

Configuration de XR Embedded Packet Tracer dans ASR 9000

Table des matières

[Introduction](#)

[Conditions préalables](#)

[Exigences](#)

[Composants Utilisés](#)

[Informations générales.](#)

[Restrictions et limitations de XR Embedded Packet Tracer](#)

[Workflow Packet Tracer](#)

[Configurer](#)

[Clear Packet Tracer - Compteurs Et Conditions](#)

[Démarrer/Arrêter le suivi des paquets](#)

[Conditions de Packet Tracer](#)

[Conditions de Packet Tracer - Interfaces](#)

[Conditions de Packet Tracer - Décalage/Valeur/Masque](#)

[Exemple de configuration :](#)

[Informations connexes](#)

Introduction

Ce document décrit le logiciel XR Embedded Packet Tracer. Il permet de suivre les flux de paquets personnalisés pour la validation et le dépannage des services.

Conditions préalables

Exigences

Le logiciel XR Embedded Packet Tracer est initialement disponible à partir de la version 7.1.2 de Cisco IOS® XR et est pris en charge par la gamme ASR 9000. D'autres gammes de produits XR devraient être prises en charge dans les futures mises à jour.

Composants utilisés

XR Embedded Packet Tracer est indépendant de protocoles spécifiques et est compatible avec tous les types de paquets de monodiffusion et de multidiffusion.

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.

L'infrastructure XR Embedded Packet Tracer a considérablement simplifié la validation du flux de services et le dépannage des problèmes de transfert de paquets.

Lorsque le suivi des paquets est activé sur une interface, le processeur réseau (NP) évalue les paquets entrants pour déterminer s'ils répondent aux critères définis. Si un paquet satisfait à la condition spécifiée, un identificateur est ajouté à son en-tête interne. Cet identificateur facilite le suivi du paquet à travers tous les composants impliqués dans le chemin de données et le chemin ponctuel au sein du routeur.

La condition fait référence à un ensemble de critères ou de règles qui définissent quels paquets peuvent être suivis lorsqu'ils traversent le routeur. Ces conditions aident le système à identifier et à surveiller des flux de paquets spécifiques pour le dépannage ou la validation de service.

Les conditions sont constituées des éléments suivants :

1. Interface(s) physique(s) :

- Spécifie les interfaces réseau sur lesquelles les paquets doivent arriver.
- Exemple : `packet-trace condition interface Gi0/0/0/1`

2. Triplets de décalage/valeur/masque :

- Définit les critères de correspondance des parties spécifiques d'un en-tête de paquet (ou charge utile).
- Ces triplets permettent à l'infrastructure d'être indépendante du protocole, car ils peuvent représenter n'importe quelle partie d'un en-tête de protocole.

Exemple :

- Décalage : Position à l'intérieur du paquet (décalage d'octets).
- Valeur: Valeur attendue à cette position.
- Masque : Bits à assortir pour la précision.

Restrictions et limitations de XR Embedded Packet Tracer

XR version 7.1.2 :

Le marquage des paquets est pris en charge sur les cartes de ligne Lightspeed Plus, Lightspeed et Tomahawk.

Le suivi des paquets est pris en charge dans le type de cartes de ligne mentionné précédemment.

