Dépannage de l'utilisation CPU élevée dans les plates-formes de commutation Catalyst exécutant IOS-XE 16.x

Table des matières

Introduction

Ce document décrit comment résoudre les problèmes d'utilisation élevée du CPU, principalement dus aux interruptions, sur les nouvelles plates-formes Cisco IOS®-XE qui exécutent les versions 16.x (également appelées Polaris). En outre, ce document introduit plusieurs nouvelles commandes sur cette plate-forme qui sont intégrées afin de dépanner de tels problèmes.

Informations générales

Il est important de comprendre comment Cisco IOS®-XE est conçu. Avec Cisco IOS®-XE, Cisco est passé à un noyau Linux et tous les sous-systèmes ont été décomposés en processus. Tous les sous-systèmes qui se trouvaient auparavant dans Cisco IOS®, tels que les pilotes de modules, la haute disponibilité (HA), etc., s'exécutent désormais en tant que processus logiciels dans le système d'exploitation Linux. Cisco IOS® s'exécute en tant que démon dans le système d'exploitation Linux (IOSd). Cisco IOS®-XE conserve non seulement la même apparence que la plate-forme Cisco IOS® classique, mais également son fonctionnement, sa prise en charge et sa gestion.

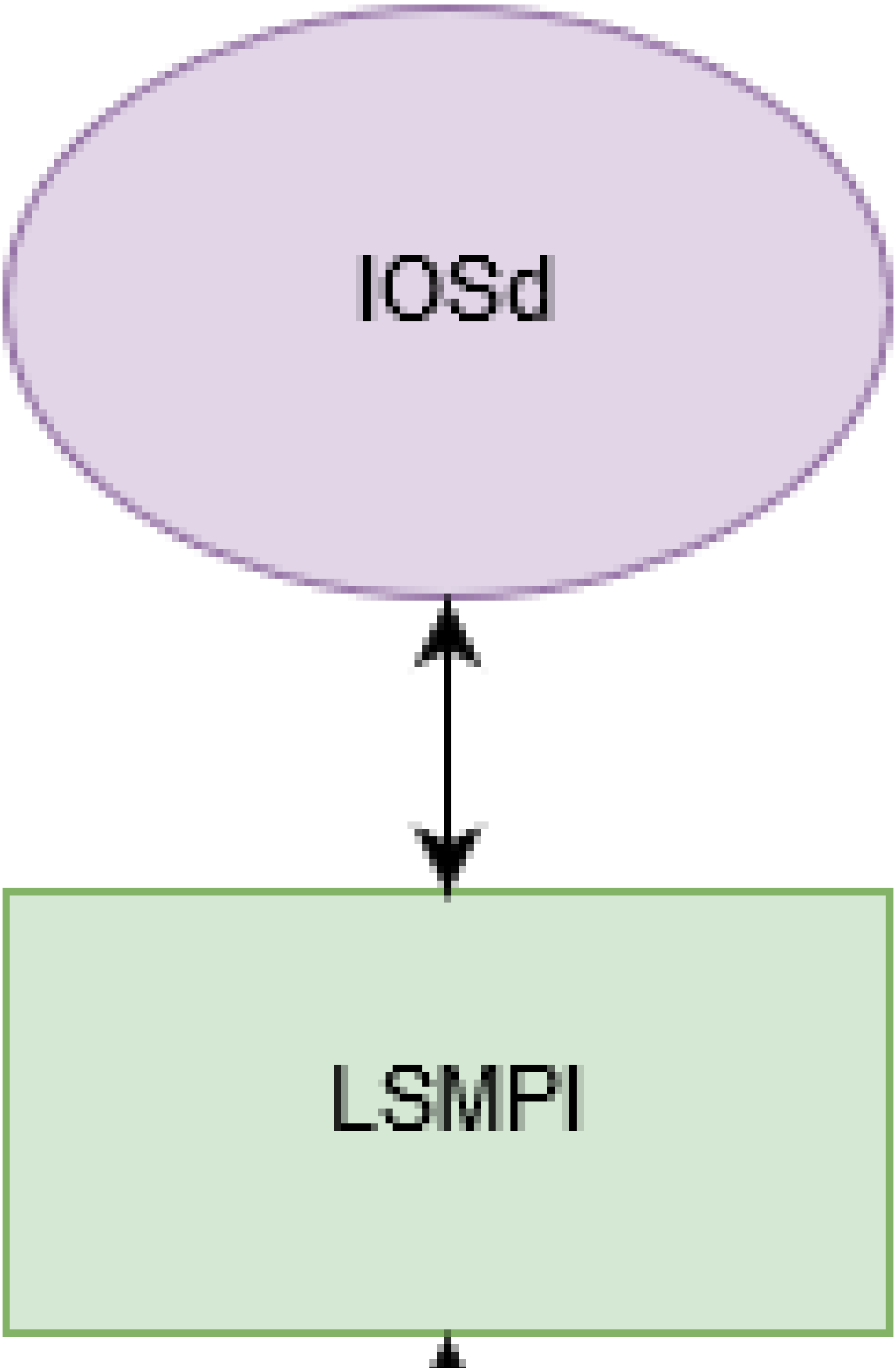
Voici quelques définitions utiles :

