Solucionar problemas de hoja de referencia de Nexus para principiantes

Contenido

[Introducción](#)
[Overview](#)
[Herramientas Nexus](#)
[Etanizador](#)
[TRAMO](#)
[Dmirror](#)
[ELAM](#)
[N9K Packet Tracer](#)
[Traceroute y Pings](#)
[PACL/RACL/VACL](#)
[OBFL](#)
[Historiales de eventos](#)
[Depuraciones](#)
[EEM](#)

Introducción

Este documento describe las diferentes herramientas disponibles para resolver problemas de los productos Nexus que puede utilizar para diagnosticar y solucionar un problema.

Overview

Es importante comprender qué herramientas están disponibles y en qué escenario las utilizaría para obtener el máximo beneficio. De hecho, a veces una cierta herramienta no es factible simplemente porque está diseñada para trabajar en otra cosa.

Esta tabla recopila las distintas herramientas para solucionar problemas de la plataforma Nexus y sus funciones. Para obtener más información y ejemplos de CLI, consulte la sección Herramientas de Nexus.

HERRAMIENTAS	FUNCIÓN	EJEMPLO DE CASOS PRÁCTICOS	PROS	CONS	PERSISTENCIA	PLANO EFECTUADO	COMANDOS DE UTILIZADOS
Etanizador	Capturar el tráfico destinado a o desde la CPU	Problemas de lentitud, latencia y congestión del tráfico	Excelente para problemas de lentitud, congestión y latencia	Normalmente, solo ve tráfico del plano de control, velocidad limitada	N/A	Plano de control. Se puede utilizar	#ethanalyzer int local en banda #ethanalyzer loc interface [interfa ID] display filter [WORD]

						para el plano de datos en algunos escenarios (SPAN a CPU)	ejemplo: #ethanalyzer loc interface Ethern display filter ICM
TRAMO	Capture y refleje un grupo de paquetes	Error ping s, paquetes fuera de servicio, etc.	Excelente para pérdida intermitente de tráfico	Requiere un dispositivo externo que ejecute software de sabueso Requiere recursos TCAM	La sesión SPAN debe configurarse y activarse/ desactivarse	Control + Datos	#monitor sessio #description [NA #source interfac [port ID] #destin interface [port ID shut
EspejoDM	Capturar tráfico destinado a la CPU o procedente de ella solo para dispositivos Broadcom Nexus	Problemas de lentitud, latencia y congestión del tráfico	Excelente para problemas de lentitud, congestión y latencia	Solo para dispositivos Broadcom Nexus. Velocidad limitada (CloudScale Nexus 9000 no tiene SPAN a CPU)	N/A	Plano de control. Se puede utilizar para el plano de datos en algunos escenarios	Varía según la plataforma; cons Descripción ger de ELAM: Cisco
ELAM	Captura un único paquete que entra [o sale, si Nexus 7K] en el switch Nexus	Verifique que el paquete llega al Nexus, verifique las decisiones de reenvío, verifique si el paquete presenta alteraciones, verifique la interfaz/VLAN del paquete, etc	Excelente para problemas de reenvío y flujo de paquetes. No intrusivo	Requiere un conocimiento profundo del hardware. Utiliza mecanismos de activación exclusivos específicos de cada arquitectura. Útil sólo si sabe qué tráfico desea inspeccionar	N/A	Control + Datos	# attach module [MODULE NUM # debug platform internal <>
Packet Tracer Nexus	Detectar la ruta del	Problemas de conectividad y pérdida de	Proporciona un contador para	No se puede capturar tráfico ARP. Solo	N/A	Datos y control	# test packet-tra src_IP [SOURC dst_IP