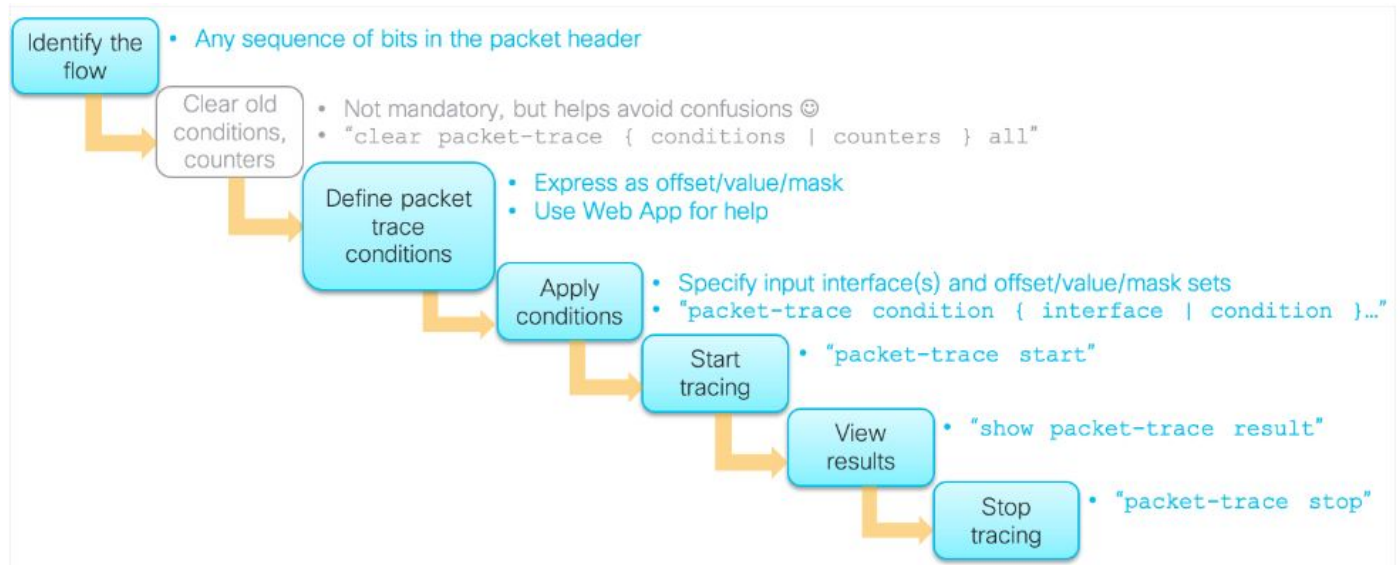
Vous pouvez spécifier un maximum de trois jeux de décalages/valeurs/masques de 4 octets.

XR version 7.5.2 :

Packet Tracer résout automatiquement les membres du lot au moment où la condition est définie vous pouvez maintenant suivre les paquets sur le chemin de point dans SPP, NetIO, UDP, TCP

Workflow Packet Tracer

Ce schéma illustre le fonctionnement du workflow Packet Tracer.



