Fiche de dépannage Nexus pour débutants

Contenu

Introduction

Ce document décrit les différents outils disponibles pour dépanner les produits Nexus que vous pouvez utiliser afin de diagnostiquer et de résoudre un problème.

Aperçu

Il est important de comprendre quels outils sont disponibles et dans quel scénario vous les utiliseriez pour un gain maximal. En fait, parfois, un certain outil n'est pas faisable simplement parce qu'il est conçu pour travailler sur quelque chose d'autre.

Ce tableau regroupe les différents outils de dépannage de la plate-forme Nexus et leurs fonctionnalités. Pour plus d'informations et des exemples CLI, reportez-vous à la section Outils Nexus.

OUTILS	FONCTION	EXEMPLES D'UTILISATION	POUR	CONTRE	PERSISTANCE	PLAN AFFECTÉ	COMMANDES UTILISÉES
Ethanalyseur	Capturer le trafic à destination ou en provenance du processeur	Problèmes de ralentissement du trafic, latence et congestion	Excellent pour les problèmes de lenteur, de congestion et de latence	Ne voit généralement que le trafic du plan de contrôle, à débit limité	S/O	Plan de contrôle. Peut être utilisé pour le plan de	#ethanalyzer interface locale intrabande #ethanalyzer interface locale interface] displa [WORD] exemple : #ethanalyzer interface locale

						données dans certains scénarios (SPAN vers CPU)	Ethernet 6/4 filtr d'affichage ICM
PORTÉE	Capturer et mettre en miroir un ensemble de paquets	Échec ping s, paquets dans le désordre, etc.	Excellent pour les pertes de trafic intermittentes	Nécessite un périphérique externe qui exécute un logiciel renifleur Nécessite des ressources TCAM	La session SPAN doit être configurée et activée/désactivée	Contrôle + Données	#monitor sessio #description [NO #source interfac de port] #destina interface [ID de #no shut
DMirror	Capturer le trafic destiné au processeur ou en provenance de celui-ci pour les périphériques Broadcom Nexus uniquement	Problèmes de ralentissement du trafic, latence et congestion	Excellent pour les problèmes de lenteur, de congestion et de latence	Uniquement pour les périphériques Broadcom Nexus. Débit limité (CloudScale Nexus 9k ne dispose pas de SPAN vers CPU)	S/O	Plan de contrôle. Peut être utilisé pour le plan de données dans certains scénarios	Varie selon la pl forme, voir Présentation d'E - Cisco
ELAM	Capture un seul paquet qui entre [ou sort, si Nexus 7K] dans le commutateur Nexus	Vérifier que le paquet atteint le Nexus, vérifier les décisions de transfert, vérifier les modifications apportées au paquet, vérifier l'interface/le VLAN du paquet, etc	Excellent pour les problèmes de flux et de transfert de paquets. Non intrusif	Nécessite une compréhension approfondie du matériel. Utilise des mécanismes de déclenchement uniques spécifiques à l'architecture. Utile uniquement si vous savez quel trafic vous souhaitez inspecter	S/O	Contrôle + Données	# module d'attachement [NUMÉRO DE MODULE] # deb platform interna
Packet	Détecter	Problèmes de	Fournit un	Impossible de	S/O	Donné	# test packet-tra

Tracer Nexus 9k	le chemin du paquet	connectivité et perte de paquets	compteur pour les statistiques de flux utiles pour les pertes intermittentes/complètes. Parfait pour les cartes de ligne sans découpage TCAM	capturer le trafic ARP. Fonctionne uniquement pour Nexus 9k		es + Contrôle	src_IP [SOURC dst_IP [DESTINATION test packet-trace start # test pack tracer stop # tes packet-tracer sh
Traceroute	Détecter le chemin du paquet par rapport aux sauts de couche 3	Échec des requêtes ping, impossible d'atteindre l'hôte/la destination/Internet, etc	Détecte les différents sauts du chemin pour isoler les pannes de couche 3.	Identifie uniquement l'endroit où la limite de couche 3 est rompue (n'identifie pas le problème lui-même)	S/O	Données + Contrôle	# traceroute [IP DESTINATION] Les arguments incluent : port, numéro de source, interface source-interface
Ping	Tester la connectivité entre deux points d'un réseau	Tester l'accessibilité entre les périphériques	Un outil simple et rapide pour tester la connectivité	Indique simplement si l'hôte est accessible ou non	S/O	Données + Contrôle	# ping [IP DE DESTINATION] Les arguments incluent : nombre, taille de paquet, interface source, intervall multidiffusion, bouclage, délai d'attente
PACL/RACL/VACL	Capturer le trafic entrant/sortant d'un port ou d'un VLAN donné	Perte de paquets intermittente entre les hôtes, vérifiez si les paquets arrivent/partent au niveau du Nexus, etc	Excellent pour les pertes de trafic intermittentes	Nécessite des ressources TCAM. Pour certains modules, une sculpture manuelle TCAM est requise	Persistant (appliqué à running-configuration)	Données + Contrôle	# ip access-list [NAME] # ip port access-group [A NAME] # ip acc group [ACL NAM Les arguments incluent : deny, fragments permit, remark, statistics, end, e pop, push, où
LogFlash	Stocke les données historiques du commutateur de manière globale,	Rechargement/arrêt soudain du périphérique, chaque fois qu'un périphérique est rechargé, les données de la mémoire flash du journal	Les informations sont conservées lors du rechargement du périphérique (stockage	Externe sur Nexus 7K = Doit être installé/intégré sur la plate-forme de supervision pour que ces journaux soient	Reload-Persistent	Données + Contrôle	# dir logflash:

	telles que les comptes de journaux, les fichiers de panne et les événements, quel que soit le rechargement du périphérique	fournissent des informations qui peuvent être utiles pour l'analyse	permanent)	collectés (ne s'applique pas à 3K/9K car logflash est une partition du périphérique de stockage interne)			
OBFL	Stocke les données historiques d'un module spécifique, telles que les informations sur les pannes et l'environnement	Rechargement/arrêt soudain du périphérique, chaque fois qu'un périphérique est rechargé, les données de la mémoire flash du journal fournissent des informations utiles	Les informations sont conservées lors du rechargement du périphérique (stockage permanent)	Prend en charge un nombre limité de lectures et d'écritures	Reload-Persistent	Données + Contrôle	# show logging onboard module Les arguments incluent : boot-uptime, ca boot-history, ca first-power-on, counter-stats, d version, endtime environment-his error-stats, exception-log, internal, interrup stats, obfl-histor stat-trace, startt status
Historiques des événements	Lorsque vous avez besoin d'informations sur un processus spécifique en cours d'exécution	Chaque processus dans nexus a ses propres historiques d'événements tels que CDP, STP, OSPF, EIGRP, BGP, vPC, LACP, etc	Dépanner un processus spécifique s'exécutant sur Nexus	Les informations sont perdues une fois le périphérique rechargé (non persistant)	Non-persistant	Données + Contrôle	# show [PROCE internal event-hi [ARGUMENT] Les arguments incluent : Contiguïté, cli, événement, inondation, ha, l ldp, lsa, msgs, objstore, redistribution, ri segrt, spf, spf-tr statistics, te
Déboguages	Lorsque vous avez besoin d'informa	Le débogage sur chaque processus dans nexus peut être fait comme CDP,	Dépanner un processus spécifique s'exécutant	Peut affecter les performances du réseau	Non-persistant	Données + Contrôle	# debug proces [PROCESS] exemple : # debug ip ospf

	tions en temps réel/en direct plus précises pour un processus spécifique	STP, OSPF, IGRP, BGP, vPC, LACP, et ainsi de suite	sur Nexus en temps réel pour plus de granularité				
OR	Fournit des diagnostics de démarrage, d'exécution et à la demande sur les composants matériels (tels que les E/S et les modules de supervision)	Testez du matériel tel que USB, Bootflash, OBFL, mémoire ASIC, PCIE, Port loopback, NVRAM, etc	Peut détecter les défaillances du matériel et prendre les mesures correctives nécessaires uniquement sur les versions 6(2)8 et ultérieures	Détecte uniquement les problèmes matériels	Non-persistant	S/O	# show diagnos content module show diagnostic description mod [#] tester tout
EEM	Surveiller les événements sur le périphérique et prendre les mesures nécessaires	Toute activité de périphérique nécessitant une action, une solution de contournement ou une notification, telle qu'un arrêt de l'interface, un dysfonctionnement du ventilateur, l'utilisation du processeur, etc	Prend en charge les scripts Python	Doit disposer de privilèges d'administrateur réseau pour configurer EEM	Le script et le déclencheur EEM résident dans la configuration	S/O	Varie, voir Configuration du gestionnaire d'événements in

Outils Nexus

Si vous avez besoin de plus de précisions sur diverses commandes et leur syntaxe ou options, reportez-vous à la section Commutateurs Cisco Nexus 9000 - Références des commandes - Cisco.

- **Ethanalyseur**

Ethanalyzer est un outil NX-OS conçu pour capturer le trafic CPU des paquets. Tout ce qui touche le processeur, que ce soit en entrée ou en sortie, peut être capturé avec cet outil. Il est basé sur l'analyseur de protocole réseau open source largement utilisé Wireshark. Pour plus de détails sur cet outil, veuillez vous reporter au Guide de dépannage d'Ethanalyzer sur Nexus 7000 - Cisco

Il est important de noter qu'en général, Ethanalyzer capture tout le trafic en provenance et à destination du superviseur, c'est-à-dire qu'il ne prend pas en charge les captures spécifiques à l'interface. Des améliorations d'interface spécifiques sont disponibles pour certaines plates-formes dans des points de code plus récents. En outre, Ethanalyzer capture uniquement le trafic commuté par le processeur et non par le matériel. Par exemple, vous pouvez capturer le trafic sur l'interface intrabande, l'interface de gestion ou un port du panneau avant (si pris en charge) :

```
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
```