9000	paquete	paquetes	estadísticas de flujo útiles para pérdidas intermitentes /completas. Perfecto para tarjetas de línea sin grabados TCAM	funciona para Nexus 9000			[DESTINATION test packet-trace start # test pack tracer stop # tes packet-tracer sh
Traceroute	Detectar la trayectoria del paquete con respecto a los saltos L3	pings con errores, no se puede alcanzar el host/destino/Internet, etc.	Detecta los diversos saltos en la trayectoria para aislar fallas L3.	Sólo identifica dónde se ha roto el límite L3 (no identifica el problema en sí)	N/A	Datos y control	# traceroute [IP DESTINO] Los argumentos incluyen: puerto, número puerto, origen, interfaz, vrf, inte de origen
Ping	Probar la conectividad entre dos puntos de una red	Probar la disponibilidad entre dispositivos	Una herramienta rápida y sencilla para probar la conectividad	Solo identifica si el host es accesible o no	N/A	Datos y control	# ping [IP DE DESTINO] Los argumentos incluyen: count, packet-si source interface interval, multica loopback, timeo
PACL/RACL/VACL	Capturar la entrada/salida del tráfico de un puerto o VLAN determinados	Pérdida intermitente de paquetes entre hosts, confirmar si los paquetes llegan o salen del Nexus, etc	Excelente para pérdida intermitente de tráfico	Requiere recursos TCAM. Para algunos módulos se requiere tallado manual TCAM	Persistente (aplicado a running-configuración)	Datos y control	# ip access-list [NAME] # ip port access-group [A NAME] # ip acc group [ACL NA Los argumentos incluyen: deny, fragments permit, remark, statistics, end, e pop, push, wher
LogFlash	Almacena datos históricos para el switch de forma global, como registros de cuentas, archivos de bloqueo	La recarga/apagado repentino del dispositivo, cada vez que se recarga un dispositivo, los datos flash del registro proporcionan cierta información que puede ser útil en el análisis	La información se conserva en la recarga del dispositivo (almacenamiento persistente)	Externo en Nexus 7000 = se debe instalar/integrar en la plataforma del supervisor para recopilar estos registros (con no se aplica a 3K/9K ya que logflash es una partición del	Persistente a la recarga	Datos y control	# dir logflash:

	y eventos independientemente de la recarga del dispositivo			dispositivo de almacenamiento interno)			
OBFL	Almacena datos históricos en un módulo específico como, por ejemplo, información sobre fallos y entorno	La recarga/apagado repentino del dispositivo, cada vez que se recarga un dispositivo, los datos flash del registro proporcionan cierta información que puede ser útil	La información se conserva en la recarga del dispositivo (almacenamiento persistente)	Admite un número limitado de lecturas y escrituras	Persistente a la recarga	Datos y control	# show logging onboard module Los argumentos incluyen: boot-uptime, ca boot-history, ca first-power-on, counter-stats, d version, endtime environment-his error-stats, exception-log, internal, interru stats, obfl-histor stat-trace, startt status
Historiales de eventos	Cuando necesite información para un proceso específico que se está ejecutando actualmente	Cada proceso de nexus tiene sus propios historiales de eventos, como CDP, STP, OSPF, EIGRP, BGP, vPC, LACP, etc	Solucionar problemas de un proceso específico que se ejecuta en Nexus	La información se pierde una vez que se vuelve a cargar el dispositivo (no persistente)	No persistente	Datos y control	# show [PROCE internal event-hi [ARGUMENT] Los argumentos incluyen: Adyacencia, cli, evento, inundac ha, hello, ldp, ls msgs, objstore, redistribution, ri segrt, spf, spf-tr statistics, te
Depuraciones	Cuando necesite información en tiempo real y en directo más granular para un	Se puede realizar una depuración de cada proceso de nexus, como CDP, STP, OSPF, IGRP, BGP, vPC, LACP, etc	Solucionar problemas de un proceso específico que se ejecuta en Nexus en tiempo real para obtener	Puede afectar al rendimiento de la red	No persistente	Datos y control	# debug proces [PROCESS] ejemplo: # debug ip ospf

	proceso específico		una mayor granularidad				
ORO	Proporciona diagnósticos de inicio, tiempo de ejecución y bajo demanda en componentes de hardware (como módulos de supervisor y E/S)	Pruebe hardware como USB, Bootflash, OBFL, memoria ASIC, PCIE, loopback de puerto, NVRAM, etc	Puede detectar fallos en el hardware y tomar las medidas correctivas necesarias solo en la versión 6(2)8 y posteriores	Solo detecta problemas de hardware	No persistente	N/A	# show diagnos content module show diagnostic description mod [#] test all
EEM	Supervise los eventos en el dispositivo y realice las acciones necesarias	Cualquier actividad del dispositivo que requiera alguna acción/solución alternativa/notificación como el apagado de la interfaz, el mal funcionamiento del ventilador, el uso de la CPU, etc	Admite scripts Python	Debe tener privilegios de administrador de red para configurar EEM	El script EEM y el disparador residen en la configuración	N/A	Varía; consulte Configuración d Embedded Even Manager

Herramientas Nexus

Si necesita más información sobre varios comandos y su sintaxis u opciones, consulte Switches Nexus de Cisco serie 9000 - Referencias de comandos - Cisco.