- Pilote FED (Forwarding Engine Driver) : il s'agit du coeur du commutateur Cisco Catalyst et il est responsable de la programmation/transmission de tout le matériel
- IOSd : démon Cisco IOS® qui s'exécute sur le noyau Linux. Il est exécuté en tant que processus logiciel dans le noyau
- Packet Delivery System (PDS) : architecture et processus de livraison des paquets vers et

depuis les différents sous-systèmes. Par exemple, il contrôle la manière dont les paquets sont livrés de la FED à l'IOSd et vice versa

- Plan de contrôle (CP) : le plan de contrôle est un terme générique utilisé pour regrouper les fonctions et le trafic qui impliquent le CPU du commutateur Catalyst. Cela inclut le trafic tel que le protocole Spanning Tree (STP), le protocole HSRP (Hot Standby Router Protocol) et les protocoles de routage qui sont destinés au commutateur, ou envoyés à partir du commutateur. Cela inclut également les protocoles de couche application tels que Secure Shell (SSH) et Simple Network Management Protocol (SNMP) qui doivent être gérés par le processeur
- Plan de données (DP) : généralement, le plan de données englobe les circuits ASIC matériels et le trafic qui est transféré sans assistance du plan de contrôle
- Punt : paquet de contrôle de protocole entrant intercepté par le protocole DP et envoyé au PC pour le traiter
- Injection : paquet de protocole généré par le protocole CP envoyé au protocole DP pour sortir sur les interfaces E/S
- LSMPI : interface de point de mémoire partagée Linux

Schéma de haut niveau du chemin de communication entre le plan de données et le plan de contrôle :


IOSd
LSMPI

est utilisée afin d'afficher l'état actuel du processus à l'intérieur du démon IOSd. Lorsque vous ajoutez la sortie, modifiez | exclure 0.00, il filtrera les processus qui sont actuellement inactifs.

Il y a deux éléments d'information précieux de ce résultat :

- Utilisation du processeur pendant cinq secondes : 91 %/30 %
  - Le premier chiffre (91 %) correspond à l'utilisation totale du processeur du commutateur
  - Le deuxième nombre (30 %) correspond à l'utilisation provoquée par les interruptions du plan de données
- Le processus ARP (Address Resolution Protocol) Input est actuellement le principal processus IOS® qui consomme les ressources :

```
<#root>

Switch#

show processes cpu sort | ex 0.00


CPU utilization for five seconds:

91%/30%

; one minute: 30%; five minutes: 8%
 PID Runtime(ms)     Invoked      uSecs   5Sec   1Min   5Min TTY Process
  37       14645         325

45061 59.53% 18.86%  4.38%   0 ARP Input


 137        2288         115      19895  1.20%  0.14%  0.07%   0 Per-minute Jobs
 373        2626       35334         74  0.15%  0.11%  0.09%   0 MMA DB TIMER
 218        3123       69739         44  0.07%  0.09%  0.12%   0 IP ARP Retry Age
 404        2656       35333         75  0.07%  0.09%  0.09%   0 MMA DP TIMER
```