Configurer

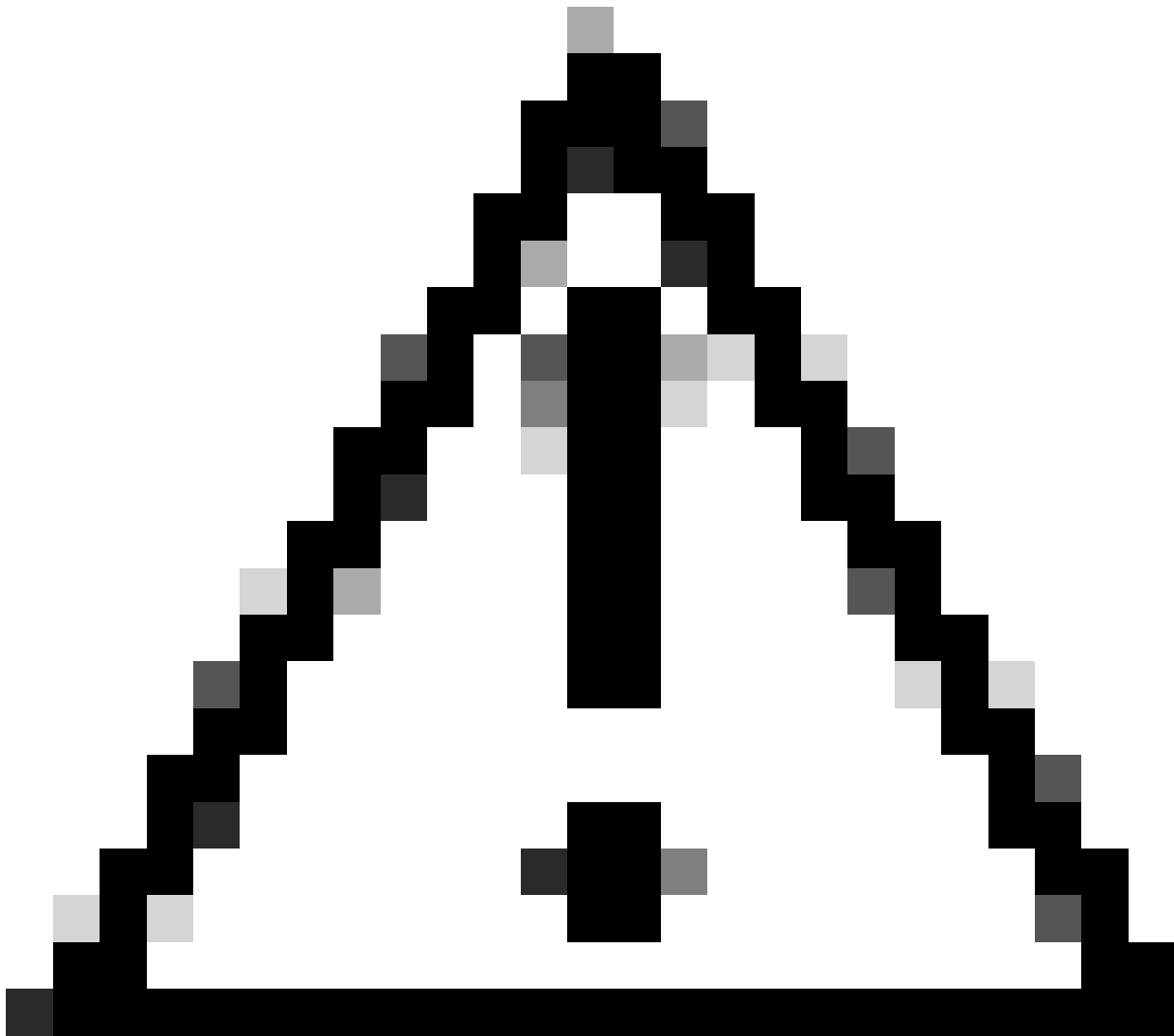
Clear Packet Tracer - Compteurs Et Conditions

Commande de réinitialisation des compteurs du traceur de paquets ; Les compteurs Packet Tracer peuvent être réinitialisés si nécessaire :

```
clear packet-trace counters all
```

Pour supprimer toutes les conditions du traceur de paquets, utilisez la commande suivante :

```
clear packet-trace conditions all
```



Mise en garde : Par conception, les conditions du traceur de paquets ne peuvent être effacées que lorsque le traçage de paquets est inactif.

Démarrer/Arrêter le suivi des paquets

Nécessité de spécifier manuellement le début et la fin de la trace du paquet :

```
RP/0/RP0/CPU0:Device# packet-trace start
RP/0/RP0/CPU0:Device# packet-trace stop
```

