# パケットカラーリングテクニックまたはプラットフォームカウンタを使用したパケットドロップのトラブルシューティング

## 内容



## はじめに

このドキュメントでは、パケットカラーリングテクニックを使用してネットワークフローを追跡する方法について説明します。

## 前提条件

### 要件

- ACIの基礎知識
- エンドポイントグループと契約
- Wiresharkの基礎知識

### 使用するコンポーネント

このドキュメントは、特定のハードウェアやソフトウェアのバージョンに限定されるものではありません。

使用デバイス：

- バージョン5.3(2)を実行しているCisco ACI
- Span宛先
- Gen2スイッチ

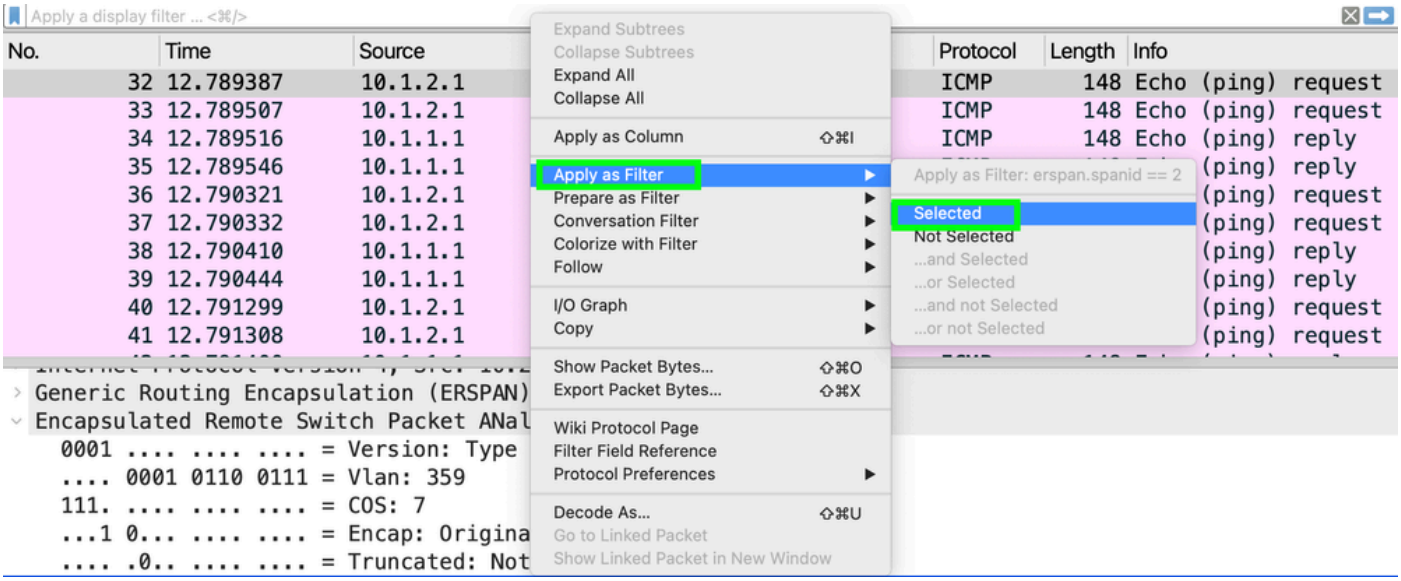
このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