La commande show processes cpu platform sorted est utilisée pour afficher à quoi ressemble l'utilisation du processus à partir du noyau Linux. D'après le résultat, on peut observer que le processus FED est élevé, ce qui est dû aux requêtes ARP envoyées au processus IOSd :

```
<#root>

Switch#

show processes cpu platform sorted

CPU utilization for five seconds: 38%, one minute: 38%, five minutes: 40%
Core 0: CPU utilization for five seconds: 39%, one minute: 37%, five minutes: 39%
Core 1: CPU utilization for five seconds: 41%, one minute: 38%, five minutes: 40%
Core 2: CPU utilization for five seconds: 30%, one minute: 38%, five minutes: 40%
Core 3: CPU utilization for five seconds: 37%, one minute: 39%, five minutes: 41%
   Pid    PPid    5Sec    1Min    5Min  Status        Size  Name
--------------------------------------------------------------------------------
 22701   22439     89%     88%
```

```
88%  R       2187444224  linux_iosd-imag

 11626   11064     46%     47%

48%  S       2476175360  fed main event

  4585       2      7%      9%      9%  S               0  lsmpi-xmit
  4586       2      3%      6%      6%  S               0  lsmpi-rx
```

Étape 2. Examinez pourquoi FED envoie des paquets au plan de contrôle

À l'étape 1, vous pouvez conclure que le processus IOSd/ARP s'exécute à un niveau élevé, mais qu'il est victime du trafic introduit à partir du plan de données. Il est nécessaire d'étudier plus en détail pourquoi le processus FED envoie le trafic au processeur et d'où provient ce trafic.

Le résumé de la cause punt active du commutateur alimenté par le logiciel show platform donne une vue d'ensemble de haut niveau de la raison punt. Tout nombre incrémenté sur plusieurs exécutions de cette commande :

<#root>

```
Switch#

show platform software fed switch active punt cause summary

Statistics for all causes

Cause    Cause Info                                          Rcvd         Dropped
--------------------------------------------------------------------------------

7        ARP request or response                             18444227     0

11       For-us data                                         16           0
21       RP<->QFP keepalive                                  3367         0
24       Glean adjacency                                     2            0
55       For-us control                                      6787         0
60       IP subnet or broadcast packet                       14           0
96       Layer2 control protocols                            3548         0
--------------------------------------------------------------------------------
```

Les paquets qui sont envoyés au plan de contrôle depuis FED utilisent une structure de file d'attente partagée afin de garantir un trafic de contrôle de priorité élevée. Il ne se perd pas derrière le trafic de priorité inférieure, comme ARP. Une vue d'ensemble de haut niveau de ces files d'attente peut être visualisée avec l'utilisation de l'interface cpu active du commutateur alimenté par le logiciel de la plate-forme show. Après avoir exécuté cette commande plusieurs fois, il peut être constaté que la file d'attente Forus Resolution (Forus - qui signifie le trafic destiné au CPU) s'incrémente rapidement.

<#root>

```
Switch#
```

show platform software fed switch active cpu-interface

queue	retrieved	dropped	invalid	hol-block
Routing Protocol	8182	0	0	0
L2 Protocol	161	0	0	0
sw forwarding	2	0	0	0
broadcast	14	0	0	0
icmp gen	0	0	0	0
icmp redirect	0	0	0	0
logging	0	0	0	0
rpf-fail	0	0	0	0
DOT1X authentication	0	0	0	0
Forus Traffic	16	0	0	0
Forus Resolution	**24097779**	**0**	**0**	**0**
Inter FED	0	0	0	0
L2 LVX control	0	0	0	0
EWLC control	0	0	0	0
EWLC data	0	0	0	0
L2 LVX data	0	0	0	0
Learning cache	0	0	0	0
Topology control	4117	0	0	0
Proto snooping	0	0	0	0
DHCP snooping	0	0	0	0
Transit Traffic	0	0	0	0
Multi End station	0	0	0	0
Webauth	0	0	0	0
Crypto control	0	0	0	0
Exception	0	0	0	0
General Punt	0	0	0	0
NFL sampled data	0	0	0	0
Low latency	0	0	0	0
EGR exception	0	0	0	0
FSS	0	0	0	0
Multicast data	0	0	0	0
Gold packet	0	0	0	0

