

Vérifiez le chemin lent de transfert des paquets NCS6K utilisant le test de ping

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Informations générales](#)

[Vérifiez](#)

[Cisco relatif prennent en charge des discussions de la Communauté](#)

Introduction

Le test de paquet de ping est test utilisé généralement pour dépanner des problèmes de connectivité. Ce document montrera une approche systématique pour l'usage du test de ping pour vérifier le transfert des paquets lent du système 6000 (NCS6K) de convergence de réseau.

Conditions préalables

Conditions requises

Les lecteurs de ce document devraient avoir connaissance des sujets suivants :

- Routage IP de base.
- Système d'exploitation XR.

[Composants utilisés](#)

Ce document est créé pour la plate-forme NCS6K.

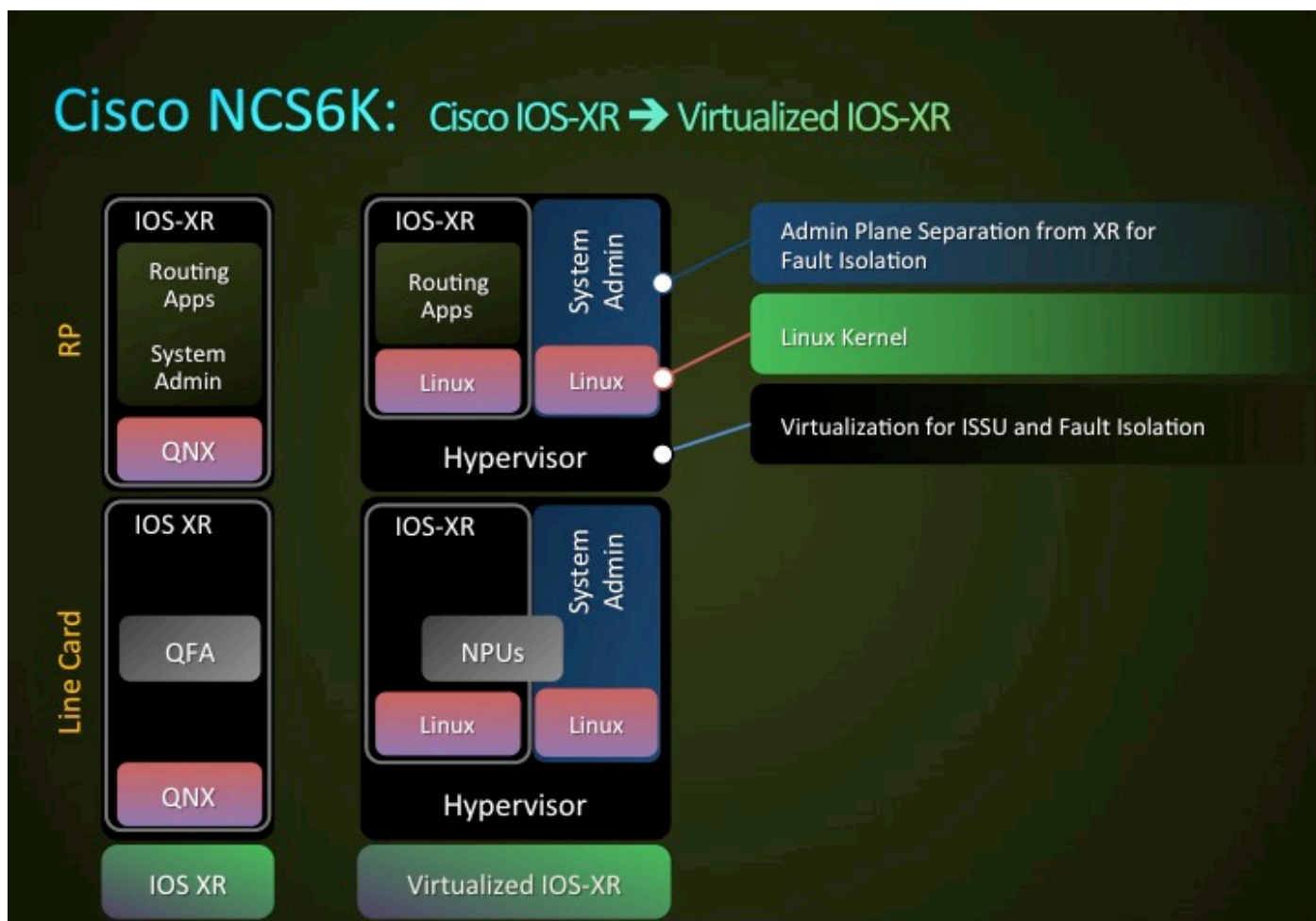
Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

[Informations générales](#)

Il y a une différence principale entre NCS6K et plate-forme traditionnelle IOS-XR : NCS6K utilise la technologie de virtualisation pour accumuler le système. Chaque noeud, conduisant le processeur (RP) ou le linecard (LC), peut exécuter plusieurs ordinateurs de Virtual (VM) comme la VM d'admin de système, l'IOS-XR VM1, l'IOS-XR VM2 etc., qui a combiné pour créer ensemble a entièrement - noeud fonctionnel XR. La figure suivante affiche à un exemple où le RP et le LC

exécutent une VM IOS-XR :