## 背景説明

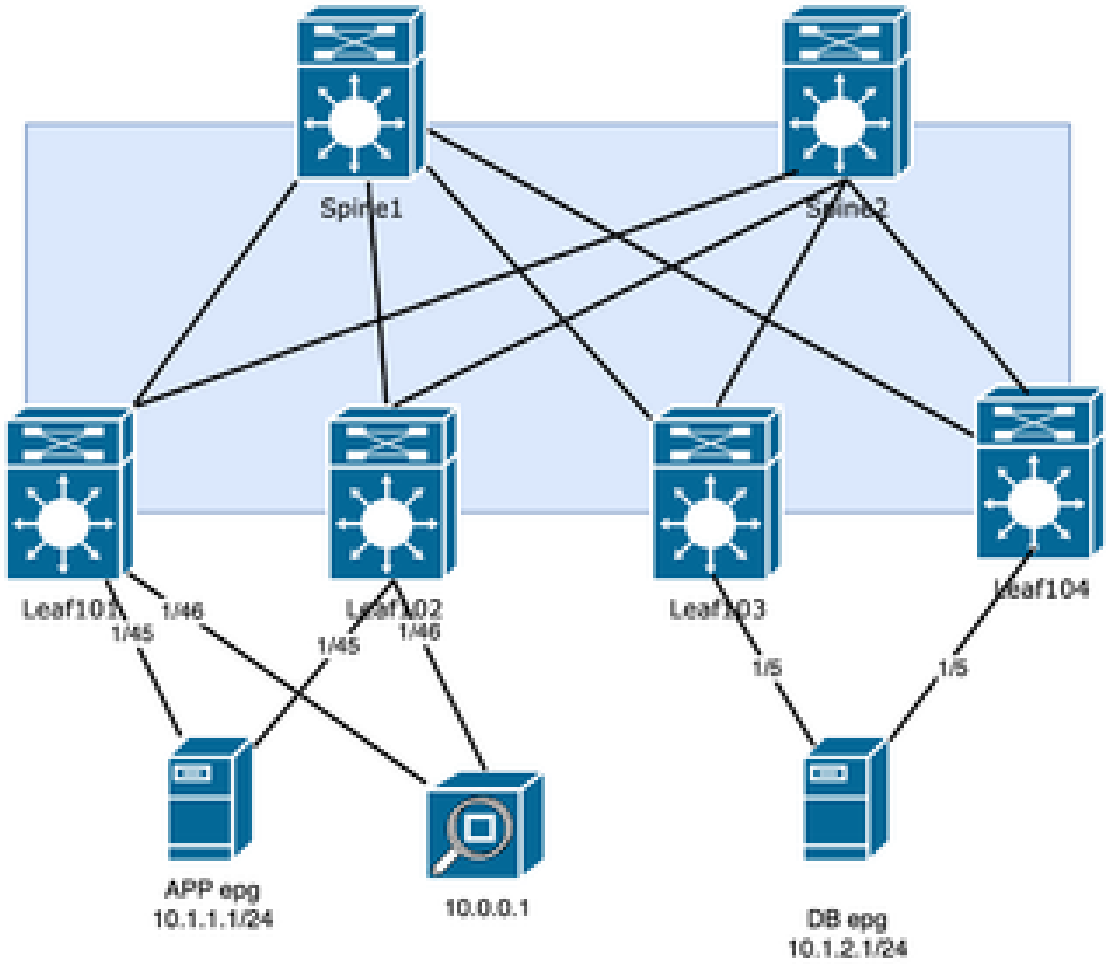
Wiresharkでのフィルタの作成方法

キャプチャを開きます。Encapsulated Remote Switch Packet(ERSP)内のフレームを使用して、SpanID行を選択し、右クリックします。

