

在ASR 9000中配置XR嵌入式Packet Tracer

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[XR嵌入式Packet Tracer的限制和限制](#)

[Packet Tracer工作流程](#)

[配置](#)

[清除Packet Tracer计数器 and 条件](#)

[开始/停止数据包跟踪](#)

[Packet Tracer条件](#)

[Packet Tracer条件 — 接口](#)

[Packet Tracer条件 — 偏移/值/掩码](#)

[配置示例：](#)

[相关信息](#)

简介

本文档介绍XR嵌入式Packet Tracer。它有助于跟踪自定义数据包流，以进行服务验证和故障排除。

先决条件

要求

XR嵌入式Packet Tracer最初从Cisco IOS® XR版本7.1.2开始提供，ASR 9000系列支持该软件。计划在未来更新中获得其他XR产品系列的支持。

使用的组件

XR嵌入式Packet Tracer独立于特定协议，与所有类型的单播和组播数据包兼容。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

背景信息

XR嵌入式Packet Tracer框架大大简化了服务流验证和数据包转发问题的故障排除。

当在接口上激活数据包跟踪时，网络处理器(NP)会评估传入数据包，以确定它们是否满足定义的标准。如果数据包满足指定的条件，则在其内部报头中添加标识符。此标识符有助于在路由器内的数据路径和点进路径涉及的所有组件之间跟踪数据包。

条件是指定义数据包通过路由器时可以跟踪的一组条件或规则。这些情况有助于系统识别和监控特定数据包流，以便进行故障排除或服务验证。

这些条件包括以下组件：

1. 物理接口：

- 指定数据包预期到达的网络接口。
- 示例： `packet-trace condition interface Gi0/0/0/1`

2. 偏移/值/掩码三重：

- 定义匹配数据包报头（或负载）的特定部分的标准。
- 这些三重式允许框架与协议无关，因为它们可以代表任何协议报头的任何部分。

示例：

- 偏移：在数据包中的位置（字节偏移）。
- 值：该位置的预期值。
- 掩码：精确度要匹配的位。

XR嵌入式Packet Tracer的限制和限制

XR版本7.1.2:

Lightspeed Plus、Lightspeed和Tomahawk线卡支持数据包标记。

前面提到的线卡类型支持数据包跟踪。

最多可以指定三个4个八位组的偏移/值/掩码集。

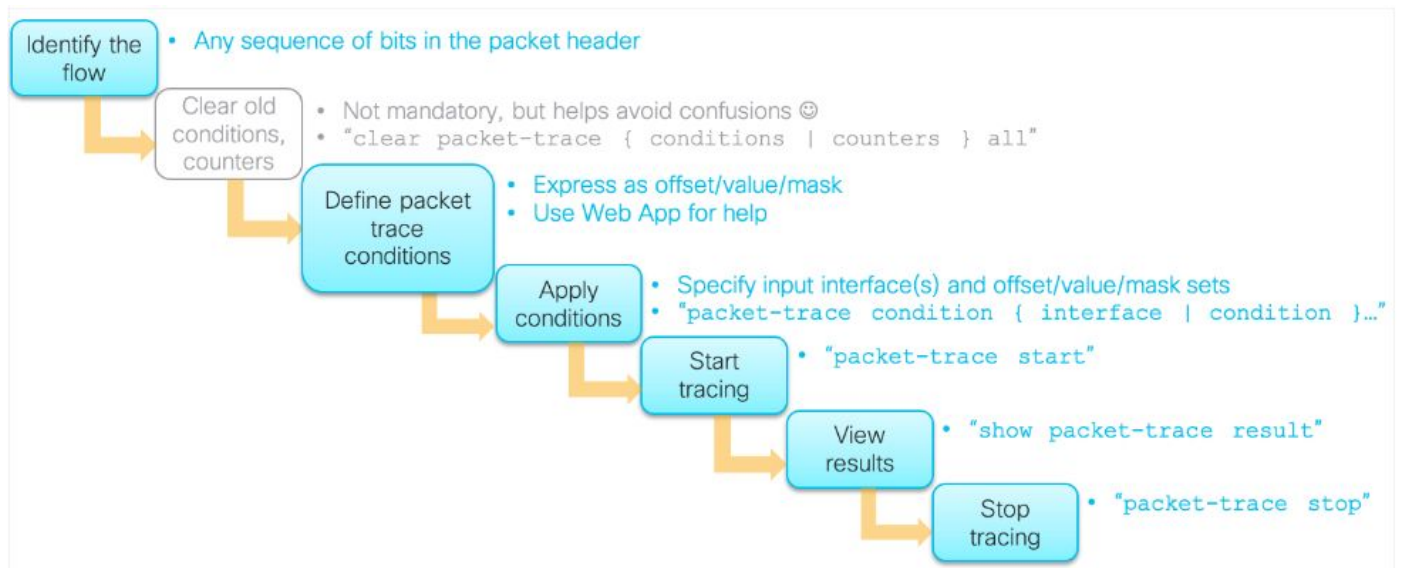
XR版本7.5.2:

Packet tracer会在设置条件时自动解析捆绑成员

您现在可以在SPP、NetIO、UDP、TCP中跟踪分支路径上的数据包

Packet Tracer工作流程

下图说明了Packet Tracer工作流的运行方式。



配置

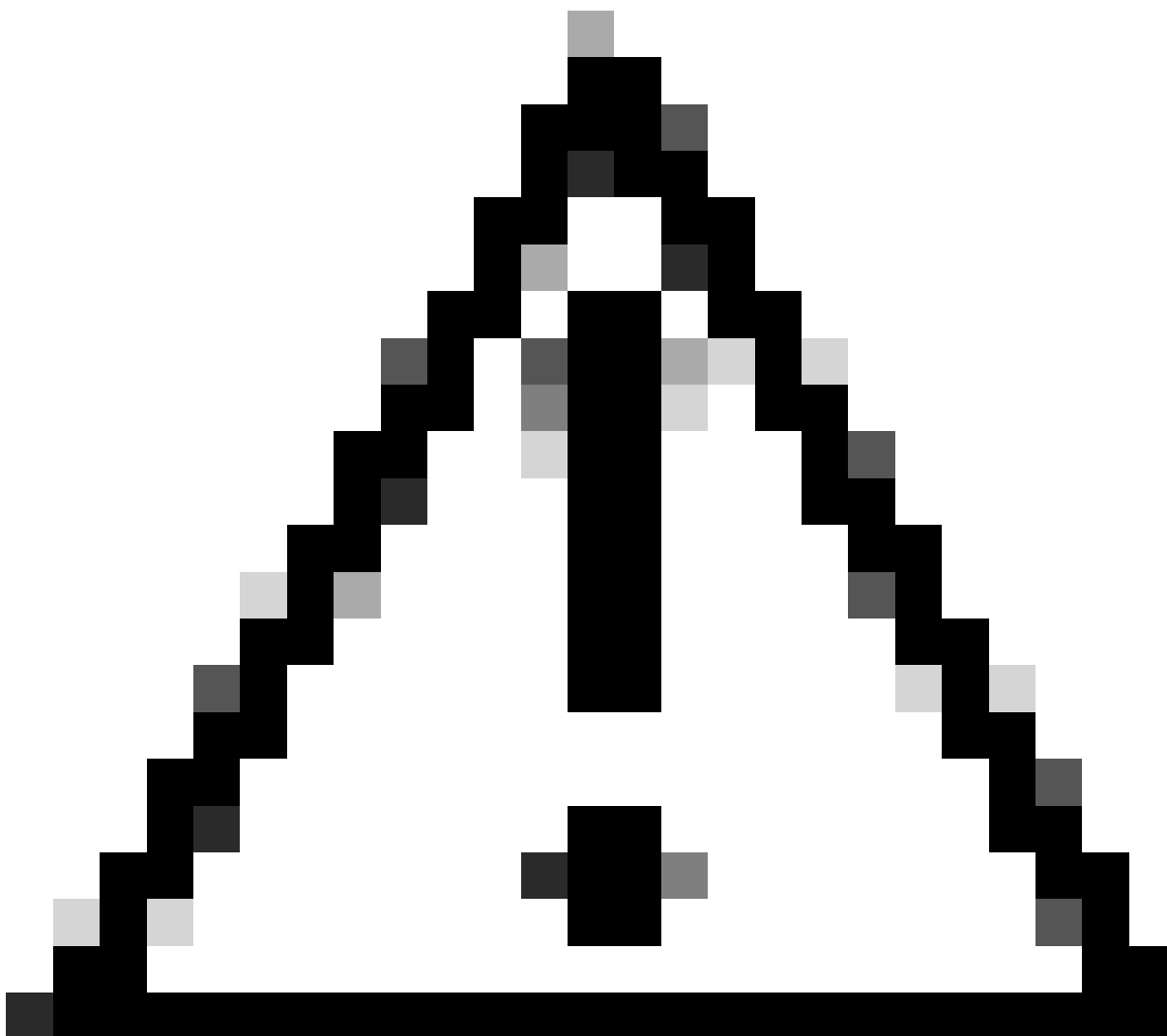
清除Packet Tracer计数器 and 条件

命令重置Packet Tracer计数器；Packet Tracer计数器可在需要时重置：

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