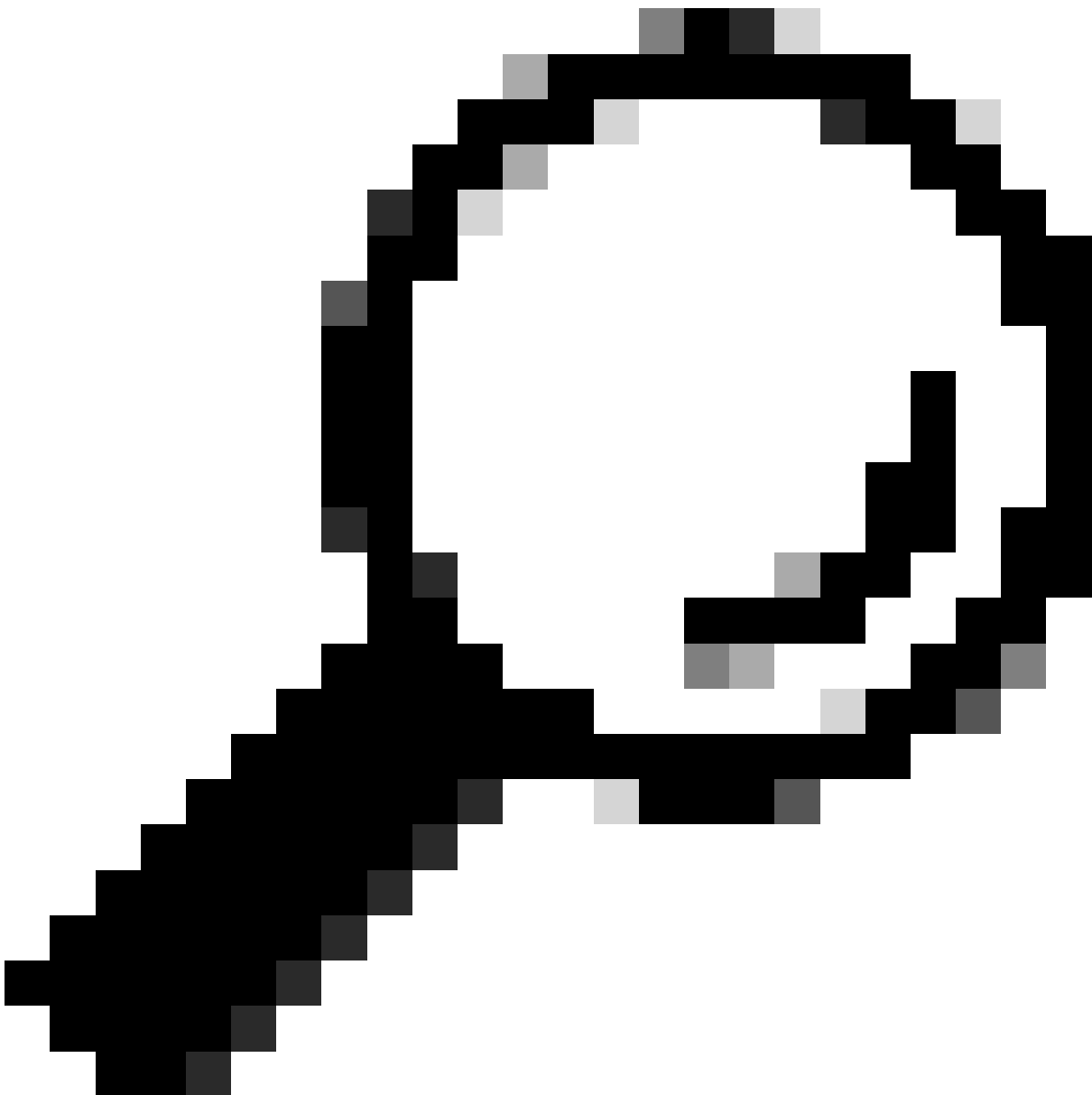
Conditions de Packet Tracer

Les conditions sont les suivantes :

1. Physical Interface(s) : indiquez la ou les interfaces physiques sur lesquelles les paquets doivent être reçus.
2. Triplets de décalage/valeur/masque. Aide à définir le flux d'intérêt.

Conditions de Packet Tracer - Interfaces

```
RP/0/RP0/CPU0:Device#packet-trace condition interface GigE0/0/0/0  
RP/0/RP0/CPU0:Device#packet-trace condition interface GigE0/0/0/1
```



