

# Dépannage des pertes de paquets avec les techniques de coloration des paquets ou les compteurs de plate-forme

## Table des matières

---

[Introduction](#)

[Conditions préalables](#)

[Exigences](#)

[Composants utilisés](#)

[Informations générales](#)

[Topologie](#)

[Option 1. Configuration d'ERSPAN avec Flow-id](#)

[Étape 1. Configuration de la destination ESPAN](#)

[Étape 2a. Créer une source d'étendue pour le trafic directement connecté à la SRC](#)

[Étape 2b. Création d'une source étendue pour le trafic directement connecté à l'heure d'été](#)

[Étape 3. Analyse rapide de Wireshark](#)

[Option 2. Compteurs de plate-forme](#)

[Effacer les compteurs de plateforme](#)

[Identification d'une taille de paquet avec des paquets faibles ou nuls](#)

[Suivi du flux de trafic](#)

[Informations connexes](#)

---

## Introduction

Ce document décrit comment suivre un flux réseau à l'aide de techniques de coloration des paquets.

## Conditions préalables

### Exigences

- Connaissances de base de l'ACI
- Groupes de terminaux et contrat
- Connaissances de base de Wireshark

### Composants utilisés

Ce document n'est pas limité à des versions matérielles et logicielles spécifiques.

Périphériques utilisés :

- Cisco ACI version 5.3(2)
- Couvrir la destination
- Commutateurs Gen2

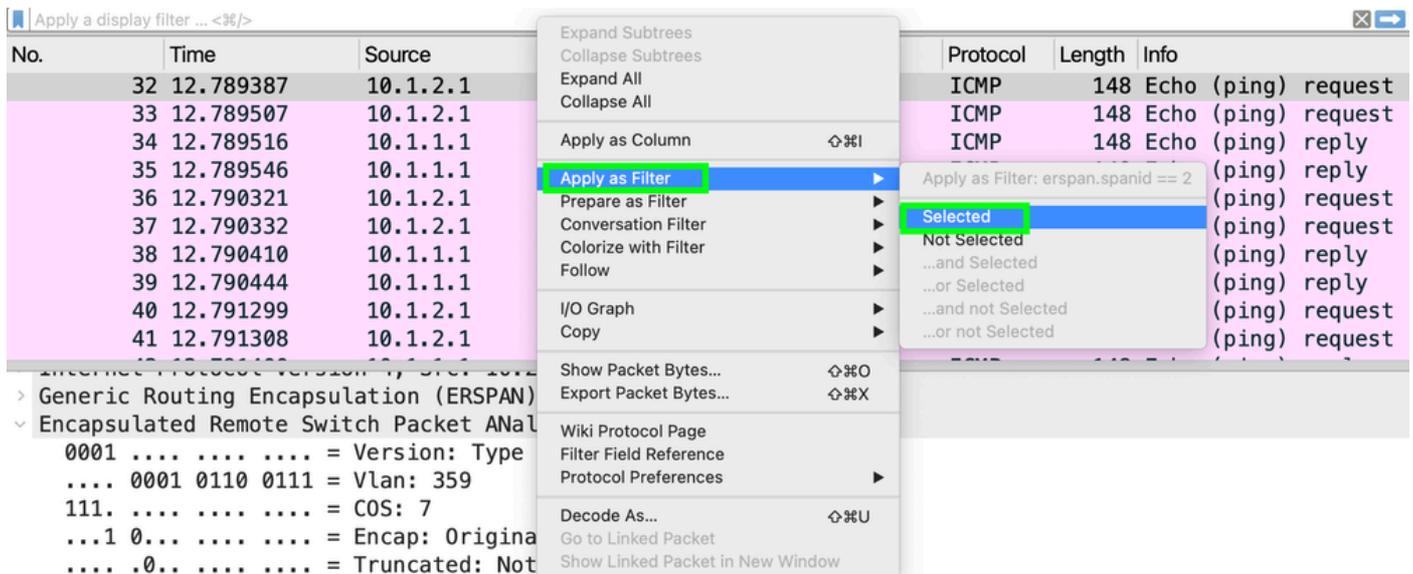
The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Si votre réseau est en ligne, assurez-vous de bien comprendre l'incidence possible des commandes.