次の図に示すように、Apply as Filter > Selectedの順に選択します。




## トポロジ



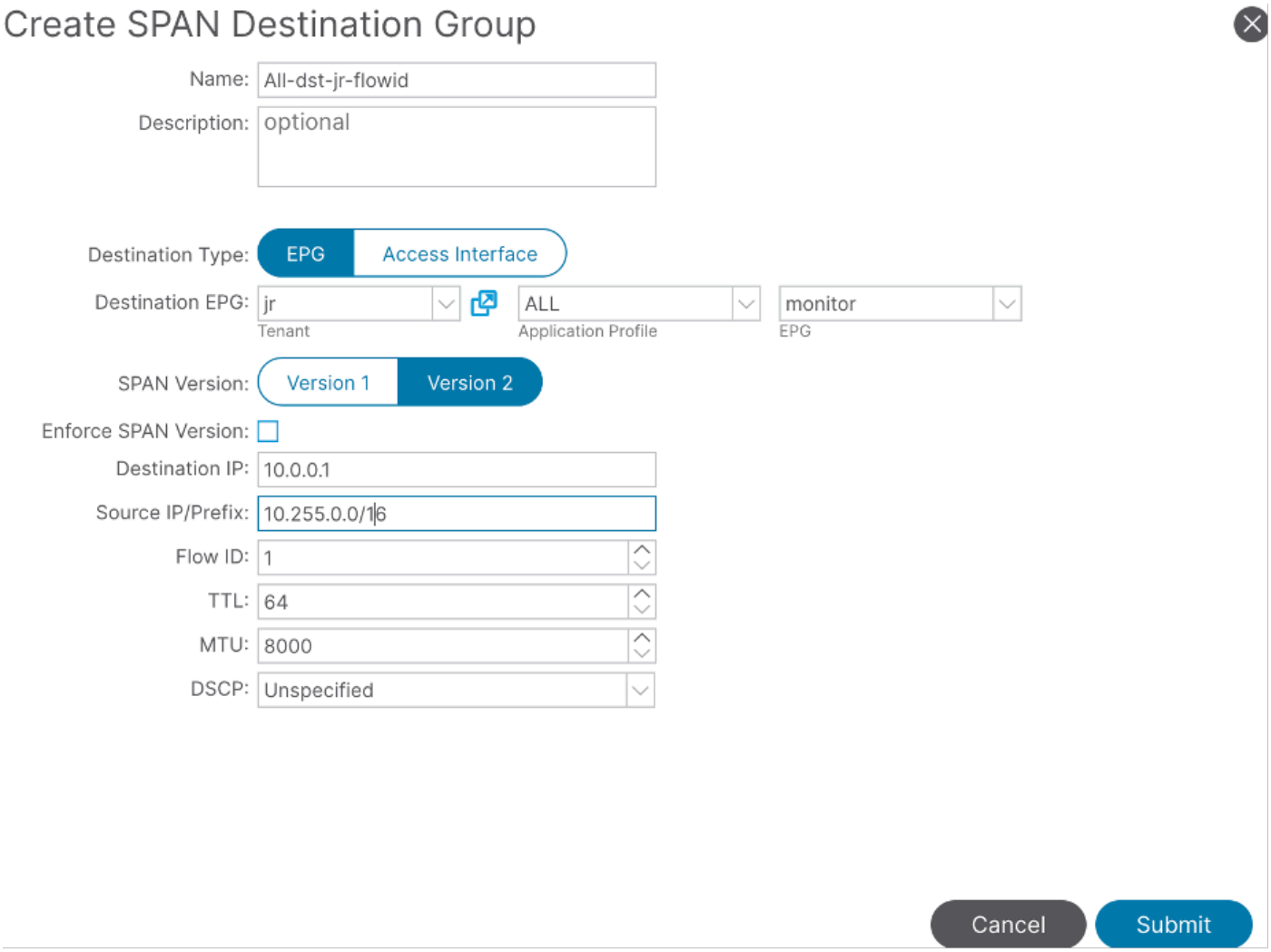

# オプション 1Flow-idを使用したERSPANの設定

宛先サーバがすべてのトラフィックを処理できる場合、ERSPANヘッダーにはフローIDを定義するオプションが含まれます。