- **Etanizador**

Ethanalyzer es una herramienta NX-OS diseñada para capturar paquetes y tráfico de CPU. Cualquier cosa que llegue a la CPU, ya sea entrada o salida, puede ser capturada con esta herramienta. Se basa en el ampliamente utilizado analizador de protocolos de red de código

abierto Wireshark. Para obtener más detalles sobre esta herramienta, consulte la Guía de solución de problemas de Ethanalyzer en Nexus 7000 - Cisco

Es importante tener en cuenta que generalmente, Ethanalyzer captura todo el tráfico hacia y desde el supervisor, es decir, no soporta capturas específicas de la interfaz. Existen mejoras de interfaz específicas para determinadas plataformas en puntos de código más recientes. Además, Ethanalyzer solo captura el tráfico conmutado por CPU, no por hardware conmutado. Por ejemplo, puede capturar el tráfico en la interfaz dentro de la banda, la interfaz de administración o un puerto del panel frontal (donde sea compatible):

```
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
```

Este resultado muestra pocos de los mensajes que se pueden capturar con Ethanalyzer. Tenga en cuenta que, de forma predeterminada, Ethanalyzer solo captura hasta 10 paquetes. Sin embargo, puede utilizar este comando para solicitar a la CLI que capture paquetes indefinidamente. Utilice CTRL+C para salir del modo de captura.

```
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
```

También puede utilizar filtros con Ethanalyzer para centrarse en tráfico específico. Hay dos tipos de filtros que puede utilizar con ethanalzyer, que se conocen como filtros de captura y filtros de visualización. Un filtro de captura solo captura el tráfico que coincide con los criterios definidos en el filtro de captura. Un filtro de visualización todavía captura todo el tráfico, pero sólo se muestra el tráfico que coincide con los criterios definidos en el filtro de visualización.

```
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
```

También puede capturar paquetes con la opción de detalle y verlos en su terminal, de manera similar a como lo haría en Wireshark. Esto le permite ver la información completa del encabezado basada en el resultado del disector de paquetes. Por ejemplo, si una trama está cifrada, no podrá ver la carga cifrada. Observe este ejemplo:

```
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
```

Con Ethanalyzer puede:

- Escriba el resultado (un archivo PCAP) en el nombre de archivo especificado en varios sistemas de archivos de destino: bootflash, logflash, USB, etc... A continuación, puede transferir el archivo guardado fuera del dispositivo y verlo en Wireshark, según sea necesario.
- Lea un archivo de bootflash y muéstrelo en su terminal. Al igual que cuando lee directamente desde la interfaz de la CPU, también puede mostrar la información completa del paquete si utiliza la palabra clave detail.

Vea ejemplos de esto para diversas fuentes de interfaz y opciones de salida:

```
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]
```

- TRAMO

SPAN significa analizador de puertos de switch y se utiliza para capturar todo el tráfico de una interfaz y duplicar ese tráfico a un puerto de destino. El puerto de destino normalmente se conecta a una herramienta de análisis de red (como un PC que ejecuta Wireshark) que le permite analizar el tráfico que atraviesa esos puertos. Puede SPAN para el tráfico de un solo puerto o de varios puertos y VLAN.

Las sesiones SPAN incluyen un puerto de origen y un puerto de destino. Un puerto de origen puede ser un puerto Ethernet (sin subinterfaces), canales de puerto e interfaces en banda del supervisor y no puede ser un puerto de destino simultáneamente. Además, para algunos

dispositivos como la plataforma 9300 y 9500, también se admiten puertos FEX (Fabric Extender). Un puerto de destino puede ser un puerto Ethernet (de acceso o troncal), un canal de puerto (de acceso o troncal) y, en algunos dispositivos, como los puertos de enlace ascendente 9300, también se admiten mientras que los puertos FEX no se admiten como destino.

Puede configurar varias sesiones SPAN para que sean de entrada/salida/ambas. Hay un límite en el número total de sesiones SPAN que puede admitir un dispositivo individual. Por ejemplo, un Nexus 9000 puede admitir hasta 32 sesiones, mientras que un Nexus 7000 solo admite 16. Puede comprobarlo en la CLI o consultar las guías de configuración de SPAN del producto que utilice.