## Informations générales

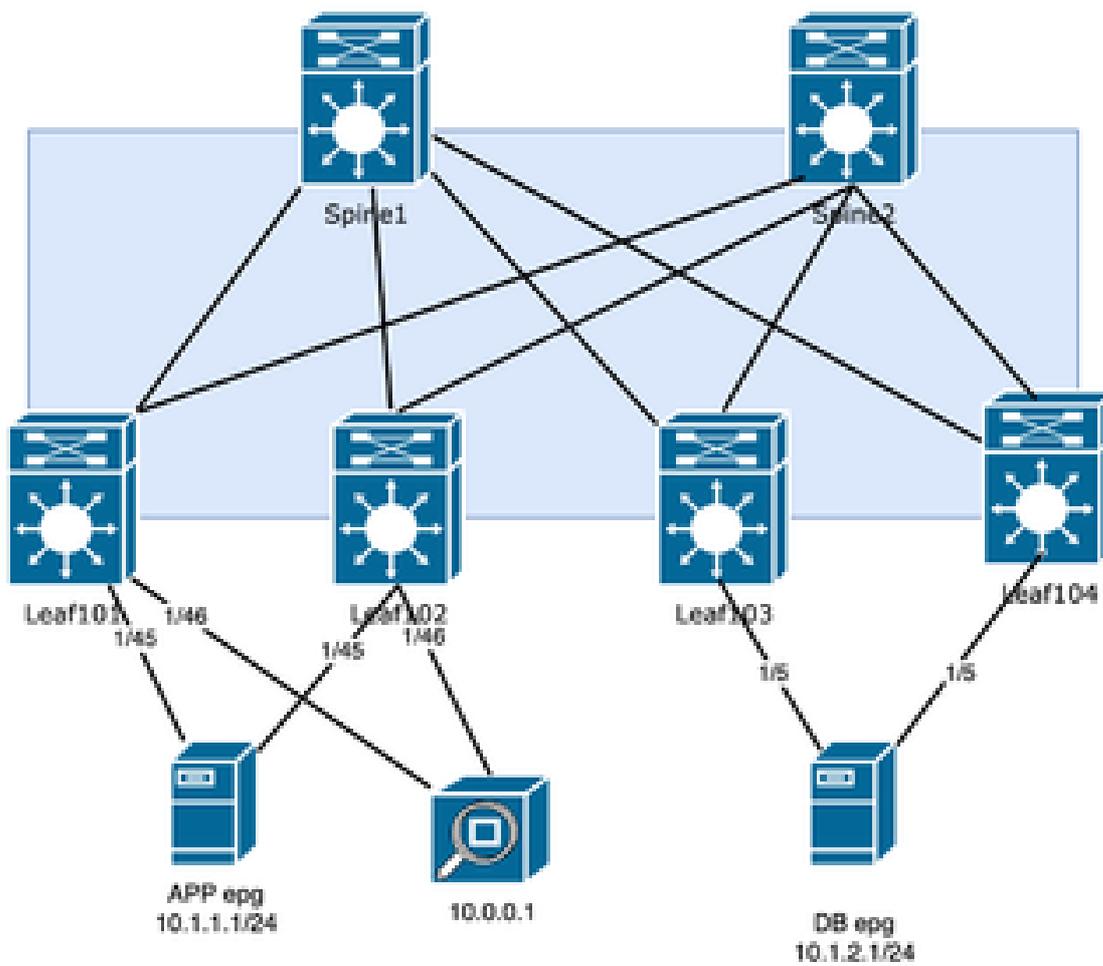
Comment créer des filtres dans Wireshark.

Ouvrez la capture. À l'aide d'une trame dans le paquet de commutateur distant encapsulé, sélectionnez la ligne SpanID et cliquez avec le bouton droit de la souris.

Sélectionnez Appliquer comme filtre > Sélectionné comme le montre l'image :



## Topologie



## Option 1. Configuration d'ERSPAN avec Flow-id

Si un serveur de destination est capable de gérer tout le trafic, l'en-tête ERSPAN inclut une option permettant de définir un ID de flux. Cet ID de flux peut être configuré pour identifier le trafic entrant vers le fabric, tandis qu'un autre ID de flux peut être configuré pour le trafic sortant.

### Étape 1. Configuration de la destination ESPAN

L'ID de flux d'un groupe de destinations sera égal à 1

Sous Fabric > Access Policies > Politiques > Troubleshooting > SPAN > SPAN Destination Groups

## Create SPAN Destination Group



Name: All-dst-jr-flowid

Description: optional

Destination Type: **EPG** Access Interface

Destination EPG: jr  ALL  monitor   
Tenant Application Profile EPG

SPAN Version: **Version 1** Version 2

Enforce SPAN Version:

Destination IP: 10.0.0.1

Source IP/Prefix: 10.255.0.0/16

Flow ID: 1

TTL: 64

MTU: 8000

DSCP: Unspecified

Cancel

Submit

Sur le deuxième groupe de destinations, configurez l'ID de flux 2 :

## Create SPAN Destination Group



Name:

Description:

Destination Type:  EPG  Access Interface

Destination EPG:

Tenant Application Profile EPG

SPAN Version:  Version 1  Version 2

Enforce SPAN Version:

Destination IP:

Source IP/Prefix:

Flow ID:

TTL:

MTU:

DSCP:

Cancel

Submit

Étape 2a. Créer une source étendue pour le trafic directement connecté à la SRC

Sous Fabric > Access Policies > Politiques > Troubleshooting > SPAN > SPAN Source Groups

## Create SPAN Source Group



Name:

Description:

Admin State:  Disabled  Enabled

Filter Group:

Destination Group:

### Create Sources



Name	Direction	Source EPG	Source Paths
------	-----------	------------	--------------

Filtrez davantage le trafic en ajoutant le chemin et l'EPG. L'exemple de TP est Tenant jr Application Profile ALL et EPG app.