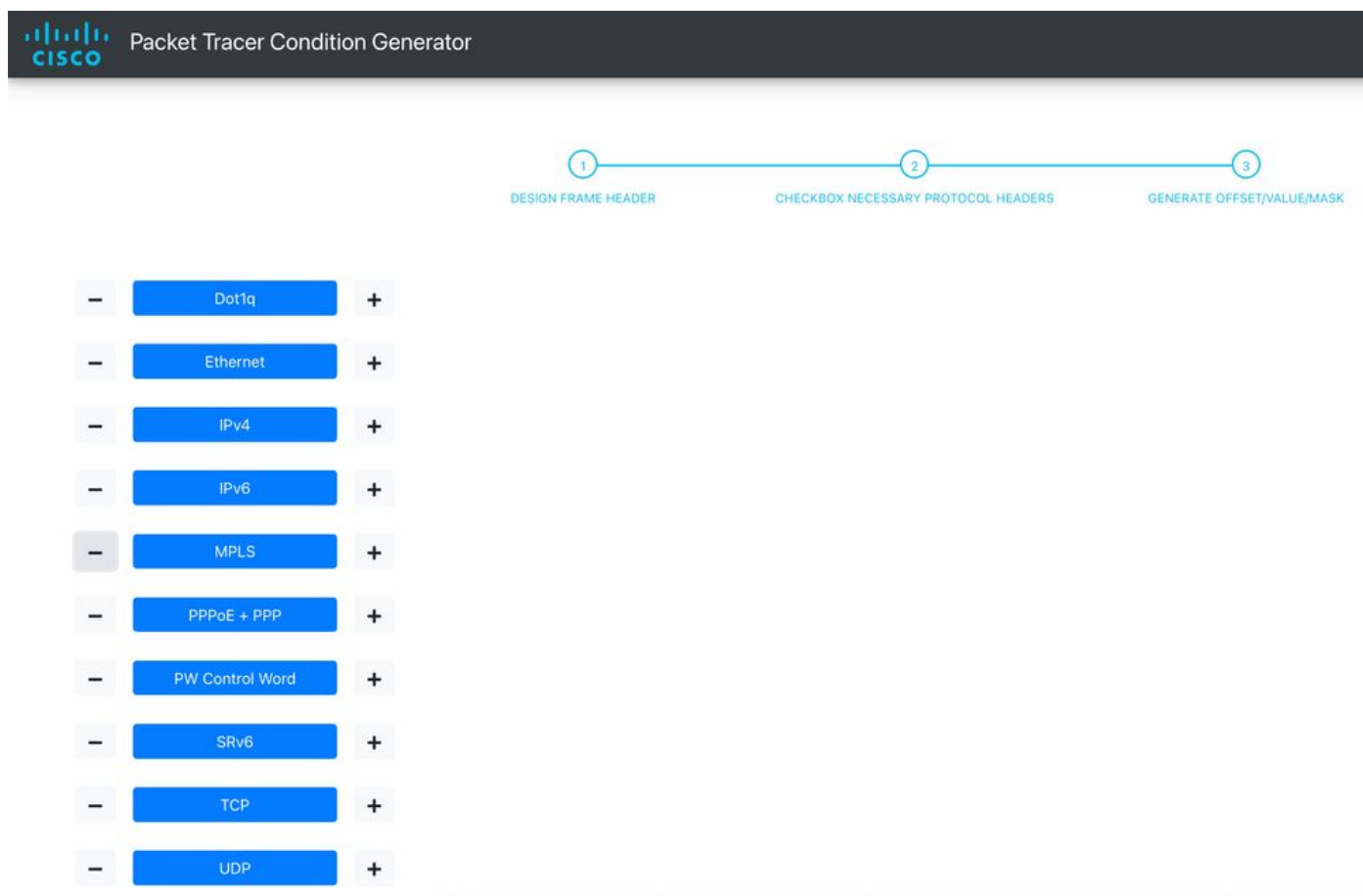
Conseil : Lors du suivi sur les sous-interfaces, la spécification Offset/Value/Mask doit prendre en compte l'encapsulation dot1q ou QinQ.

Conditions de Packet Tracer - Décalage/Valeur/Masque

L'application Web Générateur de conditions XR Packet Tracer fournit un outil permettant de créer des conditions Packet Tracer.