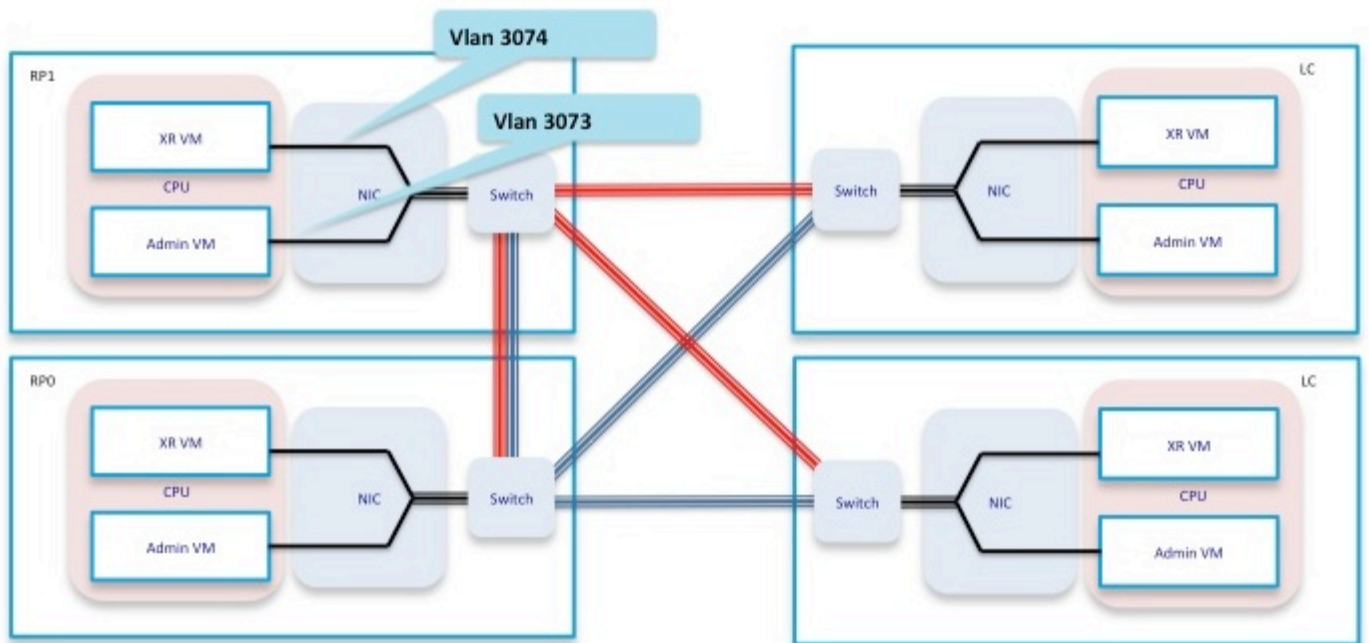
Figure 1



Il y a un réseau Ethernet de contrôle pour connecter la RPS et le LCS. Le trafic d'avion de contrôle entre la RPS et le LCS traversera ce réseau Ethernet de contrôle. Puisque c'est un environnement de virtualization, est-ce que questions comme comment ces le paquet sont livrés à la VM de particularité et comment le Nicantic (NIC) dans le RP ou le LC sait un paquet est destinées à eux ?

En un mot, des VLAN sont utilisés pour différencier le trafic de différentes VMs et ce processus est fait par le NIC. La figure 2 affiche comment le NIC fournira le trafic VLAN 3074 à la VM IOS-XR, et la trafic VLAN 3073 à la VM d'admin.