命令可删除所有Packet Tracer条件，请使用以下命令：

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



警告：根据设计，只有在数据包跟踪处于非活动状态时，才能清除Packet Tracer条件。

开始/停止数据包跟踪

需要手动指定数据包跟踪的开头和结尾：

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

Packet Tracer条件

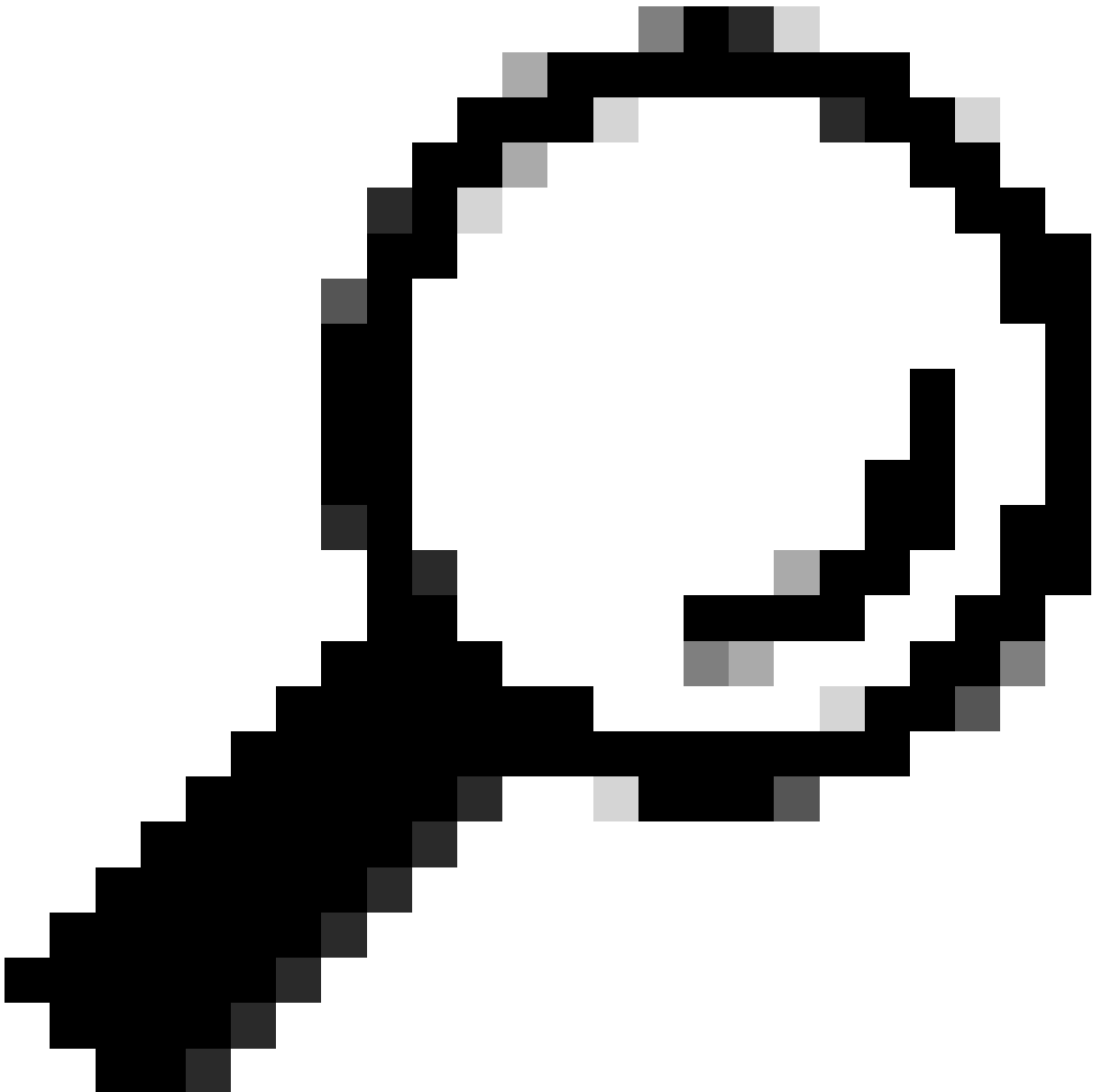
条件包括：

1. Physical Interface(s):指示预期接收数据包的物理接口。

2. 偏移量/值/掩码三重。帮助定义兴趣流。

Packet Tracer条件 — 接口

```
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
```



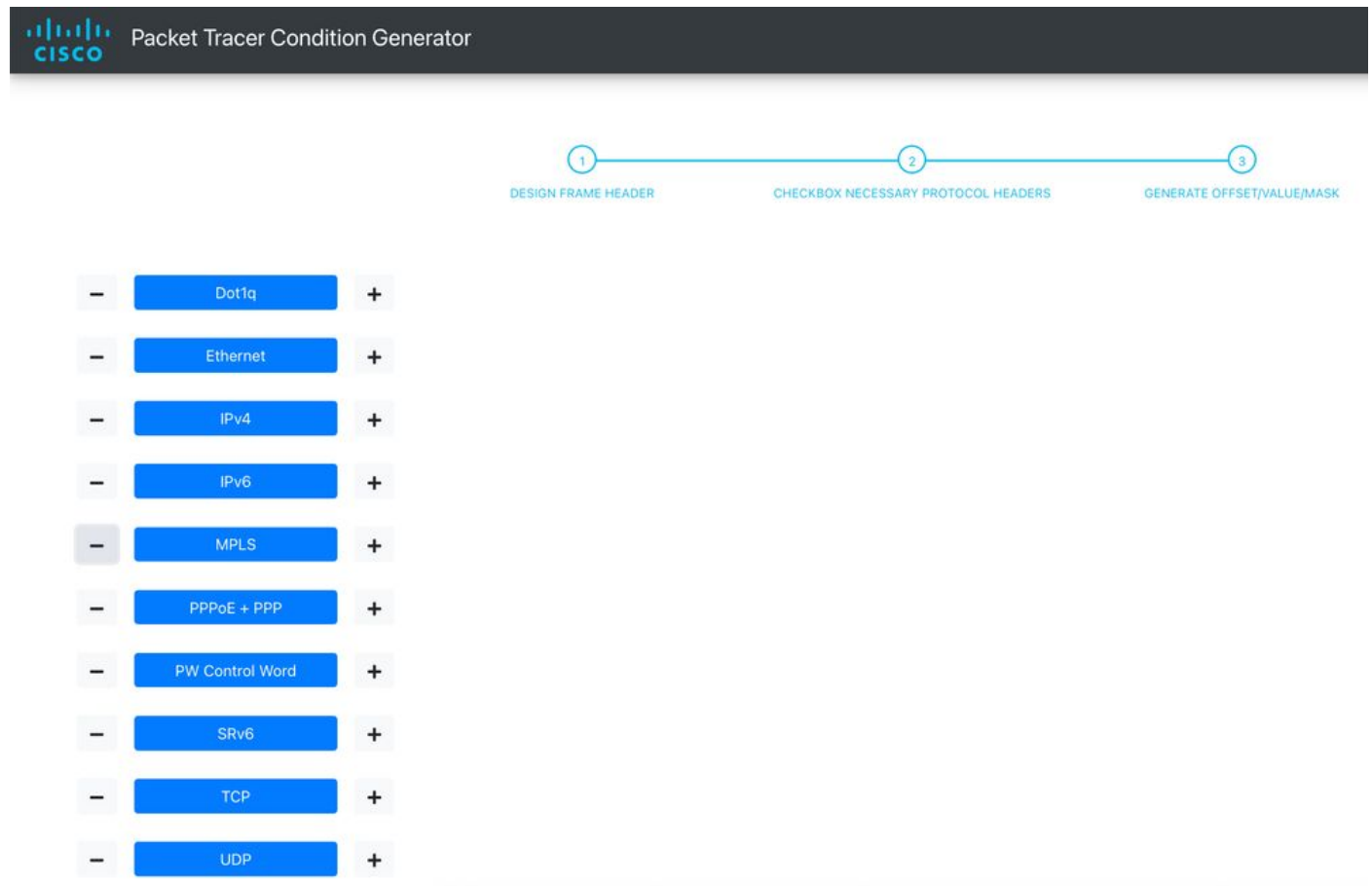
提示：在子接口上跟踪时，Offset/Value/Mask规范必须考虑dot1q或QinQ封装。