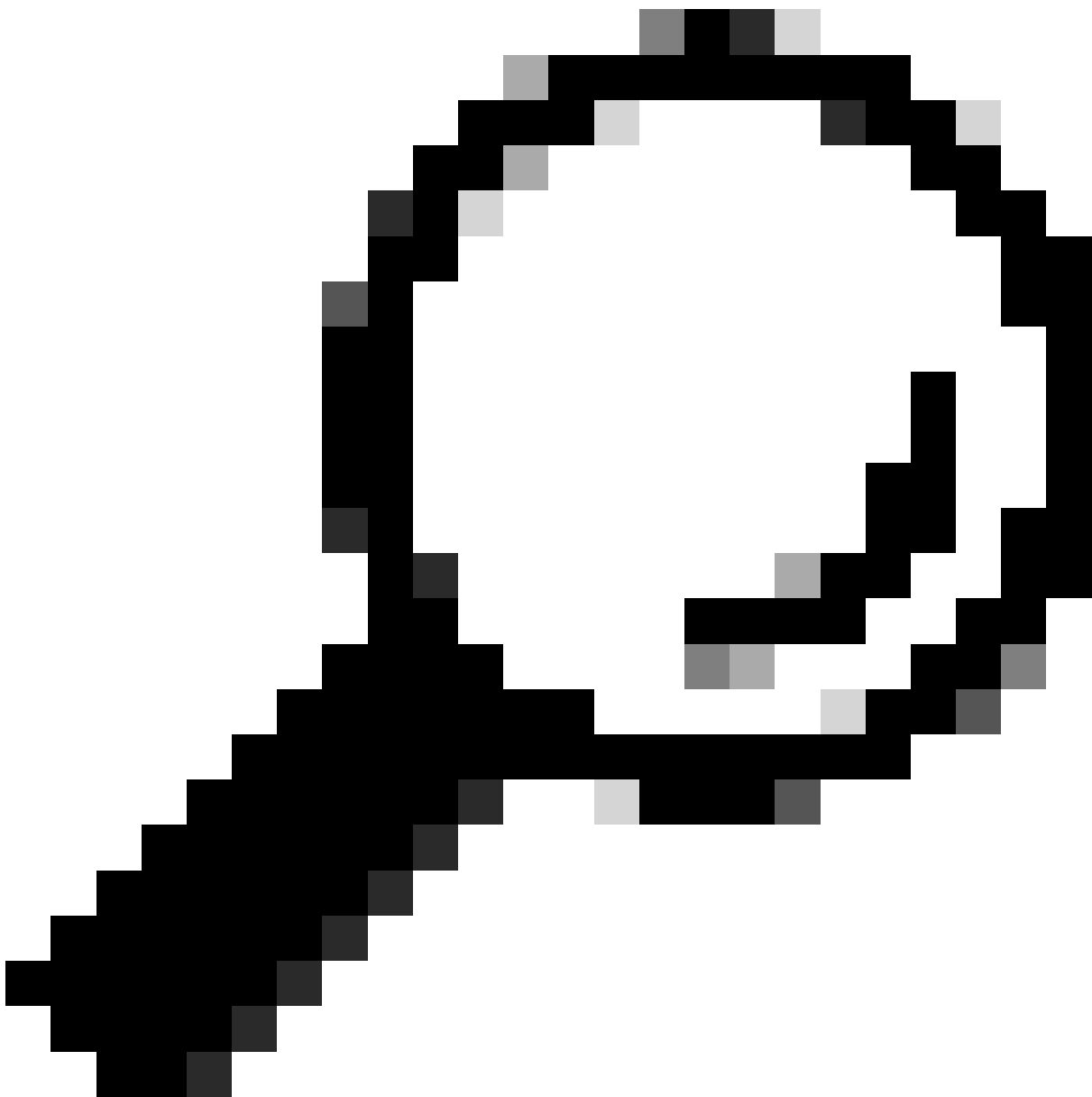
Son code source et ses instructions d'installation sont accessibles sur GitHub sous le nom : [XR Embedded Packet Tracer - Condition Generator](#) .