このフローIDは、ファブリックへの着信トラフィックを識別するように設定できますが、発信トラフィックには異なるフローIDを設定できます。

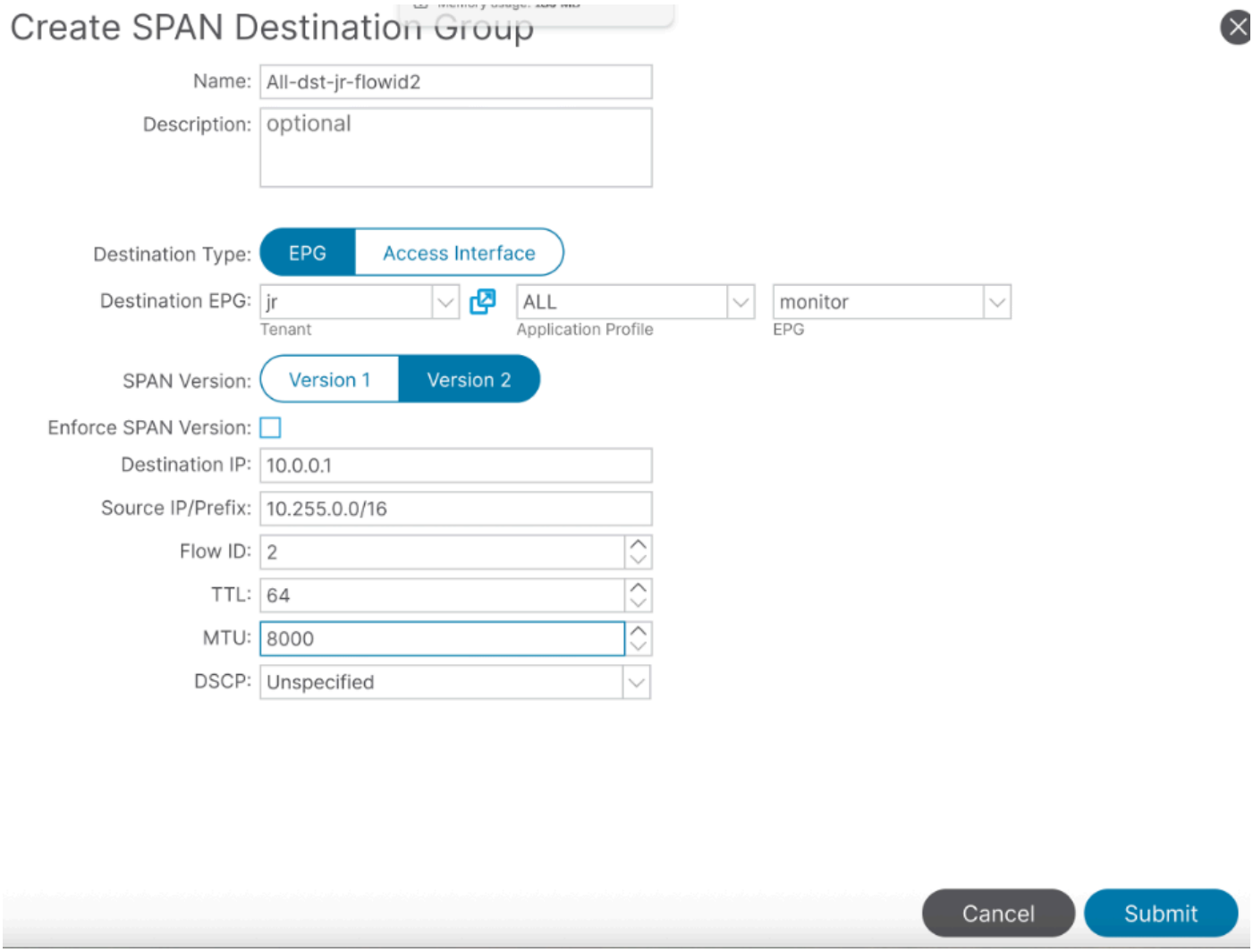
## ステップ 1：ESPAN接続先の設定

1つの宛先グループのフローIDは1になります

Fabric > Access Policies > Policies > Troubleshooting > SPAN > SPAN Destination Groupsの下で