Tenga en cuenta que, para cada versión de NX-OS y el tipo de producto, los tipos de interfaces compatibles y la funcionalidad difieren. Consulte las directrices y limitaciones de configuración más recientes para el producto y la versión que utiliza. Estos son los enlaces para Nexus 9000 y Nexus 7000, respectivamente:

Guía de configuración de la gestión del sistema NX-OS de Cisco Nexus serie 9000, versión 9.3(x) - Configuración de SPAN [switches Nexus de Cisco serie 9000] - Cisco

Guía de configuración de la gestión del sistema NX-OS de Cisco Nexus serie 7000 - Configuración de SPAN [switches Nexus de Cisco serie 7000] - Cisco

Hay varios tipos de sesiones SPAN. A continuación se enumeran algunos de los tipos más comunes:

- SPAN local: tipo de sesión SPAN en la que tanto el host de origen como el de destino son locales para el switch. En otras palabras, toda la configuración necesaria para configurar la sesión SPAN se aplica a un solo switch, el mismo switch donde residen los puertos host de origen y destino.
- SPAN remoto (RSPAN): tipo de sesión SPAN en la que el host de origen y de destino no son locales para el switch. En otras palabras, usted configura las sesiones RSPAN de origen en un switch y RSPAN de destino en el switch de destino y extiende la conectividad con la VLAN RSPAN.

Nota: Nexus no admite RSPAN

- SPAN remoto extendido (ERSPAN): El switch encapsula la trama copiada con un encabezado de túnel GRE (Generic Routing Encapsulation) y enruta el paquete al destino configurado. Las sesiones de origen y destino se configuran en los switches de encapsulación y desencapsulación (dos dispositivos diferentes). Esto nos ofrece la capacidad de SPAN del tráfico en una red de capa 3.
- SPAN-to-CPU: nombre dado a un tipo especial de sesión SPAN donde el puerto de destino es el supervisor o la CPU. Es una forma de sesión SPAN local y se puede utilizar en casos en los que no se puede utilizar una sesión SPAN estándar. Algunas de las razones más comunes son: no hay puertos de destino SPAN disponibles o adecuados, no se puede acceder al sitio o no se puede administrar el sitio, no hay dispositivos disponibles que se puedan conectar al puerto de destino SPAN, etc. Para obtener más información, consulte este enlace Procedimiento de SPAN a CPU de ASIC NX-OS de la escala de nube de Nexus 9000 - Cisco. Es importante recordar que la velocidad de SPAN a CPU está limitada por CoPP (Control Plane Policing), por lo que sniffing una o más interfaces de origen que exceden el regulador pueden dar lugar a caídas para la sesión SPAN a CPU. Si esto sucede, los datos no reflejan al 100% lo que está en el cable, por lo que SPAN a CPU no siempre es apropiado

para la resolución de problemas en escenarios con alta velocidad de datos y/o pérdida intermitente. Una vez que configure un SPAN para la sesión de CPU y lo habilite administrativamente, debe ejecutar Ethanalyzer para ver el tráfico que se envía a la CPU para realizar el análisis correspondiente.

Este es un ejemplo de cómo puede configurar una sesión SPAN local simple en un switch Nexus 9000:

```
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
```

Este ejemplo muestra la configuración de una sesión SPAN a CPU que se ha activado y, a continuación, el uso de Ethanalyzer para capturar el tráfico:

```
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 es un tipo de sesión SPAN-TO-CPU para plataformas Nexus basadas en Broadcom. El concepto es el mismo que para SPAN a CPU y su velocidad está limitada a 50 pps (paquetes por segundo). La función se implementó para depurar la ruta de datos interna con la CLI de bcm-shell. Debido a las limitaciones asociadas, no existe una CLI de NX-OS que permita a los usuarios configurar sesiones SPAN para el supervisor, ya que puede afectar al tráfico de control y consumir clases CoPP.

- ELAM

ELAM significa módulo Embedded Logic Analyzer. Proporciona la capacidad de examinar el ASIC y determinar qué decisiones de reenvío se toman para un **SOLO** paquete. Por lo tanto, con ELAM puede identificar si el paquete alcanza el motor de reenvío y en qué información de puertos/VLAN. También puede verificar la estructura de paquetes L2 - L4 y si se realizaron alteraciones en el paquete o no.

Es importante comprender que ELAM depende de la arquitectura y que el procedimiento para capturar un paquete varía de una plataforma a otra en función de la arquitectura interna. Debe conocer las asignaciones ASIC del hardware para aplicar correctamente la herramienta. Para Nexus 7000, se toman dos capturas para un solo paquete, una antes de tomar la decisión **Data BUS (DBUS**) y otra después de tomar la decisión **Result BUS (RBUS)**. Al ver la información de DBUS, puede ver qué/dónde se recibió el paquete, así como la información de la capa 2 a la 4. Los resultados de la RBUS pueden mostrar a dónde se reenvía el paquete y si se alteró la trama. Debe configurar disparadores para DBUS y RBUS, asegurarse de que estén listos y luego intentar capturar el paquete en tiempo real. Los procedimientos para varias tarjetas de línea son los siguientes:

Para obtener más información sobre los distintos procedimientos de ELAM, consulte los enlaces de esta tabla:

DESCRIPCIÓN GENERAL DE ELAM	Descripción general de ELAM: Cisco
Módulo Nexus 7K F1	Procedimiento ELAM del módulo F1 Nexus 7000 - Cisco
Módulo Nexus 7K F2	Procedimiento ELAM del módulo F2 Nexus 7000 - Cisco
Módulo Nexus 7K F3	Ejemplo de F3- ELAM
Módulo Nexus 7K M	Procedimiento ELAM del módulo Nexus serie 7000 M - Cisco
Módulo Nexus 7K M1/M2 y F2	ELAM Nexus 7K para M1/M2 y F2 y Ethanalyzer
Módulo Nexus 7000 M3	Procedimiento ELAM del módulo Nexus 7000 M3 - Cisco

ELAM para Nexus 7000 - M1/M2 (plataforma Eureka)

- Verifique el número de módulo con el comando **show module**.
- Adjunte al módulo con **attach module x**, donde x es el número de módulo.
- Verifique la correspondencia interna de ASIC con el comando **show hardware internal dev-port-map** y verifique si L2LKP y L3LKP.

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

- Primero, captura el paquete en L2 y ve si la decisión de reenvío es correcta. Para hacer esto, observe la columna de mapeos L2LKP e identifique el número de instancia ASIC que corresponde al puerto.
- A continuación, ejecute ELAM en esta instancia con el comando **elam asic eureka instance** xdonde x es el número de instancia de ASIC y configura nuestros disparadores para DBUS y RBUS. Verifique el estado de los disparadores con el comando **status** y confirme que los disparadores se han configurado.