Avec l'utilisation de la commande show platform software fed switch active punt cpuq tout donne une vue plus détaillée de ces files d'attente. La file d'attente 5 est responsable du protocole ARP et, comme prévu, elle s'incrémente sur plusieurs exécutions de la commande. La commande show plat soft fed sw active inject cpuq clear peut être utilisée pour effacer les compteurs pour une lecture plus facile.

<#root>

Switch#

show platform software fed switch active punt cpuq all

<snip>
CPU Q Id : 5

CPU Q Name : CPU_Q_FORUS_ADDR_RESOLUTION

Packets received from ASIC : 21018219
Send to IOSd total attempts : 21018219

```
Send to IOSd failed count      : 0
RX suspend count               : 0
RX unsuspend count             : 0
RX unsuspend send count        : 0
RX unsuspend send failed count : 0
RX consumed count              : 0
RX dropped count               : 0
RX non-active dropped count    : 0
RX conversion failure dropped  : 0
RX INTACK count                : 1050215
RX packets dq'd after intack   : 90
Active RxQ event               : 3677400
RX spurious interrupt          : 1050016
<snip>
```

De là, il y a quelques options. Le protocole ARP est un trafic de diffusion. Vous pouvez donc rechercher les interfaces qui présentent un taux anormalement élevé de trafic de diffusion (également utile pour dépanner les boucles de couche 2). Il peut être nécessaire d'exécuter cette commande plusieurs fois afin de déterminer quelle interface s'incrémente activement.

<#root>

```
Switch#
```

show interfaces counters

```
Port            InOctets    InUcastPkts    InMcastPkts
```

InBcastPkts

```
Gi1/0/1    1041141009678              9              0
```

16267828358

```
Gi1/0/2             1254             11              0              1
Gi1/0/3                0              0              0              0
Gi1/0/4                0              0              0              0
```

L'autre option consiste à utiliser l'outil Embedded Packet Capture (EPC) afin de recueillir un échantillon des paquets qui sont vus au niveau du plan de contrôle.

<#root>

```
Switch#
```

monitor capture cpuCap control-plane in match any file location flash:cpuCap.pcap

```
Switch#
```

show monitor capture cpuCap

```
Status Information for Capture cpuCap
  Target Type:
 Interface: Control Plane, Direction: IN
```

```
   Status : Inactive
  Filter Details:
    Capture all packets
  Buffer Details:
   Buffer Type: LINEAR (default)
  File Details:
   Associated file name: flash:cpuCap.pcap
  Limit Details:
   Number of Packets to capture: 0 (no limit)
   Packet Capture duration: 0 (no limit)
   Packet Size to capture: 0 (no limit)
   Packet sampling rate: 0 (no sampling)
```

Cette commande configure une capture interne sur le commutateur afin de capturer tout trafic qui est envoyé au plan de contrôle. Ce trafic est enregistré dans un fichier de la mémoire flash. Il s'agit d'un fichier pcap normal de wireshark qui peut être exporté à partir d'un commutateur et ouvert dans wireshark pour une analyse plus approfondie.

Démarrez la capture et laissez-la s'exécuter pendant quelques secondes et arrêtez la capture :

<#root>

```
Switch#
monitor capture cpuCap start
Enabling Control plane capture may seriously impact system performance. Do you want to continue? [yes/n
yes
Started capture point : cpuCap
*Jun 14 17:57:43.172: %BUFCAP-6-ENABLE: Capture Point cpuCap enabled.
Switch#
monitor capture cpuCap stop
Capture statistics collected at software:
        Capture duration - 59 seconds
Packets received - 215950
        Packets dropped - 0
        Packets oversized - 0
Bytes dropped in asic - 0
Stopped capture point : cpuCap
Switch#
*Jun 14 17:58:37.884: %BUFCAP-6-DISABLE: Capture Point cpuCap disabled.
```