Packet Tracer条件 — 偏移/值/掩码

“XR Packet Tracer Condition Generator Web App”提供了创建Packet Tracer条件的工具。

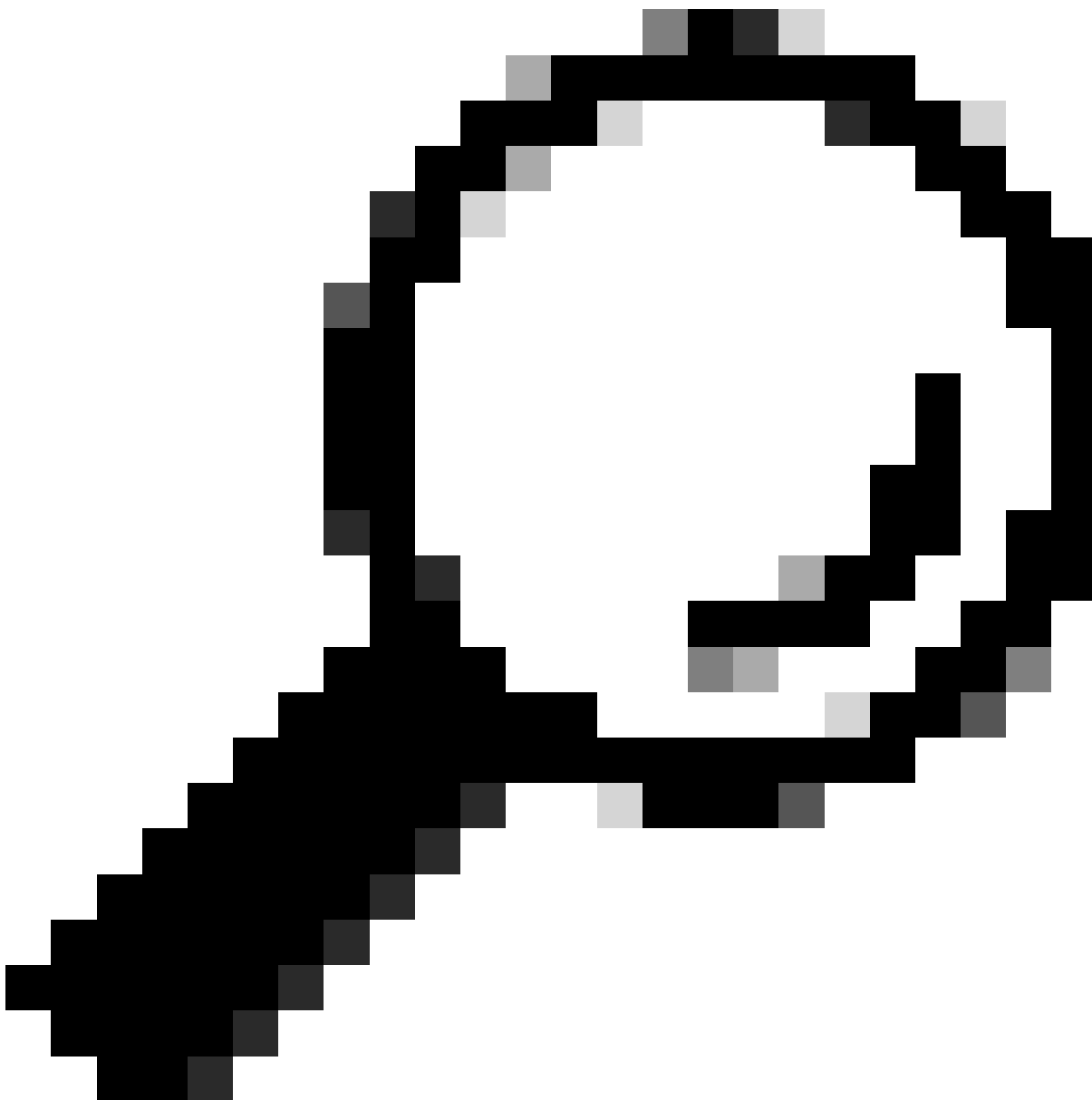
可以在GitHub上访问其源代码和安装指南，名称是：[XR Embedded Packet Tracer - Condition Generator](#)。