Figure 2

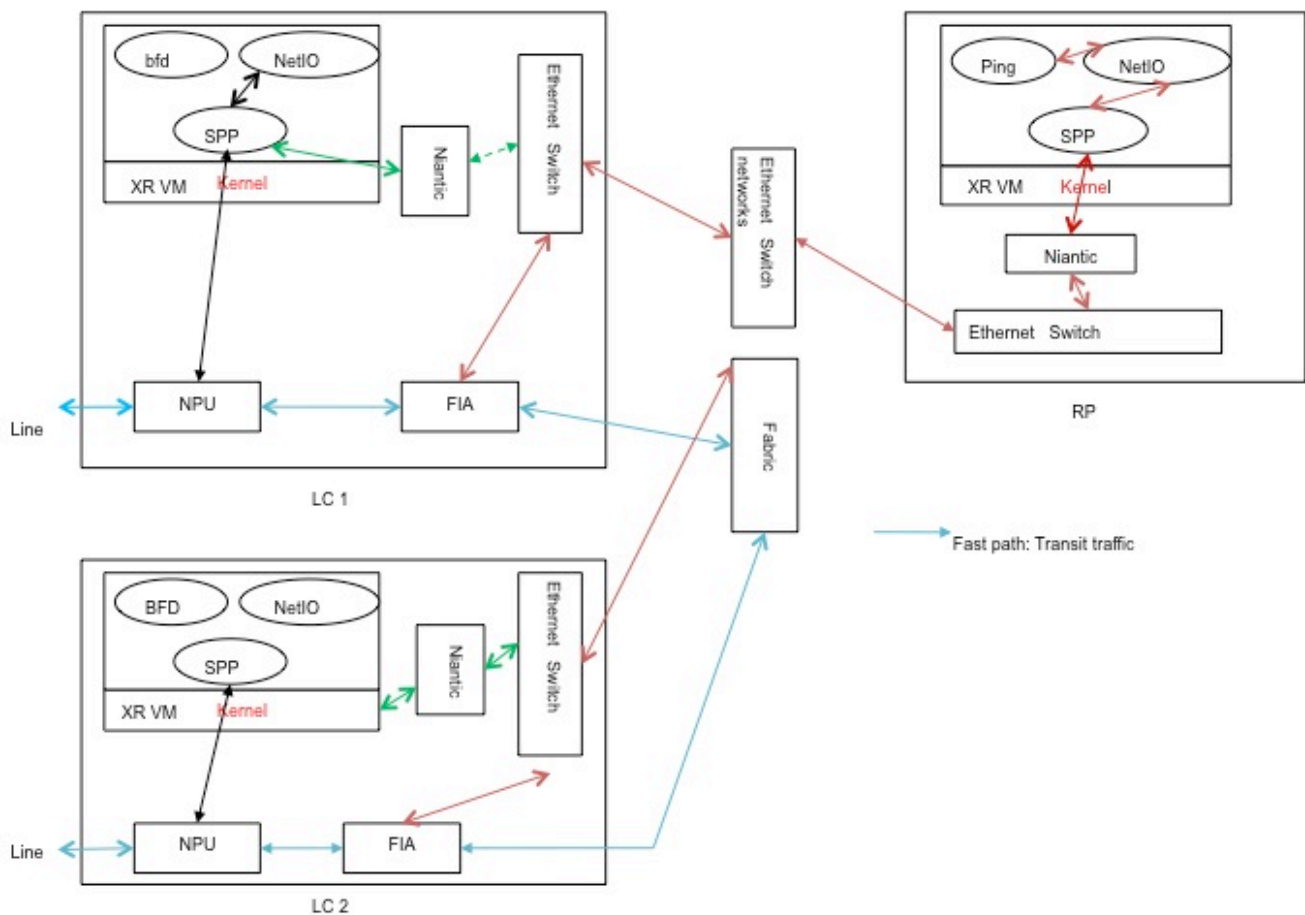


Remontant ces derniers composant d'expédition, vous obtenez un chemin de transfert simplifié pour le scénario de test de ping suivant les indications de la figure 3.

En faire un test de ping de RP, les paquets prennent le chemin de transfert suivant à l'intérieur de la case :

RP_PING <—> RP_NETIO <—> RP_SPP <—> RP_Linux_Kernel_Socket <—> commutateur
 <—> LC_FIA <—> LC_NPU (incluez PSE, PLIM_ASIC) <—> ligne

Figure 3



Vérifiez

Pour le reste du document, un scénario où un ping serait initié du RP sera pris comme exemple. Le ping serait initié directement à un hôte connecté sur Te0/0/0/2/0. Les étapes suivantes afficheront qu'une approche pas à pas vérifiait le chemin de ce paquet de ping.

```
RP/0/RP0/CPU0:NCS6k-Deploy#show ip interface brief
```

Interface	IP-Address	Status	Protocol
Bundle-Ether671	10.67.2.2	Up	Up
Bundle-Ether672	10.67.3.2	Down	Down
Loopback0	10.17.17.17	Up	Up
MgmtEth0/RP0/CPU0/0	10.7.54.11	Up	Up
TenGigE0/0/0/2/0	10.67.1.2	Up	Up
TenGigE0/0/0/2/1	unassigned	Up	Up
TenGigE0/0/0/2/2	unassigned	Up	Up
TenGigE0/0/0/2/3	unassigned	Up	Up
TenGigE0/0/0/2/4	unassigned	Up	Up
TenGigE0/0/0/2/5	unassigned	Down	Down

[snip]

```
RP/0/RP0/CPU0:NCS6k-Deploy#show run interface Ten 0/0/0/2/0
interface TenGigE0/0/0/2/0
  ipv4 address 10.67.1.2 255.255.255.252
  load-interval 30
```

```
RP/0/RP0/CPU0:NCS6k-Deploy#ping 10.67.1.1
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 10.67.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 5/6/7 ms
```

1. le « **show ipv4 traffic** » parent sur le noeud RP, affichera combien échos de Protocole ICMP (Internet Control Message Protocol) ont été envoyés et combien la réponse d'ICMP ont renvoyé.
RP/0/RP0/CPU0:NCS6k-Deploy#**show ipv4 traffic**

```
IP statistics:
  Rcvd: 1495334 total, 80112 local destination
        0 format errors, 0 bad hop count
        23 unknown protocol, 0 not a gateway
        0 security failures, 0 bad source, 0 bad header
        133207 with options, 0 bad, 0 unknown
  Opts: 0 end, 0 nop, 0 basic security, 0 extended security
        0 strict source rt, 0 loose source rt, 0 record rt
        0 stream ID, 0 timestamp, 133207 alert, 0 cipso
  Frags: 0 reassembled, 0 timeouts, 0 couldn't reassemble, 0 fragments received
        0 fragmented, 0 fragment count, 0 fragment max drop
  Bcast: 0 sent, 0 received
  Mcast: 1361652 sent, 1376283 received
        Drop: 0 encapsulation failed, 237 no route, 0 too big
        Sent: 1437435 total
```

```
ICMP statistics:
  Sent: 0 admin unreachable, 63 network unreachable
        8 host unreachable, 0 protocol unreachable
        16 port unreachable, 0 fragment unreachable
        0 time to live exceeded, 0 reassembly ttl exceeded
        24 echo request, 30024 echo reply
        0 mask request, 0 mask reply
        0 parameter error, 0 redirects
        30131 total
  Rcvd: 0 admin unreachable, 21 network unreachable
        0 host unreachable, 0 protocol unreachable
        0 port unreachable, 0 fragment unreachable
        0 time to live exceeded, 0 reassembly ttl exceeded
        30024 echo request, 15 echo reply
        0 mask request, 0 mask reply
        0 redirect, 0 parameter error
        0 source quench, 0 timestamp, 0 timestamp reply
        0 router advertisement, 0 router solicitation
        30063 total, 0 checksum errors, 0 unknown
```