Ce résultat montre quelques-uns des messages qui peuvent être capturés avec Ethanalyzer. Notez que par défaut, Ethanalyzer ne capture que 10 paquets au maximum. Cependant, vous pouvez utiliser cette commande pour demander à l'interface de ligne de commande de capturer des paquets indéfiniment. Utilisez CTRL+C pour quitter le mode de capture.

```
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
```

Vous pouvez également utiliser des filtres avec Ethanalyzer pour vous concentrer sur un trafic spécifique. Il existe deux types de filtres que vous pouvez utiliser avec ethanalzyer : les filtres Capture et les filtres Display. Un filtre de capture capture capture uniquement le trafic correspondant aux critères définis dans le filtre de capture. Un filtre d'affichage capture toujours

tout le trafic, mais seul le trafic qui correspond aux critères définis dans le filtre d'affichage est affiché.

```
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
```

Vous pouvez également capturer des paquets avec l'option detail et les afficher dans votre terminal, comme vous le feriez avec Wireshark. Cela vous permet d'afficher les informations d'en-tête complètes en fonction du résultat du dissecteur de paquets. Par exemple, si une trame est chiffrée, vous ne pouvez pas voir la charge utile chiffrée. Reportez-vous à l'exemple suivant :

```
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
```

Avec Ethanalyzer, vous pouvez :

- Écrivez le résultat (un fichier PCAP) dans le nom de fichier spécifié sur divers systèmes de fichiers cibles : bootflash, logflash, USB, etc... Vous pouvez ensuite transférer le fichier enregistré à l'extérieur du périphérique et l'afficher dans Wireshark, si nécessaire.
- Lisez un fichier du bootflash et affichez-le sur votre terminal. Tout comme lorsque vous lisez directement à partir de l'interface du processeur, vous pouvez également afficher l'intégralité des informations du paquet si vous utilisez le mot clé detail.

Reportez-vous à des exemples pour diverses sources d'interface et options de sortie :

```
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]
```

- **PORTÉE**

La fonctionnalité SPAN (SwitchPort Analyzer) est utilisée pour capturer tout le trafic d'une interface et le mettre en miroir sur un port de destination. Le port de destination se connecte généralement à un outil d'analyse de réseau (tel qu'un PC exécutant Wireshark) qui vous permet d'analyser le trafic qui traverse ces ports. Vous pouvez effectuer une analyse SPAN pour le trafic

provenant d'un seul port ou de plusieurs ports et VLAN.

Les sessions SPAN incluent un port source et un port de destination. Un port source peut être un port Ethernet (sans sous-interfaces), des canaux de port, des interfaces intrabandes de superviseur et ne peut pas être un port de destination simultanément. En outre, pour certains périphériques tels que la plate-forme 9300 et 9500, les ports FEX (Fabric Extender) sont également pris en charge. Un port de destination peut être un port Ethernet (accès ou agrégation), un canal de port (accès ou agrégation) et, pour certains périphériques tels que les ports de liaison ascendante 9300, les ports FEX ne sont pas pris en charge.

Vous pouvez configurer plusieurs sessions SPAN en tant qu'entrées/sorties/les deux. Le nombre total de sessions SPAN qu'un périphérique individuel peut prendre en charge est limité. Par exemple, un Nexus 9000 peut prendre en charge jusqu'à 32 sessions, tandis qu'un Nexus 7000 ne peut en prendre en charge que 16. Vous pouvez le vérifier dans l'interface de ligne de commande ou consulter les guides de configuration SPAN du produit que vous utilisez.

Notez que pour chaque version de NX-OS et pour chaque type de produit, les types d'interfaces et les fonctionnalités pris en charge diffèrent. Reportez-vous aux dernières consignes et restrictions de configuration pour le produit et la version que vous utilisez. Voici les liens pour Nexus 9000 et Nexus 7000 respectivement :

Guide de configuration de la gestion du système NX-OS de la gamme Cisco Nexus 9000, version 9.3(x) - Configuration de la fonctionnalité SPAN [Commutateurs de la gamme Cisco Nexus 9000] - Cisco

Guide de configuration de la gestion du système NX-OS de la gamme Cisco Nexus 7000 - Configuration de la fonctionnalité SPAN [Commutateurs Cisco Nexus 7000] - Cisco

Il existe différents types de sessions SPAN. Certains des types les plus courants sont répertoriés ici :

- Local SPAN : type de session SPAN dans laquelle l'hôte source et l'hôte de destination sont locaux au commutateur. En d'autres termes, toute la configuration requise pour configurer la session SPAN est appliquée à un seul commutateur, le même commutateur où résident les ports hôtes source et de destination.
- Remote SPAN (RSPAN) : type de session SPAN dans lequel les hôtes source et de destination ne sont pas locaux au commutateur. En d'autres termes, vous configurez des sessions RSPAN source sur un commutateur et RSPAN de destination sur le commutateur de destination et étendez la connectivité avec le VLAN RSPAN.