通过此应用程序，您可以直观地构建所需数据包流的协议栈，选择用于定义条件的相关层，并输入描述要跟踪的特定流的值（带有可选掩码）。



Web应用的登录页面显示配置支持的协议报头的列表。

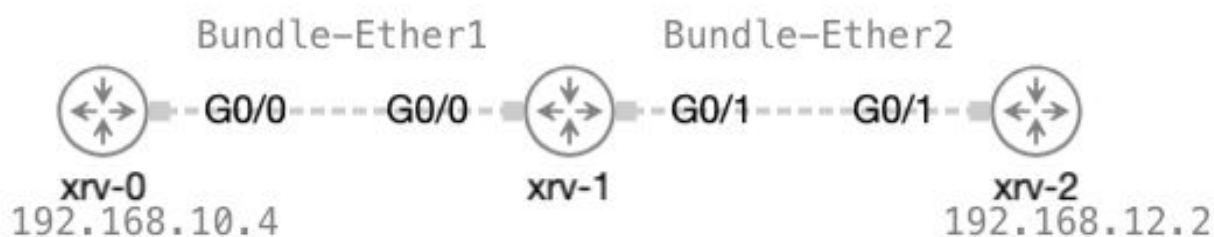
请确保在要匹配流量的信头之前包含所有必要的信头，因为偏移计算取决于信头的顺序。



提示：切记要包括PW控制字头（如果正在使用）。

配置示例：

以下是拓扑示例。我们的目标是验证数据包是否正确接收并通过XRV1设备传输：



1. — 为要监控的特定接口设置数据包跟踪条件。

```
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. — 生成偏移/值/掩码，选中要匹配的报头旁边的复选框。如有必要，您可以选择多个信头。对于每个选定的报头，相应的帧将显示在右侧。输入所需的值并在帧中蒙版，然后单击Submit按钮完成配置。

3. — 将“偏移/值/掩码”复制到剪贴板后，使用它来定义条件：

```
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. — 检查数据包跟踪状态：

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. — 启动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. — 让我们等待几分钟，以便捕获流量：

7. — 检查结果：

```
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      NPO          PACKET_MARKED                         P   GigE0_0_0_0
0/0/CPU0      NPO          PACKET_TO_FABRIC                     P
0/0/CPU0      NPO          PACKET_TO_PUNT                       P
0/0/CPU0      NPO          PACKET_FROM_FABRIC                   P
0/0/CPU0      NPO          PACKET_TO_INTERFACE                  P   GigE0_0_0_1
RP/0/RP0/CPU0:xrv-1#
```

8. — 您可以使用show packet-trace description命令检查注册到Packet Tracer框架的所有计数器及其说明：

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

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

9. — 停止数据包跟踪：

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

相关信息

[XR嵌入式Packet Tracer](#)

[思科技术支持和下载](#)

[了解ASR 9000系列线卡类型](#)

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。