Troubleshooting de Nexus Cheat Sheet para Iniciantes

Contents

Introduction

Este documento descreve as diferentes ferramentas disponíveis para solucionar problemas de produtos Nexus que você pode utilizar para diagnosticar e corrigir um problema.

Overview

Éimportante entender quais ferramentas estão disponíveis e em que cenário você as usaria para obter o máximo de ganho. Na verdade, algumas vezes uma determinada ferramenta não é viável simplesmente porque é projetada para trabalhar em outra coisa.

Esta tabela compila as várias ferramentas para solucionar problemas na plataforma Nexus e seus recursos. Para obter detalhes e exemplos de CLI, consulte a seção Nexus Tools.

FERRAMENTAS	FUNÇÃO	EXEMPLOS DE CASOS DE USO	PROS	CONS	PERSISTÊNCIA	PLANO AFETADO	COMANDOS CLI USADOS
Ethanalyzer	Capturar o tráfego destinado à CPU ou proveniente dela	Problemas de lentidão de tráfego, latência e congestionamento	Excelente para problemas de lentidão, congestionamento e latência	Geralmente vê apenas o tráfego do plano de controle, taxa limitada	N/A	Plano de controle. Pode ser usado para	#ethanalyzer interface local in #ethanalyzer loc interface [interfa ID] display filter [WORD] exemplo: #ethanalyzer IC

						plano de dados em alguns cenários (SPAN para CPU)	de filtro de exibi de Ethernet 6/4 interface local
SPAN	Capturar e espelhar vários pacotes	Falha ping s, pacotes fora de ordem e assim por diante	Excelente para perda intermitente de tráfego	Requer um dispositivo externo que execute o software farejador Requer recursos TCAM	A sessão de SPAN precisa ser configurada e habilitada/ desabilitada	Controle + Dados	#monitor sessio #description [NA #source interfac [port ID] #destin interface [port ID shut
DMirror	Capturar o tráfego destinado à CPU ou proveniente dela somente para dispositivos Broadcom Nexus	Problemas de lentidão de tráfego, latência e congestionamento	Excelente para problemas de lentidão, congestionamento e latência	Somente para dispositivos Broadcom Nexus. Taxa limitada (o CloudScale Nexus 9k tem SPAN para CPU)	N/A	Plano de controle. Pode ser usado para plano de dados em alguns cenários	Varia de acordo a plataforma, consulte Visão geral do E - Cisco
ELAM	Captura um único pacote que entra [ou sai, se Nexus 7K] do switch Nexus	Verificar se o pacote alcança o Nexus, verificar decisões de encaminhamento, verificar alterações no pacote, verificar interface/VLAN do pacote e assim por diante	Excelente para problemas de fluxo e encaminhamento de pacotes. Não intrusivo	Requer compreensão profunda do hardware. Utiliza mecanismos de acionamento exclusivos que são específicos da arquitetura. Útil somente se você souber qual tráfego deseja inspecionar	N/A	Controle + Dados	# attach module [MODULE NUM # debug platform internal <>
Packet Tracer do Nexus 9k	Detectar caminho do	Problemas de conectividade e perda de pacotes	Fornece contador para	Não é possível capturar o tráfego ARP.	N/A	Dados + control	# test packet-tra src_IP [SOURC dst_IP

	pacote		estatísticas de fluxo úteis para perda intermitente/ completa. Perfeito para placas de linha sem entalhes TCAM	Funciona somente no Nexus 9k		e	[DESTINATION test packet-trace start # test pack tracer stop # tes packet-tracer sh
Traceroute	Detecta o caminho do pacote em relação aos saltos L3	Falha nos pings, não é possível acessar host/destino/inter net e assim por diante	Detecta os vários saltos no caminho para isolar falhas de L3.	Identifica apenas onde o limite L3 foi quebrado (não identifica o problema em si)	N/A	Dados + control e	# traceroute [IP DESTINO] Os argumentos incluem: porta, número d porta, origem, interface, vrf, interface de orig
Ping	Testar a conectivi dade entre dois pontos em uma rede	Testar a acessibilidade entre dispositivos	Uma ferramenta rápida e simples para testar a conectividad e	Apenas identifica se o host está acessível ou não	N/A	Dados + control e	# ping [DESTINATION Os argumentos incluem: contagem, tama do pacote, inter de origem, inter multicast, loopb tempo limite
PACL/RAC L/VACL	Capturar a entrada/ saída de tráfego em uma determin ada porta ou VLAN	Perda intermitente de pacotes entre hosts, confirmar se os pacotes chegam/saem do Nexus e assim por diante	Excelente para perda intermitente de tráfego	Requer recursos TCAM. Para alguns módulos é necessário o entalhe TCAM manual	Persistent e (aplicado a running-configuraç ão)	Dados + control e	# ip access-list [NAME] # ip port access-group [A NAME] # ip acc group [ACL NAM Os argumentos incluem: deny, fragments permit, remark, statistics, end, e pop, push, wher
LogFlash	Armazen a globalme nte dados históricos para o switch, como logs de contas, arquivos de	Recarregamento/ desligamento repentino do dispositivo, sempre que um dispositivo é recarregado, os dados de log flash fornecem algumas informações que podem ser úteis na análise	As informações são retidas durante o recarregame nto do dispositivo (armazenam ento persistente)	Externo no Nexus 7K = Deve ser instalado/integr ado na plataforma do supervisor para que esses registros sejam coletados (con não se aplica a 3K/9K, pois logflash é	Reload-Persistent	Dados + control e	# dir logflash:

	travamento e eventos, independentemente da recarga do dispositivo			uma partição do dispositivo de armazenamento interno)			
OBFL	Armazena dados históricos em um módulo específico, como informações sobre falhas e ambiente	Recarregamento/ desligamento repentino do dispositivo, sempre que um dispositivo é recarregado, os dados de log flash fornecem algumas informações que podem ser úteis	As informações são retidas durante o recarregamento do dispositivo (armazenamento persistente)	Suporta número limitado de leituras e gravações	Reload-Persistente	Dados + controle	# show logging onboard module Os argumentos incluem: boot-uptime, ca boot-history, ca first-power-on, counter-stats, d version, endtime environmental-history, error-sta exception-log, internal, interrup stats, obfl-histor stat-trace, startt status
Histórico de eventos	Quando você precisa de informações para um processo específico que está em execução no momento	Cada processo no Nexus tem seu próprio histórico de eventos, como CDP, STP, OSPF, EIGRP, BGP, vPC, LACP e assim por diante	Solucionar problemas de um processo específico executado no Nexus	As informações são perdidas quando o dispositivo é recarregado (não persistente)	Não Persistente	Dados + controle	# show [PROCE internal event-hi [ARGUMENTO] Os argumentos incluem: Adjacência, cli, evento, inundaç ha, hello, ldp, ls msgs, objstore, redistribuição, ri segrt, spf, spf-tr statistics, te
Debugs	Quando você precisa de informações em tempo real/ao	A depuração em todos os processos no Nexus pode ser feita, como CDP, STP, OSPF, IGRP, BGP, vPC, LACP e assim	Solucionar problemas de um processo específico executado no Nexus em tempo	Pode afetar o desempenho da rede	Não Persistente	Dados + controle	# debug proces [PROCESS] exemplo: # debug ip ospf

	vivo mais granulares es para um processo específico	por diante	real para obter mais granularidade				
OURO	Fornece Inicialização, tempo de execução e diagnósticos sob demanda nos componentes de hardware (como Módulos de E/S e Supervisor)	Teste de hardware como USB, Bootflash, OBFL, memória ASIC, PCIE, loopback de porta, NVRAM e assim por diante	Pode detectar falhas no hardware e tomar as ações corretivas necessárias somente na versão 6(2)8 e posterior	Detecta apenas problemas de hardware	Não Persistente	N/A	# show diagnos content module show diagnostic description mod [#] test all
EEM	Monitorar eventos no dispositivo e tomar as ações necessárias	Qualquer atividade de dispositivo que exija alguma ação/solução/notificação, como desligamento da interface, mau funcionamento do ventilador, utilização da CPU etc.	Suporta scripts Python	Deve ter privilégios de administrador de rede para configurar o EEM	O script e o acionador do EEM residem na configuração	N/A	Varia, consulte Configuração do Gerenciador de Eventos Inserid

Ferramentas Nexus

Se precisar de mais esclarecimentos sobre vários comandos e sua sintaxe ou opções consulte Switches Cisco Nexus 9000 Series - Referências de comando - Cisco.

