](#)

[EEM](#)

簡介

本文檔介紹可用於對Nexus產品進行故障排除的各種工具，您可以使用這些工具來診斷和修復問題。

概觀

瞭解可用工具以及在什麼情況下使用它們以獲得最大收益是很重要的。事實上，有時候，僅僅因為某個工具是為了處理其他事情而設計的，所以它就不可行。

此表彙編了在Nexus平台上排除故障的各種工具及其功能。有關詳細資訊和CLI示例，請參閱Nexus工具部分。

工具	功能	使用案例示例	優點	缺點	持續性	受影響的平面	使用的CLI命令
Ethanalyzer	擷取目的地為或來自CPU的流量	流量緩慢問題、延遲和擁塞	非常適用於慢速、擁塞和延遲問題	通常只看到控制平面流量，速率受限	不適用	控制平面。可用於某些情況下的資料層面（SPAN到CPU）	#ethanalyzer lo inband #ethanalyzer lo [interface ID] dis [WORD]

							範例： #ethalyzer lo Ethernet 6/4 dis ICMP
範圍	捕獲和映象大量資料包	失敗 ping 的、無序資料包等	適用於間斷性流量丟失	需要運行嗅探器軟體的外部裝置 需要TCAM資源	需要設定和啟用/停用SPAN作業階段	Control +資料	#monitor sessio #description [N interface [port I #destination inte ID] #no shut
錯誤	僅捕獲Broadcom Nexus裝置的CPU發往或發往CPU的流量	流量緩慢問題、延遲和擁塞	非常適用於慢速、擁塞和延遲問題	僅適用於Broadcom Nexus裝置。速率限制 (CloudScale Nexus 9k具有SPAN到CPU)	不適用	控制平面。在某些情況下可用於資料層面	因平台而異，請 ELAM概述-思科
伊蘭	捕獲進入[或離開 (如果Nexus 7K) Nexus交換機的單個資料包	驗證資料包是否到達Nexus，檢查轉發決策，檢查資料包是否更改，驗證資料包的介面/VLAN等	非常適用於資料包流量和轉發問題。非侵入式	需要深入瞭解硬體。使用架構特定的獨特觸發機制。只有在您知道要檢查的流量時，才有用	不適用	Control +資料	# attach module NUMBER] # de internal <>
Nexus 9k Packet Tracer	檢測資料包的路徑	連線問題和資料包丟失	提供用於間斷/完全丟失的流統計資料的計數器。最適合無TCAM雕刻的線卡	無法捕獲ARP流量。僅適用於Nexus 9k	不適用	資料+控制	# test packet-tra IP] dst_IP [目標 packet-tracer st packet-tracer st packet-tracer sh
Traceroute	檢測資料包相	ping失敗、	檢測路徑	僅標識L3邊界	不適用	資料+控制	# traceroute [目

	對於L3跳的路徑	無法到達主機/目標/國際網路等	中的各個躍點以隔離L3故障。	中斷的位置 (不標識問題本身)			引數包括： 埠、埠號、源、介面
Ping	測試網路中兩點之間的連通性	測試裝置之間的可接通性	測試連通性的快速而簡單的工具	只辨識主機是否可連線	不適用	資料+控制	# ping [目標IP] 引數包括： count, packet-interface, inter-multicast, loop-timeout
PACL/RAACL/VACL	擷取從特定連線埠或VLAN輸入/輸出的流量	主機之間的問題性資料包丟失，確認資料包是否到達/離開Nexus等	適用於間斷性流量丟失	需要TCAM資源。對於某些模組，需要手動TCAM刻制	持續(套用至running-組態)	資料+控制	# ip access-list # ip port access-NAME] # ip acco[ACL NAME] 引數包括： deny, fragmen-permit, remark-statistics, end-pop, push, w
LogFlash	全局儲存交換機的歷史資料，例如日誌帳戶、崩潰檔案和事件，而不考慮裝置重新載入	裝置突然重新載入/關閉，每當重新載入裝置時，日誌快閃記憶體資料都會提供一些有助於分析的資訊	資訊保留在裝置重新載入上(永久儲存)	Nexus 7K上的外部=必須在管理引擎平台上安裝/整合，以便收集這些日誌 (con不適用於3K/9K，因為logflash是內部儲存裝置的分割槽)	重新載入-持續	資料+控制	# dir logflash :
OBFL	儲存特定模組的歷史資料，如故障和環境資訊	裝置突然重新載入/關閉，每當重新載入裝置時，日誌快	資訊保留在裝置重新載入上(永久儲存)	支援有限數量的讀取和寫入	重新載入-持續	資料+控制	# show logging module [#] 引數包括： boot-uptime, o