Note: RSPAN n'est pas pris en charge sur Nexus

- SPAN distant étendu (ERSPAN) : Le commutateur encapsule la trame copiée avec un en-tête de tunnel GRE (Generic Routing Encapsulation) et achemine le paquet vers la destination configurée. Vous configurez les sessions source et de destination sur les commutateurs d'encapsulation et de décapsulation (deux périphériques différents). Cela nous donne la possibilité d'effectuer une analyse SPAN du trafic sur un réseau de couche 3.
- SPAN-to-CPU : nom donné à un type spécial de session SPAN où votre port de destination est le superviseur ou le CPU. Il s'agit d'une forme de session SPAN locale qui peut être utilisée lorsque vous ne pouvez pas utiliser une session SPAN standard. Voici quelques-unes des raisons courantes : aucun port de destination SPAN disponible ou approprié, site non accessible ou site non géré, aucun périphérique disponible pouvant se connecter au port de

destination SPAN, etc. Pour plus de détails, consultez ce lien Procédure ASIC NX-OS SPAN-to-CPU pour l'évolutivité du cloud Nexus 9000 - Cisco. Il est important de se rappeler que le débit SPAN vers CPU est limité par CoPP (Control Plane Policing), donc sniffing une ou plusieurs interfaces source qui dépassent le régulateur peuvent entraîner des pertes de session SPAN vers CPU. Si cela se produit, les données ne reflètent pas à 100 % ce qui se trouve sur le câble, de sorte que la fonctionnalité SPAN vers CPU n'est pas toujours appropriée pour les scénarios de dépannage avec un débit de données élevé et/ou une perte intermittente. Une fois que vous avez configuré une session SPAN vers CPU et que vous l'avez activée administrativement, vous devez exécuter Ethanalyzer pour voir le trafic qui est envoyé au CPU pour effectuer l'analyse en conséquence.

Voici un exemple de configuration d'une session SPAN locale simple sur un commutateur 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
```

Cet exemple montre la configuration d'une session SPAN vers CPU qui a été activée, puis l'utilisation d'Ethanalyzer pour capturer le trafic :

```
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 est un type de session SPAN-TO-CPU pour plates-formes Nexus Broadcom. Le concept

est le même que pour SPAN-to-CPU et son débit est limité à 50 pps (paquets par seconde). La fonctionnalité a été implémentée pour déboguer le chemin de données interne avec l'interface de ligne de commande bcm-shell. En raison des limitations associées, il n'existe pas d'interface de ligne de commande NX-OS permettant aux utilisateurs de configurer des sessions SPAN sur le superviseur, car cela peut affecter le trafic de contrôle et consommer des classes CoPP.

- **ELAM**

ELAM est l'acronyme de Embedded Logic Analyzer Module. Il permet d'examiner l'ASIC et de déterminer les décisions de transmission prises pour un paquet **UNIQUE**. Ainsi, avec ELAM, vous pouvez identifier si le paquet atteint le moteur de transfert et sur quels ports/informations VLAN. Vous pouvez également vérifier la structure des paquets L2 - L4 et si des modifications ont été apportées au paquet ou non.

Il est important de comprendre qu'ELAM dépend de l'architecture et que la procédure de capture d'un paquet varie d'une plate-forme à l'autre en fonction de l'architecture interne. Vous devez connaître les mappages ASIC du matériel pour appliquer correctement l'outil. Pour Nexus 7000, deux captures sont effectuées pour un seul paquet, l'une avant la prise de décision **Data BUS (DBUS**) et l'autre après la prise de décision **Result BUS (RBUS)**. Lorsque vous affichez les informations DBUS, vous pouvez voir ce que/où le paquet a été reçu, ainsi que les informations de couche 2 à 4. Les résultats dans le RBUS peuvent vous montrer où le paquet est transféré et si la trame a été modifiée. Vous devez configurer des déclencheurs pour DBUS et RBUS, vous assurer qu'ils sont prêts, puis essayer de capturer le paquet en temps réel. Les procédures pour les différentes cartes de ligne sont les suivantes :

Pour plus d'informations sur les différentes procédures ELAM, reportez-vous aux liens de ce tableau :

PRÉSENTATION D'ELAM	Présentation d'ELAM - Cisco
Module F1 Nexus 7K	Procédure ELAM du module F1 Nexus 7000 - Cisco
Module F2 Nexus 7K	Procédure ELAM du module F2 Nexus 7000 - Cisco
Module F3 Nexus 7K	Exemple F3- ELAM
Module Nexus 7K M	Procédure ELAM du module Nexus 7000 série M - Cisco
Module Nexus 7K M1/M2 et F2	Nexus 7K ELAM pour M1/M2 et F2 et Ethanalyzer
Module Nexus 7K M3	Procédure ELAM du module Nexus 7000 M3 - Cisco