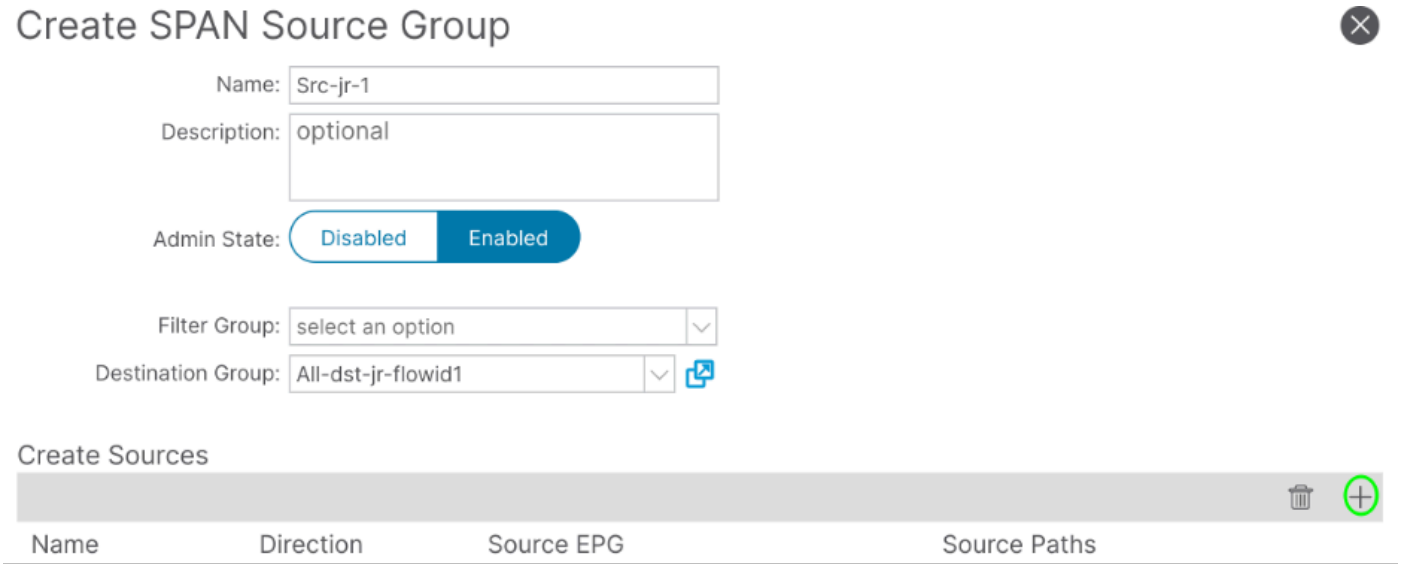



2番目の宛先グループで、flow-idを2に設定します。

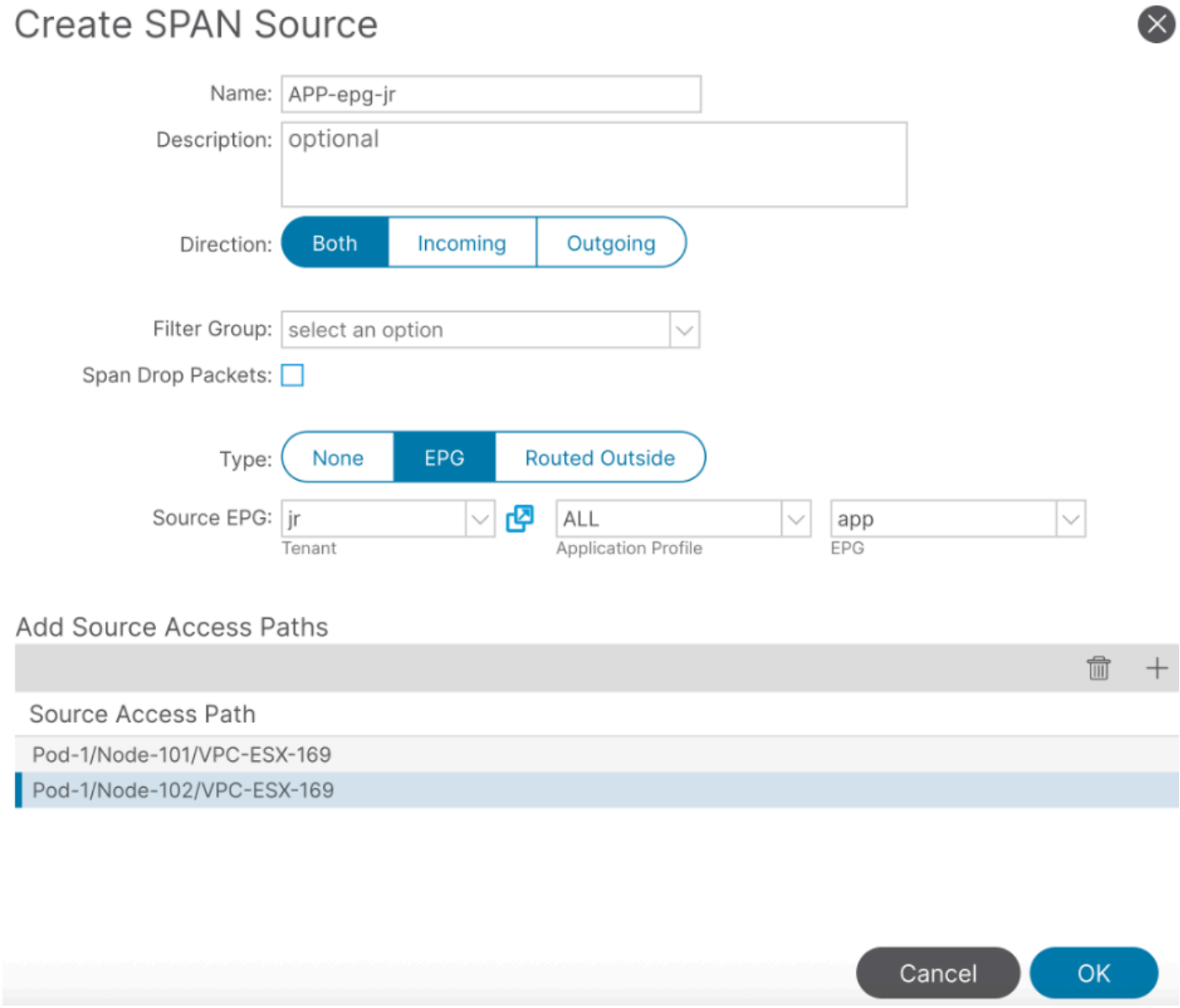



## ステップ 2aSRCに直接接続されたトラフィックのSPANソースの作成

Fabric > Access Policies > Policies > Troubleshooting > SPAN > SPAN Source Groupsの下で