		閃記憶體資料都會提供一些有用的資訊					history , card-f on , counter-st version , endtin environmental-h stats , exceptio internal , intern obfl-history , st starttime , stat
事件記錄	當您需要目前執行之特定處理作業的資訊時	Nexus中的每個進程都有自己的事件歷史記錄，例如 CDP、STP、OSPF、EIGRP、BGP、vPC、LACP等	排除 Nexus上運行的特定流程的故障	重新載入裝置後，資訊將丟失（非永續性）	非持續	資料+控制	# show [PROCE event-history [5 引數包括： 鄰接， cli， eve flooding， ha， lsa， msgs， o 分配， rib， se spf-trigger， sta
調試	當您需要更精細的即時/即時資訊用於特定流程時	可以對 nexus中的每個進程進行調試，如 CDP、STP、OSPF、IGRP、BGP、vPC、LACP等	即時對 Nexus上運行的特定流程進行故障排除，從而獲得更精細的體驗	可能影響網路效能	非持續	資料+控制	# debug proces [PROCESS] 範例： # debug ip ospf
金牌	提供硬體元件（例如 I/O 和 Supervisor 模組）的啟動、執行階段和隨選診斷	測試硬體，例如 USB、Bootflash、OBFL、ASIC 記憶體、PCIE、連線埠回送、NVRAM 等	只能在版本 6(2)8 及更高版本上檢測硬體中的故障並採取必要的糾正措施	僅檢測硬體問題	非持續	不適用	# show diagnos module all # sho description mod

EEM	監視裝置上的事件並採取必要的操作	任何需要某些動作/解決方法/通知的裝置活動，例如介面關閉、風扇故障、CPU利用率等	支援Python指令碼	必須具有網路管理員許可權才能配置EEM	EEM指令碼和觸發器駐留在配置中	不適用	視情況而定，請 配置嵌入式事件
-----	------------------	---	-------------	---------------------	------------------	-----	---------------------------------

Nexus工具

如果您需要瞭解有關各種命令及其語法或選項的更多資訊，請參閱[Cisco Nexus 9000系列交換機-命令參考- Cisco](#)。

Ethalyzer

Ethalyzer是一種NX-OS工具，旨在捕獲資料包CPU流量。使用此工具可以捕獲任何敲擊CPU（入口或出口）的內容。它基於廣泛使用的開源網路協定分析器Wireshark。有關此工具的詳細資訊，請參閱[Nexus 7000上的Ethalyzer故障排除指南-思科](#)

必須注意的是，一般來說，Ethalyzer會捕獲所有進出主管的流量。也就是說，它不支援特定於介面的捕獲。特定介面增強功能適用於較新程式碼點中的特定平台。此外，Ethalyzer僅捕獲CPU交換而不是硬體交換的流量。例如，您可以在頻內介面、管理介面或前面板連線埠（支援的地方）上擷取流量：