ELAM pour Nexus 7000 - M1/M2 (plateforme Eureka)

- Vérifiez le numéro de module avec la commande **show module**.
- Fixez au module avec **attach module x**, où x est le numéro de module.
- Vérifiez le mappage ASIC interne avec la commande **show hardware internal dev-port-map** et vérifiez L2LKP et 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
+---------------------------------------------------------------------------+
+---------------------------------------------------------------------------+
```

- Tout d'abord, vous capturez le paquet dans L2 et vous vérifiez si la décision de transfert est correcte. Pour ce faire, consultez la colonne des mappages L2LKP et identifiez le numéro d'instance ASIC correspondant au port.
- Exécutez ensuite ELAM sur cette instance avec la commande **elam asic eureka instance x**où x est le numéro d'instance ASIC et configurez nos déclencheurs pour DBUS et RBUS. Vérifiez l'état des déclencheurs avec la commande **status** et confirmez que les déclencheurs ont été

configurés.

```
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
```

- Activez les déclencheurs avec la commande **start** et vérifiez que l'état des déclencheurs avec l'**état de** la commande pour confirmer que les déclencheurs sont armés.

```
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
```

- Une fois que l'état indique que les déclencheurs sont armés, ils sont prêts à être capturés. À ce stade, vous devez envoyer le trafic et vérifier à nouveau l'état pour voir si vos déclencheurs ont été réellement déclenchés.

```
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
```

- Une fois déclenchés, vérifiez le numéro de séquence des paquets pour rbus et dbus afin de confirmer qu'ils ont capturé le même paquet. Pour ce faire, utilisez la commande **show dbus | i seq ; show rbus | i seq**. Si le numéro d'ordre correspond, vous pouvez afficher le contenu de dbus et de rbus. Si ce n'est pas le cas, réexécutez la capture jusqu'à ce que vous puissiez capturer le même paquet.

  Note: Pour plus de précision, exécutez toujours ELAM plusieurs fois pour confirmer les problèmes de transfert.

- Vous pouvez afficher le contenu de rbus et dbus avec les commandes **show dbus** et **show rbus**. L'élément important dans la capture est la séquence # et l'index source/destination. Dbus affiche l'index source qui vous indique le port sur lequel il a reçu le paquet. Rbus affiche l'index de destination du port vers lequel le paquet est transféré. En outre, vous pouvez également consulter les adresses IP/MAC source et de destination ainsi que les informations VLAN.
- Avec l'index source et de destination (également connu sous le nom d'index LTL), vous pouvez vérifier le port du panneau avant associé avec la commande **show system internal pixm info ltl #**.

ELAM pour Nexus 7000 - M1/M2 (plate-forme Lamira)

La procédure est la même pour la plateforme Lamira, mais il y a quelques différences :

- Vous exécutez ELAM avec le mot clé Lamira **elam asic lamira instance x**.
- Les commandes permettant de déclencher l'ELAM sont les suivantes :

```
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
```

- Vous vérifiez l'état à l'aide de la commande **status** et assurez-vous qu'ils sont Armés avant d'envoyer du trafic et déclenchés après l'avoir capturé.
- Vous pouvez alors interpréter les sorties de dbus et show bus de la même manière que pour Eureka.

ELAM pour Nexus 7000 - F2/F2E (Clipper Platform)

Là encore, la procédure est similaire, seuls les déclencheurs sont différents. Les quelques différences sont les suivantes :

- Vous exécutez ELAM avec le mot clé Clipper **elam asic clipper instance x** et spécifiez le mode Couche 2 ou Couche 3.

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

- Les commandes permettant de déclencher l'ELAM sont les suivantes :

```
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
```

- Vous vérifiez l'état à l'aide de la commande **status** et assurez-vous qu'ils sont Armés avant d'envoyer du trafic et déclenchés après l'avoir capturé.
- Vous pouvez alors interpréter les sorties de dbus et show bus de la même manière que pour Eureka.

ELAM pour Nexus 7000 - F3 (Flanker Platform)

Là encore, la procédure est similaire, seuls les déclencheurs sont différents. Les quelques différences sont les suivantes :

- Vous exécutez ELAM avec le mot clé Flanker **elam asic flanker instance x** et spécifiez le mode de couche 2 ou de couche 3.

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

```
module-4(flanker-elam)#
```

- Les commandes permettant de déclencher l'ELAM sont les suivantes :