Il est également possible d'afficher le fichier de capture sur le commutateur :

<#root>

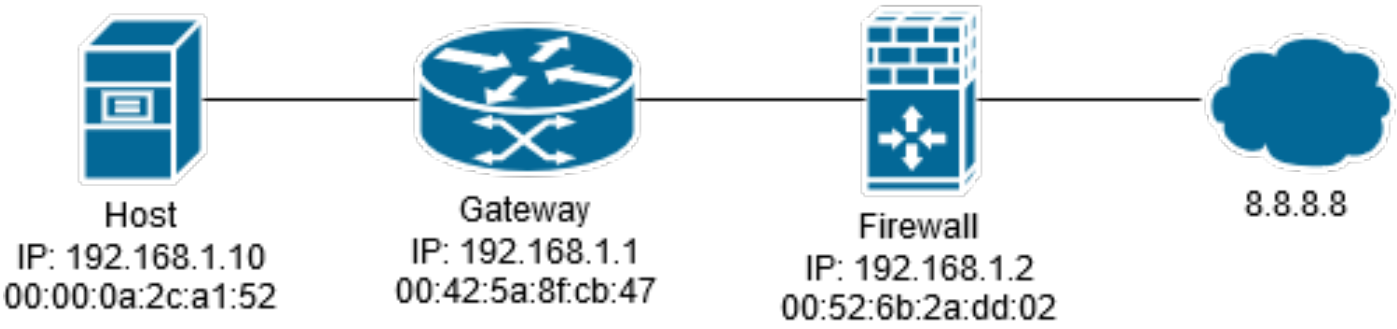
```
Switch#

show monitor capture file flash:cpuCap.pcap

Starting the packet display ........ Press Ctrl + Shift + 6 to exit

  1   0.000000 Xerox_d7:67:a1 -> Broadcast    ARP 60 Who has 192.168.1.24? Tell 192.168.1.2
  2   0.000054 Xerox_d7:67:a1 -> Broadcast    ARP 60 Who has 192.168.1.24? Tell 192.168.1.2
  3   0.000082 Xerox_d7:67:a1 -> Broadcast    ARP 60 Who has 192.168.1.24? Tell 192.168.1.2
  4   0.000109 Xerox_d7:67:a1 -> Broadcast    ARP 60 Who has 192.168.1.24? Tell 192.168.1.2
  5   0.000136 Xerox_d7:67:a1 -> Broadcast    ARP 60 Who has 192.168.1.24? Tell 192.168.1.2
  6   0.000162 Xerox_d7:67:a1 -> Broadcast    ARP 60 Who has 192.168.1.24? Tell 192.168.1.2
  7   0.000188 Xerox_d7:67:a1 -> Broadcast    ARP 60 Who has 192.168.1.24? Tell 192.168.1.2
  8   0.000214 Xerox_d7:67:a1 -> Broadcast    ARP 60 Who has 192.168.1.24? Tell 192.168.1.2
  9   0.000241 Xerox_d7:67:a1 -> Broadcast    ARP 60 Who has 192.168.1.24? Tell 192.168.1.2
```

D'après ce résultat, il est évident que l'hôte 192.168.1.2 est la source des ARP constants qui provoquent la CPU élevée sur le commutateur. Avec l'utilisation des commandes show ip arp et show mac address-table address pour suivre l'hôte et soit le supprimer du réseau, soit adresser les ARP. Il est également possible d'obtenir un détail complet de chaque paquet capturé à l'aide de l'option detail de la commande capture view, show monitor capture file flash:cpuCap.pcap detail. Référez-vous à ce guide pour plus d'informations sur les captures de paquets sur un commutateur Catalyst.

Étude de cas 2. Redirections IP avec CoPP

Par défaut, les commutateurs Catalyst de dernière génération sont protégés par le protocole CoPP (Control Plane Policing). Le protocole CoPP est utilisé pour protéger le processeur contre les attaques malveillantes et les erreurs de configuration susceptibles de compromettre la capacité des commutateurs à maintenir des fonctions critiques telles que le protocole Spanning Tree et les protocoles de routage. Ces protections peuvent conduire à des scénarios où le commutateur n'a qu'un processeur légèrement élevé et des compteurs d'interface clairs, mais où le trafic est abandonné pendant qu'il traverse le commutateur. Il est important de noter l'utilisation de base du processeur sur votre périphérique au moment d'un fonctionnement normal. L'utilisation élevée du CPU n'est pas nécessairement un problème, et cela dépend des fonctionnalités activées sur le périphérique, mais lorsque cette utilisation augmente sans modification de la configuration, cela peut être un signe de préoccupation.

Considérez ce scénario. Les hôtes qui ne sont pas connectés au commutateur de passerelle signalent des vitesses de téléchargement lentes et des pertes de requêtes ping sur Internet. Un contrôle d'intégrité général du commutateur ne montre aucune erreur sur les interfaces ou perte de la requête ping lorsqu'elle provient du commutateur de passerelle.



Lorsque vous vérifiez le processeur, il affiche des nombres légèrement élevés en raison d'interruptions.

<#root>

```
Switch#

show processes cpu sorted | ex 0.00

CPU utilization for five seconds:

 8%/7%

; one minute: 8%; five minutes: 8%
 PID Runtime(ms)     Invoked      uSecs   5Sec   1Min   5Min TTY Process
 122      913359     1990893        458  0.39%  1.29%  1.57%   0

IOSXE-RP Punt

 Se
 147        5823       16416        354  0.07%  0.05%  0.06%   0 PLFM-MGR IPC pro
 404       13237      183032         72  0.07%  0.08%  0.07%   0 MMA DP TIMER
```

Lorsque vous vérifiez l'interface du processeur, vous voyez que le compteur de redirection ICMP est incrémenté de manière active.

<#root>

```
Switch#

show platform software fed switch active cpu-interface

queue                       retrieved   dropped     invalid     hol-block
------------------------------------------------------------------------------
Routing Protocol            12175       0           0           0
L2 Protocol                 236         0           0           0
sw forwarding               714673      0           0           0
broadcast                   2           0           0           0
icmp gen                    0           0           0           0

icmp redirect               2662788     0           0           0

logging                     7           0           0           0
rpf-fail                    0           0           0           0
DOT1X authentication        0           0           0           0
Forus Traffic               21776434    0           0           0
Forus Resolution            724021      0           0           0
```

```
Inter FED                 0           0           0           0
L2 LVX control            0           0           0           0
EWLC control              0           0           0           0
EWLC data                 0           0           0           0
L2 LVX data               0           0           0           0
Learning cache            0           0           0           0
Topology control          6122        0           0           0
Proto snooping            0           0           0           0
DHCP snooping             0           0           0           0
Transit Traffic           0           0           0           0
```

Bien qu'aucune baisse ne soit observée dans FED, si vous cochez CoPP, des chutes peuvent être observées dans la file d'attente de redirection ICMP.

<#root>

```
Switch#
```

show platform hardware fed switch 1 qos queue stats internal cpu policer

```
                          CPU Queue Statistics
============================================================================================
                                              (default)  (set)
```

Queue

```
QId PlcIdx  Queue Name               Enabled   Rate     Rate
```

Drop(Bytes)

```
----------------------------------------------------------------------------
0    11     DOT1X Auth                 Yes     1000      1000     0
1    1      L2 Control                 Yes     2000      2000     0
2    14     Forus traffic              Yes     4000      4000     0
3    0      ICMP GEN                   Yes     600       600      0
4    2      Routing Control            Yes     5400      5400     0
5    14     Forus Address resolution   Yes     4000      4000     0

6    0      ICMP Redirect              Yes     600       600      463538463

7    16     Inter FED Traffic          Yes     2000      2000     0
8    4      L2 LVX Cont Pack           Yes     1000      1000     0
<snip>
```

CoPP est essentiellement une politique QoS placée sur le plan de contrôle du périphérique. CoPP fonctionne comme n'importe quelle autre QoS sur le commutateur : lorsque la file d'attente pour un trafic spécifique est épuisée, le trafic qui utilise cette file d'attente est abandonné. À partir de ces résultats, vous savez que le trafic est commuté par logiciel en raison de redirections ICMP et que ce trafic est abandonné en raison de la limite de débit sur la file d'attente de redirection ICMP. Vous pouvez effectuer une capture sur le plan de contrôle afin de valider que les paquets qui atteignent le plan de contrôle proviennent des utilisateurs.