## Create SPAN Source



Name:

Description:

Direction:  Both  Incoming  Outgoing

Filter Group:

Span Drop Packets:

Type:  None  EPG  Routed Outside

Source EPG:

Tenant                      Application Profile                      EPG

### Add Source Access Paths

Source Access Path		
Pod-1/Node-101/VPC-ESX-169		
Pod-1/Node-102/VPC-ESX-169		

Étape 2b. Création d'une source étendue pour le trafic directement connecté à l'heure d'été

Sous Fabric > Access Policies > Politiques > Troubleshooting > SPAN > SPAN Source Groups

## Create SPAN Source

Description: optional

Direction: **Both** Incoming Outgoing

Filter Group: select an option

Span Drop Packets:

Type: None **EPG** Routed Outside

Source EPG: jr Tenant ALL Application Profile db EPG

### Add Source Access Paths

Source Access Path
Pod-1/Node-103/eth1/6

Filtrez davantage le trafic en ajoutant non seulement le chemin, mais aussi la base de données EPG :

## Create SPAN Source Group

Name: Src-epg-2

Description: optional

Admin State: Disabled **Enabled**

Filter Group: select an option

Destination Group: All-dst-jr-flowid2

### Create Sources

Name	Direction	Source EPG	Source Paths
------	-----------	------------	--------------

## Étape 3. Analyse rapide de Wireshark

Dans cet exemple, vous vérifiez que le nombre de paquets de requête ICMP correspond au nombre de paquets de réponse ICMP, en vous assurant qu'il n'y a pas d'abandon de paquets dans le fabric ACI.

Ouvrez la capture sur wireshark pour créer le filtre à l'aide de l'ID de SPAN /ID de flux configuré avec l'IP SRC et DST :

```
<#root>  
(erspan.spanid ==  
  
and  
  
) && (ip.src==  
  
and ip.dst ==  
  
)
```

Filtre utilisé pour le flux testé en laboratoire :

```
<#root>  
(erspan.spanid == 1 and icmp) && (ip.src== 10.1.2.1 and ip.dst == 10.1.1.1)
```

Vérifiez que le paquet affiché est de la même quantité que celui envoyé :

(erspan.spanid == 1 and icmp) && (ip.src == 10.1.2.1 and ip.dst == 10.1.1.1)

No.	Time	Source	Destination	Protocol	Length	Info
33	12.789507	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
37	12.790332	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
41	12.791308	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
45	12.792088	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
49	12.792891	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
53	12.793663	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
57	12.794455	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
61	12.795259	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
65	12.796080	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request
69	12.796812	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) request

> Frame 33: 148 bytes on wire (1184 bits), 148 bytes captured (1184 bits)  
 > Ethernet II, Src: Cisco\_f8:19:ff (00:22:bd:f8:19:ff), Dst: VMware\_b7:4c:66 (00:50:56:b7:4c:66)  
 > Internet Protocol Version 4, Src: 10.255.0.102, Dst: 10.0.0.1  
 > Generic Routing Encapsulation (ERSPAN)  
 > Encapsulated Remote Switch Packet ANalysis Type II  
 0001 .... = Version: Type II (1)  
 .... 1010 0111 1110 = Vlan: 2686  
 000. .... = COS: 0  
 ...1 0... = Encap: Originally 802.1Q encapsulated (2)  
 .... 0... = Truncated: Not truncated (0)  
 .... ..00 0000 0001 = SpanID: 1  
 0000 0000 0000 = Reserved: 0

SpanID (erspan.spanid), 10 bits      Packets: 4109    Displayed: 1000 (24.3%)    Profile:

L'ID SPAN suivant doit avoir le même montant ; dans le cas contraire, le paquet a été abandonné dans le fabric.