- **Ethanalyzer**

O Ethanalyzer é uma ferramenta NX-OS projetada para capturar o tráfego da CPU dos pacotes. Qualquer coisa que atinja a CPU, seja de ingresso ou saída, pode ser capturada com essa ferramenta. Ele é baseado no amplamente utilizado analisador de protocolo de rede de código aberto Wireshark. Para obter mais detalhes sobre essa ferramenta, consulte o Guia de solução de

[problemas do Ethanalyzer no Nexus 7000 - Cisco]

Éimportante observar que, em geral, o Ethanalyzer captura todo o tráfego de e para o supervisor, ou seja, ele não suporta capturas específicas de interface. Melhorias específicas de interface estão disponíveis para plataformas selecionadas em pontos de código mais recentes. Além disso, o Ethanalyzer captura apenas o tráfego que é comutado pela CPU, e não pelo hardware. Por exemplo, você pode capturar o tráfego na interface inband, na interface de gerenciamento ou em uma porta do painel frontal (onde houver suporte):

```
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
```

Esta saída mostra algumas das mensagens que podem ser capturadas com o Ethanalyzer. Observe que, por padrão, o Ethanalyzer captura apenas até 10 pacotes. No entanto, você pode usar esse comando para solicitar que a CLI capture pacotes indefinidamente. Use CTRL+C para sair do 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
```

Você também pode usar filtros com o Ethanalyzer para se concentrar em tráfego específico. Há dois tipos de filtros que você pode usar com o ethanalzyer, conhecidos como filtros de captura e filtros de exibição. Um filtro de captura só captura o tráfego que corresponde aos critérios definidos no filtro de captura. Um filtro de exibição ainda captura todo o tráfego, mas somente o tráfego que corresponde aos critérios definidos no filtro de exibição é mostrado.

```
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
```

Você também pode capturar pacotes com a opção detail e visualizá-los em seu terminal, semelhante ao que você faria no Wireshark. Isso permite que você veja as informações completas do cabeçalho com base no resultado do setor do pacote. Por exemplo, se um quadro for criptografado, você não poderá ver o payload criptografado. Veja este exemplo:

```
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
```

Com o Ethanalyzer você pode:

- Grave a saída (um arquivo PCAP) no nome de arquivo especificado em vários sistemas de arquivos de destino: bootflash, logflash, USB, etc... Você pode transferir o arquivo salvo para fora do dispositivo e visualizá-lo no Wireshark, conforme necessário.
- Leia um arquivo do flash de inicialização e exiba-o no terminal. Assim como ao ler diretamente da interface da CPU, você também pode exibir as informações completas do pacote se usar a palavra-chave detail.

Veja exemplos disso para várias fontes de interface e opções de saída:

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

```
        590    Jan 10 14:21:08 2019  MDS20190110082155835.lic
       1164    Feb 18 02:18:15 2020  TEST.PCAP
>>>>>>>Output Clipped

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

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

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

RTP-SUG-BGW-1# ethanalyzer local read bootflash:e1-1.pcap detail
Frame 1: 53 bytes on wire (424 bits), 53 bytes captured (424 bits) on interface Eth1-1, id 0
    Interface id: 0 (Eth1-1)
        Interface name: Eth1-1
    Encapsulation type: Ethernet (1)
    Arrival Time: Jul 15, 2022 19:59:50.696219656 UTC
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1657915190.696219656 seconds
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 0.000000000 seconds]
    Frame Number: 1
    Frame Length: 53 bytes (424 bits)
    Capture Length: 53 bytes (424 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:llc:stp]
```

- SPAN

SPAN significa SwitchPort Analyzer e é usado para capturar todo o tráfego de uma interface e espelhar esse tráfego para uma porta de destino. A porta de destino geralmente se conecta a uma ferramenta de análise de rede (como um PC que executa o Wireshark) que permite analisar o tráfego que atravessa essas portas. Você pode SPAN para tráfego de uma única porta ou de várias portas e VLANs.

As sessões de SPAN incluem uma porta origem e uma porta destino. Uma porta de origem pode ser uma porta Ethernet (sem subinterfaces), canais de porta, interfaces inband do supervisor e não pode ser uma porta de destino simultaneamente. Além disso, para alguns dispositivos como a plataforma 9300 e 9500, portas FEX (Fabric Extender) também são suportadas. Uma porta de destino pode ser uma porta Ethernet (acesso ou tronco), canal de porta (acesso ou tronco) e,

para alguns dispositivos como as portas de uplink 9300, também são suportadas enquanto portas FEX não são suportadas como destino.

Você pode configurar várias sessões de SPAN para serem uma entrada/saída/ambas. Há um limite para o número total de sessões de SPAN que um dispositivo individual pode suportar. Por exemplo, um Nexus 9000 pode suportar até 32 sessões, enquanto um Nexus 7000 pode suportar apenas 16. Você pode verificar isso na CLI ou consultar os guias de configuração de SPAN para o produto que você usa.

Observe que, para cada versão do NX-OS e tipo de produto, os tipos de interfaces e a funcionalidade compatíveis são diferentes. Consulte as últimas diretrizes e limitações de configuração para o produto e a versão que você usa. Estes são os links para o Nexus 9000 e o Nexus 7000, respectivamente:

Guia de configuração de gerenciamento do sistema NX-OS do Cisco Nexus 9000 Series, versão 9.3(x) - configuração de SPAN [switches Cisco Nexus 9000 Series] - Cisco

Guia de configuração de gerenciamento do sistema NX-OS do Cisco Nexus 7000 Series - configuração de SPAN [switches Cisco Nexus 7000 Series] - Cisco

Há vários tipos de sessões de SPAN. Alguns dos tipos mais comuns estão listados aqui:

- SPAN Local: Um tipo de sessão de SPAN em que os hosts origem e destino são locais para o switch. Em outras palavras, toda a configuração necessária para configurar a sessão de SPAN é aplicada a um único switch, o mesmo switch onde as portas do host origem e destino residem.
- SPAN Remoto (RSPAN): Um tipo de sessão de SPAN em que o host origem e destino não são locais para o switch. Em outras palavras, você configura as sessões de RSPAN de origem em um switch e o RSPAN de destino no switch de destino e estende a conectividade com o RSPAN VLAN.

Note: RSPAN não é suportado no Nexus

- SPAN remoto estendido (ERSPAN): O switch encapsula o quadro copiado com um cabeçalho de túnel GRE (Generic Routing Encapsulation) e roteia o pacote para o destino configurado. Você configura as sessões de origem e destino nos switches de encapsulamento e desencapsulamento (dois dispositivos diferentes). Isso nos dá a capacidade de tráfego SPAN em uma rede de camada 3.
- SPAN-para-CPU: um nome dado a um tipo especial de sessão de SPAN em que sua porta de destino é o supervisor ou a CPU. É uma forma de sessão de SPAN local e pode ser usada nos casos em que você não pode utilizar uma sessão de SPAN padrão. Algumas das razões comuns são: nenhuma porta de destino de SPAN disponível ou adequada, site não acessível ou site não gerenciado, nenhum dispositivo disponível que possa se conectar à porta de destino de SPAN e assim por diante. Para obter detalhes, consulte este link Procedimento de SPAN para CPU do Nexus 9000 Cloud Scale ASIC NX-OS - Cisco. É importante lembrar que a SPAN para CPU é limitada pela taxa de CoPP (Políticas de plano de controle), portanto sniffing uma ou mais interfaces de origem que excedem o vigilante podem resultar em quedas da sessão SPAN para CPU. Se isso acontecer, os dados não refletirão 100% do que está no fio, portanto, SPAN para CPU nem sempre é apropriado para cenários de solução de problemas com alta taxa de dados e/ou perda intermitente. Depois de configurar uma sessão de SPAN para CPU e ativá-la administrativamente, você precisa executar o Ethanalyzer para

ver o tráfego enviado à CPU para executar a análise de acordo.

Este é um exemplo de como você pode configurar uma sessão de SPAN local simples em um 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 exemplo mostra a configuração de uma sessão de SPAN para CPU que foi ativada e o uso do Ethanalyzer para capturar o tráfego:

```
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

O Dmirror é um tipo de sessão SPAN-TO-CPU para plataformas Nexus baseadas em Broadcom. O conceito é o mesmo de SPAN para CPU e sua taxa é limitada a 50 pps (pacotes por segundo). O recurso foi implementado para depurar o caminho de dados interno com a CLI bcm-shell. Devido às limitações associadas, não há CLI do NX-OS para permitir que os usuários configurem sessões de SPAN para o Sup porque ele pode afetar o tráfego de controle e consumir classes CoPP.

- ELAM

ELAM significa Embedded Logic Analyzer Module. Ele permite examinar o ASIC e determinar quais decisões de encaminhamento são tomadas para um pacote SINGLE. Com o ELAM, você pode identificar se o pacote alcança o mecanismo de encaminhamento e em que

portas/informações de VLAN. Você também pode verificar a estrutura do pacote L2 - L4 e se foram feitas alterações no pacote ou não.

Éimportante entender que o ELAM depende da arquitetura e que o procedimento para capturar um pacote varia de plataforma para plataforma com base na arquitetura interna. Você deve conhecer os mapeamentos ASIC do hardware para aplicar corretamente a ferramenta. Para o Nexus 7000, duas capturas são feitas para um único pacote, uma antes da decisão ser tomada **Data BUS (DBUS**) e outra depois da decisão ter sido tomada **Result BUS (RBUS)**. Ao visualizar as informações de DBUS, você pode ver o que/onde o pacote foi recebido, bem como as informações das camadas 2 a 4. Os resultados no RBUS podem mostrar para onde o pacote é encaminhado e se o quadro foi alterado. Você precisa configurar os disparadores para DBUS e RBUS, garantir que estejam prontos e tentar capturar o pacote em tempo real. Os procedimentos para várias placas de linha são os seguintes:

Para obter detalhes sobre vários procedimentos do ELAM, consulte os links nesta tabela:

VISÃO GERAL DO ELAM	Visão geral do ELAM - Cisco
Módulo Nexus 7K F1	Procedimento ELAM do módulo Nexus 7000 F1 - Cisco
Módulo Nexus 7K F2	Procedimento do ELAM do módulo Nexus 7000 F2 - Cisco
Módulo Nexus 7K F3	F3- Exemplo de ELAM
Módulo Nexus 7K M	Procedimento ELAM do módulo Nexus 7000 M-Series - Cisco
Módulo Nexus 7K M1/M2 e F2	ELAM Nexus 7K para M1/M2 e F2 e Ethanalyzer
Módulo Nexus 7K M3	Procedimento do módulo ELAM do Nexus 7000 M3 - Cisco

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

- Verifique o número do módulo com o comando **show module**.
- Anexe ao módulo com **attach module x**, onde x é o número do módulo.
- Verifique o mapeamento ASIC interno com o comando **show hardware internal dev-port-map** e verifique L2LKP e 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
+---------------------------------------------------------------------+
+---------------------------------------------------------------------+
```

- Primeiro, você captura o pacote em L2 e vê se a decisão de encaminhamento está correta. Para fazer isso, examine a coluna mapeamentos L2LKP e identifique o número da instância do ASIC que corresponde à porta.
- Em seguida, você executa o ELAM nessa instância com o comando **elam asic eureka instance** xonde x é o número da instância ASIC e configure nossos disparadores para DBUS e RBUS. Verifique o status dos disparadores com o comando **status** e confirme se os disparadores foram configurados.

```
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
```

- Ative os disparadores com o comando **start** e verifique o status dos disparadores com o comando **status** para confirmar se os disparadores estão 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
```

- Quando o status mostrar que os acionadores estão armados, eles estarão prontos para capturar. Nesse momento, você deve enviar o tráfego e verificar o status novamente para ver se os disparadores foram realmente acionados.

```
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
```

- Uma vez disparado, verifique o número de sequência do pacote para rbus e dbus para confirmar se ambos capturaram o mesmo pacote. Isso pode ser feito com o comando **show dbus | i seq. show rbus | i seq**. Se o número de sequência coincidir, você poderá ver o conteúdo do dbus e rbus. Caso contrário, execute novamente a captura até conseguir capturar o mesmo pacote.

  Note: Para maior precisão, sempre execute o ELAM várias vezes para confirmar problemas de encaminhamento.

- Você pode visualizar o conteúdo de rbus e dbus com os comandos **show dbus** e **show rbus**. O importante na captura é o número de sequência e o índice de origem/destino. O Dbus mostra o índice de origem que informa a porta na qual o pacote foi recebido. O Rbus mostra o índice de destino da porta para a qual o pacote é encaminhado. Além disso, você também pode examinar os endereços IP/MAC origem e destino, bem como informações de VLAN.
- Com o índice de origem e destino (também conhecido como índice LTL), você pode verificar a porta do painel frontal associada com o comando **show system internal pixm info ltl #**.

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

O procedimento é o mesmo para a plataforma Lamira, no entanto, existem algumas diferenças:

- Você executa o ELAM com a palavra-chave Lamira **elam asic lamira instance x**.
- Os comandos para acionar o ELAM são:

```
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
```

- Você verifica o status com o comando **status** e garante que eles estejam Armados antes de

enviar o tráfego e acionados depois de capturá-lo.
- Você pode interpretar as saídas de dbus e show bus de forma semelhante à mostrada para Eureka.

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

Novamente, o procedimento é semelhante, apenas os acionadores são diferentes. As poucas diferenças são as seguintes:

- Você executa o ELAM com a palavra-chave Clipper **elam asic clipper instance x** e especifica o modo da camada 2 ou da camada 3.

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

- Os comandos para acionar o ELAM são os seguintes:

```
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
```

- Você verifica o status com o comando **status** e garante que eles estejam Armados antes de enviar o tráfego e acionados depois de capturá-lo.
- Você pode interpretar as saídas de dbus e show bus de forma semelhante à mostrada para Eureka.

ELAM para Nexus 7000 - F3 (plataforma Flanker)

Novamente, o procedimento é semelhante, apenas os acionadores são diferentes. As poucas diferenças são as seguintes:

- Você executa o ELAM com a palavra-chave Flanker **elam asic flanker instance x** e especifica o modo de Camada 2 ou Camada 3.

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

- Os comandos para acionar o ELAM são os seguintes:

```
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
```

- Você verifica o status com o comando **status** e garante que eles estejam Armados antes de enviar tráfego e acionados depois de capturá-lo.
- Você pode interpretar as saídas de dbus e rbus de forma semelhante à mostrada para Eureka.

ELAM para Nexus 9000 (plataforma Tahoe)

No Nexus 9000, o procedimento é um pouco diferente do Nexus 7000. Para o Nexus 9000, consulte o link Nexus 9000 Cloud Scale ASIC (Tahoe) NX-OS ELAM - Cisco

- Primeiro, verifique o mapeamento de interface com o comando **show hardware internal tah interface #**. As informações mais importantes nesta saída são **ASIC #**, **Slice #** e **source ID (srcid) #**.
- Além disso, você também pode verificar essas informações com o comando **show system internal ethpm info interface # | i src**. O importante aqui, além do que foi listado anteriormente, são os valores dpid e dmod.
- Verifique o número do módulo com o comando **show module**.
- Anexe ao módulo com **attach module x**, onde x é o número do módulo.
- Execute o ELAM no módulo com o comando **module-1# debug platform internal tah elam asic #**
- Configure o disparador interno ou externo com base no tipo de tráfego que deseja capturar (L2, L3, tráfego encapsulado como GRE ou 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
```

- Depois que os acionadores estiverem definidos, inicie o ELAM com o comando **start**, envie tráfego e visualize a saída com o comando **report**. A saída do relatório mostra as interfaces de saída e de entrada juntamente com o ID da vlan e o endereço IP/MAC de origem e 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)

O procedimento para a plataforma NorthStar é o mesmo da plataforma Tahoe, a única diferença é que a palavra-chave **ns** é usada em vez de **tah** quando o modo ELAM é inserido:

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

- **Packet Tracer N9K**

A ferramenta Packet Tracer do Nexus 9000 pode ser usada para rastrear o caminho do pacote e, com seus contadores integrados para estatísticas de fluxo, torna-o uma ferramenta valiosa para cenários de perda de tráfego intermitente/completa. Seria muito útil quando os recursos do TCAM são limitados ou não estão disponíveis para executar outras ferramentas. Além disso, essa ferramenta não pode capturar o tráfego ARP e não exibe detalhes do conteúdo dos pacotes como o Wireshark.

Para configurar o packet tracer, use estes 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 obter detalhes, consulte o link Nexus 9000: Ferramenta Packet Tracer explicada - Cisco

- **Traceroute e Pings**

Esses comandos são os mais úteis que permitem identificar rapidamente problemas de conectividade.

O ping usa o Internet Control Message Protocol (ICMP) para enviar mensagens de eco ICMP ao destino específico e aguarda respostas de eco ICMP desse destino. Se o caminho entre o host funcionar bem sem problemas, você poderá ver as respostas voltarem e os pings tiverem êxito. Por padrão, o comando ping envia mensagens de eco ICMP 5x (tamanho igual em ambas as direções) e, se tudo funcionar bem, você poderá ver respostas de eco ICMP 5x. Às vezes, a solicitação de eco inicial falha quando os switches aprendem o endereço MAC durante a solicitação do Address Resolution Protocol (ARP). Se você executar o ping novamente logo após, não haverá perda inicial do ping. Além disso, você também pode definir o número de pings, o tamanho do pacote, a origem, a interface de origem e os intervalos de tempo limite com estas palavras-chave:

```
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
```

O traceroute é usado para identificar os vários saltos que um pacote leva antes de chegar ao seu destino. É uma ferramenta muito importante porque ajuda a identificar o limite de L3 onde a falha ocorre. Você também pode usar a porta, a interface de origem e a interface de origem com estas palavras-chave:

```
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 significa Access Control List. É uma ferramenta importante que permite filtrar o tráfego com base em um critério relevante definido. Quando a ACL estiver preenchida com entradas para critérios de correspondência, ela poderá ser aplicada para capturar tráfego de entrada ou de saída. Um aspecto importante da ACL é sua capacidade de fornecer contadores para estatísticas de fluxo. Os termos PACL/RACL/VACL referem-se a várias implementações dessas ACLs que permitem usar a ACL como uma poderosa ferramenta de solução de problemas, especialmente para perda intermitente de tráfego. Estes termos são descritos resumidamente aqui:

- PACL significa Port Access Control List: Quando você aplica uma lista de acesso a uma interface/porta de switch L2, essa lista de acesso é conhecida como PACL.
- RACL significa Router Access Control List: Quando você aplica uma lista de acesso a uma porta/interface roteada L3, essa lista de acesso é conhecida como RACL.
- VACL significa VLAN Access Control List: Você pode configurar VACLs para aplicar a todos os pacotes que são roteados para dentro ou para fora de uma VLAN ou que estão interligados dentro de uma VLAN. As VACLs são estritamente para filtros de pacotes de segurança e para redirecionar o tráfego para interfaces físicas específicas. As VACLs não são definidas pela direção (entrada ou saída).

Esta tabela fornece uma comparação entre as versões das ACLs.

TIPO DE ACL	PACL	RACL	VACL
FUNÇÃO	Filtrar o tráfego recebido em uma interface L2.	Filtrar o tráfego recebido em uma interface L3	Filtrar tráfego vLAN
APLICADO EM	- Interfaces/portas L2. - Interfaces de canal de porta L2. - Se aplicada em uma porta de tronco, a ACL filtra o tráfego em todas as VLANs permitidas nessa porta de tronco.	- Interfaces VLAN. - Interfaces físicas L3. - Subinterfaces L3. - Interfaces de canal de porta L3. - Interfaces de gerenciamento.	Uma vez ativada, a ACL é aplicada a todas as portas nessa VLAN (inclui portas tronco).
DIREÇÃO APLICADA	Somente entrada.	Entrada ou saída	-

Aqui está um exemplo de como você pode configurar uma lista de acesso. Para obter detalhes, consulte o link Guia de configuração de segurança do Cisco Nexus 9000 Series NX-OS, versão 9.3(x) - Configurando ACLs IP [Switches Cisco Nexus 9000 Series] - 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 é um tipo de armazenamento persistente disponível nas plataformas Nexus como flash compacto externo, um dispositivo USB ou um disco incorporado no supervisor. Se for removido do switch, o sistema notificará periodicamente o usuário de que o LogFlash está ausente. O Logflash é instalado no supervisor e armazena dados históricos, como logs de contabilidade, mensagens de syslog, depurações e saídas Embedded Event Manager (EEM). O EEM será discutido posteriormente neste artigo. Você pode verificar o conteúdo do LogFlash com 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
```

No caso de um usuário recarregar o dispositivo ou recarregá-lo repentinamente por conta própria devido a um evento, todas as informações de log serão perdidas. Nesses cenários, o LogFlash pode fornecer dados históricos que podem ser revisados para identificar uma causa provável do problema. Obviamente, é necessária uma diligência adicional para identificar a causa raiz que fornece dicas sobre o que procurar caso esse evento ocorra novamente.

Para obter informações sobre como instalar o logflash no dispositivo, consulte o link Nexus 7000 Logging Capabilities - Cisco.

- **OBFL**

OBFL significa OnBoard Failure Logging. É um tipo de armazenamento persistente disponível para switches Nexus Top of Rack e Modular. Assim como o LogFlash, as informações são retidas quando o dispositivo é recarregado. O OBFL armazena informações como falhas e dados ambientais. As informações variam para cada plataforma e módulo, no entanto, aqui está um exemplo de saída do módulo 1 da plataforma Nexus 93108 (ou seja, um chassi fixo com apenas um 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
```

Novamente, essas informações são úteis no caso de um dispositivo que é recarregado de propósito pelo usuário ou devido a um evento que disparou um recarregamento. Nesse caso, as informações de OBFL podem ajudar a identificar o que deu errado da perspectiva de uma placa de linha. O comando **show logging onboard** é um bom ponto de partida. Lembre-se de que você deve capturar o contexto do módulo para obter tudo o que precisa. Certifique-se de usar **show logging onboard module x** ou **attach mod x ; show logging onboard.**

- ## Históricos de eventos

Os históricos de eventos são uma das ferramentas eficientes que podem fornecer informações sobre vários eventos que ocorrem em um processo executado no Nexus. Em outras palavras, cada processo executado em uma plataforma Nexus tem históricos de eventos que são executados em segundo plano e armazenam informações sobre vários eventos desse processo (pense neles como depurações que são executadas constantemente). Esses históricos de eventos não são persistentes e todas as informações armazenadas são perdidas após o recarregamento do dispositivo. Eles são muito úteis quando você identifica um problema com um determinado processo e deseja solucionar esse processo. Por exemplo, se o protocolo de roteamento OSPF não funcionar corretamente, você poderá usar históricos de eventos associados ao OSPF para identificar onde o processo OSPF falha. Você pode encontrar históricos de eventos associados a quase todos os processos na plataforma Nexus, como CDP/STP, UDLD, LACP/OSPF, EIGRP/BGP e assim por diante.

Éassim que você normalmente verifica os históricos de eventos de um processo com exemplos de referência. Cada processo tem várias opções, então use **?** para verificar várias opções disponíveis em um processo.

```
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
```

- **Debugs**

As depurações são ferramentas eficientes no NX-OS que permitem executar eventos de solução de problemas em tempo real e registrá-los em um arquivo ou exibição na CLI. É altamente recomendável registrar as saídas de depuração em um arquivo, pois elas afetam o desempenho da CPU. Tenha cuidado antes de executar uma depuração diretamente no CLI.

As depurações geralmente são executadas apenas quando você identifica um problema como sendo um único processo e gostaria de verificar como esse processo se comporta em tempo real com o tráfego real na rede. Você precisa ativar um recurso de depuração com base nos privilégios de conta de usuário definidos.

Assim como os históricos de eventos, você pode executar depurações para cada processo em um dispositivo Nexus, como CDP/STP, UDLD, LACP/OSPF, EIGRP/BGP e assim por diante.

Éassim que você normalmente executa uma depuração para um processo. Cada processo tem várias opções, então use **?** para verificar várias opções disponíveis em um processo.

```
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
```

- OURO

GOLD significa Generic OnLine Diagnostics. Como o nome sugere, esses testes são geralmente usados como uma verificação de integridade do sistema e são usados para verificar ou verificar o hardware em questão. Há vários testes on-line que são realizados e baseados na plataforma em uso, alguns desses testes causam interrupções, enquanto outros não causam interrupções. Esses testes on-line podem ser categorizados da seguinte forma:

- **Diagnóstico de inicialização:** Esses testes são os que são executados quando o dispositivo é inicializado. Eles também verificam a conectividade entre o supervisor e os módulos, o que inclui a conectividade entre os dados e o plano de controle para todos os ASICs. Testes como ManagementPortLoopback e EOBCLoopback interrompem, enquanto testes para OBFL e USB não interrompem.
- **Diagnósticos de monitoramento de integridade ou de tempo de execução:** Esses testes fornecem informações sobre a integridade do dispositivo. Esses testes não causam interrupções e são executados em segundo plano para garantir a estabilidade do hardware. Você pode ativar/desativar esses testes conforme necessário ou para fins de solução de problemas.
- **Diagnóstico sob demanda:** Todos os testes mencionados podem ser executados novamente sob demanda para localizar um problema.

Você pode verificar os vários tipos de testes on-line disponíveis para seu switch com 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 exibir o que cada um dos 17 testes mencionados faz, você pode usar 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. É uma ferramenta poderosa que permite programar seu dispositivo para executar tarefas específicas no caso de um determinado evento acontecer. Ele monitora vários eventos no dispositivo e, em seguida, executa a ação necessária para solucionar o problema e possivelmente recuperar. O EEM consiste em três componentes principais, cada um dos quais é descrito resumidamente aqui:

- **Declaração do evento:** Esses são os eventos que você deseja monitorar e deseja que o Nexus execute uma determinada ação, como fazer uma solução alternativa ou simplesmente notificar um servidor SNMP ou exibir um log CLI, e assim por diante.
- **Instruções da ação:** Essas seriam as etapas que o EEM executaria quando um evento fosse disparado. Essas ações podem ser simplesmente desabilitar uma interface ou executar alguns comandos show e copiar saídas para um arquivo no servidor ftp, enviar um e-mail e assim por diante.
- **Políticas:** É basicamente um evento em combinação com uma ou mais instruções de ação que você pode configurar no supervisor via CLI ou um script bash. Você também pode chamar o EEM com um script python. Quando a política for definida no supervisor, ela será enviada para o módulo relevante.

Para obter detalhes sobre o EEM, consulte o link Guia de configuração de gerenciamento do sistema Cisco Nexus 9000 Series NX-OS, versão 9.2(x) - Configurando o gerenciador de eventos incorporado [switches Cisco Nexus 9000 Series] - Cisco.

Sobre esta tradução
A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês (link fornecido) seja sempre consultado.