2. Composant de l'entrée sortie de réseau de contrôle (NETIO).L'étape suivante est de vérifier le compteur de chaîne RP FINT NETIO. Vous devez voir « » le compteur du noeud d'ipv4 dans la chaîne de netio. S'il incrémente, il signifie que les paquets ont atteint le composant NETIO et sont envoyés du composant NETIO.
check initial NETIO counter value.

```
RP/0/RP0/CPU0:NCS6k-Deploy#sh netio chains FINT loc 0/rp0/cpu0 | in Stats
<Protocol number> (name) Stats
<6> (fint_n2n) Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
<10> (clns) Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
<12> (ipv4) Stats IN: 2788 pkts, 115373 bytes; OUT: 2816 pkts, 117933 bytes
<13> (mpls) Stats IN: 16482 pkts, 2467508 bytes; OUT: 0 pkts, 0 bytes
<18> (lpts) Stats IN: 47234 pkts, 10381065 bytes; OUT: 0 pkts, 0 bytes
<19> (ipv6) Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
<30> (ipv4_preroute) Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
<32> (ipv6_preroute) Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
<34> (fint_proto_tp) Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
<36> (l2transport) Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
```

Initiate 10 ping packets.

```
RP/0/RP0/CPU0:NCS6k-Deploy#ping 10.67.1.1 coun 10
Type escape sequence to abort.
Sending 10, 100-byte ICMP Echos to 10.67.1.1, timeout is 2 seconds:
!!!!!!!!!!!!
Success rate is 100 percent (10/10), round-trip min/avg/max = 4/7/8 ms
```

Check NETIO counter again. You would see increment of 10 packets.

```
RP/0/RP0/CPU0:NCS6k-Deploy#sh netio chains FINT loc 0/rp0/cpu0 | in Stats
<Protocol number> (name) Stats
<6> (fint_n2n) Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
<10> (clns) Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
<12> (ipv4) Stats IN: 2788 pkts, 115373 bytes; OUT: 2826 pkts, 118933 bytes
<13> (mpls) Stats IN: 16482 pkts, 2467508 bytes; OUT: 0 pkts, 0 bytes
<18> (lpts) Stats IN: 47234 pkts, 10381065 bytes; OUT: 0 pkts, 0 bytes
<19> (ipv6) Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
<30> (ipv4_preroute) Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
<32> (ipv6_preroute) Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
<34> (fint_proto_tp) Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
<36> (l2transport) Stats IN: 0 pkts, 0 bytes; OUT: 0 pkts, 0 bytes
```

Vous pouvez également utiliser la commande « show_netio_fwder_stats de KornShell (ksh) - g » de vérifier s'injectez/contre- incréments de coup de volée ou pas. Remarque: Dans l'environnement de production, il peut y avoir l'autre trafic de fond qui le rend dur pour vérifier si les paquets de ping atteignaient ce composant ou pas. Comme contournement, vous pouvez utiliser le grand nombre de paquets avec le délai d'attente 0 : « temps 0" du compte 10000 du ping x.x.x.x et contrôle si le compteur incrémente soudainement ou a un pic.**Check initial counter value.**

```
RP/0/RP0/CPU0:NCS6k-Deploy#run show_netio_fwder_stats -g
RECEIVE STATISTICS SUMMARY:
rx_pkts: 2224455
punt_pkts: 2224447
ingress_total_drops: 8
TRANSMIT STATISTICS SUMMARY:
inject_pkts: 2077319
tx_pkts: 2058041
egress_total_drops: 2
RECEIVE STATISTICS DETAILS:
Rx Pkt type stats:
  lpts_pkts: 2220753
Rx Listener tag stats:
  ipv4: 1116092
  ipv6: 658627
  clns: 112549
  ipv4_l1: 286252
  raw4: 23
  raw6: 43984
  ospf_mc4: 45
  ospf_mc6: 2
  udp4: 7
  tcp4: 405
  isis: 2767
Rx Punt reason stats:
  IFIB: 2220753
Rx Drop stats:
  null_fint_ifh_drops: 8
  ingress_total_drops: 8
TRANSMIT STATISTICS DETAILS:
Tx Pkt type stats:
  ipv4: 2852
```

```
mpls: 42647
osi: 78760
ipv4_preroute: 1339401
ipv6_preroute: 613659
Tx Protocol Id stats:
  clns: 78760
  ipv4: 2852
  mpls: 42647
  ipv4_preroute: 1339401
  ipv6_preroute: 613659
Tx Drop stats:
  invalid_queue_drops: 2
  hdr_init_drops: 2
  egress_total_drops: 2
```

Initiate 10 ping packets.

```
RP/0/RP0/CPU0:NCS6k-Deploy#ping 10.67.1.1 coun 10
Type escape sequence to abort.
Sending 10, 100-byte ICMP Echos to 10.67.1.1, timeout is 2 seconds:
!!!!!!!!!!!!
Success rate is 100 percent (10/10), round-trip min/avg/max = 3/4/7 ms
```

Check counter again to check to se increment of 10 packets.

```
RP/0/RP0/CPU0:NCS6k-Deploy#run show_netio_fwder_stats -g
```

```
RECEIVE STATISTICS SUMMARY:
```

```
rx_pkts: 2224465
```

```
punt_pkts: 2224457
```

```
ingress_total_drops: 8
```

```
TRANSMIT STATISTICS SUMMARY:
```

```
inject_pkts: 2077332
```

```
tx_pkts: 2058051
```

```
egress_total_drops: 2
```

```
RECEIVE STATISTICS DETAILS:
```

```
Rx Pkt type stats:
```

```
  lpts_pkts: 2220763
```

```
Rx Listener tag stats:
```

```
  ipv4: 1116102
```

```
  ipv6: 658627
```

```
  clns: 112549
```

```
  ipv4_l: 286252
```

```
  raw4: 23
```

```
  raw6: 43984
```

```
  ospf_mc4: 45
```

```
  ospf_mc6: 2
```

```
  udp4: 7
```

```
  tcp4: 405
```

```
  isis: 2767
```

```
Rx Punt reason stats:
```

```
  IFIB: 2220763
```

```
Rx Drop stats:
```

```
  null_fint_ifh_drops: 8
```

```
  ingress_total_drops: 8
```

```
TRANSMIT STATISTICS DETAILS:
```

```
Tx Pkt type stats:
```

```
  ipv4: 2865
```

```
  mpls: 42647
```

```
  osi: 78760
```

```
  ipv4_preroute: 1339401
```

```
  ipv6_preroute: 613659
```

```
Tx Protocol Id stats:
```

```
  clns: 78760
```

```
  ipv4: 2865
```

```
  mpls: 42647
```

```

    ipv4_preroute: 1339401
    ipv6_preroute: 613659
Tx Drop stats:
    invalid_queue_drops: 2
    hdr_init_drops: 2
    egress_total_drops: 2
RP/0/RP0/CPU0:NCS6k-Deploy#