Pour connaître la logique de correspondance utilisée par chaque classe, vous disposez d’une interface de ligne de commande (CLI) qui vous aide à identifier les types de paquets qui atteignent

une file d'attente donnée. Par exemple, si vous voulez savoir ce qui atteindrait la classe system-cpp-routing-control :

<#root>

```
Switch#
```

show platform software qos copp policy-info

```
Default rates of all classmaps are displayed:
 policy-map system-cpp-policy
 class system-cpp-police-routing-control
 police rate 5400 pps

Switch#
```

show platform software qos copp class-info

```
ACL representable classmap filters are displayed:
```

class-map match-any system-cpp-police-routing-control

```
 description Routing control and Low Latency
 match access-group name system-cpp-mac-match-routing-control
 match access-group name system-cpp-ipv4-match-routing-control
 match access-group name system-cpp-ipv6-match-routing-control
 match access-group name system-cpp-ipv4-match-low-latency
 match access-group name system-cpp-ipv6-match-low-latency


mac access-list extended system-cpp-mac-match-routing-control
 permit any host 0180.C200.0014
 permit any host 0900.2B00.0004
 ip access-list extended system-cpp-ipv4-match-routing-control
 permit udp any any eq rip
 <...snip...>
 ipv6 access-list system-cpp-ipv6-match-routing-control
 permit ipv6 any FF02::1:FF00:0/104
 permit ipv6 any host FF01::1
<...snip...>
 ip access-list extended system-cpp-ipv4-match-low-latency
 permit udp any any eq 3784
 permit udp any any eq 3785
 ipv6 access-list system-cpp-ipv6-match-low-latency
 permit udp any any eq 3784
 permit udp any any eq 3785
 <...snip...>
```

<#root>

```
Switch#
```

monitor capture cpuSPan control-plane in match any file location flash:cpuCap.pcap

```
Control-plane direction IN is already attached to the capture
Switch#
```

```
monitor capture cpuSpan start


Enabling Control plane capture may seriously impact system performance. Do you want to continue? [yes/n
Started capture point : cpuSpan
Switch#
*Jun 15 17:28:52.841: %BUFCAP-6-ENABLE: Capture Point cpuSpan enabled.
Switch#

monitor capture cpuSpan stop

Capture statistics collected at software:
        Capture duration - 12 seconds
        Packets received - 5751
        Packets dropped - 0
        Packets oversized - 0

Bytes dropped in asic - 0

Stopped capture point : cpuSpan
Switch#
*Jun 15 17:29:02.415: %BUFCAP-6-DISABLE: Capture Point cpuSpan disabled.
Switch#

show monitor capture file flash:cpuCap.pcap detailed

Starting the packet display ........ Press Ctrl + Shift + 6 to exit

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
<snip>

Ethernet II, Src: OmronTat_2c:a1:52 (00:00:0a:2c:a1:52), Dst: Cisco_8f:cb:47 (00:42:5a:8f:cb:47)


<snip>

Internet Protocol Version 4, Src: 192.168.1.10, Dst: 8.8.8.8



<snip>
```

Lorsque cet hôte envoie une requête ping vers la version 8.8.8.8, il envoie la requête ping à l’adresse MAC des passerelles, car l’adresse de destination se trouve en dehors du VLAN. Le commutateur de passerelle détecte que le saut suivant se trouve dans le même VLAN et réécrit l'adresse MAC de destination dans le pare-feu et transfère le paquet. Ce processus peut se produire dans le matériel, mais une exception à ce transfert matériel est le processus de redirection IP. Lorsque le commutateur reçoit la requête ping, il détecte qu’il achemine le trafic sur le même VLAN et envoie le trafic au processeur afin de générer un paquet de redirection vers l’hôte. Ce message de redirection informe l’hôte qu’il existe un chemin plus optimal vers la destination. Dans ce cas, le saut suivant de couche 2 est par conception et prévu, le commutateur doit être configuré pour ne pas envoyer les messages de redirection et transférer les paquets dans le matériel. Ceci est fait lorsque vous désactivez les redirections sur l'interface VLAN.