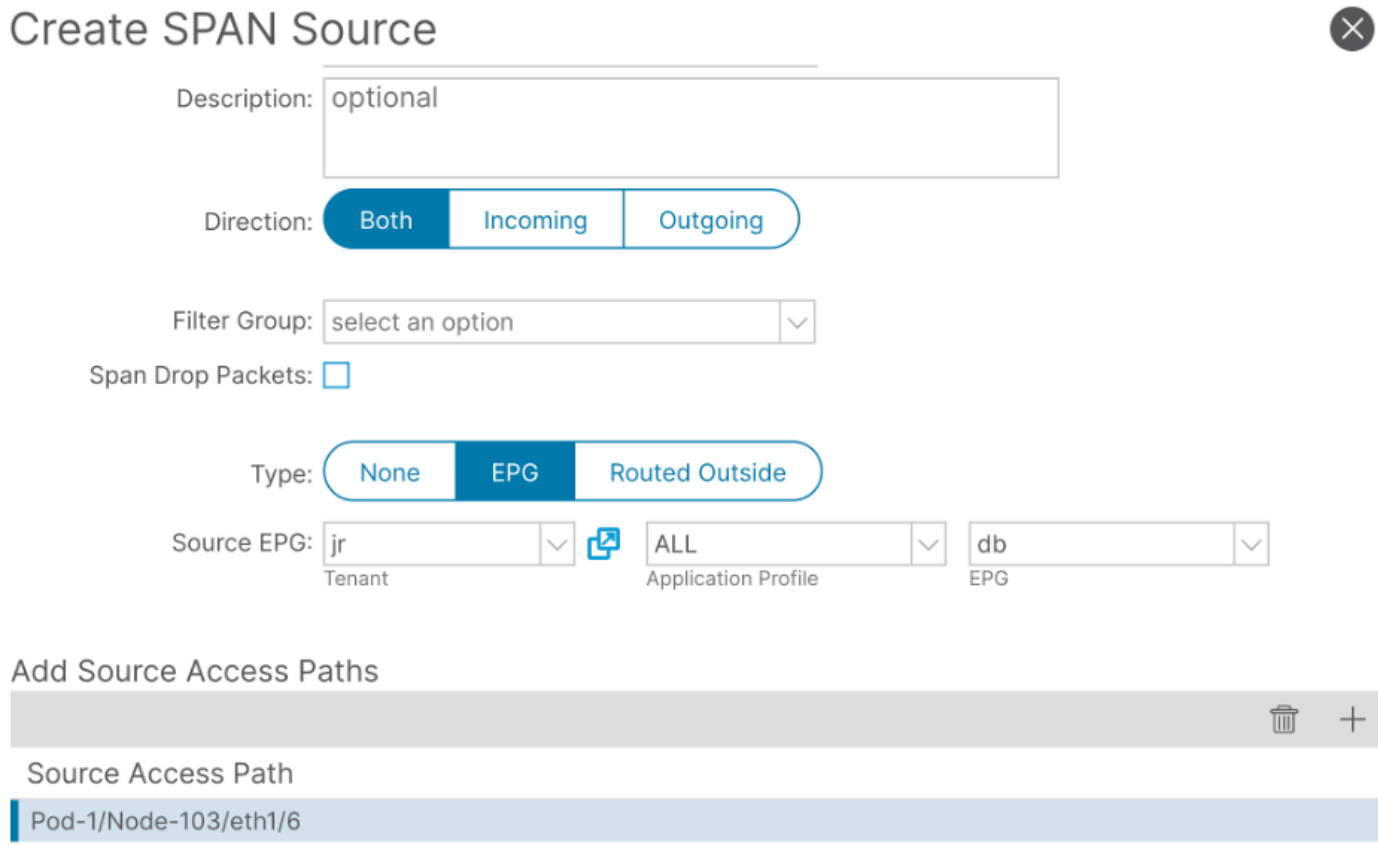



パスとEPGを追加して、トラフィックをさらにフィルタリングします。このラボの例は、Tenant jr Application Profile ALLおよびEPGアプリケーションです。

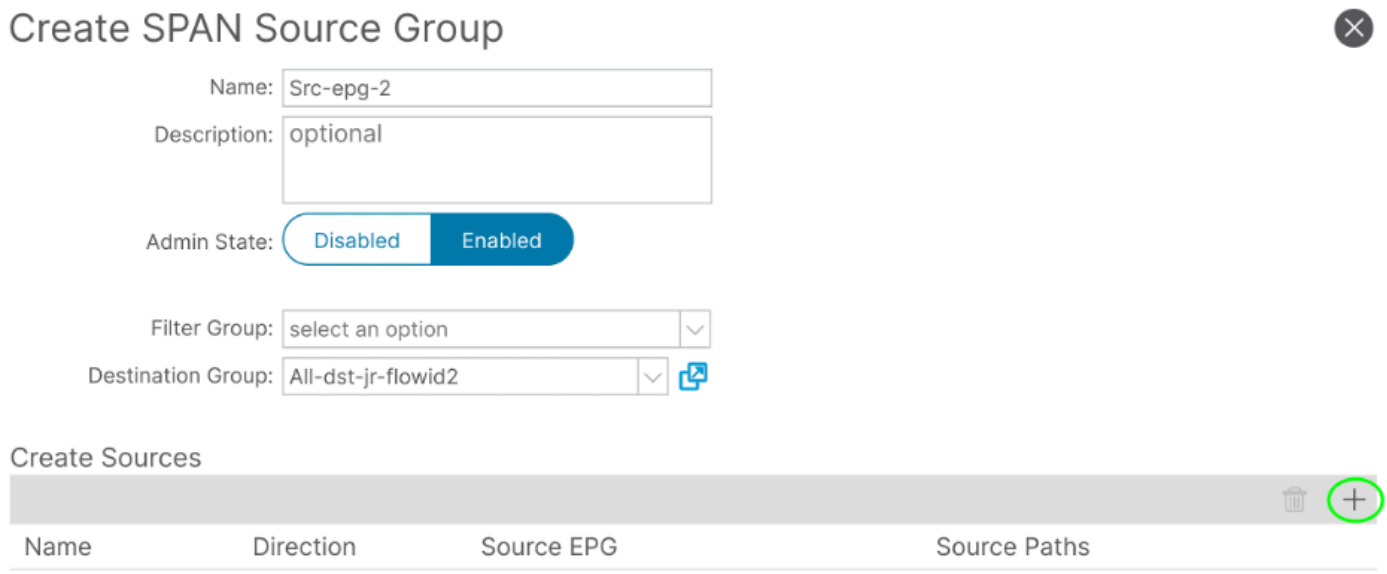



ステップ2b:DSTに直接接続されたトラフィックのSPANソースの作成

Fabric > Access Policies > Policies > Troubleshooting > SPAN > SPAN Source Groupsの下で