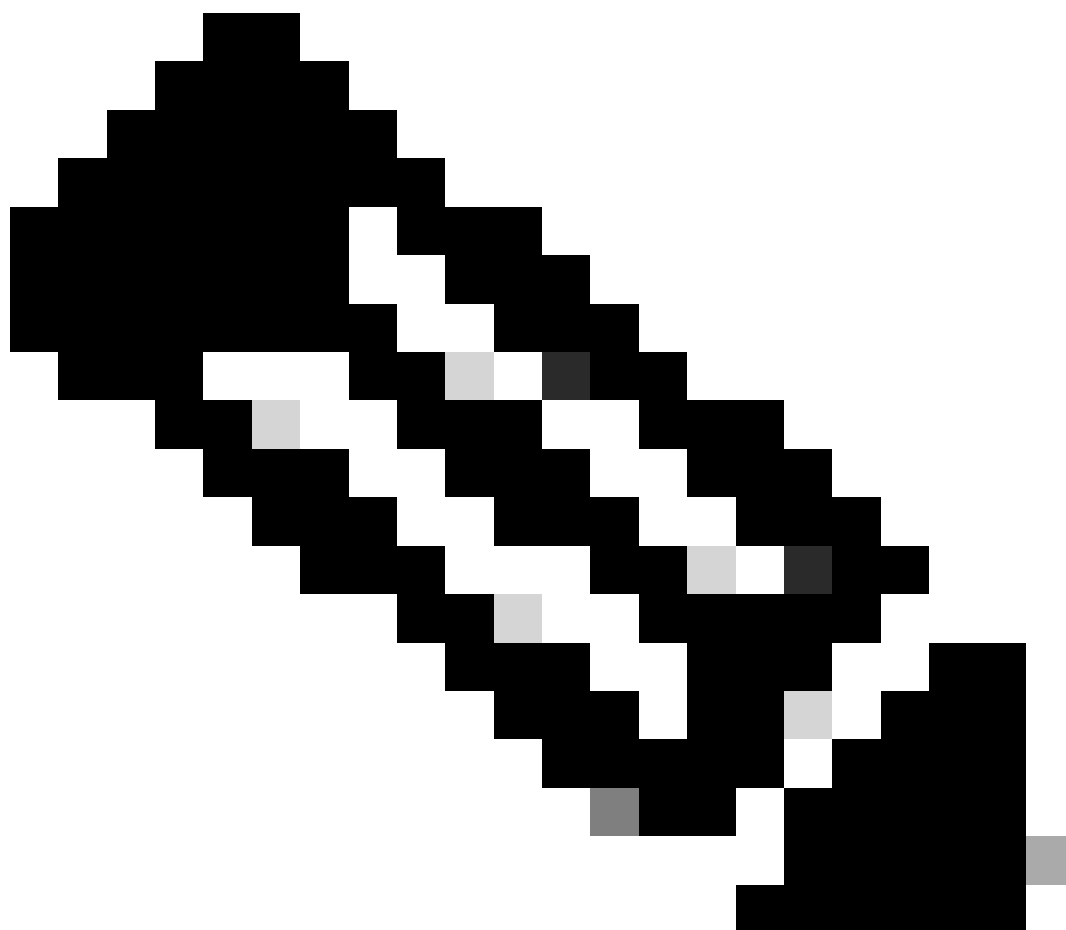
```
Nexus9000_A(config-if-range)# ethalyzer local interface inband
Capturing on inband
2020-02-18 01:40:55.183177 cc:98:91:fc:55:8b -> 01:80:c2:00:00:00 STP RST. Root = 32768/1/cc:98:91:fc:55:8b
2020-02-18 01:40:55.184031 f8:b7:e2:49:2d:f2 -> 01:80:c2:00:00:0e LLC U, func=UI; SNAP, OUI 0x00000C (Cisco LLC)
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 LLC)
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 LLC)
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 LLC)
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:f9:85
2020-02-18 01:40:56.365722 0.0.0.0 -> 255.255.255.255 DHCP DHCP Discover - Transaction ID 0xc82a6d
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 LLC)
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:55:8b
2020-02-18 01:40:57.367890 0.0.0.0 -> 255.255.255.255 DHCP DHCP Discover - Transaction ID 0xc82a6d
10 packets captured
```

```
Nexus9000_A(config-if-range)# ethalyzer 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:00:00
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:00:00
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:00:00
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:00:00
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:00:00
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:00:00
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:00:00
2020-02-18 01:53:20.391215 00:be:75:5b:d9:00 -> 01:00:0c:cc:cc:cc CDP Device ID: Nexus9000_A(FD021512ZE)
```

```
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:
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:
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:b8
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:b8
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:b8
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:b8
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:b8
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:b8
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:b8
```

此輸出顯示可以透過Ethanalyzer捕獲的消息很少。



注意：預設情況下，Ethanalyzer僅捕獲最多10個資料包。但是，您可以使用此命令提示CLI無限期捕獲資料包。使用CTRL+C退出捕獲模式。

```
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 (C
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 (C
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 (C
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 (C
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:f
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 (C
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 (C
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 (C
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:5
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 (C
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:f
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:5
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 (C
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 (C
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 (C
15 packets captured
```

您還可以將過濾器與Ethanalyzer配合使用，以專注於特定流量。乙烷醇過濾器有兩種型別，它們稱為捕獲過濾器和顯示過濾器。捕獲過濾器只捕獲與捕獲過濾器中所定義條件匹配的流量。顯示過濾器仍會捕獲所有流量，但僅顯示與顯示過濾器中所定義標準匹配的流量。

```
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
```

您還可以使用詳細資訊選項捕獲資料包，並在終端中檢視它們，這與您在Wireshark中的操作方法類似。這樣您就可以根據資料包丟棄結果檢視完整的報頭資訊。例如，如果幀已加密，您將無法看到加密的有效負載。請參閱此範例：

```
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
```

使用Ethanalyzer，您可以：

- 將輸出 (PCAP檔案) 寫入各種目標檔案系統上的指定檔案名稱：bootflash、logflash、USB等。
- 然後，您可以將儲存的檔案傳輸至裝置外部，並根據需要在Wireshark中檢視。
- 從bootflash讀取檔案並在終端上顯示。就像直接從CPU介面讀取一樣，如果使用detail關鍵字，您還可以顯示完整的資料包資訊。

如需各種介面來源和輸出選項，請參閱以下範例：

```
Nexus9000_A# ethanalyzer local interface mgmt capture-filter "host 10.82.140.107" write bootflash:TEST.
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
```