Cette application vous permet de construire visuellement la pile de protocoles pour le flux de paquets souhaité, de sélectionner les couches appropriées pour définir les conditions et d'entrer les valeurs (avec des masques facultatifs) qui décrivent le flux spécifique que vous souhaitez tracer.



La page de renvoi de l'application Web affiche la liste des en-têtes de protocole pris en charge pour la configuration.

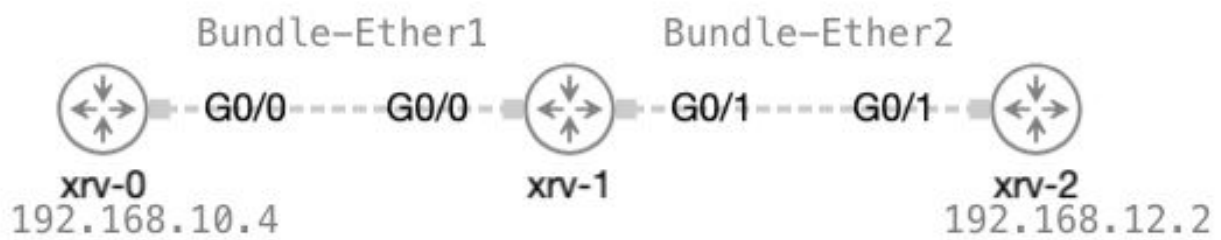
Assurez-vous d'inclure tous les en-têtes nécessaires avant celui avec lequel vous souhaitez faire correspondre le trafic, car le calcul du décalage dépend de l'ordre des en-têtes.



Conseil : N'oubliez pas d'inclure l'en-tête du mot de contrôle PW s'il est utilisé.

Exemple de configuration :

Voici un exemple de topologie. Notre objectif est de vérifier si les paquets sont correctement reçus et transmis via le périphérique XRV1 :



1.- Configurez la condition packet-trace pour les interfaces spécifiques que vous souhaitez surveiller.

```
RP/0/RP0/CPU0:xrv-1#packet-trace condition interface Bundle-Ether1
RP/0/RP0/CPU0:xrv-1#packet-trace condition interface Bundle-Ether2
```

2.- Générer le décalage/valeur/masque, Cochez les cases en regard des en-têtes que vous souhaitez faire correspondre. Vous pouvez choisir plusieurs en-têtes si nécessaire. Pour chaque en-tête sélectionné, un cadre correspondant est affiché sur le côté droit. Entrez la valeur et le masque souhaités dans le cadre, puis cliquez sur le bouton Submit pour finaliser votre configuration.

3.- Une fois que le décalage/valeur/masque a été copié dans le Presse-papiers, utilisez-le pour définir les conditions :

```
RP/0/RP0/CPU0:xrv-1#packet-trace condition 1 Offset 30 Value 0xc0a80a Mask 0xffffffff
RP/0/RP0/CPU0:xrv-1#packet-trace condition 5 Offset 34 Value 0xc0a80c Mask 0xffffffff
```