パスだけでなくEPG DBも追加して、トラフィックのフィルタリングを強化します。




## ステップ 3：迅速なWireshark分析

この例では、ICMP要求パケットの数がICMP応答パケットの数と一致することを確認し、ACIファブリック内でパケットドロップが発生していないことを確認します。

Wiresharkでキャプチャを開き、SRCおよびDST IPとともに設定されたSPAN ID /Flow-IDを使用してフィルタを作成します。

<#root>

**(erspan.spanid ==**

**and**

**) && (ip.src==**

**and ip.dst ==**

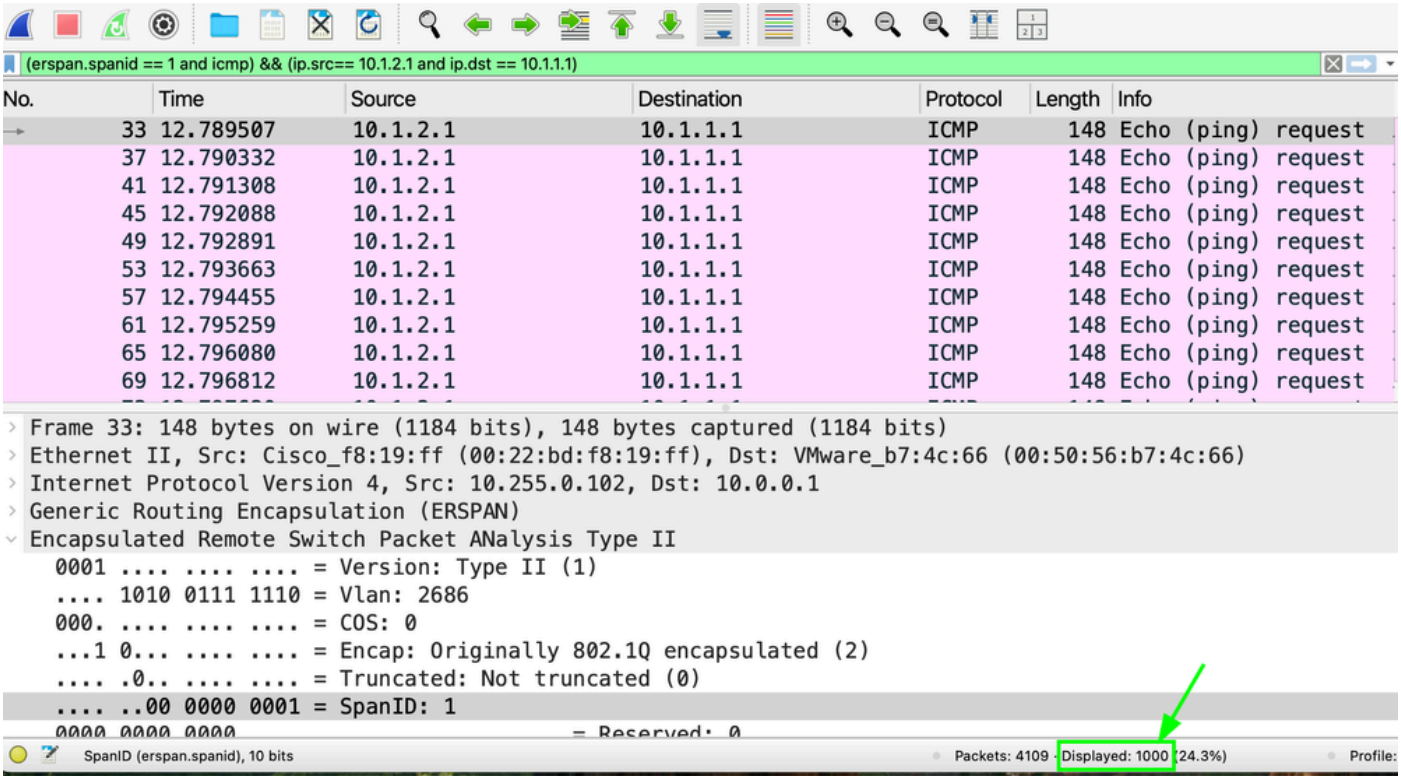
**)**

ラボでテストしたフローに使用するフィルタ：

<#root>

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

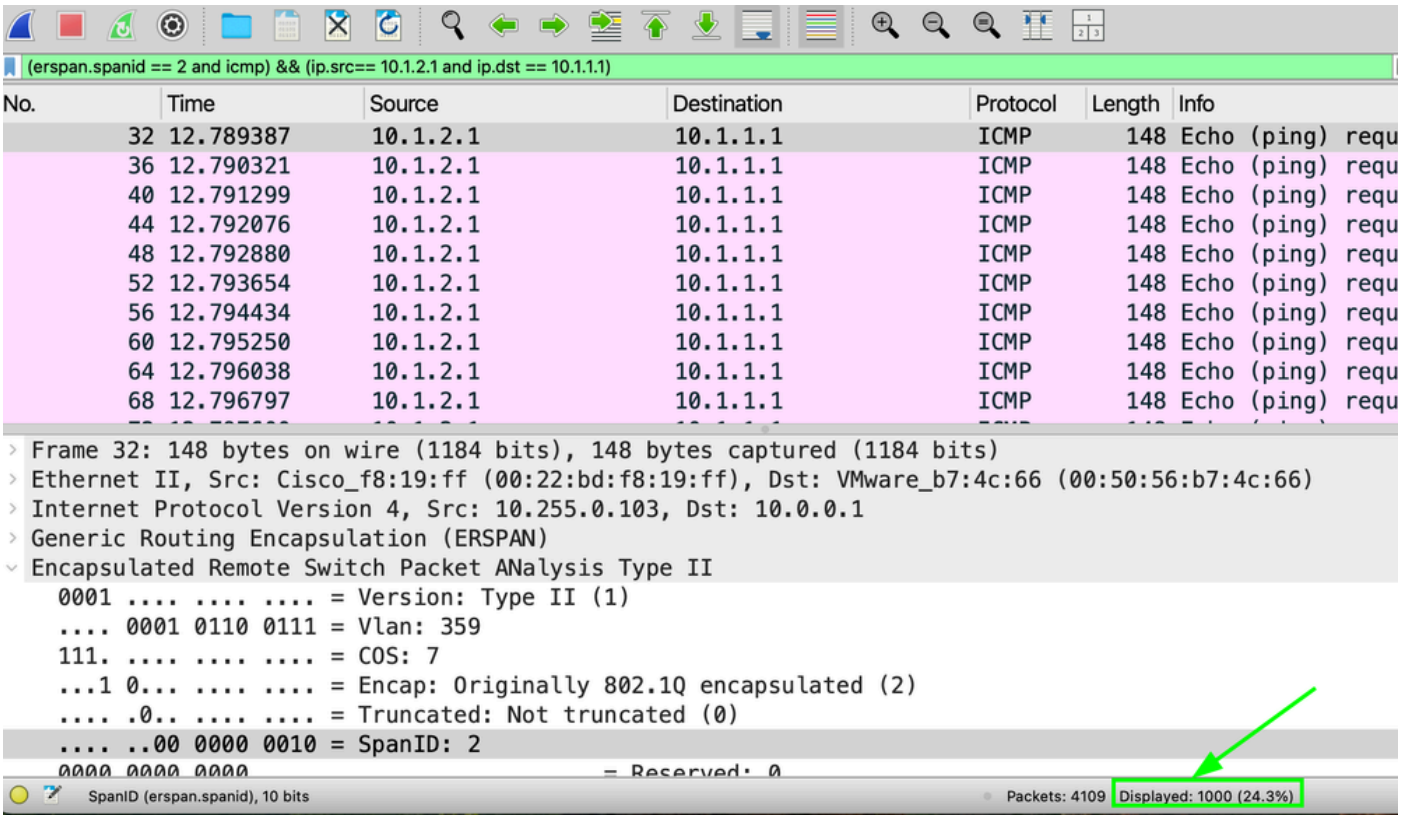
表示されたパケットが送信されたものと同じ量であることを確認します。




次のSPAN IDの値は同じでなければなりません。同じでない場合、パケットはファブリック内で廃棄されています。

[Filter]：

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




# オプション 2プラットフォームカウンタ

この方法は、Nexusがパケットサイズの異なる個々のインターフェイスのパフォーマンスを追跡していることを利用しますが、ゼロではないにしても、少なくともキューのトラフィック量が少なくないことが必要です。

## プラットフォームカウンタのクリア

個々のスイッチに移動し、デバイスに接続されている個々のインターフェイスをクリアします。

<#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”
```

## パケットが少ない、またはゼロのパケットサイズの特定

RXとTXの両方のすべてのリーフにカウンタがない可能性のあるパケットサイズを見つけます。

<#root>

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


       ’ | grep X_PKT
```

次の例では、パケットサイズが512より大きく1024より小さい場合です。

<#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