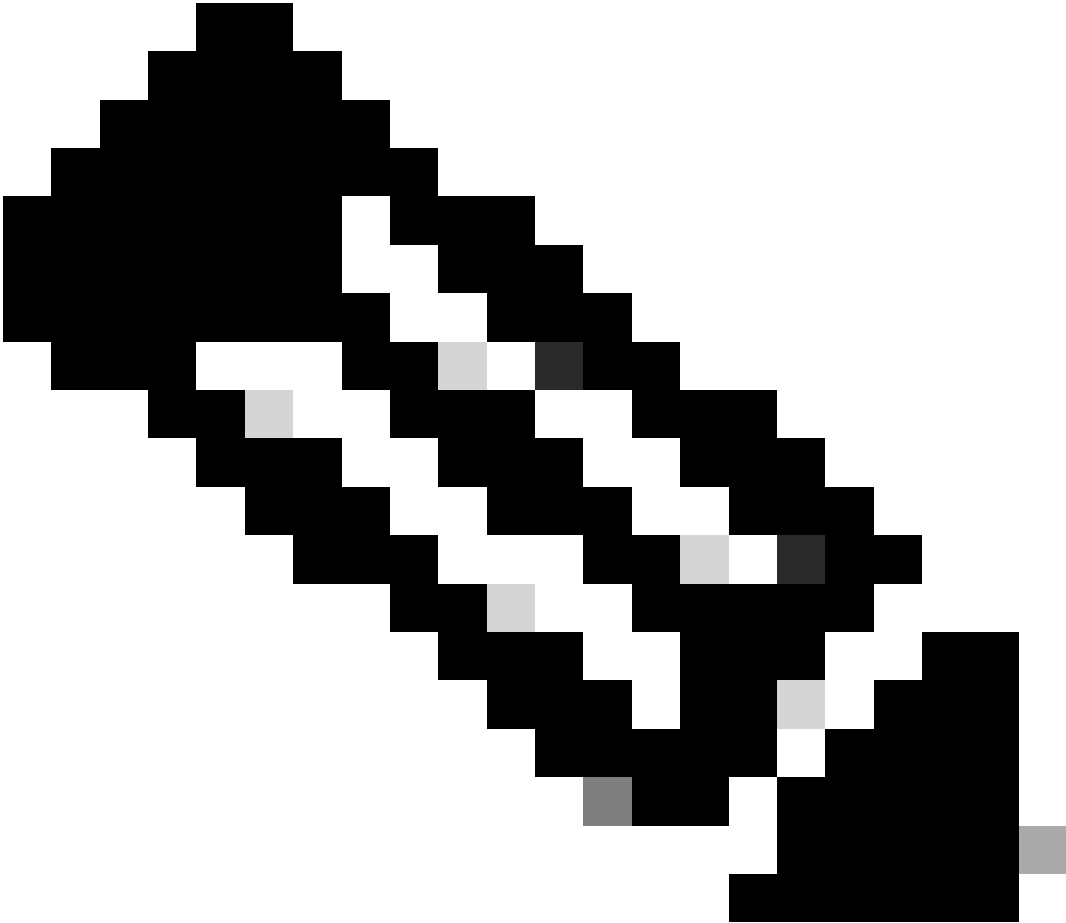
```
RTP-SUG-BGW-1# ethanalyzer local read bootflash:e1-1.pcap detail
Frame 1: 53 bytes on wire (424 bits), 53 bytes captured (424 bits) on interface Eth1-1, id 0
  Interface id: 0 (Eth1-1)
    Interface name: Eth1-1
      Encapsulation type: Ethernet (1)
        Arrival Time: Jul 15, 2022 19:59:50.696219656 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]
```

範圍

交換器連線埠分析器(SPAN)用於擷取來自介面的所有流量，並將該流量映象到目的地連線埠。目的地連線埠通常會連線到網路分析器工具（例如執行Wireshark的PC），讓您分析透過這些連線埠的流量。您可以對來自單一連線埠或多個連線埠和VLAN的流量執行SPAN。

SPAN作業階段包括來源連線埠和目的地連線埠。來源連線埠可以是乙太網路連線埠（無子介面）、連線埠通道、Supervisor Inband介面，但不能同時作為目的地連線埠。此外，某些裝置（例如9300和9500平台）也支援光纖通路擴充模組(FEX)連線埠。目的地連線埠可以是乙太網路連線埠（存取或中繼）、連線埠通道（存取或中繼），對於某些裝置（例如9300上行鏈路連線埠）也支援，而對於FEX連線埠則不支援。

您可以將多個SPAN作業階段設定為輸入/輸出/雙向。單個裝置支援的SPAN會話總數有限制。例如，Nexus 9000最多支援32個會話，而Nexus 7000僅支援16個會話。您可以在CLI上進行檢查，或參閱所用產品的SPAN設定指南。



注意：每個NX-OS版本、產品型別、支援的介面型別和功能各不相同。請參閱您使用的產品和版本的最新配置指南和限制。

以下是Nexus 9000和Nexus 7000的連結：

[Cisco Nexus 9000系列NX-OS系統管理配置指南，版本9.3\(x\) -配置SPAN \[Cisco Nexus 9000系列交換機\] -思科](#)

[Cisco Nexus 7000系列NX-OS系統管理配置指南-配置SPAN \[Cisco Nexus 7000系列交換機\] -思科](#)