<#root>

```
interface Vlan1
```

```
 ip address 192.168.1.1 255.255.255.0

no ip redirects

end
```

Lorsque les redirections IP sont désactivées, le commutateur réécrit l'adresse MAC et transfère les données dans le matériel.

Étude de cas 3. Processeur élevé intermittent

Dans le cas où la CPU élevée sur le commutateur est intermittente, il est possible de configurer un script sur le commutateur pour exécuter automatiquement ces commandes au moment des événements de CPU élevée. Pour ce faire, vous devez utiliser le gestionnaire d'événements intégré (EEM) Cisco IOS®.

Le paramètre entry-val permet de déterminer la hauteur du processeur avant le déclenchement du script. Le script surveille l'OID SNMP moyen de 5 secondes du processeur. Deux fichiers sont écrits dans la mémoire flash : tac-cpu-<timestamp>.txt contient les sorties de commande et tac-cpu-<timestamp>.pcap contient la capture d'entrée du processeur. Ces dossiers peuvent ensuite être examinés à une date ultérieure.

```
config t
no event manager applet high-cpu authorization bypass
event manager applet high-cpu authorization bypass
event snmp oid 1.3.6.1.4.1.9.9.109.1.1.1.1.3.1 get-type next entry-op gt entry-val 80 poll-interval 1 r
action 0.01 syslog msg "High CPU detected, gathering system information."
action 0.02 cli command "enable"
action 0.03 cli command "term exec prompt timestamp"
action 0.04 cli command "term length 0"
action 0.05 cli command "show clock"
action 0.06 regex "([0-9]|[0-9][0-9]):([0-9]|[0-9][0-9]):([0-9]|[0-9][0-9])" $_cli_result match match1
action 0.07 string replace "$match" 2 2 "."
action 0.08 string replace "$_string_result" 5 5 "."
action 0.09 set time $_string_result
action 1.01 cli command "show proc cpu sort | append flash:tac-cpu-$time.txt"
action 1.02 cli command "show proc cpu hist | append flash:tac-cpu-$time.txt"
action 1.03 cli command "show proc cpu platform sorted | append flash:tac-cpu-$time.txt"
action 1.04 cli command "show interface | append flash:tac-cpu-$time.txt"
action 1.05 cli command "show interface stats | append flash:tac-cpu-$time.txt"
action 1.06 cli command "show log | append flash:tac-cpu-$time.txt"
action 1.07 cli command "show ip traffic | append flash:tac-cpu-$time.txt"
action 1.08 cli command "show users | append flash:tac-cpu-$time.txt"
action 1.09 cli command "show platform software fed switch active punt cause summary | append flash:tac
action 1.10 cli command "show platform software fed switch active cpu-interface | append flash:tac-cpu-
action 1.11 cli command "show platform software fed switch active punt cpuq all | append flash:tac-cpu-
action 2.08 cli command "no monitor capture tac_cpu"
action 2.09 cli command "monitor capture tac_cpu control-plane in match any file location flash:tac-cpu
action 2.10 cli command "monitor capture tac_cpu start" pattern "yes"
action 2.11 cli command "yes"
action 2.12 wait 10
action 2.13 cli command "monitor capture tac_cpu stop"
```

```
action 3.01 cli command "term default length"
action 3.02 cli command "terminal no exec prompt timestamp"
action 3.03 cli command "no monitor capture tac_cpu"
```

Informations connexes

- Cisco IOS-XE 16 - En quelques mots
- Dépannage lors de l'utilisation élevée du processeur des commutateurs de la gamme Catalyst 3850
- Exemple de configuration de la capture de paquets intégrée pour Cisco IOS® et IOS®-XE
- Configuration de la capture de paquets par injection/injection FED sur les commutateurs Catalyst exécutant IOS-XE 16.X
- Assistance et documentation techniques - Cisco Systems

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.