                RX_PKT_512           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

TX_PKT_512            0 <<


TX_PKT_1024          10
TX_PKT_1519          3
TX_PKT_2048          662
TX_PKT_4096          0
TX_PKT_8192          0
TX_PKT_GT9216        0
```

このステップは、パケットが転送されるリンクで実行する必要があります。

## トラフィックフローの追跡

サーバ10.1.2.1から、1000個のパケットが520のパケットサイズで送信されます。

トラフィックがRXで開始されるリーフ103インターフェイス1/6で確認します。

<#root>

```
MXS2-LF103#

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


                RX_PKT_512         1000
                TX_PKT_512          647
```

1000パケットのRXが送信されましたが、応答として送信されたのは647パケットだけです。

次に、他のサーバの発信インターフェイスを確認します。

Leaf102の場合：

<#root>

```
MXS2-LF102#

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


                RX_PKT_512            0
                TX_PKT_512         1000
```

ファブリックは要求をドロップしませんでした。

リーフ101の場合、RXパケットは647で、ACIによるTXと同じ量のパケットです。

<#root>

```
MXS2-LF101#

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


                RX_PKT_512         647
                TX_PKT_512           0
```

# 関連情報

ACI Intra-Fabric Forwardingのトラブルシューティング：断続的なドロップ

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。