```

3. Espèces de contrôle composantes. Employez les espèces CLI pour voir si le paquet atteignait des espèces ou pas. **Check initial counter value.**

```

RP/0/RP0/CPU0:NCS6k-Deploy#sh spp node-counters
0/0/CPU0:
pdma/rx
    slicel high pkts:          10
-----
pdma/tx
    slicel low pkts:          10
-----
panini/classify
    forwarded to spp clients: 10
-----
client/inject
    pkts injected into spp:   10
-----
client/punt
    punted to client:         10
-----

0/RP0/CPU0:
panini/classify
    forwarded to spp clients: 22070
-----
client/inject  pkts injected into spp: 4640
-----
socket/rx
    ce low pkts:              45
    mgmt interface pkts:      22025
-----
socket/tx
    ce pkts:                  45
    mgmt interface pkts:      4595
-----
client/punt  punted to client: 22070
-----

```

Initiate 100 ping packets.

```

RP/0/RP0/CPU0:NCS6k-Deploy#ping 10.67.1.1 count 100
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 10.67.1.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Success rate is 100 percent (100/100), round-trip
min/avg/max = 3/3/8 ms

```

Check counter again to see increment of 100 packets.

```

RP/0/RP0/CPU0:NCS6k-Deploy#sh spp node-counters
0/0/CPU0:
pdma/rx
    slicel high pkts:          10
-----
pdma/tx
    slicel low pkts:          10
-----
panini/classify

```



```

        forwarded to spp clients:                10
-----
client/inject
        pkts injected into spp:                10
-----
client/punt
        punted to client:                      10
-----

0/RP0/CPU0:
panini/classify
        forwarded to spp clients:              22172
-----
client/inject  pkts injected into spp: 4740
-----
socket/rx
        ce low pkts:                          145
        mgmt interface pkts:                  22027
-----
socket/tx
        ce pkts:                              145
        mgmt interface pkts:                  4595
-----
client/punt punted to client:    22172
-----

```

4. Utilisez les outils de tcpdump pour vider le paquet du composant de kernel Linux. De la sortie ci-dessous, sous la VM ksh NCS6K XR, vous pouvez voir plusieurs sous interfaces

```

:RP/0/RP0/CPU0:NCS6008-SJ#
RP/0/RP0/CPU0:NCS6008-SJ#run
Tue Jun 24 10:51:51.972 UTC
[xr-vm_node0_RP0_CPU0:/]$
[xr-vm_node0_RP0_CPU0:/]$ ifconfig -a
eth-vf1  Link encap:Ethernet  HWaddr 46:91:EE:A5:48:A8
        inet6 addr: fe80::4491:eeff:fea5:48a8/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:9700  Metric:1
        RX packets:518403076C3 errors:0 dropped:0 overruns:0 frame:0 TX packets:969599306
errors:0 dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:1000 RX bytes:138405352234
(128.9 GiB) TX bytes:242828863250 (226.1 GiB) eth-vf1.514 Link encap:Ethernet HWaddr
4C:4E:35:B6:63:68 inet6 addr: fe80::4e4e:35ff:feb6:6368/64 Scope:Link UP BROADCAST RUNNING
MULTICAST MTU:9700 Metric:1 RX packets:13547000 errors:0 dropped:0 overruns:0 frame:0 TX
packets:116957 errors:0 dropped:10 overruns:0 carrier:0 collisions:0 txqueuelen:0 RX
bytes:623478135C3 (594.5 MiB) TX bytes:26876899 (25.6 MiB) eth-vf1.3073 Link encap:Ethernet
HWaddr 4C:4E:35:B6:63:69 inet addr:192.0.0.4 Bcast:192.255.255.255 Mask:255.0.0.0 inet6
addr: fe80::4e4e:35ff:feb6:6369/64 Scope:Link UP BROADCAST RUNNING MULTICAST MTU:9700
Metric:1 RX packets:102364757 errors:0 dropped:0 overruns:0 frame:0 TX packets:100689507
errors:0 dropped:3 overruns:0 carrier:0 collisions:0 txqueuelen:0 RX bytes:29925046692
(27.8 GiB) TX bytes:7562528012 (7.0 GiB) eth-vf1.3074 Link encap:Ethernet HWaddr
4E:41:50:00:10:01 inet addr:172.0.16.1 Bcast:172.255.255.255 Mask:255.0.0.0 inet6 addr:
fe80::4c41:50ff:fe00:1001/64 Scope:Link UP BROADCAST RUNNING MULTICAST MTU:9700 Metric:1 RX
packets:402491385 errors:0 dropped:0 overruns:0 frame:0 TX packets:350389778 errors:0
dropped:6 overruns:0 carrier:0 collisions:0 txqueuelen:0 RX bytes:100599198478 (93.6 GiB)
TX bytes:96834116492 (90.1 GiB) lo Link encap:Local Loopback inet addr:127.0.0.1
Mask:255.0.0.0 inet6 addr: ::1/128 Scope:Host UP LOOPBACK RUNNING MTU:16436 Metric:1 RX
packets:1029861486 errors:0 dropped:0 overruns:0 frame:0 TX packets:1029861486 errors:0
dropped:0 overruns:0 carrier:0 collisions:0 txqueuelen:0 RX bytes:201624257033 (187.7 GiB)
TX bytes:201624257033 (187.7 GiB)