```
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
```

- Active los disparadores con el comando **start** y verifique el estado de los disparadores con el comando **status** para confirmar que los disparadores están armados.

```
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
```

- Una vez que el estado muestra que los disparadores están armados, están listos para capturar. En este momento, debe enviar el tráfico a través de y comprobar el estado de nuevo para ver si sus disparadores fueron realmente disparados.

```
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
```

- Una vez disparado, verifique el número de secuencia de paquetes para rbus y dbus para confirmar que ambos han capturado el mismo paquete. Esto se puede hacer con el comando **show dbus | i seq ; show rbus | i seq**. Si el número de secuencia coincide, puede ver el contenido de dbus y rbus. Si no, vuelva a ejecutar la captura hasta que pueda capturar el mismo paquete.

  Nota: Para mayor precisión, ejecute siempre ELAM varias veces para confirmar los problemas de reenvío.

- Puede ver el contenido de rbus y dbus con los comandos **show dbus** y **show rbus**. Lo importante en la captura es el número de secuencia y el índice de origen/destino. Dbus muestra el índice de origen que le indica el puerto en el que recibió el paquete. Rbus muestra el índice de destino del puerto al que se reenvía el paquete. Además, también puede consultar las direcciones IP/MAC de origen y destino, así como la información de VLAN.
- Con el índice de origen y destino (también conocido como índice LTL), puede verificar el puerto asociado del panel frontal con el comando **show system internal pixm info ltl #**.

ELAM para Nexus 7000 - M1/M2 (plataforma Lamira)

El procedimiento es el mismo para la plataforma Lamira también, sin embargo, hay algunas diferencias:

- Ejecutas ELAM con la palabra clave Lamira **elam asic lamira instance x**.
- Los comandos para activar el ELAM son:

```
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
```

- Usted verifica el estado con el comando **status** y se asegura de que estén Armados antes de enviar tráfico y se activen después de capturarlo.
- A continuación, puede interpretar las salidas de dbus y show bus de manera similar a como se muestra para Eureka.

ELAM para Nexus 7000 - F2/F2E (plataforma Clipper)

De nuevo, el procedimiento es similar, solo los desencadenadores son diferentes. Las pocas diferencias son las siguientes:

- Ejecute ELAM con la palabra clave Clipper **elam asic clipper instance x** y especifique el modo de Capa 2 o Capa 3.

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

- Los comandos para activar el ELAM son los siguientes:

```
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
```

- Usted verifica el estado con el comando **status** y se asegura de que estén Armados antes de enviar tráfico y se activen después de capturarlo.
- A continuación, puede interpretar las salidas de dbus y show bus de manera similar a como se muestra para Eureka.

ELAM para Nexus 7000 - F3 (plataforma Flanker)

De nuevo, el procedimiento es similar, solo los desencadenadores son diferentes. Las pocas diferencias son las siguientes:

- Ejecute ELAM con la palabra clave Flanker **elam asic flanker instance x** y especifique el modo de Capa 2 o Capa 3.

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

- Los comandos para activar el ELAM son los siguientes:

```
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
```

- Usted verifica el estado con el comando **status** y se asegura de que estén Armados antes de enviar tráfico y se activen después de capturarlo.
- A continuación, puede interpretar las salidas de dbus y rbus de manera similar a como se muestra para Eureka.

ELAM para Nexus 9000 (plataforma Tahoe)

En Nexus 9000, el procedimiento es un poco diferente al de Nexus 7000. Para Nexus 9000, consulte el enlace [ASIC de la escala de nube Nexus 9000 (Tahoe) NX-OS ELAM - Cisco]

- Primero, verifique la asignación de la interfaz con el comando **show hardware internal tah interface #** . La información más importante en esta salida es el **ASIC #**, **Slice #** y el **source ID (srcid) #**.
- Además, también puede verificar nuevamente esta información con el comando **show system internal ethpm info interface # | i i src**. Lo importante aquí, además de lo que se enumeró anteriormente, son los valores dpid y dmod.
- Verifique el número de módulo con el comando **show module**.
- Adjunte al módulo con **attach module x**, donde x es el número de módulo.
- Ejecute ELAM en el módulo con el comando **module-1# debug platform internal tah elam asic #**
- Configure su disparador interno o externo en función del tipo de tráfico que desea capturar (L2, L3, tráfico encapsulado como GRE o VXLAN, etc.):

```
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
```

- Una vez que se establecen los disparadores, inicie ELAM con el comando **start**, envíe el tráfico y vea el resultado con el comando **report**. La salida del informe muestra las interfaces de salida y de entrada junto con el ID de vlan, la dirección IP/MAC de origen y de destino.

```
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 para Nexus 9000 (plataforma NorthStar)

El procedimiento para la plataforma NorthStar es el mismo que para la plataforma Tahoe, la única diferencia es que la palabra clave **ns** se utiliza en lugar de **tah** cuando se ingresa en el modo ELAM:

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

- ## N9K Packet Tracer

La herramienta de seguimiento de paquetes Nexus 9000 se puede utilizar para realizar un seguimiento de la ruta del paquete y, con sus contadores integrados para las estadísticas de flujo, se convierte en una herramienta valiosa para los escenarios de pérdida de tráfico intermitente/completa. Sería muy útil cuando los recursos de TCAM son limitados o no están disponibles para ejecutar otras herramientas. Además, esta herramienta no puede capturar el tráfico ARP y no muestra detalles del contenido de los paquetes como Wireshark.

Para configurar packet tracer, utilice estos comandos:

```
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
```

Para obtener más información, consulte el enlace [Nexus 9000: Herramienta Packet Tracer explicada - Cisco]

- ## Traceroute y Pings

Estos comandos son los dos comandos más útiles que le permiten identificar rápidamente problemas de conectividad.

Ping utiliza el protocolo de mensajes de control de Internet (ICMP) para enviar mensajes de eco ICMP al destino específico y espera las respuestas de eco ICMP de ese destino. Si la trayectoria entre el host funciona correctamente sin problemas, puede ver que las respuestas regresan y los pings son exitosos. El comando ping envía de forma predeterminada 5 mensajes de eco ICMP (del mismo tamaño en ambas direcciones) y si todo funciona correctamente, puede ver 5 respuestas de eco ICMP. A veces, la solicitud de eco inicial falla cuando los switches aprenden la dirección MAC durante la solicitud del Protocolo de resolución de direcciones (ARP). Si vuelve a ejecutar el ping inmediatamente después, no se producirá ninguna pérdida de ping inicial. Además, también puede establecer el número de pings, el tamaño del paquete, el origen, la interfaz de origen y los intervalos de tiempo de espera con estas palabras clave:

```
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 se utiliza para identificar los diversos saltos que toma un paquete antes de llegar a su destino. Es una herramienta muy importante porque ayuda a identificar el límite L3 donde ocurre el fallo. También puede utilizar el puerto, el origen y la interfaz de origen con estas palabras clave:

```
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 son las siglas en inglés de Access Control List. Es una herramienta importante que permite filtrar el tráfico según un criterio definido relevante. Una vez que la ACL se llena con entradas para los criterios de coincidencia, se puede aplicar para capturar el tráfico entrante o saliente. Un aspecto importante de ACL es su capacidad de proporcionar contadores para las estadísticas de flujo. Los términos PACL/RACL/VACL se refieren a varias implementaciones de estas ACL que le permiten utilizar ACL como una herramienta de troubleshooting potente especialmente para la pérdida de tráfico intermitente. Estos términos se describen brevemente a continuación:

- PACL significa Port Access Control List: Cuando aplica una lista de acceso a una interfaz/puerto de switch L2, esa lista de acceso se conoce como PACL.
- RACL significa Router Access Control List: Cuando aplica una lista de acceso a un puerto/interfaz ruteado L3, esa lista de acceso se conoce como RACL.
- VACL significa VLAN Access Control List: Puede configurar las VACL para que se apliquen a todos los paquetes que se rutean dentro o fuera de una VLAN o que se puentean dentro de una VLAN. Las VACL son estrictamente para filtros de paquetes de seguridad y para redirigir el tráfico a interfaces físicas específicas. Las VACL no se definen por dirección (entrada o salida).