Filter :

(erspan.spanid == 2 and icmp) && (ip.src == 10.1.2.1 and ip.dst == 10.1.1.1)

No.	Time	Source	Destination	Protocol	Length	Info
32	12.789387	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) requ
36	12.790321	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) requ
40	12.791299	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) requ
44	12.792076	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) requ
48	12.792880	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) requ
52	12.793654	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) requ
56	12.794434	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) requ
60	12.795250	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) requ
64	12.796038	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) requ
68	12.796797	10.1.2.1	10.1.1.1	ICMP	148	Echo (ping) requ

> Frame 32: 148 bytes on wire (1184 bits), 148 bytes captured (1184 bits)  
 > Ethernet II, Src: Cisco\_f8:19:ff (00:22:bd:f8:19:ff), Dst: VMware\_b7:4c:66 (00:50:56:b7:4c:66)  
 > Internet Protocol Version 4, Src: 10.255.0.103, Dst: 10.0.0.1  
 > Generic Routing Encapsulation (ERSPAN)  
 > Encapsulated Remote Switch Packet ANalysis Type II  
   0001 .... .... .... = Version: Type II (1)  
   .... 0001 0110 0111 = Vlan: 359  
   111. .... .... .... = COS: 7  
   ...1 0... .... .... = Encap: Originally 802.1Q encapsulated (2)  
   .... .0.. .... .... = Truncated: Not truncated (0)  
   .... ..00 0000 0010 = SpanID: 2  
   0000 0000 0000 = Reserved: 0

SpanID (erspan.spanid), 10 bits      Packets: 4109      Displayed: 1000 (24.3%)

## Option 2. Compteurs de plate-forme

Cette méthode tire parti du fait que Nexus suit les performances des interfaces individuelles avec différentes tailles de paquets, mais la méthode ne nécessite pas qu'au moins une file d'attente ait un faible volume de trafic, si ce n'est zéro.

### Effacer les compteurs de plateforme

Accédez au commutateur individuel et effacez l'interface individuelle qui se connecte aux périphériques.

```
<#root>
```

```
Switch#
```

```
vsh_lc -c "clear platform internal counters port
```

```
"
```

```
<#root>
```

```
LEAF3#
```

```
vsh_lc -c "clear platform internal counters port 6"
```

```
LEAF1#
```

```
vsh_lc -c "clear platform internal counters port 45"
```

```
LEAF2#
```

```
vsh_lc -c "clear platform internal counters port 45"
```

## Identification d'une taille de paquet avec des paquets faibles ou nuls

Trouvez une taille de paquet qui n'a peut-être aucun compteur dans toutes les Leafs pour RX et TX :

```
<#root>
```

```
vsh_lc -c 'show platform internal counters port
```

```
' | grep X_PKT
```

Dans l'exemple suivant, taille de paquet supérieure à 512 et inférieure à 1024 :

```
<#root>
```

```
LEAF101#
```

```
vsh_lc -c "show platform internal counters port 45 " | grep X_PKT
```

RX_PKTOK	1187
RX_PKTTOTAL	1187
RX_PKT_LT64	0
RX_PKT_64	0
RX_PKT_65	1179
RX_PKT_128	8
RX_PKT_256	0
<del>RX_PKT_512</del>	0 <<
RX_PKT_1024	0
RX_PKT_1519	0
RX_PKT_2048	0
RX_PKT_4096	7

RX_PKT_8192	43
RX_PKT_GT9216	0
TX_PKTOK	3865
TX_PKTTOTAL	3865
TX_PKT_LT64	0
TX_PKT_64	0
TX_PKT_65	3842
TX_PKT_128	17
TX_PKT_256	6
<b>TX_PKT_512</b>	<b>0 &lt;&lt;</b>
TX_PKT_1024	10
TX_PKT_1519	3
TX_PKT_2048	662
TX_PKT_4096	0
TX_PKT_8192	0
TX_PKT_GT9216	0

L'étape doit être effectuée sur la liaison vers laquelle les paquets sont transférés.

## Suivi du flux de trafic

À partir du serveur 10.1.2.1, 1 000 paquets sont envoyés avec une taille de paquet de 520.

Vérifiez sur l'interface Leaf 103 1/6, où le trafic est initié sur RX :

```
<#root>
```

```
MXS2-LF103#
```

```
vsh_lc -c "show platform internal counters port 6 " | grep X_PKT_512
```

RX_PKT_512	1000
TX_PKT_512	647

1 000 paquets RX, mais seulement 647 ont été envoyés en réponse.

L'étape suivante consiste à vérifier les interfaces sortantes des autres serveurs :

Pour Leaf102 :

```
<#root>
```

```
MXS2-LF102#
```

```
vsh_lc -c "show platform internal counters port 45 " | grep X_PKT_512
```

RX_PKT_512	0
TX_PKT_512	1000

Le fabric n'a pas abandonné la requête.

Pour le Leaf 101, les paquets RX 647 et 648 représentent la même quantité de paquets TX par l'ACI.

```
<#root>
```

```
MXS2-LF101#
```

```
vsh_lc -c "show platform internal counters port 45 " | grep X_PKT_512
```

```
      RX_PKT_512      647
      TX_PKT_512      0
```

## Informations connexes

[Dépannage du transfert intra-fabric ACI - Pertes intermittentes](#)

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.