SPAN作業階段有各種型別。下面列出了一些較為常見的型別：

- 本地SPAN：一種型別的SPAN會話，其中源主機和目標主機均位於交換機的本地。換句話說，設定SPAN作業階段所需的所有組態會套用到單一交換器，也就是來源和目的地主機連線埠所在的同一交換器。
- 遠端SPAN (RSPAN)：來源和目的地主機不在交換器本地的一種SPAN作業階段。換句話說，您要在同一台交換器上設定來源RSPAN作業階段，並在目的地交換器上設定目的地RSPAN，並使用RSPAN VLAN擴充連線。



注意：Nexus不支援RSPAN。

-
- 擴充遠端SPAN (ERSPAN)：交換器使用GRE (通用路由封裝) 通道標頭封裝複製的訊框，並將封包路由到設定的目的地。您可以在封裝和解除封裝交換器 (兩種不同的裝置) 上設定來源和目的地作業階段。這使我們能夠跨越第3層網路的SPAN流量。
 - SPAN到CPU：指定給目的地連線埠為Supervisor或CPU的特殊型別SPAN作業階段的名稱。這是本地SPAN作業階段的形式，可用於無法使用標準SPAN作業階段的情況。一些常見原因包括：沒有可用或適當的SPAN目的地連線埠、月台無法存取或不受管理的月台、沒有可連線到SPAN目的地連線埠的可用裝置等。有關詳細資訊，請參閱此連結[Nexus 9000雲擴展ASIC NX-OS SPAN到CPU過程- Cisco](#)。請務必記住，SPAN到CPU的速率受到控制平面管制 (CoPP)的限制，因此超過監察器的 sniffing 一個或多個源介面可能導致SPAN到CPU會話的丟棄。如果發生這種情況，資料並非100%反映線路上的內容，因此SPAN到CPU並不總是適用於資料速率高和/或間斷性遺失的疑難排解案例。配置SPAN到CPU會話並管理啟用後，您需要運行Ethanalyzer檢視傳送到CPU的流量以便相應地執行分析。

以下示例說明如何在Nexus 9000交換機上配置簡單的本地SPAN會話：

```
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
```

本示例顯示已啟動的SPAN到CPU會話配置，然後使用Ethanalyzer捕獲流量：

```
<#root>
```

```
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是基於Broadcom的Nexus平台的SPAN到CPU會話型別。其概念與SPAN到CPU相同，且速率限制為50 pps（每秒封包數）。

實現該功能是為了使用bcm-shell CLI調試內部資料路徑。由於相關的限制，沒有NX-OS CLI允許使用者配置到Sup的SPAN會話，因為它可能影響控制流量並使用CoPP類。

伊蘭

嵌入式邏輯分析器模組(ELAM)可以檢視ASIC並確定為單個資料包做出的轉發決策。因此，使用ELAM，您可以確定資料包是否到達轉發引擎以及到達哪些埠/VLAN資訊。您還可以檢查L2 - L4資料包結構，以及資料包是否進行了更改。

必須瞭解ELAM依賴於體系結構，捕獲資料包的過程因平台而異，具體取決於內部體系結構。您必須知道硬體的ASIC對應，才能正確套用工具。對於Nexus 7000，對單個資料包進行兩次捕獲，一次是在做出資料匯流排(DBUS)決策之前，另一次是在做出結果匯流排(RBUS)決策之後。檢視DBUS資訊時，您可以看到接收資料包的內容/位置，以及第2層到第4層資訊。RBUS中的結果可以顯示資料包轉發到的位置，以及幀是否被更改。您需要為DBUS和RBUS設定觸發器，確保它們已準備就緒，然後嘗試即時捕獲資料包。各種線卡的步驟如下：

有關各種ELAM過程的詳細資訊，請參閱下表中的連結：

ELAM概述	ELAM概述-思科
Nexus 7K F1模組	Nexus 7000 F1模組ELAM流程-思科
Nexus 7K F2模組	Nexus 7000 F2模組ELAM流程-思科
Nexus 7K F3模組	F3 - ELAM示例
Nexus 7K M模組	Nexus 7000 M系列模組ELAM流程-思科
Nexus 7K M1/M2和F2模組	Nexus 7K ELAM，用於M1/M2、F2和Ethanalyzer
Nexus 7K M3模組	Nexus 7000 M3模組ELAM流程-思科

適用於Nexus 7000 - M1/M2的ELAM (Eureka平台)

- 使用命令**show module**檢查模組號。
- 使用附加模組x附加到模組，其中x是模組編號。
- 使用命令**show hardware internal dev-port-map**檢查內部ASIC對映，並檢查L2LKP和L3LKP。

<#root>

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

- 一旦狀態顯示觸發程式已準備就緒，便可隨時擷取。此時，您必須透過傳送流量，並再次檢查狀態，檢視觸發器的觸發是否真正被觸發。

<#root>

```
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
```

- 觸發之後，請檢查rbus和dbus的資料包序列號，以確認它們是否捕獲了同一資料包。這可以透過命令show dbus完成|i序列；show rbus|i seq。如果序列號匹配，則可以檢視dbus和rbus的內容。如果不能，請重新運行捕獲，直到能夠捕獲相同的資料包。



注意：為了提高準確性，請始終多次運行ELAM以確認轉發問題。

- 您可以使用命令show dbus和show rbus檢視rbus和dbus的內容。捕獲中的重要資訊是序列號(sequence #)和源/目標索引。Dbus會顯示源索引，它告訴您接收資料包的埠。Rbus顯示資料包轉發到的埠的目標索引。此外，您還可以檢視源和目標IP/MAC地址以及VLAN資訊。
- 有了源索引和目標索引（也稱為LTL索引），您可以使用命令show system internal pixm info ltl #檢查關聯的前面板埠。

適用於Nexus 7000 - M1/M2的ELAM (Lamira平台)

Lamira平台的流程也一樣，但也有一些差異：

- 使用關鍵字Lamiraelam asic lamira instance x運行ELAM。
- 用於觸發ELAM的命令有：

<#root>


```
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 <ife|ofe> ip if elam-match 1
```

- 您使用status命令檢查狀態，並確保它們在傳送資料流之前為「Armed」並在捕獲流量之後觸發。
- 然後，您可以解釋dbus和show bus的輸出，其方式與Eureka的顯示方式類似。

適用於Nexus 7000 - F2/F2E的ELAM (Clipper平台)

同樣，過程是相似的，只是觸發因素不同。少數差異如下：

- 您使用關鍵字Clipperelemasic clipper instance x執行ELAM，並指定第2層或第3層模式。

```
<#root>
```

```
module-4# elam asic clipper instance 1 module-4(clipper-elam)# <layer2/Layer3>
```

- 用於觸發ELAM的命令如下：

```
<#root>
```

```
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
```

- 您使用status命令檢查狀態，並確保它們在傳送資料流之前為Armed，並在捕獲流量之後觸發。
- 然後，您可以解釋dbus和show bus的輸出，其方式與Eureka的顯示方式類似。

適用於Nexus 7000 - F3的ELAM (側翼平台)

同樣，過程相似，只有觸發器不同。少數差異如下：

- 使用關鍵字Flanker **elamasic flanker instance x**運行ELAM並指定第2層或第3層模式。

```
module-4# elamasic flanker instance 1
module-4(flanker-elam)# <layer2/Layer3>
```

- 用於觸發ELAM的命令如下：

<#root>

```
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
```

- 您使用**status**命令檢查狀態，並確保它們在傳送資料流之前為Armed，並在捕獲資料流之後觸發。
- 然後，您可以用類似於Eureka的方式解釋dbus和rbus的輸出。

適用於Nexus 9000的ELAM (Tahoe平台)

在Nexus 9000中，過程與Nexus 7000略有不同。對於Nexus 9000，請參閱[Nexus 9000雲規模ASIC \(Tahoe\) NX-OS ELAM - 思科鏈路](#)

- 首先，使用**show hardware internal tah interface #**命令檢查介面對映。此輸出中最重要資訊是ASIC #、Slice #和source ID (srcid) #。
- 此外，還可以使用**show system internal ethpm info interface #**命令仔細檢查此資訊 | 我src。除了前面列出的值外，此處的重要事項還有dpid和dmod值。
- 使用命令**show module**檢查模組號。
- 使用**attach module x**連線到模組，其中x是模組編號。
- 使用命令**module-1# debug platform internal tah elamasic #**在模組上運行ELAM
- 根據要捕獲的流量型別(L2、L3、封裝的流量 (例如GRE或VXLAN等))，配置內部或外部觸發器：

<#root>

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-inse16)#

reset

module-1(TAH-elam-inse16)#

set outer ipv4 dst_ip 192.0.2.1 src_ip 192.0.2.2

- 設定觸發器後，使用start命令啟動ELAM，傳送資料流，並使用report命令檢視輸出。報告的輸出顯示了傳出和傳入介面以及vlan ID、源和目標IP/MAC地址。

<#root>

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
```

適用於Nexus 9000的ELAM (NorthStar平台)

NorthStar平台的過程與Tahoe平台相同，唯一的區別是在進入ELAM模式時，使用了關鍵字ns而不是tah：

```
<#root>
```

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

N9K Packet Tracer

Nexus 9000 Packet Tracer工具可用於跟蹤資料包的路徑，其內建的流量統計計數器使其成為適用於間斷/完全流量丟失情況的寶貴工具。在TCAM資源有限或無法運行其他工具的情況下，它非常有用。此外，此工具無法捕獲ARP流量，並且不顯示資料包內容（如Wireshark）的詳細資訊。

要配置Packet Tracer，請使用以下命令：

```
<#root>
```

```
N9K-9508#
```

```
test packet-tracer src_ip <src_ip> dst_ip <dst_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
```

有關詳細資訊，請參閱鏈路[Nexus 9000 : Packet Tracer工具介紹- Cisco](#)

Traceroute和Ping

這些命令是兩個最有用的命令，可用於快速確定連線問題。

Ping操作使用Internet控制消息協定(ICMP)將ICMP回應消息傳送到特定目標，並等待該目標的ICMP回應應答。如果主機之間的路徑工

作正常且沒有問題，您可以看到應答返回，ping操作成功。預設情況下，ping命令傳送5x ICMP回應消息（兩個方向的大小相等），如果一切運行正常，您可以看到5x ICMP回應應答。有時，當交換機在地址解析協定(ARP)請求期間獲知MAC地址時，初始回應請求會失敗。如果之後立即再次運行ping，則不會出現初始ping丟失。此外，還可以使用以下關鍵字設定ping次數、資料包大小、源、源介面和超時間隔：

```
<#root>
```

```
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用於辨識資料包在到達目的地之前經過的各種跳。這是一個非常重要的工具，因為它有助於確定發生故障的L3邊界。您也可以將連線埠、來源和來源介面與下列關鍵字搭配使用：

```
<#root>
```

```
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
```

PAACL/RACL/VACL

訪問控制清單(ACL)是一個重要工具，允許您根據相關定義的標準過濾流量。在ACL中填入符合條件的條目後，可以應用它來捕獲入站或出站流量。ACL的一個重要方面是能夠提供流量統計資訊的計數器。術語PAACL/RACL/VACL是指這些ACL的各種實施，它允許您使用ACL作為功能強大的故障排除工具，特別是用於間歇性流量丟失。以下簡要介紹這些術語：

- 埠訪問控制清單(PAACL)：將訪問清單應用到L2交換機埠/介面時，該訪問清單稱為PAACL。
- 路由器訪問控制清單(RACL)：將訪問清單應用到L3路由埠/介面時，該訪問清單稱為RACL。
- VLAN訪問控制清單(VACL)：您可以將VACL配置為應用於路由進或路由出VLAN或在VLAN內橋接的所有資料包。VACL嚴格用於安全資料包過濾器並將流量重定向到特定物理介面。VACL不是由方向（入口或出口）定義的。

下表提供ACL版本之間的比較。

ACL型別	PAACL	RACL	VACL
功能	過濾在L2介面上接收的流量。	過濾在L3介面上接收的流量	過濾VLAN流量
套用於	<ul style="list-style-type: none"> - L2介面/埠。 - L2埠通道介面。 - 如果應用於中繼埠，ACL將過濾該中繼埠上允許的所有VLAN上的流量。 	<ul style="list-style-type: none"> - VLAN介面。 - 物理L3介面。 - L3子介面。 - L3埠通道介面。 - 管理介面。 	啟用後，ACL將應用到該VLAN中的所有埠（包括中繼埠）。
套用的方向	僅限入站。	入站或出站	-

以下是您可以如何設定存取清單的範例。有關詳細資訊，請參閱[Cisco Nexus 9000系列NX-OS安全配置指南9.3\(x\)版-配置IP ACL \[Cisco Nexus 9000系列交換機\]-思科](#)

```
<#root>
```

```
Nexus93180(config)#
```

```
ip access-list <Name_of_ACL>
```

```
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 <NAME_of_ACL> ? >>>>> When you configure ACL like this, it
in Inbound packets
```

```
Nexus93180(config-if)# ip access-group <NAME_of_ACL> ? >>>>> When you configure ACL like this, it
in Inbound packets
out Outbound packets
```

LOGFLASH

LogFlash是Nexus平台上提供的一種永久儲存型別，可用作外部快閃記憶體、USB裝置或Supervisor中的嵌入式磁碟。如果從交換機上刪除，系統會定期通知使用者缺少LogFlash。Logflash安裝在管理引擎上，儲存歷史資料，如記帳日誌、系統日誌消息、調試和內嵌事件管理器(EEM)輸出。本文稍後將討論EEM。您可以使用以下命令檢查LogFlash的內容：

```
<#root>
```

```
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

如果使用者重新載入裝置，或者裝置因某一事件而突然自行重新載入，所有日誌資訊都將丟失。在這種情況下，LogFlash可提供歷史資料，您可以檢視這些資料以確定問題的可能原因。當然，還需要進行進一步的盡職調查，以確定根本原因，從而提供您一些提示，告訴您如果再次發生此事件，需要尋找什麼。

有關如何在裝置上安裝logflash的資訊，請參閱[Nexus 7000日誌記錄功能- Cisco](#)連結。

OBFL

板載故障記錄(OBFL)是Nexus Top of Rack和模組化交換機都可用的永久儲存型別。與LogFlash一樣，在重新載入裝置後，資訊也會保留。OBFL會儲存失敗和環境資料等資訊。資訊因平台和模組而異，但以下是Nexus 93108平台模組1的輸出示例 (即僅有一個模組的固定機箱) :

<#root>

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

同樣，當使用者故意重新載入裝置或由於觸發重新載入的事件而重新載入裝置時，此資訊也非常有用。在這種情況下，OBFL資訊可幫助從板卡的角度確定問題所在。命令**show logging onboard**是一個很好的起點。請記住，您必須從模組內容內擷取，才能取得您需要的所有內容。確保使用**show logging onboard module x**或**attach mod x ; show logging onboard**。

事件記錄

事件歷史記錄是功能強大的工具之一，它可以為您提供有關在Nexus上運行的進程發生的各種事件的資訊。換句話說，在Nexus平台上運行的每個進程都有在後台運行的事件歷史記錄，並儲存有關該進程各種事件的資訊（將它們視為持續運行的調試）。這些事件記錄是非永久性的，並且在重新載入裝置時儲存的所有資訊都會丟失。當您發現特定流程的問題並要對該流程進行故障排除時，這些功能非常有用。例如，如果您的OSPF路由協定不能正常工作，您可以使用與OSPF關聯的事件歷史記錄來確定OSPF進程發生故障的位置。您可以找到與Nexus平台上的幾乎每個進程（如CDP/STP、UDLD、LACP/OSPF、EIGRP/BGP等）關聯的事件歷史記錄。

這是您通常使用參照範例來檢查流程的事件記錄的方式。每個進程都有多個選項，因此請使用？檢查進程下可用的各種選項。

<#root>

Nexus93180(config)#

```
show <Process> internal event-history ?
```

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
```

調試

調試是NX-OS中功能強大的工具，允許您運行即時故障排除事件並將其記錄到檔案或在CLI中顯示。強烈建議您將偵錯輸出記錄到檔案中，因為這些輸出確實會影響CPU效能。在CLI上直接運行調試之前，請務必謹慎。

調試通常僅在您確定問題為單個進程，並想檢查此進程在即時處理網路中實際流量時的行為方式時運行。您需要根據定義的使用者帳戶許可權啟用調試功能。

與事件歷史記錄一樣，您可以為Nexus裝置上的每個進程（如CDP/STP、UDLD、LACP/OSPF、EIGRP/BGP等）運行調試。

這是您通常執行程式除錯的方式。每個進程都有多個選項，因此請使用？檢查進程下可用的各種選項。

<#root>

```
Nexus93180# debug <Process> ?
```

```
Nexus93180# debug spanning-tree ?
```

```
all    Configure all debug flags of stp
bpdurx Configure debugging of stp bpdurx
bpdutx Configure debugging of stp bpdutx
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

```

金牌

顧名思義，通用線上診斷(GOLD)通常用於系統狀況檢查，並用於檢查或驗證有問題的硬體。根據所使用的平台進行了各種線上測試，其中一些測試具有破壞性，而另一些測試則不具有破壞性。這些線上測試可分為以下幾類：

- 開機診斷：這些測試是在裝置開機時執行的測試。他們還檢查管理引擎和模組之間的連通性，包括所有ASIC的資料和控制平面之間的連通性。ManagementPortLoopback和EOBCLoopback等測試會中斷，而OBFL和USB測試則不會中斷。
- 運行時或運行狀況監控診斷：這些測試提供有關裝置運行狀態的資訊。這些測試是無中斷的，可在後台運行，以確保硬體的穩定性。您可以根據需要或出於故障排除目的啟用/停用這些測試。
- 隨選診斷：上述所有測試均可依選重新執行，以便在地化問題。

使用以下命令，可以檢查交換機上可用的各種型別的線上測試：

```

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)

ID	Name	Attributes	Testing Interval (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-

```

若要顯示上述17項測試的作用，您可以使用此命令：

```
<#root>
```

```
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

內嵌式事件管理員(EEM)是一種強大的工具，可讓您設定裝置，以便在發生特定事件時執行特定工作。它會監視裝置上的各種事件，然後採取必要的操作來排查問題並可能恢復。EEM包括三個主要組成部分，下面簡要介紹每個組成部分：

- Event Statement (事件語句) : 這些事件是要監控並希望Nexus執行特定操作 (例如採取應急措施、通知SNMP伺服器或顯示CLI日誌等) 的事件。
- 操作語句 : 這些是EEM在觸發事件後採取的步驟。這些動作可以只用來停用介面或執行某些show指令，將輸出複製到ftp伺服器上的檔案、傳送電子郵件等。
- 策略 : 它基本上是一個事件，它結合了一個或多個action語句，您可以透過CLI或bash指令碼在Supervisor上配置這些語句。您也可以使用python指令碼呼叫EEM。一旦在Supervisor上定義了策略，它就會將策略推送到相關模組。

有關EEM的詳細資訊，請參閱[Cisco Nexus 9000系列NX-OS系統管理配置指南9.2\(x\)版-配置嵌入式事件管理器\[Cisco Nexus 9000系列交換機\]-思科](#)連結。

關於此翻譯

思科已使用電腦和人工技術翻譯本文件，讓全世界的使用者能夠以自己的語言理解支援內容。請注意，即使是最佳機器翻譯，也不如專業譯者翻譯的內容準確。Cisco Systems, Inc. 對這些翻譯的準確度概不負責，並建議一律查看原始英文文件（提供連結）。