```

eth-vf1.514 est utilisé pour la transmission avec l'interface de Mgmtether mais vous ne pouvez pas voir l'ipv4 adres. L'interface de Mgmtether dans la VM XR se fonde sur la pile IP d'IOS-XR au lieu de la pile IP dans le Linux.**ether-vf1.3073** est utilisé pour la transmission avec la VM d'admin.**ether-vf1.3074** est utilisé pour le trafic d'avion de contrôle associé par VM XR. Le paquet de test de ping traversera cette sous-interface (utilisant la pile de protocoles de réseau Linux). Tcpdump a associé avec le Linux a le sort

d'options sur la façon dont vider le trafic intéressant. En outre, vous pouvez utiliser des outils de tcpdump pour renifler le trafic sécurisé d'avion de contrôle de routeur de domaine (SDR) (VLAN 3074) ou pour renifler l'autre trafic comme la transmission de processus inter de la transmission (IPC) dans le VLAN 3073.

```
xr-vm_node0_RP0_CPU0:/]$ tcpdump -i eth-vf1.3074 -xx -vv
```

```
tcpdump: listening on eth-vf1.3074, link-type EN10MB (Ethernet), capture size 65535 bytes
01:49:21.798386 IP (tos 0x6,ECT(0), ttl 1, id 0, offset 0, flags [DF], proto UDP (17), length 340)
```

```
172.0.16.1.10150 > 239.255.0.4.10150: [bad udp cksum ab2a!] UDP, length 312
```

```
0x0000: 0100 5e7f 0004 4e41 5000 1001 0800 4506 ..^...NAP.....E.
0x0010: 0154 0000 4000 0111 cc8e ac00 1001 efff .T...@.....
0x0020: 0004 27a6 27a6 0140 ad56 abcd abcd 0000 ..'...'@.V.....
0x0030: 0000 0280 f502 0000 0000 0000 0000 0000 .....
0x0040: 0000 0000 0000 7856 3412 0128 0204 0000 .....xV4..(....
0x0050: 0000 5508 0100 0100 0000 3c25 2600 0000 ..U.....<%&...
0x0060: 0000 d007 0000 0000 0000 ffff 0000 0000 .....
0x0070: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0080: 0000 0000 0000 4800 0000 0200 0000 0000 .....H.....
0x0090: 0000 8800 0000 0000 0000 0000 0000 0000 .....
0x00a0: 0000 0100 0000 0000 0000 0000 0000 0000 .....
0x00b0: 0000 0000 0000 c2ca 0031 0000 0000 0000 .....l.....
0x00c0: 0000 0000 0000 0000 0000 5508 0000 6510 .....U...e.
0x00d0: 0000 ed53 4c00 0000 0000 0000 0000 0000 ...SL.....
0x00e0: 0000 0000 0000 0000 0000 0000 0000 6264 .....bd
0x00f0: 7863 0000 0000 0000 0000 0000 0000 0000 xc.....
0x0100: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0110: 0000 0100 0000 0000 0000 0000 0000 30ff .....0.
0x0120: 0002 0000 0000 0000 0000 0000 0000 0000 .....
0x0130: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0140: 0000 0000 0000 0000 0000 0c00 0000 0000 .....
0x0150: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0160: 0000 ..
```

```
01:49:21.799167 IP (tos 0x6,ECT(0), ttl 64, id 0, offset 0, flags [DF], proto UDP (17), length 380)
```

```
172.0.0.1.8197 > 172.0.16.1.8197: [udp sum ok] UDP, length 352
```

```
0x0000: 4e41 5000 1001 4e41 5000 0001 0800 4506 NAP...NAP.....E.
0x0010: 017c 0000 4000 4011 d168 ac00 0001 ac00 .|...@.
0x0040: 0000 0000 0000 7856 3412 0128 0204 0000 .....xV4..(....
0x0050: 0000 5508 0100 0100 0000 3d25 2600 0000 ..U.....=%&...
0x0060: 0000 d007 0000 0000 0000 ffff 0000 0000 .....
0x0070: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0080: 0000 0000 0000 4800 0000 0200 0000 0000 .....H.....
0x0090: 0000 8800 0000 0000 0000 0000 0000 0000 .....
0x00a0: 0000 0100 0000 0000 0000 0000 0000 0000 .....
0x00b0: 0000 0000 0000 c2ca 0031 0000 0000 0000 .....l.....
0x00c0: 0000 0000 0000 0000 0000 5508 0000 6510 .....U...e.
0x00d0: 0000 ee53 4c00 0000 0000 0000 0000 0000 ...SL.....
0x00e0: 0000 0000 0000 0000 0000 0000 0000 6264 .....bd
0x00f0: 7863 0000 0000 0000 0000 0000 0000 0000 xc.....
0x0100: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0110: 0000 0100 0000 0000 0000 0000 0000 30ff .....0.
0x0120: 0002 0000 0000 0000 0000 0000 0000 0000 .....
0x0130: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0140: 0000 0000 0000 0000 0000 0c04 0000 0000 .....
0x0150: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0160: 0000 ..
```

```
01:49:21.802982 IP (tos 0x6,ECT(0), ttl 64, id 0, offset 0, flags [DF], proto UDP (17), length 380)
```

```
172.0.0.1.8197 > 172.0.16.1.8197: [udp sum ok] UDP, length 352
```

```
0x0000: 4e41 5000 1001 4e41 5000 0001 0800 4506 NAP...NAP.....E.
0x0010: 017c 0000 4000 4011 d168 ac00 0001 ac00 .|...@.@.h.....
0x0020: 1001 2005 2005 0168 672f abcd abcd 0000 .....hg/.....
```

```

0x0030: 0000 3c80 f502 0000 0000 0000 0000 0000 ..<.....
0x0040: 0000 0000 0000 7856 3412 0411 0008 0000 .....xV4.....
0x0050: 0000 5508 0000 0100 0000 3d25 2600 0000 ..U.....=%&...
0x0060: 0000 d007 0100 0000 0000 ffff 0000 0000