Esta tabla proporciona una comparación entre las versiones de ACL.

TIPO DE ACL	PACL	RACL	VACL
FUNCIÓN	Filtrar el tráfico recibido en una interfaz L2.	Filtrar el tráfico recibido en una interfaz L3	Filtrar tráfico vLAN
APLICADO EL	- Interfaces/puertos L2. - Interfaces de canal de puerto L2. - Si se aplica en un puerto	- Interfaces VLAN. - Interfaces L3 físicas. - Subinterfaces L3. - Interfaces de canal de puerto	Una vez habilitada, la AC aplica a todos los puertos esa VLAN (incluidos los puertos troncales).

	trunk, la ACL filtra el tráfico en todas las VLAN permitidas en ese puerto trunk.	L3. - Interfaces de gestión.	
DIRECCIÓN APLICADA	Sólo entrante.	Entrante o saliente	-

Este es un ejemplo de cómo puede configurar una lista de acceso. Para obtener más información sobre Cisco, consulte la Guía de configuración de seguridad de NX-OS para Cisco Nexus serie 9000, versión 9.3(x) - Configuración de ACL IP [switches Nexus de Cisco 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 es un tipo de almacenamiento persistente disponible en las plataformas Nexus como CompactFlash externa, dispositivo USB o disco integrado en el supervisor. Si se elimina del switch, el sistema notifica periódicamente al usuario que falta LogFlash. Logflash se instala en el supervisor y contiene datos históricos como registros de cuentas, mensajes de syslog, depuraciones y salidas de Embedded Event Manager (EEM). EEM se discute más adelante en este artículo. Puede verificar el contenido de LogFlash con este comando:

```
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
```

En el caso de que un usuario recargara el dispositivo o se recargara repentinamente por sí mismo debido a un evento, se perdería toda la información de registro. En tales escenarios, LogFlash puede proporcionar datos históricos que se pueden revisar para identificar una causa probable del problema. Por supuesto, se requiere una mayor diligencia debida para identificar la causa raíz que le proporciona pistas sobre qué buscar en caso de que este evento vuelva a ocurrir.

Para obtener información sobre cómo instalar logflash en el dispositivo, consulte el enlace Capacidades de registro de Nexus 7000 - Cisco.

- ## OBFL

OBFL significa Registro de fallos OnBoard. Se trata de un tipo de almacenamiento persistente disponible tanto para switches Nexus top of rack como modulares. Al igual que LogFlash, la información se conserva una vez que se recarga el dispositivo. OBFL almacena información como fallas y datos ambientales. La información varía para cada plataforma y módulo; sin embargo, a continuación se incluye un ejemplo de resultado del módulo 1 de la plataforma Nexus 93108 (es decir, un chasis fijo con un solo módulo):

```
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
```

De nuevo, esta información es útil en el caso de un dispositivo que se recarga a propósito por el usuario o debido a un evento que desencadenó una recarga. En este caso, la información de OBFL puede ayudar a identificar qué falló desde la perspectiva de una tarjeta de línea. El comando **show logging onboard** es un buen punto de partida. Recuerde que debe capturar desde dentro del contexto del módulo para obtener todo lo que necesita. Asegúrese de utilizar **show logging onboard module x** o **attach mod x ; show logging onboard.**

- ## Historiales de eventos

Los historiales de eventos son una de las potentes herramientas que pueden proporcionarle información sobre diversos eventos que tienen lugar para un proceso que se ejecuta en Nexus. En otras palabras, cada proceso que se ejecuta en una plataforma Nexus tiene historiales de eventos que se ejecutan en segundo plano y almacenan información sobre varios eventos de ese proceso (piense en ellos como depuraciones que se ejecutan constantemente). Estos historiales de eventos no son persistentes y toda la información almacenada se pierde al volver a cargar el dispositivo. Estos son muy útiles cuando ha identificado un problema con un proceso determinado y le gustaría resolver ese proceso. Por ejemplo, si su protocolo de ruteo OSPF no funciona correctamente, puede utilizar los historiales de eventos asociados con OSPF para identificar dónde falla el proceso OSPF. Puede encontrar historiales de eventos asociados con casi todos los procesos de la plataforma Nexus, como CDP/STP, UDLD, LACP/OSPF, EIGRP/BGP, etc.

Así es como normalmente se comprueban los historiales de eventos de un proceso con ejemplos de referencia. Cada proceso tiene varias opciones, así que use **?** para comprobar si hay varias opciones disponibles en un proceso.

```
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
```

- Depuraciones

Las depuraciones son herramientas eficaces de NX-OS que permiten ejecutar eventos de solución de problemas en tiempo real y registrarlos en un archivo o mostrarlos en la CLI. Se recomienda encarecidamente registrar las salidas de depuración en un archivo ya que afectan al rendimiento de la CPU. Tenga cuidado antes de ejecutar una depuración directamente en la CLI.

Por lo general, las depuraciones se ejecutan sólo cuando se ha identificado un problema como un proceso único y se desea comprobar cómo se comporta este proceso en tiempo real con el tráfico real de la red. Debe habilitar una función de depuración basada en los privilegios de cuenta de usuario definidos.

Al igual que en los historiales de eventos, puede ejecutar depuraciones para cada proceso en un dispositivo Nexus como CDP/STP, UDLD, LACP/OSPF, EIGRP/BGP, etc.

Así es como normalmente se ejecuta una depuración para un proceso. Cada proceso tiene varias opciones, así que use **?** para comprobar si hay varias opciones disponibles en un proceso.

```
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
```

- ORO

GOLD significa Generic OnLine Diagnostics. Como su nombre indica, estas pruebas se utilizan generalmente como una comprobación del estado del sistema y se utilizan para comprobar o verificar el hardware en cuestión. Hay varias pruebas online que se llevan a cabo y basadas en la plataforma en uso, algunas de estas pruebas son disruptivas mientras que otras no lo son. Estas pruebas online se pueden clasificar de la siguiente manera:

- **Diagnósticos de Arranque:** Estas pruebas son las que se ejecutan cuando el dispositivo se está iniciando. También comprueban la conectividad entre el supervisor y los módulos, lo que incluye la conectividad entre los datos y el plano de control de todos los ASIC. Las pruebas como ManagementPortLoopback y EOBCLoopback provocan interrupciones, mientras que las pruebas para OBFL y USB no provocan interrupciones.
- **Diagnósticos de supervisión de estado o tiempo de ejecución:** Estas pruebas proporcionan información sobre el estado del dispositivo. Estas pruebas no provocan interrupciones y se ejecutan en segundo plano para garantizar la estabilidad del hardware. Puede habilitar/deshabilitar estas pruebas según sea necesario o para solucionar problemas.
- **Diagnóstico a demanda:** Todas las pruebas mencionadas se pueden volver a ejecutar a petición para localizar un problema.

Puede verificar los diversos tipos de pruebas en línea disponibles para su switch con este comando:

```
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-
```

Para mostrar lo que hace cada una de las 17 pruebas mencionadas, puede utilizar este comando:

```
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 significa Embedded Event Manager. Es una potente herramienta que le permite programar su dispositivo para realizar tareas específicas en caso de que ocurra un evento determinado. Supervisa varios eventos en el dispositivo y luego toma las medidas necesarias para resolver el problema y posiblemente recuperarse. EEM consta de tres componentes principales, cada uno de los cuales se describe brevemente aquí:

- **Declaración de evento:** Éstos son los eventos que desea supervisar y que Nexus realice una acción concreta, como una solución alternativa o simplemente notificar a un servidor SNMP o mostrar un registro de CLI, etc.
- **Declaraciones de acción:** Estos serían los pasos que EEM daría una vez que se activa un evento. Estas acciones pueden ser simplemente para inhabilitar una interfaz o ejecutar algunos comandos show y copiar salidas a un archivo en un servidor ftp, enviar un correo electrónico, y así sucesivamente.
- **Políticas:** Básicamente, se trata de un evento combinado con una o más sentencias de acción que puede configurar en el supervisor mediante CLI o una secuencia de comandos bash. También puede invocar EEM con un script de Python. Una vez definida la política en el supervisor, la envía al módulo correspondiente.

Para obtener más información sobre EEM, consulte el enlace Guía de configuración de administración del sistema NX-OS de Cisco Nexus serie 9000, versión 9.2(x) - Configuración de Embedded Event Manager [switches Nexus de Cisco serie 9000] - Cisco.

Acerca de esta traducción
Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).