```
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
```

- Vous vérifiez l'état à l'aide de la commande **status** et vous vous assurez qu'ils sont Armés avant d'envoyer du trafic et déclenchés après l'avoir capturé.
- Vous pouvez alors interpréter les sorties de dbus et rbus de la même manière que pour Eureka.

ELAM pour Nexus 9000 (plate-forme Tahoe)

Dans Nexus 9000, la procédure est légèrement différente de celle du Nexus 7000. Pour Nexus 9000, veuillez vous reporter au lien Nexus 9000 Cloud Scale ASIC (Tahoe) NX-OS ELAM - Cisco

- Tout d'abord, vérifiez le mappage d'interface avec la commande **show hardware internal tah interface #**. Les informations les plus importantes dans ce résultat sont le **numéro ASIC**, le **numéro de tranche** et l'**ID source (srcid)** .
- Vous pouvez également vérifier ces informations à l'aide de la commande **show system internal ethpm info interface # | i i src**. En plus de ce qui a été indiqué précédemment, les valeurs dpid et dmod sont importantes.
- Vérifiez le numéro de module avec la commande **show module**.
- Fixez au module avec **attach module x**, où x est le numéro de module.
- Exécutez ELAM sur le module avec la commande **module-1# debug platform internal tah elam asic #**
- Configurez votre déclencheur interne ou externe en fonction du type de trafic que vous souhaitez capturer (C2, C3, trafic encapsulé tel que 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
```

- Une fois les déclencheurs définis, démarrez ELAM avec la commande **start**, envoyez le trafic et affichez le résultat avec la commande **report**. Le résultat du rapport indique les interfaces sortantes et entrantes, ainsi que l'ID de VLAN, l'adresse IP/MAC source et de destination.

```
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 pour Nexus 9000 (plate-forme NorthStar)

La procédure pour la plateforme NorthStar est la même que pour la plateforme Tahoe, la seule différence est que le mot clé **ns** est utilisé à la place de **tah** quand le mode ELAM est entré :

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

- ### N9K Packet Tracer

L'outil Packet Tracer du Nexus 9000 peut être utilisé pour suivre le chemin du paquet et avec ses compteurs intégrés pour les statistiques de flux, il constitue un outil précieux pour les scénarios de perte de trafic intermittente/complète. Il serait très utile lorsque les ressources TCAM sont limitées ou ne sont pas disponibles pour exécuter d'autres outils. En outre, cet outil ne peut pas capturer le trafic ARP et n'affiche pas les détails du contenu des paquets comme Wireshark.

Pour configurer Packet Tracer, utilisez ces commandes :

```
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
```

Pour plus d'informations, consultez le lien Nexus 9000 : Présentation de l'outil Packet Tracer - Cisco

- ### Traceroute et Pings

Ces commandes sont les deux plus utiles pour identifier rapidement les problèmes de connectivité.

La commande ping utilise Internet Control Message Protocol (ICMP) pour envoyer des messages d'écho ICMP à la destination spécifique et attend les réponses d'écho ICMP de cette destination. Si le chemin entre l'hôte fonctionne correctement sans problème, vous pouvez voir les réponses revenir et les requêtes ping aboutir. La commande ping envoie par défaut 5 messages d'écho ICMP (de taille égale dans les deux directions) et si tout fonctionne correctement, vous pouvez voir 5 réponses d'écho ICMP. Parfois, la requête d'écho initiale échoue lorsque les commutateurs apprennent l'adresse MAC pendant la requête ARP (Address Resolution Protocol). Si vous exécutez à nouveau la requête ping immédiatement après, il n'y a aucune perte de la requête ping initiale. En outre, vous pouvez également définir le nombre de requêtes ping, la taille de paquet, la source, l'interface source et les intervalles de temporisation avec ces mots clés :

```
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
```

La commande traceroute permet d'identifier les différents sauts effectués par un paquet avant qu'il n'atteigne sa destination. Il s'agit d'un outil très important, car il permet d'identifier la limite de couche 3 où se produit la panne. Vous pouvez également utiliser le port, la source et l'interface source avec les mots clés suivants :

```
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 signifie Access Control List. C'est un outil important qui vous permet de filtrer le trafic en fonction d'un critère défini pertinent. Une fois que la liste de contrôle d’accès est remplie d’entrées pour les critères de correspondance, elle peut être appliquée pour capturer le trafic entrant ou sortant. Un aspect important de la liste de contrôle d’accès est sa capacité à fournir des compteurs pour les statistiques de flux. Les termes PACL/RACL/VACL font référence à diverses implémentations de ces listes de contrôle d’accès qui vous permettent d’utiliser les listes de contrôle d’accès comme un outil de dépannage puissant, en particulier pour les pertes de trafic intermittentes. Ces termes sont décrits brièvement ci-dessous :

- PACL signifie Port Access Control List : Lorsque vous appliquez une liste d’accès à un port/interface de commutateur de couche 2, cette liste d’accès est appelée PACL.
- RACL signifie Router Access Control List : Lorsque vous appliquez une liste d’accès à un port/une interface routé(e) de couche 3, cette liste d’accès est appelée RACL.
- VACL signifie VLAN Access Control List : Vous pouvez configurer des VACL pour qu’elles s’appliquent à tous les paquets qui sont routés vers ou depuis un VLAN ou qui sont pontés dans un VLAN. Les listes de contrôle d’accès virtuelles servent uniquement à filtrer les paquets de sécurité et à rediriger le trafic vers des interfaces physiques spécifiques. Les VACL ne sont pas définies par direction (entrée ou sortie).

Ce tableau compare les versions des listes de contrôle d'accès.

TYPE ACL	PACL	RACL	VACL
FONCTION	Filtrer le trafic reçu sur une interface L2.	Filtrer le trafic reçu sur une interface L3	Filtrer le trafic vLAN
APPLIQUÉ LE	- Interfaces/ports L2. - Interfaces port-channel L2. - Si elle est appliquée à un port agrégé, la liste de contrôle d'accès filtre le trafic sur tous les VLAN autorisés sur ce port agrégé.	- Interfaces VLAN. - Interfaces L3 physiques. - Sous-interfaces de couche 3. - Interfaces port-channel de couche 3. - Interfaces de gestion.	Une fois activée, la liste d contrôle d'accès est appli à tous les ports de ce VLA compris les ports agrégés
DIRECTION APPLIQUÉE	Entrant uniquement.	Entrant ou sortant	-

Voici un exemple de configuration d'une liste d'accès. Pour plus d'informations, reportez-vous au lien [Cisco Nexus 9000 Series NX-OS Security Configuration Guide, Release 9.3(x) - Configuring IP ACLs [Cisco Nexus 9000 Series Switches] - 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 est un type de stockage persistant disponible sur les plates-formes Nexus sous la forme

d'un compact flash externe, d'un périphérique USB ou d'un disque intégré au superviseur. S'il est supprimé du commutateur, le système avertit régulièrement l'utilisateur que LogFlash est manquant. Logflash est installé sur le superviseur et contient des données historiques telles que les journaux de comptabilité, les messages syslog, les débogages et les sorties Embedded Event Manager (EEM). L'ESEE est traitée plus loin dans cet article. Vous pouvez vérifier le contenu de LogFlash avec cette commande :

```
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
```

Si un utilisateur rechargeait le périphérique ou s'il rechargeait soudainement son propre périphérique en raison d'un événement, toutes les informations du journal seraient perdues. Dans de tels scénarios, LogFlash peut fournir des données historiques qui peuvent être examinées pour identifier une cause probable du problème. Bien sûr, une diligence raisonnable supplémentaire est nécessaire pour identifier la cause première qui vous fournit des conseils sur ce qu'il faut rechercher dans le cas où cet événement se produit à nouveau.

Pour plus d'informations sur la façon d'installer logflash sur le périphérique, veuillez consulter le lien Fonctionnalités de journalisation Nexus 7000 - Cisco.

- OBFL

OBFL signifie OnBoard Failure Logging. Il s'agit d'un type de stockage permanent disponible pour les commutateurs Nexus Top of Rack et Modular. Tout comme le LogFlash, les informations sont conservées une fois le périphérique rechargé. OBFL stocke des informations telles que les pannes et les données environnementales. Les informations varient pour chaque plate-forme et chaque module. Cependant, voici un exemple de sortie du module 1 de la plate-forme Nexus 93108 (qui est un châssis fixe avec un seul module) :

```
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
```

Encore une fois, ces informations sont utiles dans le cas d'un périphérique qui est rechargé soit intentionnellement par l'utilisateur, soit en raison d'un événement qui a déclenché un rechargement. Dans ce cas, les informations OBFL peuvent aider à identifier les erreurs du point de vue d'une carte de ligne. La commande **show logging onboard** est un bon point de départ. N'oubliez pas que vous devez capturer le contexte du module pour obtenir tout ce dont vous avez besoin. Assurez-vous d'utiliser **show logging onboard module x** ou **attach mod x ; show logging onboard.**

- **Historiques des événements**

Les historiques d'événements sont l'un des puissants outils qui peuvent vous fournir des informations sur les divers événements qui se produisent pour un processus qui s'exécute sur Nexus. En d'autres termes, chaque processus qui s'exécute sur une plate-forme Nexus a des historiques d'événements qui s'exécutent en arrière-plan et stockent des informations sur divers événements de ce processus (pensez à eux comme des débogages qui s'exécutent constamment). Ces historiques d'événements ne sont pas persistants et toutes les informations stockées sont perdues lors du rechargement du périphérique. Ils sont très utiles lorsque vous avez identifié un problème avec un certain processus et que vous souhaitez le résoudre. Par exemple, si votre protocole de routage OSPF ne fonctionne pas correctement, vous pouvez utiliser des historiques d'événements associés à OSPF pour identifier l'emplacement où le processus OSPF échoue. Vous pouvez trouver des historiques d'événements associés à presque tous les processus sur la plate-forme Nexus tels que CDP/STP, UDLD, LACP/OSPF, EIGRP/BGP, etc.

C'est ainsi que vous vérifierez généralement les historiques d'événements d'un processus avec des exemples de référence. Chaque processus a plusieurs options à utiliser **?** pour vérifier les différentes options disponibles dans un processus.

```
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
```

- ## Déboguages

Les débogages sont des outils puissants de NX-OS qui vous permettent d'exécuter des événements de dépannage en temps réel et de les consigner dans un fichier ou de les afficher dans l'interface de ligne de commande. Il est fortement recommandé de consigner les sorties de débogage dans un fichier car elles ont un impact sur les performances du processeur. Soyez prudent avant d'exécuter un débogage directement sur l'interface de ligne de commande.

Les débogages sont généralement exécutés uniquement lorsque vous avez identifié un problème comme étant un processus unique et que vous souhaitez vérifier comment ce processus se comporte en temps réel avec le trafic réel sur le réseau. Vous devez activer une fonction de débogage en fonction des privilèges de compte d'utilisateur définis.

Tout comme les historiques d'événements, vous pouvez exécuter des débogages pour chaque processus sur un périphérique Nexus tel que CDP/STP, UDLD, LACP/OSPF, EIGRP/BGP, etc.

C'est ainsi que vous exécutez généralement un débogage pour un processus. Chaque processus a plusieurs options à utiliser **?** pour vérifier les différentes options disponibles dans un processus.

```
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
```

- OR

GOLD est l'acronyme de Generic OnLine Diagnostics. Comme son nom l'indique, ces tests sont généralement utilisés pour vérifier l'état du système et pour vérifier le matériel en question. Plusieurs tests en ligne sont effectués et basés sur la plate-forme utilisée. Certains de ces tests sont perturbateurs alors que d'autres ne le sont pas. Ces tests en ligne peuvent être classés comme suit :

- **Diagnostics de démarrage :** Ces tests sont ceux qui sont exécutés lors du démarrage du périphérique. Ils vérifient également la connectivité entre le superviseur et les modules, ce qui inclut la connectivité entre les données et le plan de contrôle pour tous les ASIC. Les tests tels que ManagementPortLoopback et EOBCLoopback sont perturbateurs tandis que les tests pour OBFL et USB ne le sont pas.
- **Diagnostics d'exécution ou de contrôle d'intégrité :** Ces tests fournissent des informations sur l'intégrité du périphérique. Ces tests ne provoquent pas d'interruption et sont exécutés en arrière-plan pour garantir la stabilité du matériel. Vous pouvez activer/désactiver ces tests si

nécessaire ou à des fins de dépannage.

- **Diagnostics à la demande :** Tous les tests mentionnés peuvent être réexécutés à la demande afin de localiser un problème.

Vous pouvez vérifier les différents types de tests en ligne disponibles pour votre commutateur avec cette commande :

```
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-
```

Pour afficher les résultats de chacun des 17 tests mentionnés, vous pouvez utiliser cette commande :

```
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 signifie Embedded Event Manager. Il s'agit d'un outil puissant qui vous permet de programmer votre périphérique pour effectuer des tâches spécifiques en cas d'événement particulier. Il surveille divers événements sur le périphérique, puis prend les mesures nécessaires pour résoudre le problème et éventuellement récupérer. L'ESEE se compose de trois composants

principaux, chacun étant brièvement décrit ici :

- **Instruction d'événement :** Il s'agit des événements que vous souhaitez surveiller et que Nexus effectue une action spécifique, par exemple effectuer une solution de contournement ou simplement notifier un serveur SNMP ou afficher un journal CLI, etc.
- **Instructions d'action :** Il s'agit des étapes qu'EEM doit suivre une fois qu'un événement est déclenché. Ces actions peuvent être simplement pour désactiver une interface ou exécuter quelques commandes show et copier des sorties dans un fichier sur le serveur FTP, envoyer un e-mail, etc.
- **Politiques :** Il s'agit essentiellement d'un événement combiné à une ou plusieurs instructions d'action que vous pouvez configurer sur le superviseur via l'interface de ligne de commande ou un script bash. Vous pouvez également appeler EEM avec un script python. Une fois la stratégie définie sur le superviseur, elle la transmet au module concerné.

Pour plus d'informations sur EEM, consultez le lien Guide de configuration de la gestion du système NX-OS de la gamme Cisco Nexus 9000, version 9.2(x) - Configuration du gestionnaire d'événements intégré [Commutateurs de la gamme Cisco Nexus 9000] - Cisco.

À propos de cette traduction
Cisco a traduit ce document en traduction automatisée vérifiée par une personne dans le cadre d'un service mondial permettant à nos utilisateurs d'obtenir le contenu d'assistance dans leur propre langue.

Il convient cependant de noter que même la meilleure traduction automatisée ne sera pas aussi précise que celle fournie par un traducteur professionnel.