```

[snip] Remarque: Puisque c'est scénario VM, le trafic envoyé à la VM peut être encapsulé avec l'adresse d'interface VM dans l'en-tête externe de sorte que ce trafic puisse atteindre l'interface VM.

Le vidage mémoire ci-dessus de paquet est a été encapsulé réellement avec l'en-tête de paquet UDP avec la source/destination 172.0.16.1, qui est l'IP address eth-vf1.3074 dans la VM IOS-XR. Remarque: Les captures prises doivent expliquer l'approche et n'ont pas le trafic de Protocole ICMP (Internet Control Message Protocol).

5. Vérifier le composant FIA sur le linecard. `check initial counter value.`

```
RP/0/RP0/CPU0:NCS6k-Deploy#sh controllers fia statistics instance 1 loc 0/0/cpu0
```

```
FIA Statistics Rack: 0, Slot: 0, Asic instance: 1
```

```
FIA Rx (To Fabric) Statistics.
```

```
----- Input Pkt counters
Pkts Bytes Rx pkts from pse : 250 53000 Rx pkts from switch : 993528 349564509 bcast pkts
from switch : 0 mcast pkts from switch : 993278 ucast pkts from switch      :
```

```
250
```

```

Rx pkts enqueued(IQM)           :                500           86500
Rx pkts dequeued(IQM)           :                500           86500
Rx pkts sent to fabric           :                500

```

```
Cell counters:
```

```

Data cells sent to fabric        :                500           86500
Control cells sent to fabric     :           183039783411

```

```
Drop counters:
```

```

Rx burst error drops(NBI)       :                0
Rx error drops(Switch)          :                0
Rx error drops(pse)             :                0
Rx pkt discard drops(IQM)       :           993277           334570329
Pkt crc error drops(FDT)        :                0
Unreachable dest cell drops     :                0
Internal Error Count            :           41984110
Internal Drop Count             :                0

```

```
FIA Tx (From Fabric) Statistics
```

```
----- Cell counters:
Pkts Bytes Data cells : 500 Control cells : 179368087015 Reassembled packet counters: Pkts
received from fabric : 500 Tx Ucast pkts : 500 86500 Tx Mcast pkts : 0 0 Tx pkts (EPNI) :
500 81000 Tx pkts sent to switch : 250 53000 Bcast pkts sent to switch : 0 Mcast pkts sent
to switch : 0 Ucast pkts sent to switch : 250 Tx segments sent to pse      :
```

```
250
```

```
29000
```

```

Tx pkts sent to pse (NBI)       :                500           49000

```

```
Drop counters:
```

```

Tx pkts dropped EPNI            :                0
Tx Ucast pkts dropped           :                0
Tx Mcast pkts dropped           :                0
Tx pkts dropped in EGQ(RQP + EHP):                0
Control cell Drops              :                0
Data cell Drops                 :                0
Tx pkts dropped switch          :                0
Tx pkts dropped pse             :                0
Internal Error Count            :                0
Internal Drop Count             :                0

```


Bytes: 477000

EGRESS

From Fabric:

Packets: 2250

Bytes: 261000

To TM:

Packets: 2272

To L2 [LSIM]:

Packets: 2261

Bytes: 256962

TO/FROM CPU

To CPU:

Packets: 11

From CPU:

Packets: 11

7. Vérifier le module d'interface de couche physique (PLIM) ASIC pare. Check initial counter value.

```
RP/0/RP0/CPU0:NCS6k-Deploy#sh controllers plim asic statistics interface Te0/0/0/2/0
Node: 0/0/CPU0
```

```
-----
TenGigE0/0/0/2/0 Tx Statistics ----- Total Packets
: 2256 Total Bytes : 265884 Total Good Packets : 2256 Total Good Bytes : 265884 Unicast
```

Packets	: 2256	Multicast Packets	: 0
Broadcast Packets	: 0	64 Byte Packets	: 6
65to127 Byte Packets	: 2250	128to255 Byte Packets	: 0
256to511 Byte Packets	: 0	512to1023 Byte Packets	: 0
1024to1518 Byte Packets	: 0	1519to1522 Byte Packets	: 0
1523to1548 Byte Packets	: 0	1549to2000 Byte Packets	: 0
2001to_MRU Byte Packets	: 0	Non Pause BPDU Packets	: 0
Classic Pause Packets	: 0		
Class Based Pause Pkts 0	: 0	Class Based Pause Pkts 1	: 0
Class Based Pause Pkts 2	: 0	Class Based Pause Pkts 3	: 0
Class Based Pause Pkts 4	: 0	Class Based Pause Pkts 5	: 0
Class Based Pause Pkts 6	: 0	Class Based Pause Pkts 7	: 0

Dropped Packets

=====

Drained Packets	: 0	Abort	: 0
Length Error	: 0	Giant	: 0
Tail Drop: HP Queue	: 0	Tail Drop: LP Queue	: 0

TenGigE0/0/0/2/0 Rx Statistics

```
Total Packets : 2256 Total Bytes : 265884 Total Good Packets : 2256 Total Good Bytes :
265884 Unicast Packets : 2256 Multicast Packets : 0
```