4.- Vérifiez l'état de suivi des paquets :

```
RP/0/RP0/CPU0:xrv-1#show packet-trace status
```

```
-----  
Packet Trace Master Process:
```

```
  Buffered Conditions:
```

```
    Interface Bundle-Ether1
```

```
      Member GigE0/0/0/0
```

```
    Interface Bundle-Ether2
```

```
      Member GigE0/0/0/1
```

```
      1 Offset 30 Value 0xc0a80a Mask 0xffffffff
```

```
      5 Offset 34 Value 0xc0a80c Mask 0xffffffff
```

```
  Status: Inactive
```

```
RP/0/RP0/CPU0:xrv-1#
```

```
RP/0/RP0/CPU0:xrv-1#show packet-trace status detail
```

```
-----  
Location: 0/0/CPU0
```

```
Available Counting Modules: 4
```

```
  #1 SPP
```

```
    Last errors:
```

```
  #2 npu_server_lsp
```

```
    Last errors:
```

```
  #3 NETIO
```

```
    Last errors:
```

```
  #4 UDP
```

```
    Last errors:
```

```
Available Marking Modules: 1
```

```
  #1 npu_server_lsp
```

```
    Interfaces: 0
```

```
    Conditions: 0
```

```
    Last errors:
```

```
-----  
Packet Trace Master Process:
```

```
  Buffered Conditions:
```

```
    Interface Bundle-Ether1
```

```
      Member GigE0/0/0/0
```

```
    Interface Bundle-Ether2
```

```
      Member GigE0/0/0/1
```

```
      1 Offset 30 Value 0xc0a80a Mask 0xffffffff
```

```
      5 Offset 34 Value 0xc0a80c Mask 0xffffffff
```

```
  Status: Inactive
```

```
-----  
Location: 0/RP0/CPU0
```

```
Available Counting Modules: 3
```

```
  #1 SPP
```

```
    Last errors:
```

```
  #2 NETIO
```

```
    Last errors:
```

```
  #3 UDP
```

```
    Last errors:
```

```
Available Marking Modules: 0
```

```
RP/0/RP0/CPU0:xrv-1#
```

5.- Démarrez Packet Tracer :

```
RP/0/RP0/CPU0:xrv-1# packet-trace start
RP/0/RP0/CPU0:xrv-1#
RP/0/RP0/CPU0:xrv-1# show packet-trace status
```

```
-----
Packet Trace Master Process:
  Buffered Conditions:
    Interface Bundle-Ether1
      Member GigE0/0/0/0
    Interface Bundle-Ether2
      Member GigE0/0/0/1
      1 Offset 30 Value 0xc0a80a Mask 0xffffffff
      5 Offset 34 Value 0xc0a80c Mask 0xffffffff
  Status: Active
RP/0/RP0/CPU0:xrv-1#
```

6. - Attendons quelques minutes afin de capturer le trafic :

7.- Vérifiez les résultats :

```
RP/0/RP0/CPU0:xrv-1#show packet-trace result
T: D - Drop counter; P - Pass counter
Location      | Source      | Counter      | T | Last-Attribute
-----
0/0/CPU0      | NP0         | PACKET_MARKED | P | GigE0_0_0_0
0/0/CPU0      | NP0         | PACKET_TO_FABRIC | P |
0/0/CPU0      | NP0         | PACKET_TO_PUNT | P |
0/0/CPU0      | NP0         | PACKET_FROM_FABRIC | P |
0/0/CPU0      | NP0         | PACKET_TO_INTERFACE | P | GigE0_0_0_1
RP/0/RP0/CPU0:xrv-1#
```

8.- Vous pouvez vérifier tous les compteurs enregistrés avec le framework packet tracer avec leurs descriptions, en utilisant la commande show packet-trace description :

```
RP/0/RP0/CPU0:xrv-1#show packet-trace descriptions
NP0          PACKET_MARKED          M  Marked from ingress interface
NP0          PACKET_FROM_INJECT      P  Injected from linecard CPU
NP0          PACKET_FROM_FAB_INJECT  P  Injected from fabric
NP0          PACKET_ING_DROP         D  Dropped on ingress
NP0          PACKET_TO_FABRIC        P  Sent to router fabric
NP0          PACKET_TO_PUNT          P  Punted to linecard for CPU handling
NP0          PACKET_FROM_FABRIC      P  From router fabric
NP0          PACKET_EGR_DROP         D  Dropped on egress
NP0          PACKET_TO_INTERFACE     P  Packet sent to network interface
```

RP/0/RP0/CPU0:xrv-1#

9.- Arrêtez le suivi des paquets :

RP/0/RP0/CPU0:xrv-1#packet-trace stop

Informations connexes

[Packet Tracer intégré XR](#)

[Assistance technique de Cisco et téléchargements](#)

[Comprendre les types de cartes de ligne de la gamme ASR 9000](#)

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.