Broadcast Packets	: 0	64 Byte Packets	: 6
65to127 Byte Packets	: 2250	128to255 Byte Packets	: 0
256to511 Byte Packets	: 0	512to1023 Byte Packets	: 0
1024to1518 Byte Packets	: 0	1519to1522 Byte Packets	: 0
1523to1548 Byte Packets	: 0	1549to2000 Byte Packets	: 0
2001to_MRU Byte Packets	: 0	Non Pause BPDU Packets	: 0
Classic Pause Packets	: 0		
Class Based Pause Pkts 0	: 0	Class Based Pause Pkts 1	: 0
Class Based Pause Pkts 2	: 0	Class Based Pause Pkts 3	: 0
Class Based Pause Pkts 4	: 0	Class Based Pause Pkts 5	: 0
Class Based Pause Pkts 6	: 0	Class Based Pause Pkts 7	: 0

Dropped Packets

2001to_MRU Byte Packets	: 0	Non Pause BPDUs Packets	: 0
Classic Pause Packets	: 0		
Class Based Pause Pkts 0	: 0	Class Based Pause Pkts 1	: 0
Class Based Pause Pkts 2	: 0	Class Based Pause Pkts 3	: 0
Class Based Pause Pkts 4	: 0	Class Based Pause Pkts 5	: 0
Class Based Pause Pkts 6	: 0	Class Based Pause Pkts 7	: 0

Dropped Packets

=====

Drained Packets	: 0	Abort	: 0
Length Error	: 0	Giant	: 0
Tail Drop: HP Queue	: 0	Tail Drop: LP Queue	: 0

TenGigE0/0/0/2/0 Rx Statistics

Total Packets : 3256 Total Bytes : 383884 Total Good Packets : 3256 Total Good Bytes : 383884 **Unicast Packets : 3256** Multicast Packets : 0

Broadcast Packets	: 0	64 Byte Packets	: 6
65to127 Byte Packets	: 3250	128to255 Byte Packets	: 0
256to511 Byte Packets	: 0	512to1023 Byte Packets	: 0
1024to1518 Byte Packets	: 0	1519to1522 Byte Packets	: 0
1523to1548 Byte Packets	: 0	1549to2000 Byte Packets	: 0
2001to_MRU Byte Packets	: 0	Non Pause BPDUs Packets	: 0
Classic Pause Packets	: 0		
Class Based Pause Pkts 0	: 0	Class Based Pause Pkts 1	: 0
Class Based Pause Pkts 2	: 0	Class Based Pause Pkts 3	: 0
Class Based Pause Pkts 4	: 0	Class Based Pause Pkts 5	: 0
Class Based Pause Pkts 6	: 0	Class Based Pause Pkts 7	: 0

Dropped Packets

=====

Runts	: 0	Fragments	: 0
Jumbo	: 0	Jabber	: 0
CRC	: 0	Code Error	: 0
Code Violation	: 0	Bad Preamble	: 0
IPG Violation	: 0		
Packet HPQ QoS Ctl Drop	: 0	Bytes HPQ QoS Ctl Drop	: 0
Packet HPQ QoS HP Drop	: 0	Bytes HPQ QoS HP Drop	: 0
Packet HPQ Ctl Tail Drop	: 0	Bytes HPQ Ctl Tail Drop	: 0
Packet HPQ HP Tail Drop	: 0	Bytes HPQ HP Tail Drop	: 0
Packet LPQ LP1 Tail Drop	: 0	Bytes LPQ LP1 Tail Drop	: 0
Packet LPQ LP2 Tail Drop	: 0	Bytes LPQ LP2 Tail Drop	: 0
Packet TCAM Miss	: 0	Bytes TCAM Miss	: 0
Packet EOP Abort Drop	: 0	Bytes EOP Abort Drop	: 0
Packet Policy Deny	: 0	Bytes Policy Deny	: 0

Rx Packet Drop Details

=====

Unknown Dest MAC Pkts	: 0		
Unknown E-Type Pkts	: 0		
Unknown Encap Pkts	: 0	Unknown Encap Bytes	: 0
Unknown VLAN Pkts	: 0	Unknown VLAN Bytes	: 0
L2 Subif VLAN Deny Pkts	: 0	L2 Subif VLAN Deny Bytes	: 0

Rx Accepted Packet Details

=====

Packet HPQ CTL Sent	: 6	Bytes HPQ CTL Sent	: 384
Packet HPQ HP Sent	: 0	Bytes HPQ HP Sent	: 0
Packet LPQ LP1 Sent	: 0	Bytes LPQ LP1 Sent	: 0
Packet LPQ LP2 Sent	: 0	Bytes LPQ LP2 Sent	: 0

MTU 1514 bytes, BW 10000000 Kbit (Max: 10000000 Kbit)
reliability 255/255, txload 0/255, rxload 0/255
Encapsulation ARPA,
Full-duplex, 10000Mb/s, SR, link type is force-up
output flow control is off, input flow control is off
loopback not set,
ARP type ARPA, ARP timeout 04:00:00
Last input 00:00:00, output 00:00:00
Last clearing of "show interface" counters 22:09:38
30 second input rate 1000 bits/sec, 2 packets/sec
30 second output rate 1000 bits/sec, 2 packets/sec
4256 packets input, 484860 bytes, 0 total input drops
0 drops for unrecognized upper-level protocol
Received 0 broadcast packets, 0 multicast packets
0 runts, 0 giants, 0 throttles, 0 parity
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
4256 packets output, 484860 bytes, 0 total output drops
Output 0 broadcast packets, 0 multicast packets
0 output errors, 0 underruns, 0 applique, 0 resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions